Spreadsheet Based Model
Test

Weiqian Sun
Christopher Davey
Ford Motor Company
June 20, 2007
Acknowledgements

• Dan Card
• Doug Rosner
• Saied Rasavilar
• Jason Marx
Outline

• Software Systems Engineering Challenges
  – Complexity Growth
  – Software Defect Distribution

• Model Based Development Process/Activities

• Excel Test Design and Model Environment
  – Model Elements
  – Test Sequence Design
  – Test Sequence Visualisation

• Conclusions
Software System Challenges

• Electronics/software
  – 20% of vehicle cost today
  – 35-40% by year 2010
  – 90% new innovations from software

• System Complexity
  ~40-70 Micro-Processors and >10 million lines of code

• High Cost of Development
  – Automakers spend $2 billion to $3 billion a year fixing software problems, (Starvros Stefanis, IBM)
Software System Challenges

- Requirements related defects represent >50% of all software defects (Ford defect data aligns with industry published data)

- Gary Mogyorodi (2003) quoted study

- Ford North America Defect Distribution Data
Tranditional Development Process/Activities

• Distinct 4 phases
  – Requirement and Spec
  – Design
  – Implementation
  – Test and Verification

• Weakness: Document centric
  – Incomplete,
  – Ambiguous,
  – Easily misunderstood
  – Late Feedback/Slow Iteration
Model Based Development
Process/Activities

Model Based System Engineering

- **Documents functional requirements using block diagrams, state charts and flow diagrams** to create an easily readable/understandable requirements model
- **Enables simulation in a virtual environment** to study correctness, completeness and suitability of documented requirements to Customer usage needs
- **Provides the ability to generate test and automatically execute** for use in requirements validation and re-usable within HiL test environment
- **Supports Rapid Prototyping of innovative features/new technologies** in both a purely virtual environment or a progressively real-world, target hardware environment.
- **Generates “error-free” target software automatically** and enables validation testing of code re-using full coverage test scripts

Major Benefits

- Early identification and elimination of requirements defects
- Early discovery and mitigation of Software related failure modes
- Complete, easily understood and simulatable requirements specification
- **Supports the Electronic Consumer Lifecycle** with fast development cycles with major re-use of models & test vectors
- Enables re-use of requirements, test vectors, models and software
Model Based Design
– Systems/Holistic Approach

1. Model Based Requirements Management.

2. Architecture Models

3. Design Models (Behavioral Models)

4. Detailed Design Models (Implementation Models)

5. Test Vector Generation & Re – Use

6. Autocode

7. Component/Feature HiL

8. Sub-System HiL

9. System HiL

Vehicle Level Test

MiL – Model in the Loop
SIL – Software In the Loop
HIL – Hardware In the Loop
Excel-Model Test Environment

- Masked Subsystem uses Excel Workbook data to create Signal Builder visualization utilizing M-Script Callbacks
Feature Model Library

Context Diagram
– Input/Output Signals, Tunable Parameters and Data Definitions

Init File: Myfeat_Setup.m
State Transition Diagram

- Enumerated signals must be supported
- Timer vs. Counter Ford-EESE user preference
Model Data

- Setup model data with M-File

```matlab
%% Input
%InputSignal1
INPUTSIGNAL1_OFF = 0;
INPUTSIGNAL1_ON = 1;
...
%% Config
...
HoldTm_Cfg = 1.5; %sec
...
Decision Table

Traditional Spec: Decision Table

<table>
<thead>
<tr>
<th>Rq'n Num</th>
<th>Ignition_Status_Available</th>
<th>Ignition_Status</th>
<th>Key_In_Ignition_DbncV</th>
<th>Key_In_Ignition_Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>R: 3.1.17.7</td>
<td>AVAILABLE</td>
<td>OFF</td>
<td>IN</td>
<td>IN</td>
</tr>
<tr>
<td>R: 3.1.17.8</td>
<td>AVAILABLE</td>
<td>OFF</td>
<td>OUT</td>
<td>OUT</td>
</tr>
<tr>
<td>R: 3.1.17.9</td>
<td>AVAILABLE</td>
<td>RUN</td>
<td>IN</td>
<td>IN</td>
</tr>
<tr>
<td>R: 3.1.17.10</td>
<td>AVAILABLE</td>
<td>RUN</td>
<td>OUT</td>
<td>IN</td>
</tr>
</tbody>
</table>

- Multi-valued truth table support would reduce the modeling workload

Feature Model: Stateflow® Chart
• Traceability to/from Word, Excel, Team Centre Systems Engineering (TCsE), …
Test Harness

TestSeq: Masked Subsystem Block
## Test in Spreadsheet

<table>
<thead>
<tr>
<th>InputSignal1</th>
<th>InputSignal2</th>
<th>InputSignal2</th>
<th>Time</th>
<th>OutputSignal1</th>
<th>OutputSignal1</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF</td>
<td>OFF</td>
<td>FALSE</td>
<td>step</td>
<td>OFF</td>
<td>OFF</td>
<td>Rqmt3.2.1, this is init condition</td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>TRUE</td>
<td>step</td>
<td>ON</td>
<td>OFF</td>
<td>Rqmt3.2.2, change from state 1 to state 2</td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>FALSE</td>
<td>step</td>
<td>OFF</td>
<td>ON</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ON</td>
<td>OFF</td>
<td>TRUE</td>
<td>&lt;HoldTm_Cfg&gt; - step</td>
<td>OFF</td>
<td>OFF</td>
<td>...</td>
</tr>
<tr>
<td>ON</td>
<td>ON</td>
<td>TRUE</td>
<td>2</td>
<td>ON</td>
<td>ON</td>
<td>Rqmt3.2.5, Reach final steady state</td>
</tr>
</tbody>
</table>

- **Boolean Types**
- **Enumerated Types**
- **Adjusted sample Time**
- **Absolute time in seconds**
- **Configured Time Constant**
Test Signal

- Excel based test sequences are easily visualised within the standard Signal Builder
Test Conditions

<table>
<thead>
<tr>
<th>#SampleTime=0.25</th>
<th>Change default time step size, the unit is second.</th>
</tr>
</thead>
<tbody>
<tr>
<td>#OutputSignalPrefix = e_</td>
<td>Add a prefix string to output signals. This support test recording of both expected and model output signals.</td>
</tr>
<tr>
<td>#StartScripts = 'Switch_Cfg = SWITCH_CFG_OPT2'</td>
<td>Specify M command or script to run before the start of simulation for the test sequence. This is useful for assigning parameter and initial values for model data.</td>
</tr>
<tr>
<td>#Interpolate=[1 0 1]</td>
<td>By default all signals are treated as type of zero order of hold between sample times. With this option, one can specify a ‘1’ for linear interpolation (FOH) or ‘0’ for ZOH.</td>
</tr>
</tbody>
</table>
Simulation Condition

• Setups using checkbox
  – Simulation stop at the end of signal definition
  – Automatic solver/step size per group
  – Automatic block sampletime per group
  – Permit test group start script
    • Support use of multiple TestSeq blocks
Coverage Report

- Traditional Spec: Verify design by simply reviewing spec
- Feature Model: Review Model Coverage Report

Coverage:
- Decision
- Condition
- MCDC
- Range
- Lookup Table

- Test generation is an iterative process until desired coverage is reached
Conclusion

• Model Based Feature Design:
  – Can provide a faster, more complete and re-useable development environment

• Excel Based Test Design:
  – Builds on existing engineering tools, methods and processes
  – Provides test vector authoring capability within legacy tools
  – Enables efficient adoption of model based toolsets by less experienced users

• Next Steps:
  – Encourage The MathWorks to integrate these capabilities into their toolset
  – Pursue greater integration between their existing tools