Automated Statechart Model Checking with Promela/SPIN

Paula J. Pingree, Task Lead
Ed Benowitz, Developer
Autonomy and Control Section 345
Jet Propulsion Laboratory

Task Overview

- Extended capabilities for flight software verification by introducing formal method model checking
- Evaluated and implemented software tools that will help to automate the process
- Applied method and tools to Fault Protection (FP) flight software (FSW) implemented in StateFlow® statecharts as a prototype
- Infusing this verification technology in future projects

Team:
- Paula J. Pingree (JPL), Lead
  Micah Clark, Eddie Benowitz
  Software Engineering & Technology Infusion
  Group, Autonomy & Control Section (345)
- Erich Mikk (Erlangen, Germany), Independent Consultant
  Developer of Extended Hierarchical Automata (EHA)
- Gerard Holzmann,
  Margaret Smith, Dennis Dams
  (Bell Labs, Murray Hill, NJ), Co-Investigators
  Computer Principles Research Department
The Task in General

- Translation Programs
  Getting the statecharts of interest into Promela for SPIN

- The Environment - "Closing-the-loop"
  Creating a meaningful system for SPIN verification

- SPIN Model Checking
  Evaluating the integrated system model against user-specified correctness properties

- Maintaining the Vision
  Application to Flight Projects and Technology Infusion
  Presentations to gain interest & support
  Proposals to continue the work

Applying Model Checking to FSW

- We provide automated translation of the statechart model from Stateflow to Promela, the input modeling language of SPIN

- Key Benefits:
  - SPIN validation model and FSW code, now both auto-generated, have the same source (the Stateflow statechart)
  - Design validation can occur earlier in development cycle and without the use of valuable testbed resources

- Based on Requirements
- Expressed in Linear Temporal Logic (LTL)
Products

1. Our Toolset: HiVy
   - Translates Stateflow statecharts into Promela for SPIN
   - Includes the following programs:
     - Sfpse: implemented in Perl
     - sf2hsa: implemented in C
     - hsa2pr: implemented in C
     - HSA merge facility: implemented in C
   - Produces syntactically and semantically correct Promela models
   - Toolset is under CVS control and runs on UNIX workstations
   - C programs developed with the VDM compiler

2. Documentation (see References)
   - HiVy Toolset User’s Guide v1.0 (Draft)
   - HSA Format - to be submitted to the 2003 SPIN Workshop
   - The Abstract State Machine, (a formal semantics definition for Stateflow)
   - Conference paper for the 21st Digital Avionics Systems Conference titled
     “Validation of Mission Critical Software Design and Implementation using Model Checking”
The HiVy Toolset Work Package

1. 'SfParse' recognizes all relevant elements from Stateflow *.mdl files
2. 's2hsa' transforms parsed elements into HSA for further processing
3. 'hsa2pr' program supports:
   - Sequential automata with states, transitions and default transitions
   - Transition labels with conditions and actions varying over boolean and integer variables
   - Hierarchy
   - AND-states
   - Event handling
   - Inter-level transitions
   - Junctions
   - Condition actions
   - User-defined C code
4. HSA Merge facility allows integration of multiple HSA files into a single Promela model upon translation

Correctness Properties

- Verification: Correctness Properties

```
<> P
eventually P
<> [ ] P
eventually always P
[ ] <> P
always eventually P
```

- Correctness Properties (CP) are formal statements of the expected behavior of a system
- The accuracy of verification results depends on the accuracy and completeness of the CPs
- CP events and states must be linked to concrete events and states in the model

hsa2pr produces prop_list for generating CPs!
Alternate Environment Model

- Can be time consuming to enter diagrams
- Specify a state machine within an Excel spreadsheet
  - States
  - Transitions
  - Hierarchy

<table>
<thead>
<tr>
<th>State Name</th>
<th>Present State</th>
<th>To Initial State</th>
<th>The Final Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>power-select</td>
<td>on</td>
<td>off</td>
</tr>
<tr>
<td>off</td>
<td></td>
<td></td>
<td>power-select</td>
</tr>
<tr>
<td>power-select</td>
<td>power-select</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Transition Name</th>
<th>From State</th>
<th>To State</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>on</td>
<td>power-release</td>
</tr>
<tr>
<td>TRUE</td>
<td>power-release</td>
<td>on</td>
</tr>
<tr>
<td>TRUE</td>
<td>off</td>
<td>power-release</td>
</tr>
<tr>
<td>TRUE</td>
<td>power-release</td>
<td>off</td>
</tr>
</tbody>
</table>

The DS1 FP Response Verification - 1

FP ENGINE + RELATED COMPONENTS
(SCHEDULE OF FP's and ENV)

FP Engine
(+Manager): Promela
hand-coded

FP Engine
(+Manager): informal spec.

FP RESPONSES
(MAIN ORI OF VERIF)

FP Responses
(+Monitors): Promela
w/calls to C

HiVy
auto-generated
by SF Coder

HiVy x12hza

SPACECRAFT
(ENVIRONMENT FOR FP's)

Spacecraft: Promela
C Libraries: fpr.exe, e.g., dir(-) etc.
The DS1 FP Response Verification - 2

- Current capabilities
  - Translation of FP Response statecharts to HSA via HiHy
  - Translation of S/C environment spec. to HSA via 'xl2hsa'
  - Merge of Response and S/C HSA into integrated Promela model via HiHy
  - Integration of user-defined C-code; mechanism defined, interface tested
- Work in progress
  - Integration with Deep Impact

What's Next

- FY03 R&TD Proposal funded
  “Rapid Adoption of Model-Based Validation for Mission-Critical Flight Software Architectures & Domains”
- Under the Advanced Software Technology and Methods Initiative (ASTMI)
- Using HiHy translation to perform model checking on various domains
  - Deep Impact FP Responses
  - MDS Architecture example
  - MER Surface Ops Behavior example
- Continued support in FY03 from SQI
  - Toolset maintenance and improvement
  - Technology infusion for future projects
References

P. Pingree, E. Mikk. HiVy User’s Guide v1.0 (Draft), 2002
E. Mikk. HSA Format, 2002
E. Mikk, P. Pingree, M. Clark. The Abstract State Machine (Draft), 2002
E. Benowitz. User Level Documentation for XL2HSA (Draft), 2002


