DRYING CONTROL LOGIC DEVELOPMENT USING MODEL BASED DESIGN
Problem Definition

“To generate and deploy automatic code for Drying Control Logics compatible with new SW architecture in 6 months using MBD, a novel approach”
Key Challenges

- Drying Control logic is combination of four different modules having different Software routines.
- Generate AutoCode compatible with new Software Architecture
- Unclear requirements for Fault, Power and Safety Management
Need for Drying Control Logic

Washing machine with Drying Control

- To have optimized Drying cycle for different load types and size
- To avoid over drying and damage to laundry
- Ease of use for customer
Approach
Requirements

- In Whirlpool, all requirements are defined, managed and reported through Rational Doors Next Generation.
- Stateflow and Simulink are used for gap analysis
- Multiple iterations of review and discussions were performed
- Referencing of interfacing inputs and feedbacks
Requirements Reviews and Discussions

Modeling and Requirement Refinements:

• Added tuning parameters (timing, calibration parameter)
• While modeling missing parameters, relationships, interfaces were identified and corrected.
• Identified missing requirements required for Fault and Safety Management.
• Stateflow enabled to define transition, conditions and actions in the control logic
Plant Model Development

- Washer-Dryer plant model imported into Simulink from Dymola.
- System Models are developed by System modeling team.
- MBCD team will import these for Algorithm Validation in Simulink.
- Tools Used: - PSP Toolbox from Mathworks.
Drying Logic Top Level

Functional Unit

Drying Logic Modules

Simulink and Stateflow
Requirement linking from Simulink to DNG

- Configure Requirement Settings
- Select Project Area from DNG
- Selecting Requirement from DNG
- DNG Link of requirement in Simulink
- Simulink Implemented link in DNG

Simulink Verification & Validation
Following standard guideline checks were performed:
• MATHWORKS Automotive Advisor Board
• IEC 61508

Warnings and Failures are corrected after analyzing reports (e.g.)
• Identify signal labels and block labels that are not correct for C variable names.
• Check usage of exclusive and default states in state machines
• Identify mismatches between names of Stateflow ports and the associated signals.
Control Logic Validation with Plant Model
Control Model Verification & Validation - MIL

- **Test Case Preparation**
- **Test Harness Creation**
- **Running Test Cases**
- **Generating Test Reports**
- **Model Coverage Analysis**

**Simulink V & V**

**Simulink Test Manager**

**Simulink Verification & Validation**

**Simulink Test**
• Results of SIL are compared with Model test results
• Same test cases can be used
• Test source code on development computer
• Report of SIL includes untraceable code or model part as well as gaps between model output and generated code output
Rapid Control Prototyping
Autocode Generation and Integration

Configuration Settings

C code is generated using Simulink Embedded Coder

Code is then integrated to the RTC stream and tested in the appliance

Simulink Embedded Coder
Advantages of MBD Approach

• Direct Import of Dymola Plant Models into Simulink.
• Detecting errors in early stages
• Powerful and Formal Analysis
• Reusable Components
• Automatic Code Generation
• Highly Scalable, Ease of maintenance
• Reusability of Test Cases
• Good Test Management