Managing clone mutations in Simulink models

Yanja Dajsuren

Research Meet on 'System Engineering'
High Tech Systems Center

December 3, 2015
Eindhoven, the Netherlands
Independent Software Development

Software

Software

Software
Software Development with Conventional Reuse
Product Line Oriented Development
Simulink Clone Mutations

- According to Stephan et.al:
  - Different layout (color, position, size, other attributes) of elements
  - Different ordering of elements (blocks, lines, ports, branches)
  - Different names of elements (blocks, lines)
  - Different values of elements (blocks)
  - Added or deleted block
  - Changed block type...
Simulink Clone (Textual Format)
Variant Configuration Language (VCL)

```vcl
#set subsystem_name = "Layout2a_Position"
#set subsystem_location = "[596, 16, 1412, 554]"
#output "Layout2a_Position.mdl"
#adapt "Position.vcl"

SPC file

<adapts>

System {
    Name  @subsystem_name?
    Location  @subsystem_location?
    Open  on
    ModelBrowserVisibility  on
    ModelBrowserWidth  200
    ...
}

Position.vcl

System {
    Name  "Layout2a_Position"
    Location  [596, 16, 1412, 554]
    Open  off
    ModelBrowserVisibility  on
    ModelBrowserWidth  200
    ...
}

Layout2a_Position.mdl
```

VCL Processor
Approach

Workflow for the Model Manager:

1. Run clone detection
2. Select a clone class for unification
3. Identify variation points and variants
4. Create VCL representation and store in repository
5. Any clone classes left for unification?
   - yes: Continue with workflow
   - no: End workflow

Workflow for the Developer:

1. Locate subsystem to clone
2. Subsystem is Green?
   - yes: Customize clone from repository
   - no: Copy, paste, and modify the subsystem in the new context
3. Concrete subsystem is generated
   - The two copies of the subsystem are colored red