Improving Maintenance by Creating a DSL for Configuring a Fieldbus

Mathijs Schuts & Jozef Hooman
Philips & ESI – TNO
October, 2016
Software Life Cycle

• Different component in high-tech system have different life cycle

• Some component are legacy

• Maintenance effort 50-75% of SLC

K. Bennett and V. Rajlich, Software Maintenance and Evolution: Roadmap
DSLs for Improving Maintenance of Legacy Components

- Is it financially feasible to extend the life of a legacy component using a DSL?
- What are the pros and cons of using a DSL compared to the current way of working?
Allura Xper

• Interventional X-ray
  – Minimal invasive treatments

• Treatment of
  – Cardio, and
  – Vascular diseases
Building Blocks

• Customer can configure system

• Patient tables
  – 1-6 motorized axes

• Stands
  – Ceiling
  – Floor

• Stand mover
Fieldbus Topology

Type = Vendor + Model

Node 1  Node 2
Read  Write

Packet
Fieldbus Topology

• Topology descriptions for the master are made with a commercial tool
• Creating topology files is an error-prone time consuming activity
• Hardware set-ups need to be created to test the topology files
DSL for Fieldbus Configurations

```
topology
name ExampleTopology
network
TYPE_A1 <-> TYPE_A2 <-> TYPE_B
  <-> TYPE_B <-> TYPE_B
prev TYPE_A2 port C TYPE_C
prev TYPE_A1 port C TYPE_D
  <-> TYPE_D <-> TYPE_E
```
Checking the Topology

Validation rules:
• No two branches connected to the same port of a node
• Topology names are unique
• TYPE_A1 and TYPE_A2 are both present or not present
• TYPE_A2 is connected to TYPE_A1
DSL to Describe System Configurations

• In the future we need 2000 topology files
• A system release needs to contain support of all possible configurations, hence topologies
DSL to Describe System Configurations

building block
id            BB1
type           CeilingStand
vendor(s)     VENDOR_A

building block
id            BB2
type           Table
vendor(s)     VENDOR_A  VENDOR_B

configurations
configuration
name           Configuration1
building blocks BB1

configuration
name           Configuration2
building blocks BB1  BB2
Concluding Remarks

• Is it financially feasible to extend the life of a legacy component using a DSL?
  – Return on Investment (ROI) = (gain from investment – cost of investment) / cost of investment
    ▪ Gain: 2000 topologies * 8 hours = 16000
    ▪ Investment: 50 hours current DSL + 20 hours add support new vendors + 20 hours to create instances = 100
    ▪ ROI = (16000 - 100) / 100 = 159
Concluding Remarks (cont’d)

- What are the pros and cons of using a DSL compared to the current way of working?
- Advantages:
  - DSL is simple and easy to use
  - Faster
  - Because of validation rules, less error prone than current way of working
- Disadvantages:
  - Generator’s code need to be maintained
  - Generators are programmed in Java/Xtend while C++ is preferred programming language
  - Used IDE is Eclipse while the preferred IDE is Microsoft Visual Studio
Concluding Remarks (cont’d)

• High ROI indicates that the investment in the DSL will be preferred above the current way of working

• At Philips we applied DSLs on multiple other cases

• Given this and other experiences, we continue to use DSLs
Q&A