TOWARDS BUILDING REUSABILITY IN CONTROL SYSTEMS – A JOURNEY

ICALEPCS 15, Melbourne, Australia

Subhrojyoti C*, Puneet Patwari, Amar, Banerjee, N. Swaminathan, G Muralikrishna,
Tata Research Development and Design Centre, Tata Consultancy Services,
subhrojyoti.c@tcs.com, +91-20-66086411
Talk - Overview

- Brief about who we are

- Challenges and opportunities
  - Towards reusability

- Specific problem areas
  - Reuse across all phases of development lifecycle
  - Solution approach – MDE approach with a focus on knowledge reuse

- Conclusion
  - Where are we going with this – towards knowledge based systems
Who we are – System Map Maker

TCSL
TRDDC – One of Many innovation Labs
System Map Maker – Research team

Swami
Subhro
Murali
Puneet
Amar

Broad area of focus:

- Integrated family of models that capture all the systems engineering information related to a particular system
  - Work out information architecture to facilitate System Engineering
Collaborations - Control systems

- Translate the ITER CODAC architecture into design
- Demonstrate using prototype
- Build Mini CODAC

Engagement with ITER
Year: 2007

Collaboration with GMRT
Year: 2008-09
- Architect the NextGen GMRT Control system
- Develop reusable architecture and environment

CoDR SKA
Year: 2010-11
- Participate in the creation of the Concept of Design for SKA Supervisory Control System

Part of SKA TM – 2013-Till date
- Participate in the capacity of TM Project Engineer and TELMGT Lead
- Create detailed design and prototypes for TM of SKA

Engagement

Year: 2007
Key observation – Opportunity/Motivation

- Very robust solutions available for reuse for the purpose of implementation
  - EPICS, TANGO, ACS and so on

- Challenges and areas of opportunities
  - Significant amount of time spent in the early design life cycle
    - E.g. – ~ 3 years to design TM
  - Each project recreate the design thinking from scratch – and reinvent common M&C concerns, requirements, solution concepts and so on.
    - Usage of SysML does help communicate better, but still falls short on the reusability aspect!
  - Much of the knowledge reside with the domain experts
Solution opportunity – Start with reusable architecture M&C

Can we create a common solution architecture for M&C - requirements, non functional concerns, functional blocks and so on
Solution opportunity – Start with reusable architecture M&C

Can we create a common solution architecture for M&C - requirements, non functional concerns, functional blocks and so on

Independent of implementation technology,
Retargetable to multiple off the shelf options
Completely Data Driven
Identification of the input data

Solution to each control problem created through the composition of such nodes
Identification of the input data

Solution to each control problem created through the composition of such nodes.

Each node configured using its Self Description

---

**Identification of the input data**

**Solution to each control problem created through the composition of such nodes**

**Each node configured using its Self Description**

---

**Table:**

<table>
<thead>
<tr>
<th>Action Name</th>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DataSim Amplitude</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>DataSim Offset</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>DataSim Frequency</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>DataSim Sine</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>DataSim Square</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>DataSim Triangle</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DataSim UpdateFrequency</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DataSim Ramp</td>
<td></td>
</tr>
</tbody>
</table>

**PUMPDOWN**

<table>
<thead>
<tr>
<th>Device Name</th>
<th>Stream Name</th>
<th>Byte at location</th>
<th>Has values</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>MagnetSystem</td>
<td>Cryo Vacuum Pressure</td>
<td></td>
<td>24 VACUUM ACHIEVED</td>
<td></td>
</tr>
<tr>
<td>MagnetSystem</td>
<td>TF Temperature</td>
<td></td>
<td>20 QUENCH OCCURRED</td>
<td></td>
</tr>
<tr>
<td>MagnetSystem</td>
<td>Cryo Vacuum Pressure</td>
<td></td>
<td>6 COOLING ACHIEVED</td>
<td></td>
</tr>
</tbody>
</table>

**COOLDOWN**

<Command Name = “PUMPDOWN”/>  
<Command Name = “COOLDOWN”/>  
<Command Name = “PREPARETF”/>  
<CommandAttachment Name = “PowerSupply”/>  
<Signal Name = “Cryo Vacuum Pressure” Type>  
<UpdateRate Value = “10”/>
Identification of the input data

Solution to each control problem created through the composition of such nodes

Each node configured using its Self Description

There might be thousands of such specifications and hence needs support to maintain consistency - A development environment in its own right!!
Control system designer

Control objectives
Commands-Response, Events, Alarms

Behavioral spec
– Constraints, state machines, error handling logic and so on

Specify test and simulation related input

Incremental support to translate design into implementation

Domain Specific M&C Development Environment
M&C Architecting Environment – Implementation

Domain Specific M&C Development Environment

Controller Specification Development (DSL)
Plugins to validate the specified solution
Support to test and simulate the specified solution
Compilation to - off-the-shelf packages (e.g. TANGO)

Eclipse Modeling Framework (EMF)
XText, XTend
ATL

Based on MDD - Eclipse
Areas covered in the DSL as of now

- Specification of control structures
  - Control nodes
  - Commands, parameters
  - Responses
  - Events, Alarms
  - Data streams
  - States

- Behavior
  - Dependencies-Control Hierarchies
  - State machines
  - Command validation logic
  - Alarm detection logic
  - Alarm handling specifications
  - Data processing

- Placeholder
  - External/detailed logic
Specify M&C solution description using DSL – M&C ML
Process – Working of the environment

Specify M&C solution description using DSL – M&C ML
Process – Working of the environment

Specify M&C solution description using DSL – M&C ML

Parse into

Validation, M2M/T Translation…
Process – Working of the environment

Specify M&C solution description using DSL – M&C ML

Validation, M2M/T Translation…

Translation into TANGO implementation

Parse into

ICALEPCS 2015, Melbourne, Australia
Process – Working of the environment

Specify M&C solution description using DSL – M&C ML

Translation into TANGO implementation

Translation into TANGO implementation

Validation, M2M/T Translation…

Generation of Simulators and Test Cases
Factors influencing DSL design

Key drivers to the DSL design

- Important to understand the stakeholder concerns
  - In our case M&C designers
  - Leave out the details of the application domain (e.g. Astronomy), projects specifics

- Understanding and extracting key concerns from the domain
  - Important input is the vocabulary using while specifying M&C design

- Capturing the underlying architecture pattern
  - SACE suggests a strict hierarchical
  - Hence the flow of the DSL reflects the same
  - However, can’t be applied to other patterns, such as Agent based systems

- Decoupling the meta-modeling from the grammar definition
  - Capture the domain concerns in a structured form with relations
  - Don’t worry about the language syntax while doing so

- Careful analysis of the user intuitions for language syntax definition
  - Can be based on the combination of functional building blocks and architecture flows

- Extensibility
  - Support for incremental domain specific validations
  - Incremental addition of concepts (e.g. Radio telescope) and separation of concerns
Future direction – knowledge based engineering environment and conclusion

- We plan to try this our for uGMRT, SKA

- Such an environment is a stepping stone towards identification of domain models and capturing of knowledge –
  - Technical domain such as M&C, Networks, Security and so on
  - Application domain such as Radio astronomy Nuclear fusion

- Learnings can be used to build similar domain environment or contribute towards general capability such as SysML and so on.
Thank You!