Analyses Made to Order: Using Transformation to Rapidly Configure a Multidisciplinary Environment

Bjorn Cole
Jet Propulsion Laboratory,
California Institute of Technology
Bjorn.cole@jpl.nasa.gov

March 7, 2013
IEEE Aerospace Conference
Introducing DARPA “System F6”

• From the website:

System F6 seeks to demonstrate the feasibility and benefits of a satellite architecture wherein the functionality of a traditional “monolithic” spacecraft is delivered by a cluster of wirelessly-interconnected modules capable of sharing their resources and utilizing resources found elsewhere in the cluster. Such architecture enhances the adaptability and survivability of space systems, while shortening development timelines and reducing the barrier-to-entry for participation in the national security space industry.

The program is predicated on the development of open interface standards ...

• Flight demo of interoperability in 2015 with (currently) four spacecraft leveraging F6 infrastructure
Goals of Adaptable Systems Design and Analysis (ASDA) Tool

• “Evaluate the business case for fractionation”
• Our postulate – fractionation valuable when:
  ▫ Needs are dynamic
  ▫ The world is uncertain
• Play out the consequences of postulate and consequences through integrated modeling and analysis
We want to vary ...

- The number and types of spacecraft in the cluster
- Order of deployment
- Possible responses to unforeseen circumstances, including various ways to change the cluster
- The cluster architecture and formation to fly
- Different operating strategies
- And so on

- Too many combinations to deal with by hand!
Creating a Simple, User-Friendly Tool that is still customizable

**F6 Tool**: Create user-friendly GUI to setup and run nested experiments: 1) Architecture (Select # of MS, DS, PLs), 2) Stimuli, sizes of elements, run durations, etc.

**F6 Design Tool (ASDA)**
Why SysML?

- Need a platform for capturing the rules specifying what constitutes a cluster and how it should behave
- Open structure for SysML allows for elaboration as users and modelers get more sophisticated
- Open Modeling Group (OMG) standard that works with many others to help transition to model-based systems engineering (MBSE)
  - Query-View-Transform (QVT)
  - Business Process Modeling Notation (BPMN)
  - Real-time frameworks
The Trick

- Analysis tools typically have ill-posed data structures, either:
  - Too weak (e.g., Excel, many scripts)
  - Highly specialized (e.g., Finite Element, Modelica, etc.)
- SysML not a strong language for analysis, but excellent standardized data structure
- The trick – use SysML as Source of Truth but instance as analysis inputs via transformation!
Simplified version of Model Transformations

- Analysis Needs
  - Rules to Apply
  - Done in advance (human setup)
    - Produces: Analysis Specs
  - Done at build time (automatic)
    - Produces: Variants
  - Done at run time (automatic)
    - Produces: Populated Tradespace

We are here
What is Model Transformation?

Source Metamodel  \rightarrow  One-way Transformation Specification  \rightarrow  Target Metamodel

Model A  \rightarrow  Transformation Execution  \rightarrow  Target Model

conforms  \rightarrow  executes  \rightarrow  conforms
Transformation Flow: SysML $\rightarrow$ PHX

- Query: Gather Important Pieces of SysML Model
- Bundle into Intermediate Collections
- Transform Collections Into ModelCenter Objects
DARPA F6 Metamodel
Analysis Idiom from JPL Integrated Model-Centric Engineering (IMCE) Initiative
Analysis Context

«mission:Component»
My Spacecraft

«mission:performs»

«mission:Function»
Watch Area
Parameter Cross-Connection

Diagram:
- Thermal: F6 Thermal Model
  - PeakPower: W
  - TotalMass: kg

- Systems: Systems Sheet
  - Thermal.PeatPower: W
  - Thermal.Mass: kg

Connections:
- DirectionBlock from Thermal to Systems
- DirectionBlock from Thermal to Systems
Inheritance of Analysis Connections
Very Flexible Approach
Transformation Flow: SysML → PHX

- Query: Gather Important Pieces of SysML Model
- Bundle into Intermediate Collections
- Transform Collections Into ModelCenter Objects
Components with “Default Use”
Q: How do you keep track of what goes to which role-based instance of the system in an analysis context?
A: Inheritance (at the cost of some rules)
Transformation Flow: SysML → PHX

Query: Gather Important Pieces of SysML Model

Bundle into Intermediate Collections

Transform Collections Into ModelCenter Objects
Most Transform Effort Spent on Wiring Parameters on Analysis Instances

0-D to 2-D

0-D to 0-D

0-D to 1-D
A Final Product
What Comes Out In the End

- Instancing of a mothership-daughtership pattern with an arbitrary number of spacecraft
- Proper (and testable) connection between instances of analyses in pre-specified order
  - ASDA model runs typically involve 100-400 links per architectural combination
- (Pending final code module integration) One-click launch of architecturally distinct configurations from a SysML model