ASCEND™

16-18 November 2020 | Online

Model-Based Systems Engineering, Real-Time Operations and Autonomy

Fernando Figueroa, Lauren Underwood, and Duane Armstrong
NASA Stennis Space Center

www.ascend.events

OUTLINE

- MBSE and SysML, and NASA Platform for Autonomous Systems (NPAS)
- NPAS
- System Model Development using SysML
- G2/NPAS Model Development
- G2/NPAS software environment capabilities for integrated system model and real-time operations
- Conclusion and Recommendations

Model-Based Systems Engineering, Real-Time Operations and Autonomy

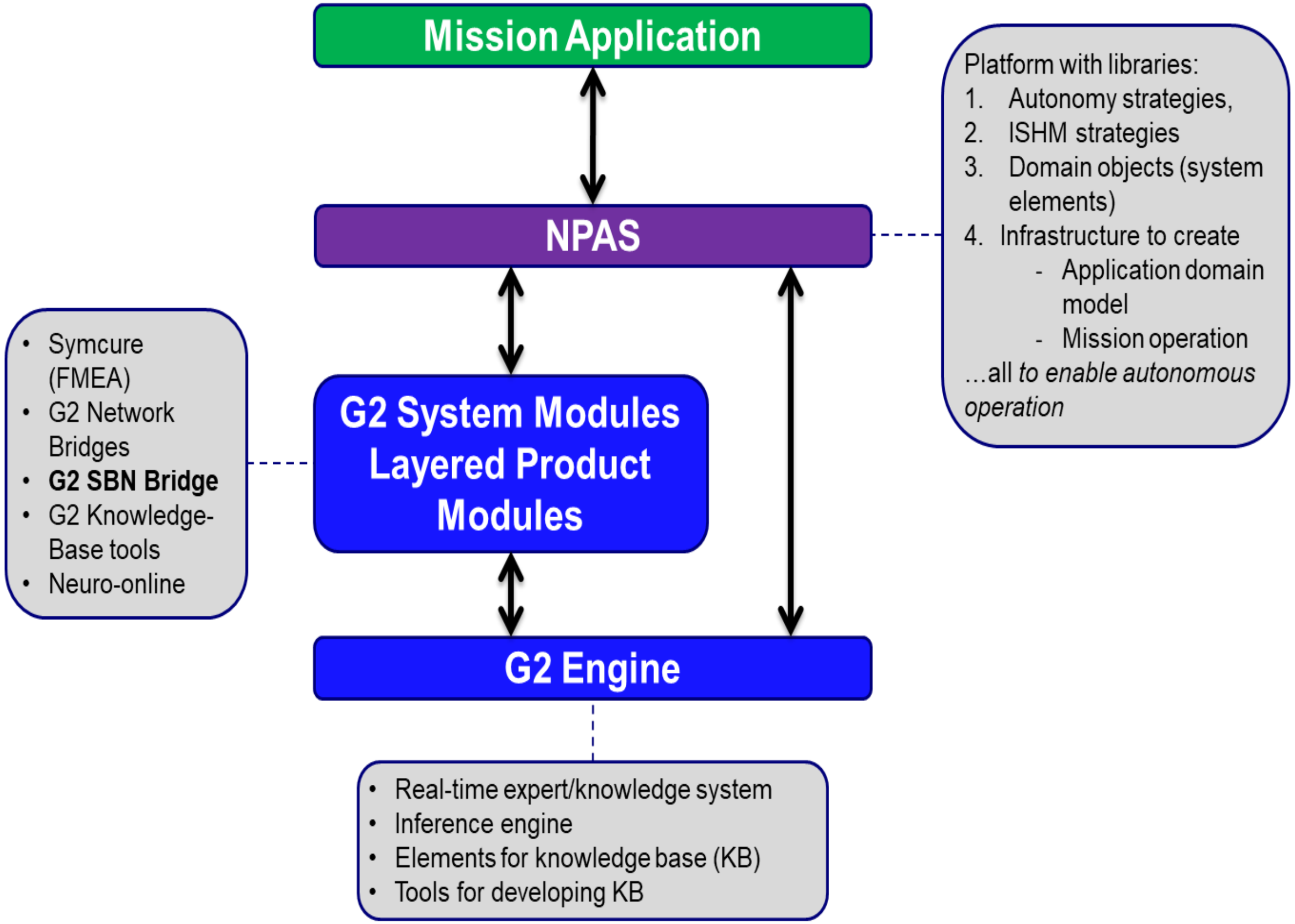
MBSE and SysML, and NASA Platform for Autonomous Systems (NPAS)

- SysML enables creation of very complete models that include structure and behavior using a standard language
- SysML models enables MBSE processes for development of systems and projects
- The NASA Platform for Autonomous Systems (NPAS) enables a paradigm for MBSE, incorporating additional content elements beyond current SysML; and models are live and can be used in real-time operations

Model-Based Systems Engineering Real-Time Operations and Autonomy

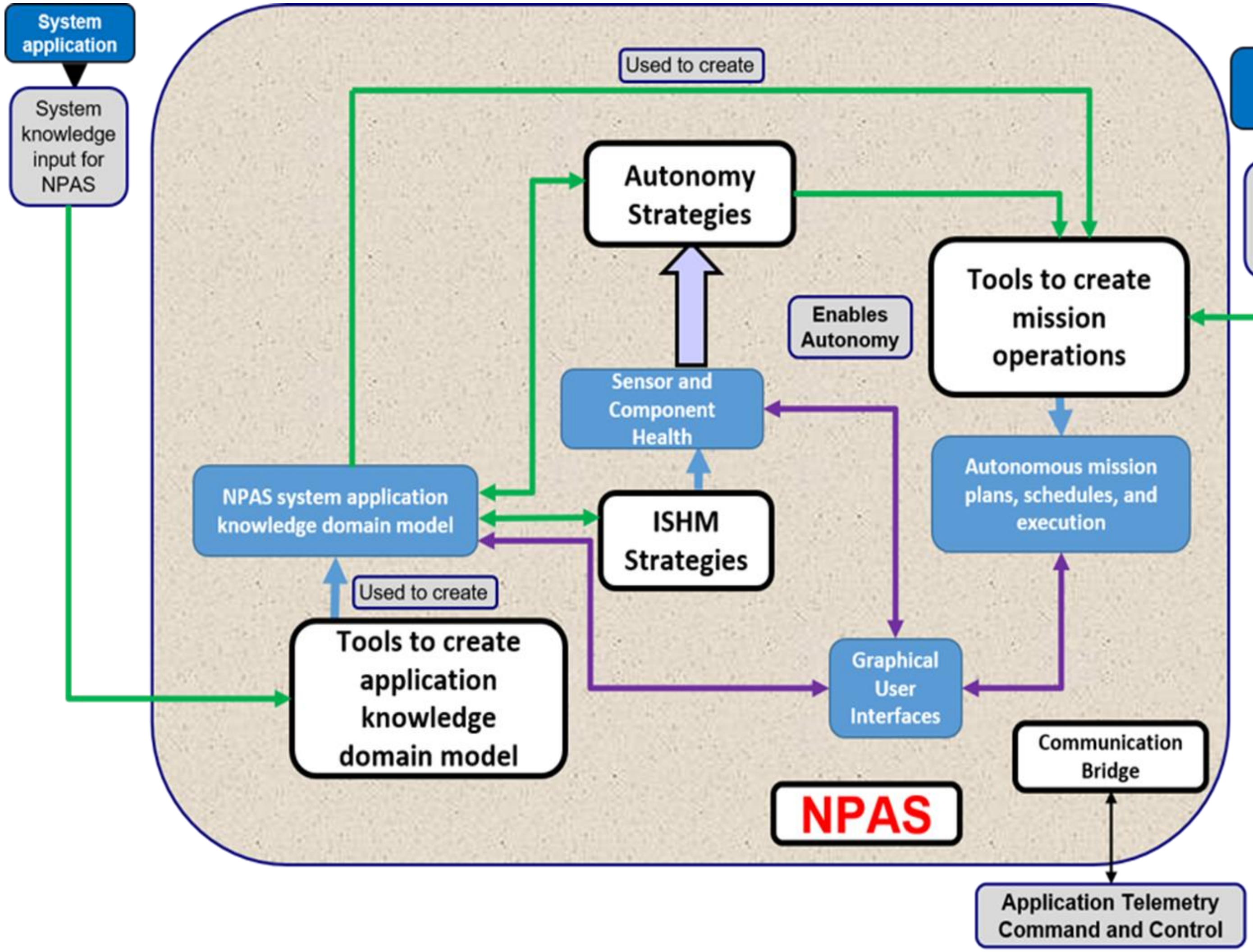
NASA Platform for Autonomous Systems (NPAS)

- Built upon G2, an object-oriented software environment for "intelligent" knowledge/expert-based real-time applications (offshoot from MIT)
- G2 tools for distributed autonomy
  - Cause-effect diagrams for FMEA (layered product)
  - G2 SBN Bridge to integrate with cFS applications
  - Neuro-online (neural nets – not being used in NPAS)
- NPAS Capabilities to implement intelligent "thinking" autonomous systems
  - Comprehensive knowledge system model
  - Integrated System Health Management
  - Autonomy strategies to accomplish goals
  - Task creation, scheduling, and execution



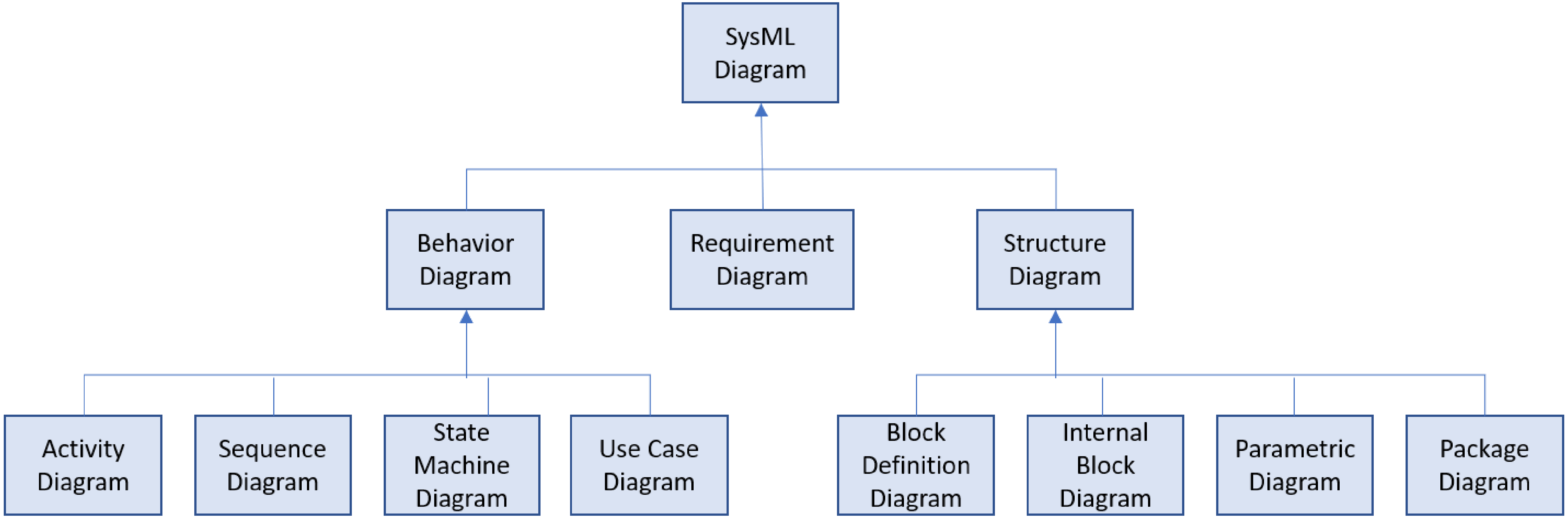
Model-Based Systems Engineering Real-Time Operations and Autonomy

NPAS Software Architecture



System Model Development using SysML

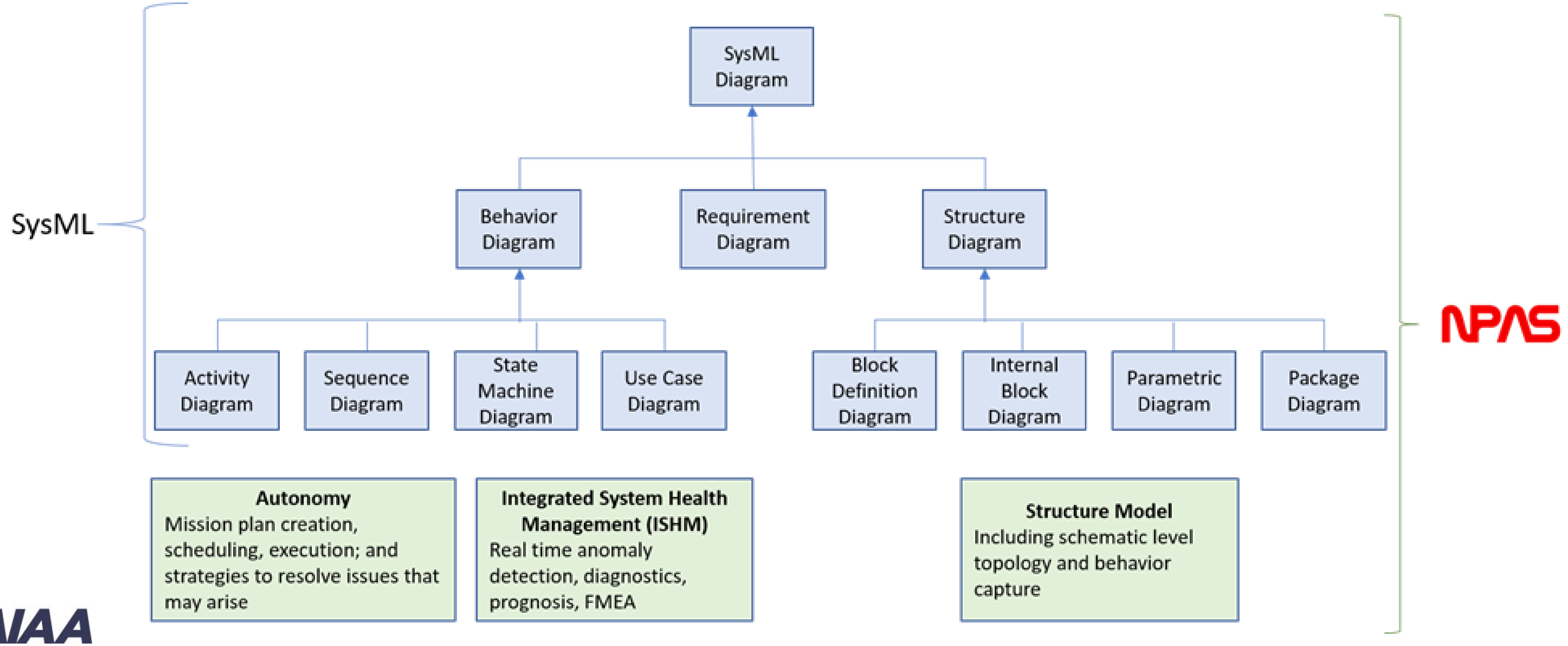
- Comprehensive models including structure, behavior, and requirements



Model-Based Systems Engineering, Real-Time Operations and Autonomy

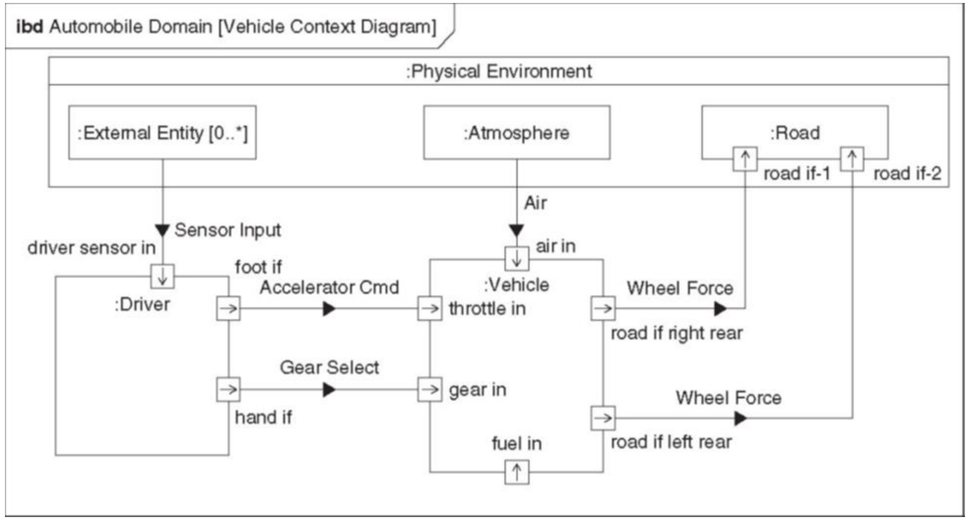
G2/NPAS Model Development

- Includes content associated with all SysML diagrams
- Augments behavior content for ISHM and Autonomy
- Augments detail of structure diagrams by including schematic details

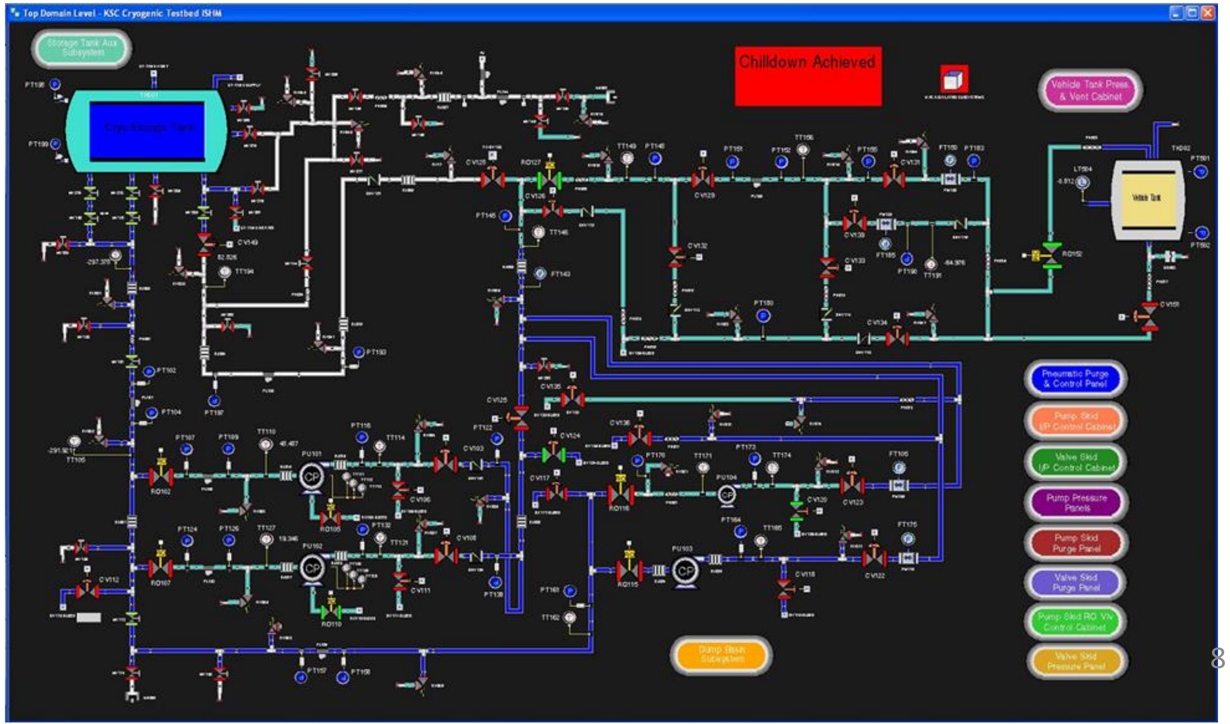


MBSE and SysML Modeling

- Purposeful decomposition into functional elements is not necessary in NPAS
- Structure and behavior are acknowledged/incorporated automatically, and recognized to be associated with concepts that are then employed in models that enable analysis and decision-making in real-time
- Content of all structure diagrams can be innately integrated on the basis of schematic level diagrams

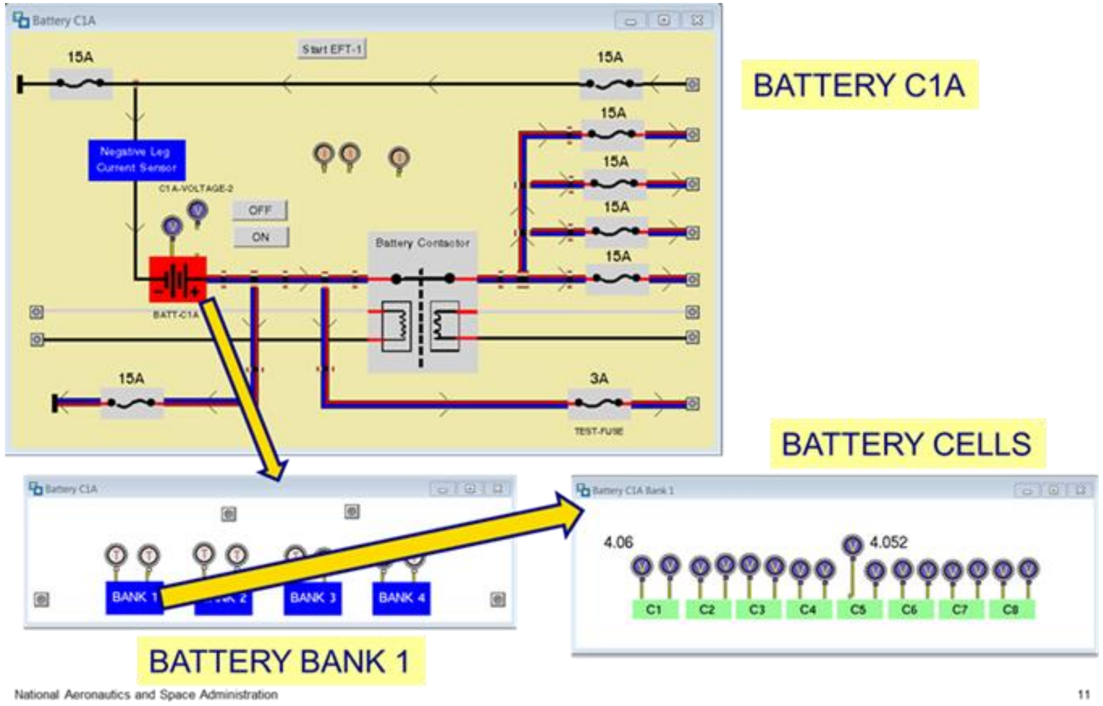


Cryogenic flow system domain



Schematic level system structure showing details that include objects, their connectivity, and the ability to implement reasoning including all objects

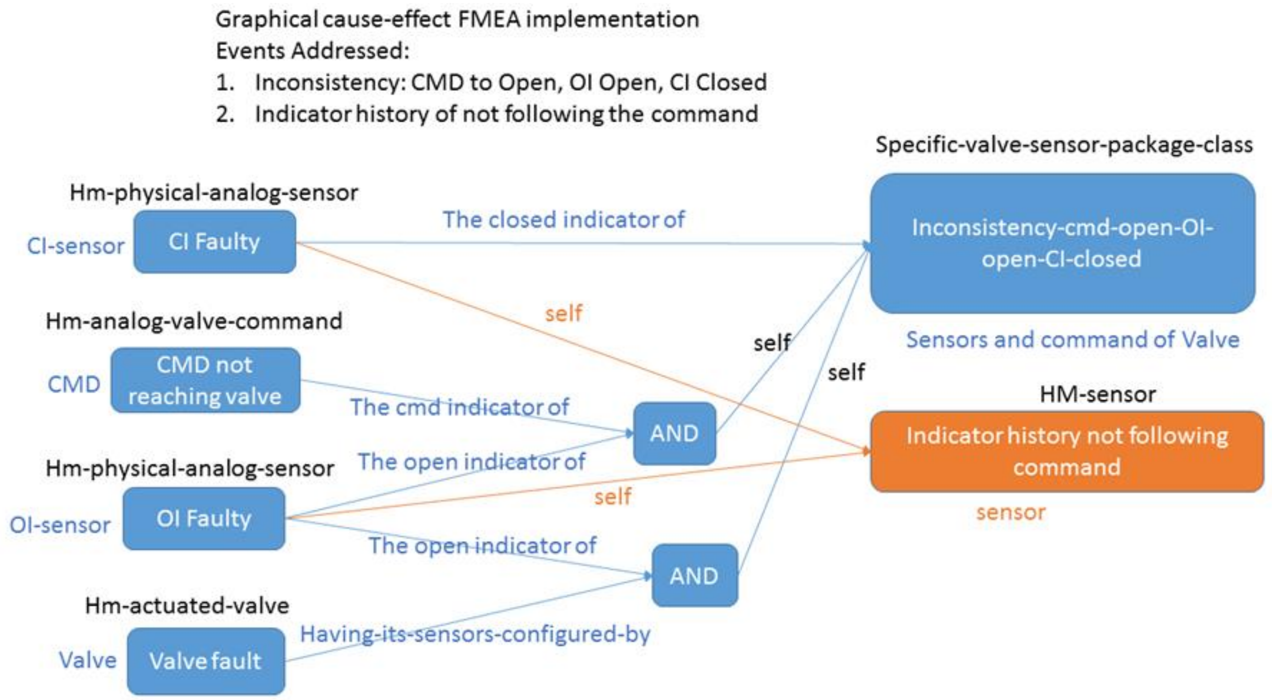
- Reasoning may say " … if the voltage of any cell of any battery is less than x then invoke low-voltage-battery-cell () …"



Relationships are created automatically when the schematic is built

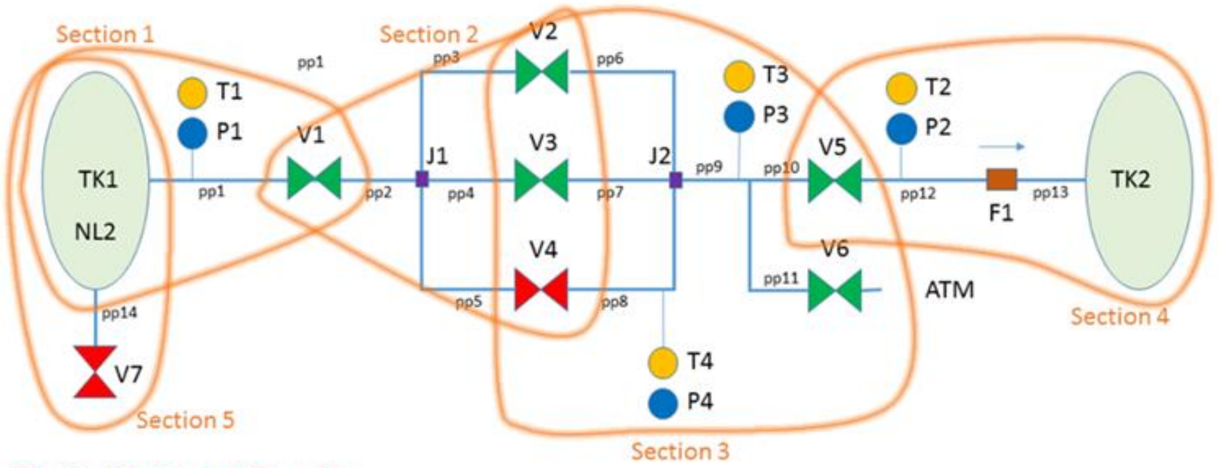
Reasoning with events associated with classes of objects to infer anomaly conditions for an instrumented valve

- The diagram shows anomalies that may cause a valve not to open when it is commanded to open
- The analysis applies to classes of sensors associated with classes of valves



Flow section concept and use of the concept for a behavior model

- Flow-section definition: collection of objects bounded by stops or flow control objects (e.g. valves) that could be considered as one piece and are therefore filled with the same commodity at any given time, and the expectation is that properties of the commodity (e.g. temperature of a fluid) will be the same throughout the flow-section



Flow Section Concept Examples
Section 1: Part of (pp1, T1, P1) Bounding: (TK1, V1)
Section 2: Part of (pp2, J1, pp3, pp4, pp5) Bounding: (V1, V2, V3, V4)
Section 3: Part of (pp6, pp7, pp8, J2, pp9, T3, P3, T4, P4, pp10, pp11) Bounding: (V2, V3, V4, V5, V6)
Section 4: Part of (T2, P2, pp12, F1, pp13) Bounding: (V5, TK2)

Model: "Throughout each flow section there is one commodity, and some physical variables of the commodity such as temperature should be similar throughout the piping section"
Similar value set 1: T3, T4

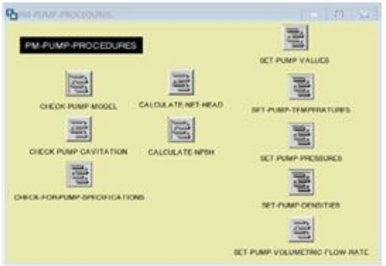
Model-Based Systems Engineering
Real-Time Operations and Autonomy

Integration of rotary pump behavior with procedural code (in structured natural language) and structure information

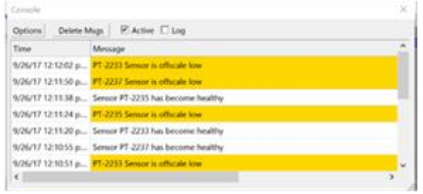
Definition of a class for a rotary pump

For every rotary pump instance

Calculating pump efficiency

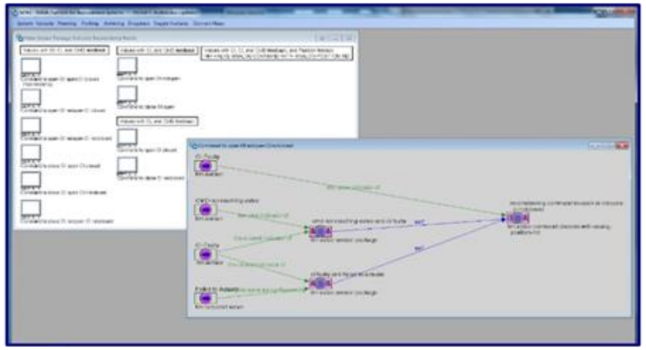


Checking consistency of models against real-time data



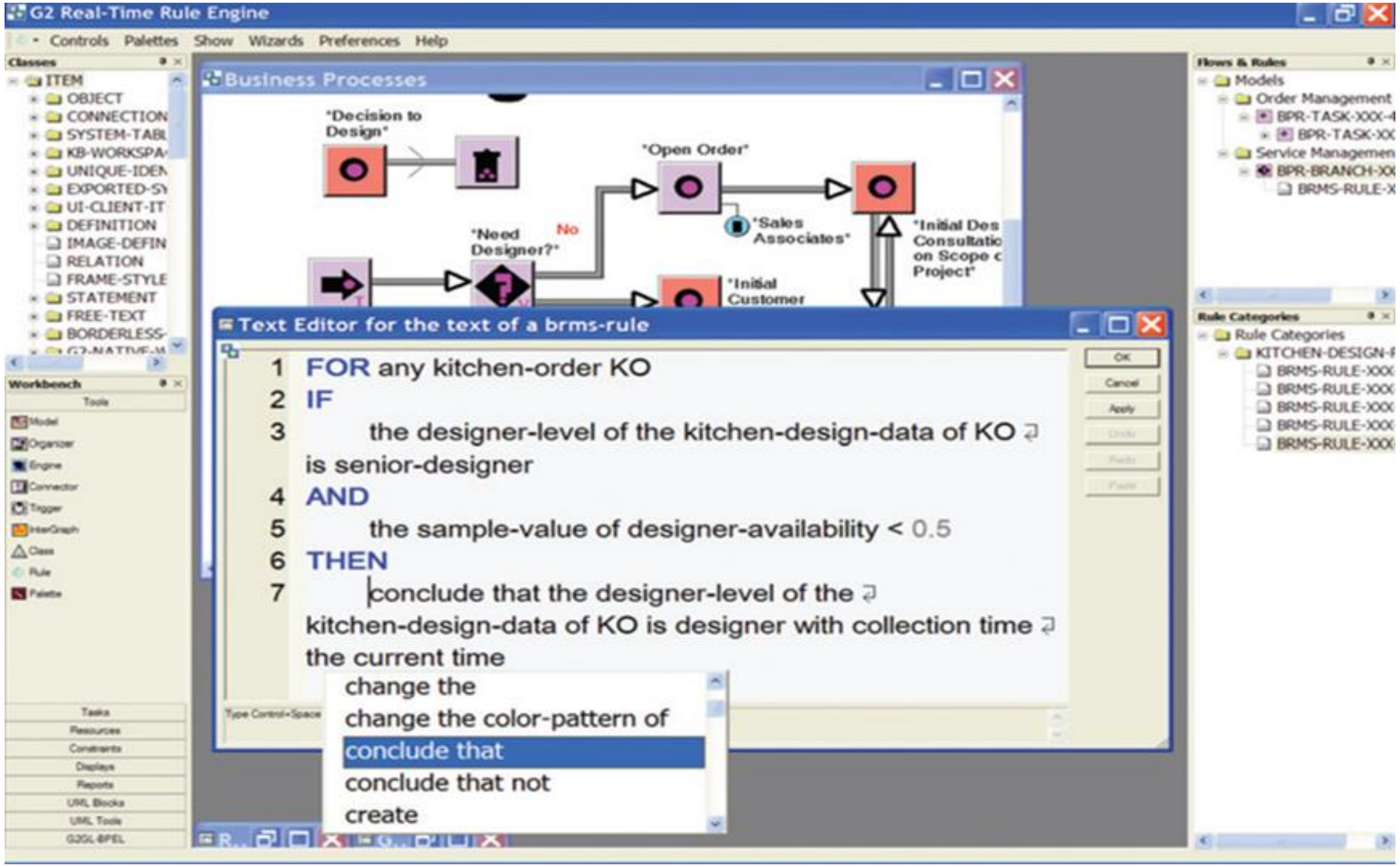
Example, of an inconsistency of a sensor, and this inconsistency is triggered by NPAS models being applied; physics, or other models in NPAS are detecting events; these events are used in root cause trees for diagnosis (as shown in example below)

Execution of models (blocks above) generate truth values on events that form part of cause-effect trees (on the right), corresponding to FMEA and/or used for other reasoning as needed by autonomous operations activities



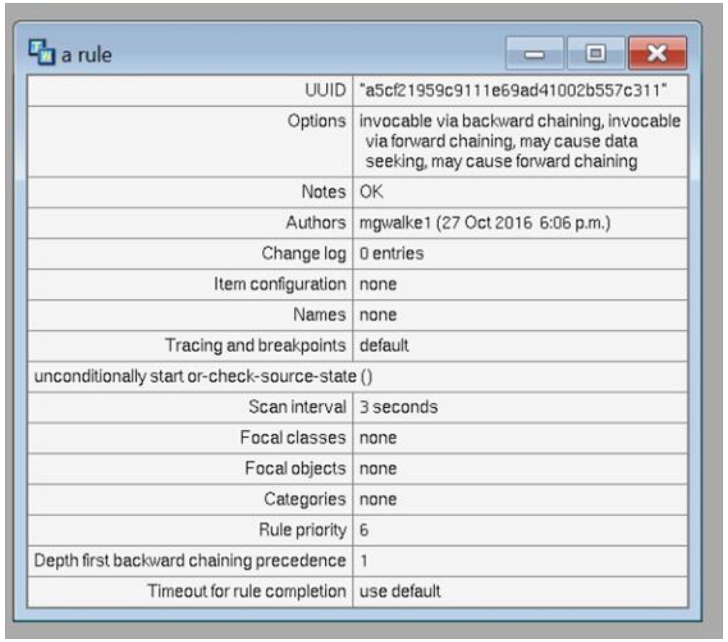
Root cause tree associated with a valve assessment model

G2 procedural language (structured natural language)



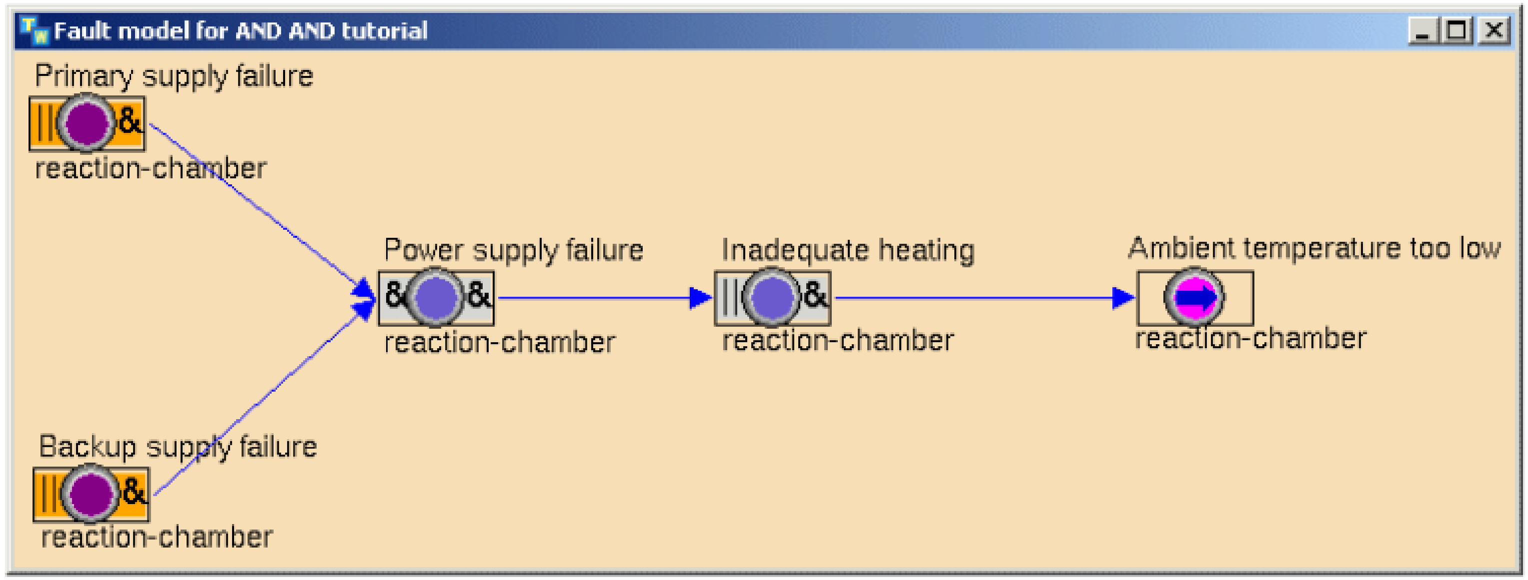
Model-Based Systems Engineering
Real-Time Operations and Autonomy

Rules and cause-effect diagrams

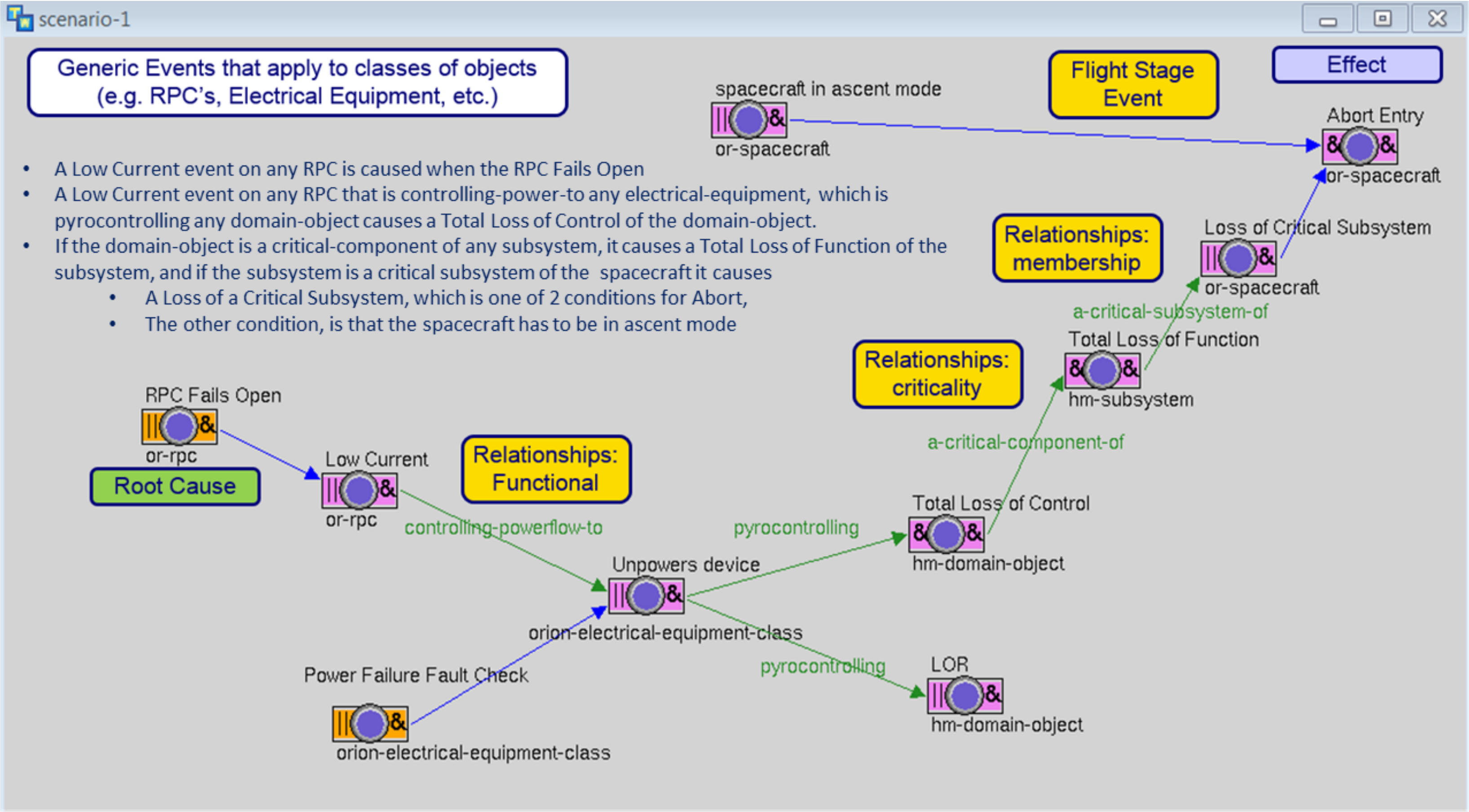


a rule

UUID	"a5cf21959c9111e69ad41002b557c311"
Options	invocable via backward chaining, invocable via forward chaining, may cause data seeking, may cause forward chaining
Notes	OK
Authors	mgwalke1 (27 Oct 2016 6:06 p.m.)
Change log	0 entries
Item configuration	none
Names	none
Tracing and breakpoints	default
unconditionally start or-check-source-state ()	
Scan interval	3 seconds
Focal classes	none
Focal objects	none
Categories	none
Rule priority	6
Depth first backward chaining precedence	1
Timeout for rule completion	use default



NPAS Cause-effect diagrams



The system can send an event to any instance represented by a node. Causes and consequences will propagate upstream and downstream to assert status of all events (true, false, suspect).

G2/NPAS software environment capabilities for integrated system model and real-time operations

- Temporal knowledge: *"if the rate of change of the pressure over the last 45 seconds …"*
- Dynamic Models: *"d/dt (the concentration of oxygen in the habitat module) = …"*
- Object Oriented knowledge representation: *"if the pressure of any habitat … and the rate of change of the temperature of the habitat …"*
- Connectivity of objects: *"if the pressure of any pressurizer PR < 150 and the temperature of PR > 450 then invoke emergency rules for the reactor connected to PR"*
- Generic knowledge: applied to classes of objects
- Procedural knowledge:

"Do in parallel [until one completes]
<statement> [;<statement>] … [;]
And
do
for T = each tank
do
for V = each valve connected to T
… etc."

- Reasoning using "meta-knowledge":

Model-Based Systems Engineering, Real-Time Operations and Autonomy

G2/NPAS software environment capabilities for integrated system model and real-time operations

- Reasoning using deep knowledge: This is a model-referenced methodology that embodies an accrual of intelligence.
- Finding the best answer in a fixed time: " *if (the first of the following that has a value, …) … then …"*
- Real-time truth maintenance: as time passes, conclusions need to have a validity time limit, even without new information/knowledge being added
- Scheduling G2 code tasks and priorities: tasks (procedures, methods, rules) are scheduled with priorities
- Structured natural language: an editor is used for creating and editing objects, definitions of rules, and other forms of knowledge.
- Defining objects: there is a structured schema to define objects
- Expanding the domain: Cloning and connecting enables rapid expansion of the domain models
- Knowledge libraries: may include tens of thousands of objects, rules, procedures, models, and other items that are re-usable
- Inspecting knowledge: knowledge contained in the model can be inspected and displayed in a temporary workspace " *Show the object hierarchy of pump"* " *Show on a workspace every rule containing the word pump"*
- Dynamic simulation testing: simulation capability can be quickly built to test behavior throughout the knowledge system.
- Interactive knowledge capture: G2 is an interactive development environment which encompasses the use of graphics and structured natural language, as well as animated graphics

Model-Based Systems Engineering, Real-Time Operations and Autonomy

Conclusions and Recommendations

- SysML modeling platforms (e.g. MagicDraw, Rational Rhapsody) enable creation of comprehensive systems models in the SysML language
- SysML modeling platforms enable application software tools for analysis during the project/system development process (e.g. a tool that determines mass and center of mass)
- SysML language is not fully satisfactory for modeling precise time requirements
- G2/NPAS graphical and structured natured language enable creation of systems' models beyond the level of complexity enabled by SysML
- G2/NPAS augment content of SysML structure and behavior frames
  - Include ISHM and Autonomy behavior frames
  - Include structure models to schematic detail fidelity
- NPAS application knowledge models are used in real-time operations: ISHM, autonomous operations, hierarchical distributed autonomous systems
- G2 can save and load XML code, thus content of SysML models could be ingested as part of NPAS models. This capability needs investment in developing a more specific domain ontology and language that can serve to further standardize both SysML and NPAS

www.ascend.events