Safety Guided Design Using STPA and Model Based System Engineering (MBSTPA)

Mike Hurley
Head of Product Safety

Jim Wankel
Systems Engineer

BAE Systems Electronic Systems

© 2019 BAE Systems – Mike Hurley, Jim Wankel

Approved for Public Release/Not Export Controlled per ES-NHQ-031819-0079

Introduction

- In Defense and other domains, customers have traditionally expected Safety analyses like FMECAs and Fault Trees that use abstractions of the design, requiring details that only exist after critical decisions are made
  - This is Safety Assessment, not Safety Design
- The adoption of Digital Engineering and Model Based Design approaches by the Department of Defense presents an opportunity to resolve the paradox: designing for Safety in the same Model-based environment as the Systems Designers eliminates the abstraction and left-shifts attention to safety
- Systems Theoretic Process Analysis (STPA) provides the means for a Systems Engineering approach to safety in a Model Based Systems Engineering (MBSE) environment - MBSTPA
  - STPA enables derivation of Safety Constraints for identified hazards
  - A Model Based Approach allows use of behavioral models early in the system's design to evaluate the system's response to unsafe control actions
    - Ensure the system will not behave in an unsafe way given the receipt UCAs

This Presentation Will Share The Implementation at BAE Systems of Design For Safety In A Model-Based Design Environment - MBSTPA

© 2019 BAE Systems – Mike Hurley, Jim Wankel

Approved for Public Release/Not Export Controlled per ES-NHQ-031819-0079

The Approach

- Apply STPA to define the Losses, Hazards, Unsafe Control Actions, and Loss Scenarios leading to definition of Safety Constraints as Design Requirements
- Design a State Machine in which the States represent performing the various required Functions or Actions of the System being designed, and where the transitions among states are governed by the Safety Constraints
- Capture the State Machine as a Behavioral Model in a SysML based tool, enabling simulation of the model to verify its behavior conforms to the Safety Constraints, under varying conditions/values of the variables involved in the Controller's Process Model

Teaching Example: Train Door System (TDS)

- The training material uses the Train Door System (TDS) as an example
- The primary Loss to be avoided here is injury or death to passengers; from a Mission perspective an additional loss must be considered in terms of the failure to convey passengers to their destination in a timely manner
- The STPA analysis is available in the STPA Primer and is not repeated here; the derived Safety Requirements are shown below

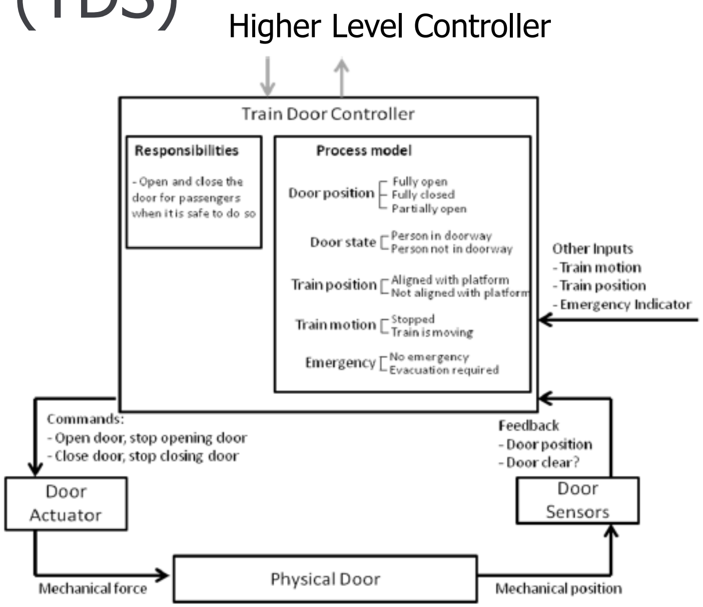


Figure 2.12. Simple Safety Control Loop for a Train Door Controller

Defining Safety Constraints:

SC-1: Controller shall not open Door while train is moving, only when stopped [H-1]

SC-2: Controller shall not close Door while a passenger is in the doorway [H-2]

SC-3: Controller shall open Door in an emergency (after train has stopped per SC-1) [H-3]

SC-4: Controller shall open Door when Train is stopped in Station [H-4]

Requirements Implied/derived from System Level Functional Causal Scenarios:

SC-5: Controller shall report the state of the Door to the next-higher level of control [H-1]*

SC-2.1 Controller shall allow time for passengers to exit & enter before attempting to close the door

SC-2.2 The Controller shall warn passengers when the door is about to open [H-5] or close [H-2]

SC-3.1 When the Train is stopped and not in a station and no emergency exists the Controller shall not open the door [H-5]

*TDS Controller does not control Train motion

© 2019 BAE Systems – Mike Hurley, Jim Wankel

Approved for Public Release/Not Export Controlled per ES-NHQ-031819-0079

MBSE at BAE Systems Electronic Systems

- BAE Systems ES is employing a tool based on the Systems Modeling Language (SysML), an Object Oriented language used for Modeling & Simulation for Systems Design
- It supports modeling of systems behavior in various ways including State Machines
  - Below is a "mapping" of key elements of STPA to SysML features

STPA	SYSML	Notes re Behavioral Modeling
System / Process	A Domain containing a collection of objects of different Classes with Classifier Behaviors defined	The model must include all objects that can contribute to a Loss either directly or through involvement in decision making by the Controller (and the Controller itself)
Unsafe Control Action (UCA)	A response by the Model causing transition to a Hazardous State	STPA is applied to derive the Safety Constraints needed to mitigate UCAs in the form of constraints on transitions between states.
Controller's Process Model	Controller's Classifier Behavior, consisting of the derived rules for state transitions (reflecting mission/performance requirements governed by Safety Constraints)	In this approach, States represent performing/not performing a Function or Action, or the resulting effects of same (E.g. Opening or Closing the Train Door).
Sensor / Feedback	A source of information (signal or variable) used by the Controller's Classifier Behavior in determining when State Transitions should or should not occur	The Train Door System will need at least two internal Sensors (Door Position and Door Clear e.g. when (In_Door == false)

Other Inputs	Can be modelled as Blocks external to the Train Door System whose states are used in the Process Model to control state transitions	Train State: Moving, Stopped;
		Train Position: In Station, Not In Station
		Emergency Status: Emergency, No Emergency;

© 2019 BAE Systems – Mike Hurley, Jim Wankel

BAE SYSTEMS

Using AND-OR Tables To Define Rules For State Transitions

- Controller Process Model States: Open the Door, Close the Door
- List of possible inputs (in this case, “Environment Variables” not under the control of this system):
  - Passenger in door/not in door
  - Train moving/stopped
  - Train in station/not in station
  - Emergency/no emergency
- Evaluate these variables for their effect (constraint) on State Transitions

	Transition:	
	Door Open to Door Closed	OR
AND	Passenger not in door	T
	Train stopped	T
	Train in station	T
	No Emergency	T

	Transition:		
	Door Closed to Door Open	OR	
AND	Passenger not in door	.	.
	Train stopped	T	T
	Train in station	T	.
	Emergency	.	T

*See: “Completeness and Consistency in Hierarchical State Machine Requirements”, Heimdahl, Mats P.E., Leveson, Nancy G., first published in: international conference on software engineering · 1996

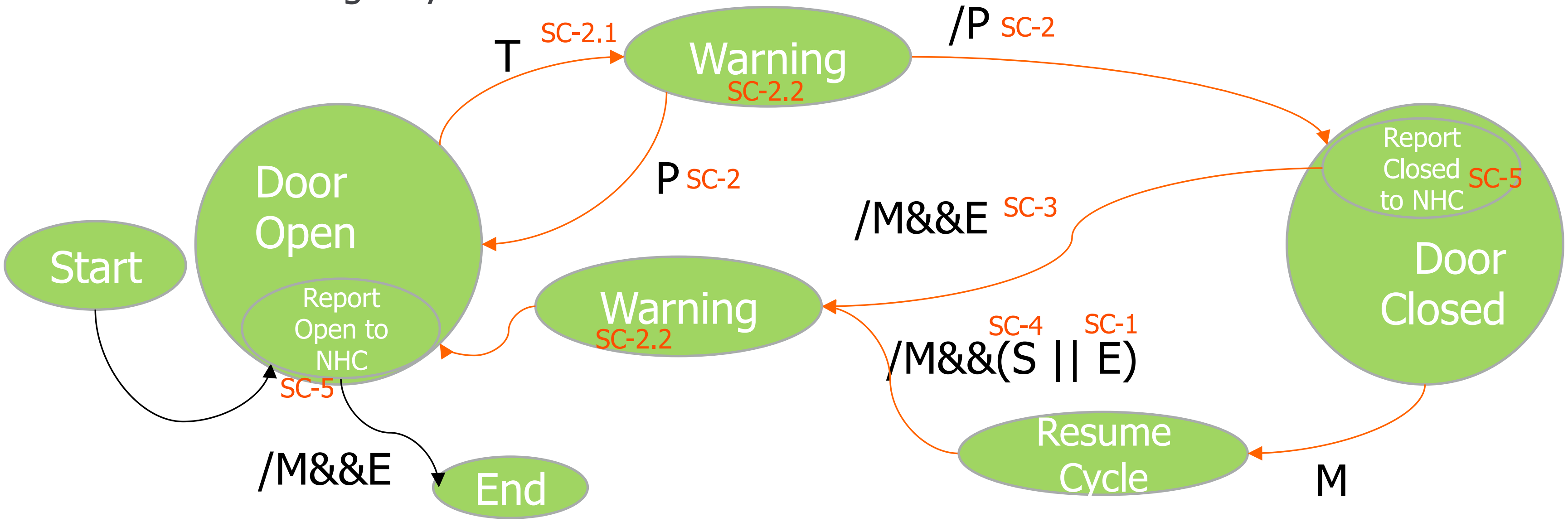
© 2019 BAE Systems – Mike Hurley, Jim Wankel

Approved for Public Release/Not Export Controlled per ES-NHQ-031819-0079

States and Transitions As A Function Of Control Variables

- Control Variables Notation:
  - T = wait interval expired
  - P = Passenger in door; /P = Passenger NOT In Door
  - M = Moving; /M = stopped
  - S = Train In Station
  - E = Emergency

&& = AND
|| = OR



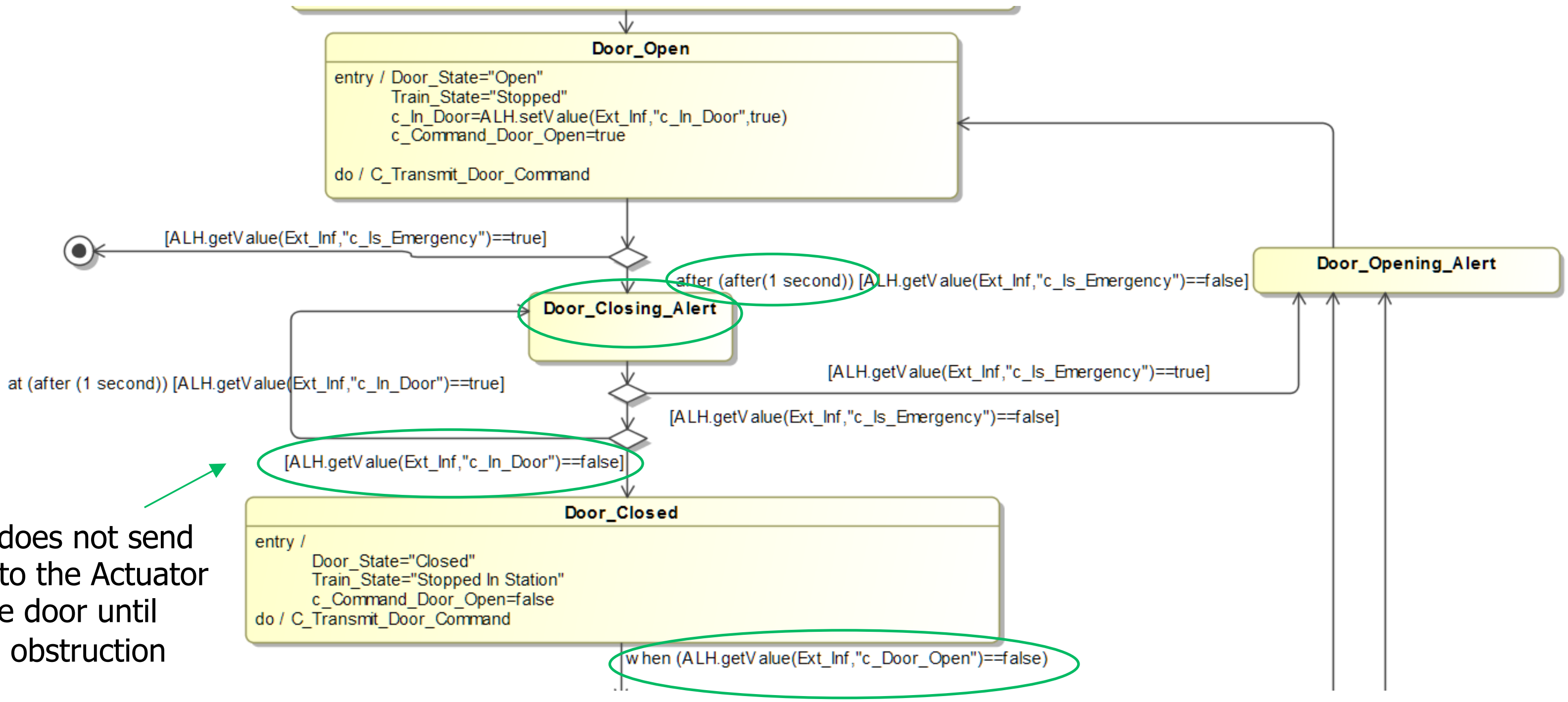
Without this, the conditions required to open the door would be immediately satisfied

© 2019 BAE Systems – Mike Hurley, Jim Wankel

BAE SYSTEMS

Approved for Public Release/Not Export Controlled per ES-NHQ-031819-0079

State Machine (Partial) Implements Safety Constraints



Door_Open

entry / Door_State="Open"
Train_State="Stopped"
c_In_Door=ALH.setValue(Ext_Inf,"c_In_Door",true)
c_Command_Door_Open=true

do / C_Transmit_Door_Command

[ALH.getValue(Ext_Inf,"c_Is_Emergency")==true]

after (after(1 second)) [ALH.getValue(Ext_Inf,"c_Is_Emergency")==false]

Door_Opening_Alert

Door_Closing_Alert

at (after (1 second)) [ALH.getValue(Ext_Inf,"c_In_Door")==true]

[ALH.getValue(Ext_Inf,"c_Is_Emergency")==true]

[ALH.getValue(Ext_Inf,"c_Is_Emergency")==false]

[ALH.getValue(Ext_Inf,"c_In_Door")==false]

Door_Closed

entry /
Door_State="Closed"
Train_State="Stopped In Station"
c_Command_Door_Open=false
do / C_Transmit_Door_Command

when (ALH.getValue(Ext_Inf,"c_Door_Open")==false)

Controller does not send command to the Actuator to close the door until there is no obstruction sensed

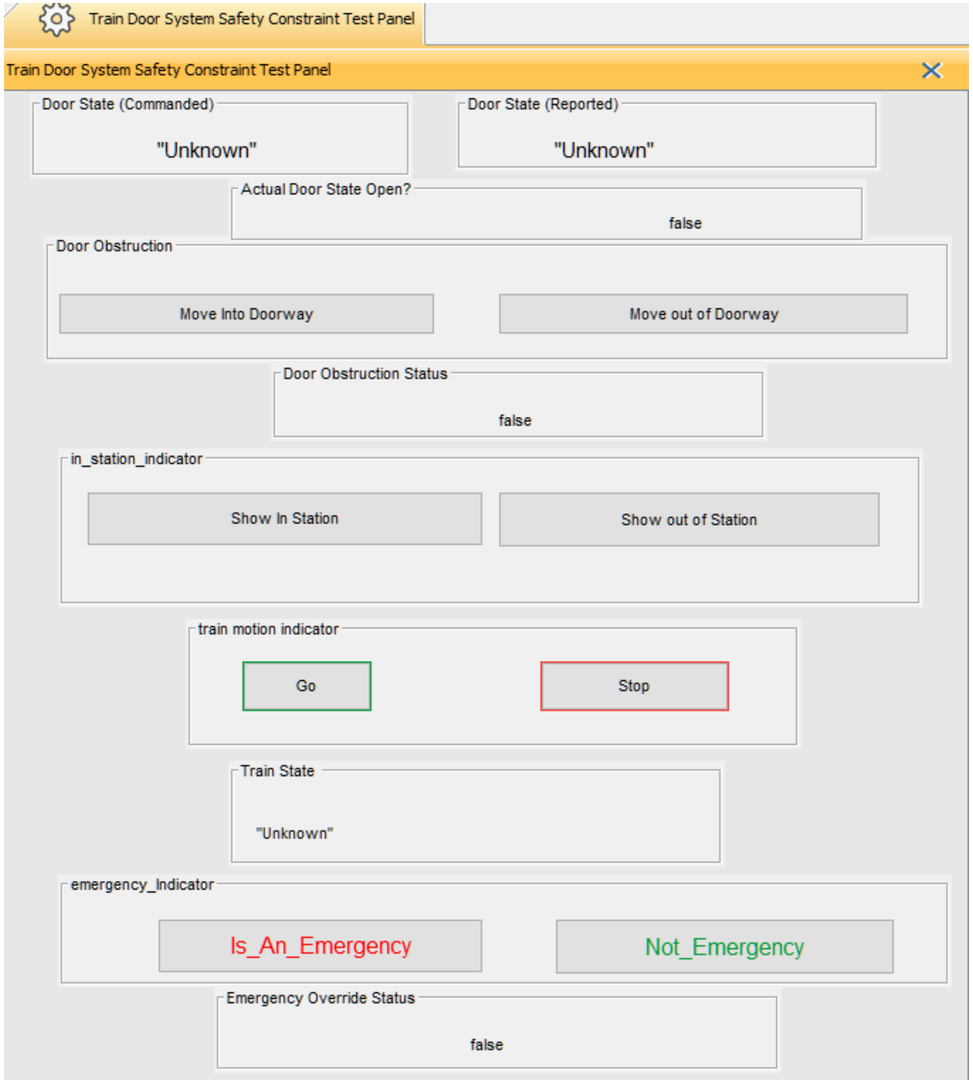
Controller does not advance until the door is actually closed

© 2019 BAE Systems – Mike Hurley, Jim Wankel

Approved for Public Release/Not Export Controlled per ES-NHQ-031819-0079

Testing The Model For Adherence To Safety Constraints

- A Graphical User Interface (GUI) can be used to control the values of the “other inputs” to verify that the Controller’s Process Model behaves within the Safety Constraints
- Various approaches to testing are possible
  - Path testing (not feasible for very large systems unless broken down into smaller individually tested pieces
  - AND-OR Table inversion
  - Use Case testing based on operational scenarios (typically part of performance specification testing)



Door State (Commanded): "Unknown"

Door State (Reported): "Unknown"

Actual Door State Open?: false

Door Obstruction Status: false

Train State: "Unknown"

Emergency Override Status: false

© 2019 BAE Systems – Mike Hurley, Jim Wankel

Summary and Next Steps

- By applying this approach during the Modeling stage of the design process, typically during early Solution Development, Safety can influence key decisions that system designers would have difficulty revising later
- Both Safety Analysts and Systems Modelers are involved
  - Working together on the same model in the same environment - not on separate/different representations of the design - so that Safety is not an (attempted) add-on but rather designed in
- Training in process for pilot projects
- The Train Door System model developed in the SysML tool provides a demonstrator of the approach for use in hands-on teaching of STPA

We Influence the Design For Safety From The Earliest Stages Of the Systems Engineering Process

© 2019 BAE Systems – Mike Hurley, Jim Wankel

Approved for Public Release/Not Export Controlled per ES-NHQ-031819-0079

Thank you

Questions?

© 2019 BAE Systems – Mike Hurley, Jim Wankel

Approved for Public Release/Not Export Controlled per ES-NHQ-031819-0079