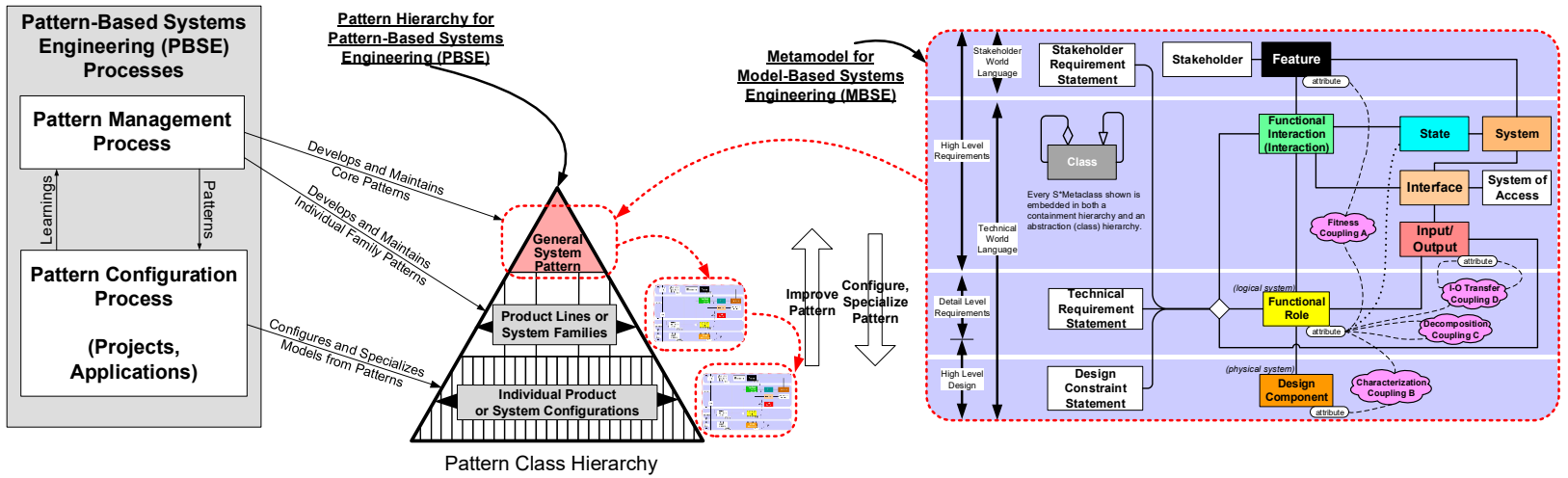
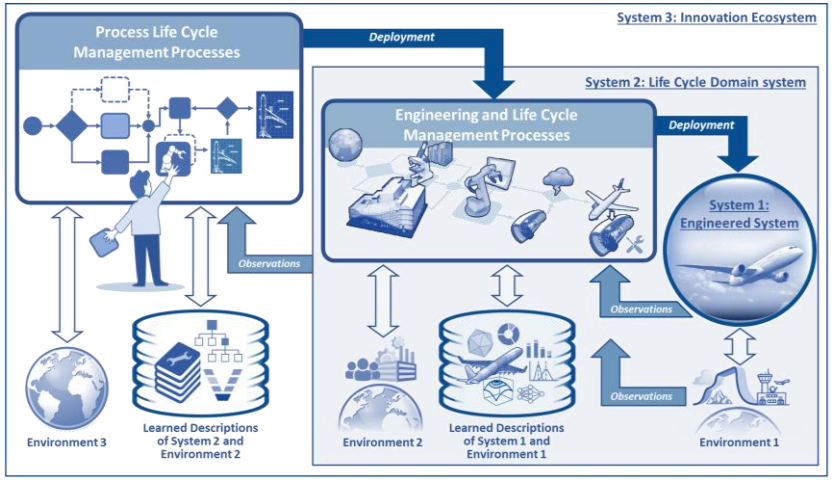
ICTT System Sciences®
Understand your systems.

<<Company/Org. Logo>>

Phase I Session 3

Introduction to S*Models, S*Metamodel, S*Patterns, Mappings to Languages & Tools, S*Configuration Wizard, ASELCM S*Pattern

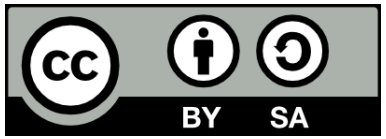




schindel@ictt.com
lewis@ictt.com

V1.2.13

Copyright 2025, System Sciences, LLC



Licensed under a Creative Commons
Attribution Share Alike-License CC BY SA International 4.0

License Link: https://creativecommons.org/licenses/by-sa/4.0/legalcode

Uses are permitted under this license without further permission from the copyright owner, provided each use (1) is clearly marked to attribute the underlying work to "S*Patterns Community", (2) provides a link to the CC BY SA license, (3) indicates if changes were made, (4) does not suggest the licensor endorses the user or use, (5) does not apply legal terms or technological measures that legally restrict others from doing anything the license permits, and (6) if you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

Permissions beyond the scope of this license are administered through contacting:

Corporate Officer
ICTT System Sciences
378 South Airport Street
Terre Haute, IN 47803
812-232-2208

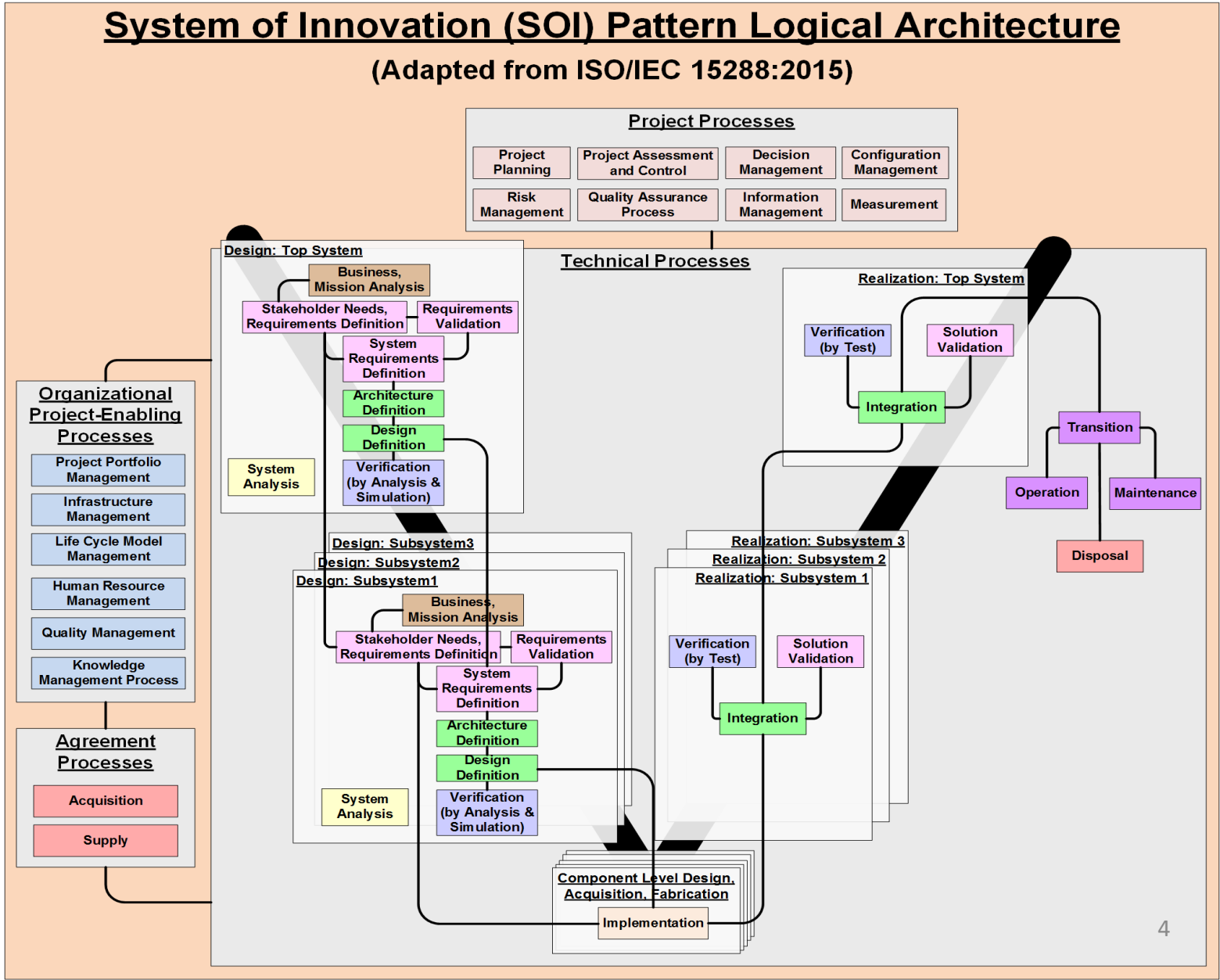
Systematica is a registered trademark of System Sciences, LLC.

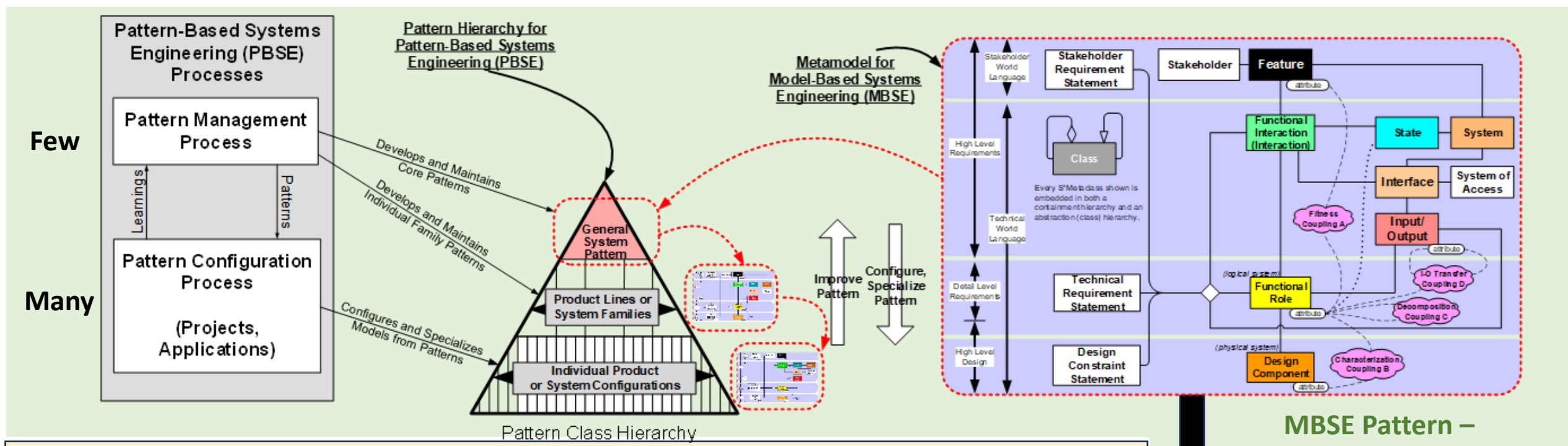
Contents

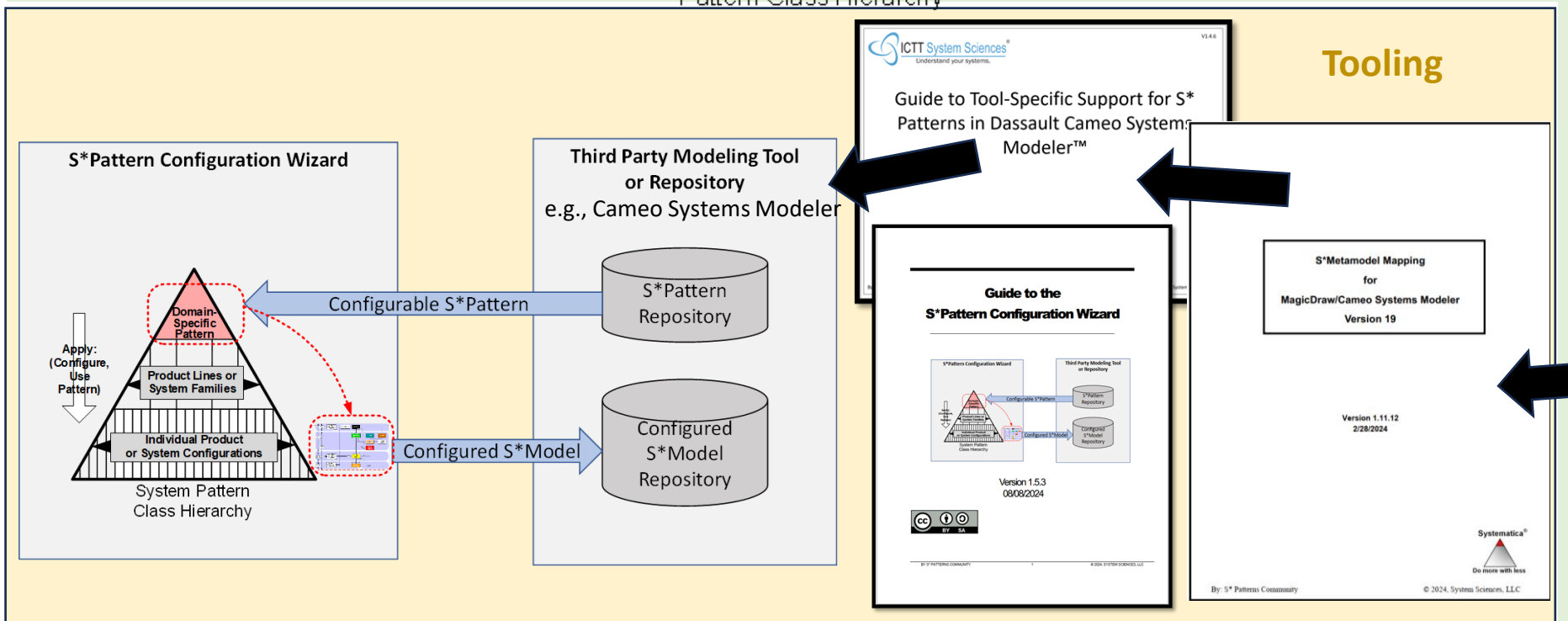
- Recap and homework:
  - Session 1, 2 Take-aways, discussion, questions
  - Session 2 Homework
- Phase I Session 3 Class Content
- Session 3 homework assignment
- References

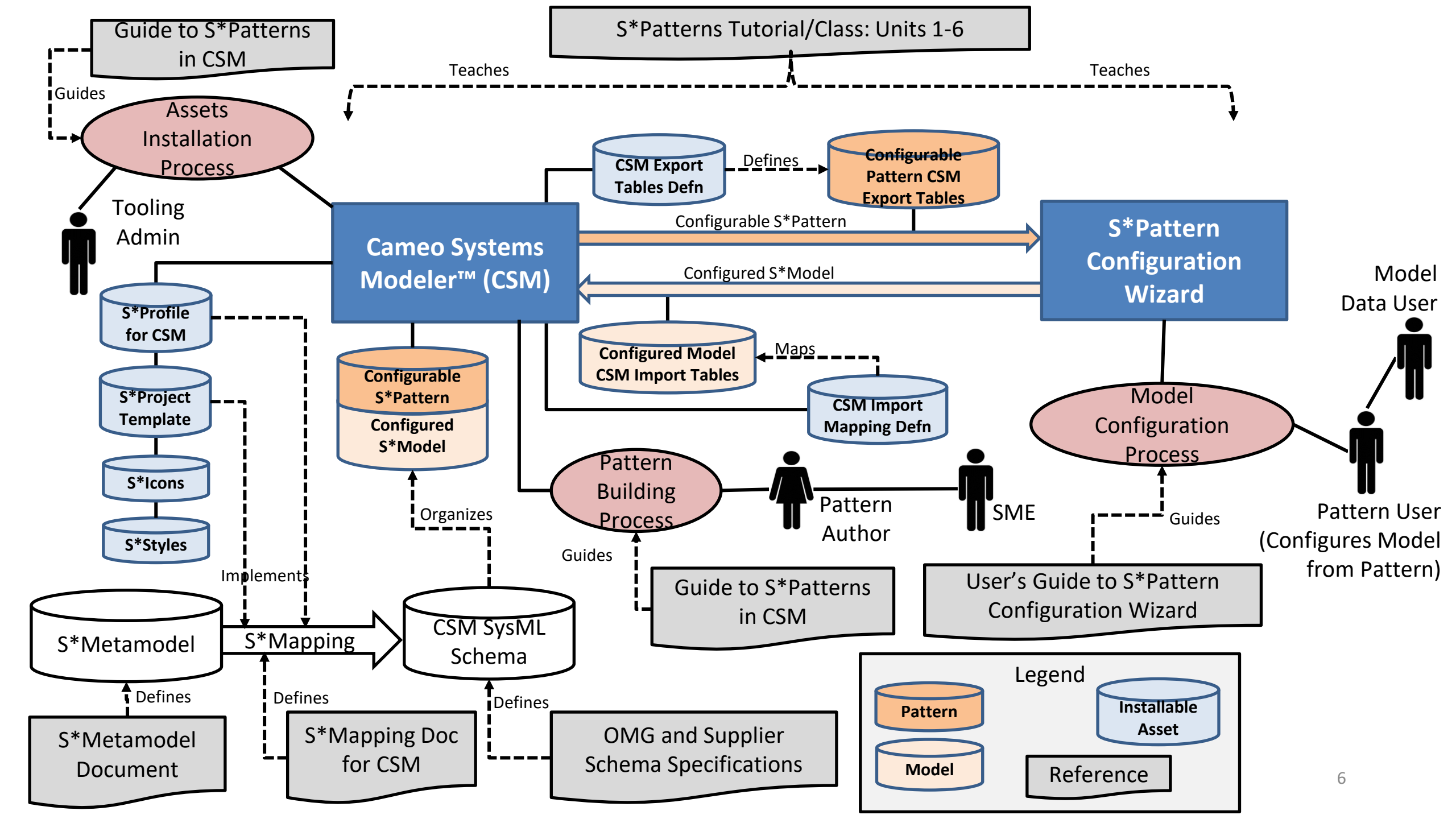
Recap of earlier sessions: Take-aways impacting actual practices

1. What is the "**Systems Engineering Vee Diagram**"? Which of its processes can be supported by MBSE? How are they related to **ISO 15288**? **ISO 13485**?
2. What are priority SE processes and information for your organization, based on targeted markets, customers, positioning?
3. Which will involve your work?




Few
Many
Pattern-Based Systems Engineering (PBSE) Processes
Pattern Management Process
Pattern Configuration Process
(Projects, Applications)
Learnings
Patterns
Develops and Maintains Core Patterns
Develops and Maintains Individual Family Patterns
Configures and Specializes Models from Patterns
Pattern Hierarchy for Pattern-Based Systems Engineering (PBSE)
General System Pattern
Product Lines or System Families
Individual Product or System Configurations
Improve Pattern
Configure, Specialize Pattern
Pattern Class Hierarchy
Metamodel for Model-Based Systems Engineering (MBSE)
Stakeholder World Language
Stakeholder Requirement Statement
Stakeholder
Feature
High Level Requirements
Class
Every S*Metaclass shown is embedded in both a containment hierarchy and an abstraction (class) hierarchy.
Technical World Language
Functional Interaction (Interaction)
State
System
Interface
System of Access
Fitness Coupling A
Input/ Output
I-O Transfer Coupling D
Decomposition Coupling C
Detail Level Requirements
Technical Requirement Statement
(logical system)
Functional Role
High Level Design
Design Constraint Statement
(physical system)
Design Component
Characterization Coupling B
MBSE Pattern – MBSE Model Concept


S*Pattern Configuration Wizard
Third Party Modeling Tool or Repository
e.g., Cameo Systems Modeler
Domain-Specific Pattern
Product Lines or System Families
Individual Product or System Configurations
Apply: (Configure, Use Pattern)
System Pattern Class Hierarchy
Configurable S*Pattern
Configured S*Model
S*Pattern Repository
Configured S*Model Repository
Tooling
Guide to Tool-Specific Support for S* Patterns in Dassault Cameo Systems Modeler™
Guide to the S*Pattern Configuration Wizard
Version 1.5.3 08/08/2024
S*Metamodel Mapping for MagicDraw/Cameo Systems Modeler Version 19
Version 1.11.12 2/28/2024
Systematica® Metamodel
Metamodel Version 8.0
2/15/2024


Guide to S*Patterns in CSM
S*Patterns Tutorial/Class: Units 1-6
Guides
Teaches
Teaches
Assets Installation Process
Tooling Admin
CSM Export Tables Defn
Defines
Configurable Pattern CSM Export Tables
Configurable S*Pattern
Configured S*Model
Cameo Systems Modeler™ (CSM)
S*Pattern Configuration Wizard
Model Data User
S*Profile for CSM
S*Project Template
S*Icons
S*Styles
Configurable S*Pattern
Configured S*Model
Configured Model CSM Import Tables
Maps
CSM Import Mapping Defn
Model Configuration Process
Pattern Building Process
Pattern Author
SME
Pattern User (Configures Model from Pattern)
Organizes
Guides
Guides
Implements
Guide to S*Patterns in CSM
User's Guide to S*Pattern Configuration Wizard
S*Metamodel
S*Mapping
CSM SysML Schema
Defines
Defines
Defines
S*Metamodel Document
S*Mapping Doc for CSM
OMG and Supplier Schema Specifications
Legend
Pattern
Installable Asset
Model
Reference

Recap of earlier sessions:
Take-aways impacting actual practices

1. What are **Systems**? Consistent use of System Domain Models;
2. What are **Interactions**? Consistent use of Interactions to describe behavior; no "naked" behavior;
3. **States**, state variables, finite state models: Consistent description of "When", temporal architecture;
4. **Input-Outputs**: Consistent representations of exchanged force, energy, material, information;
5. Stakeholder **Features**; Consistent representation of value across all stakeholders; trade space; Selection as the observable phenomenon describing value; all risk is risk to Features; use in FMEAs; use to drive PLE configuration variation; language of projects & C-Suite;
6. **Finding all the Interactions** by managing consistency of Interactions across Features, States, and External Interfaces/Actors;
7. **Functional Roles versus Design Components**, what do we mean by 'allocate"? by "physical"?
8. **Attributes**: parameterizing Features, Roles, Design Components;
9. **S*Metamodel**, mapped to languages and third-party tools, as a consistent point of reference across variable environments.



- Can a creative, talented team *align* its work?
- What should be *standard* about this, in your customer work?
- What should be *configurable* about this, in your customer work?



Recap of earlier sessions:
Take-aways impacting actual practices

1. What are **Architectural Relationships?** How can they be represented? What consistent uses shall we make of them?
2. Representing **model-based Requirement Statements** analogously to transfer functions. Consistent content across different systems, projects.
3. What are the underlying constructs of **Interfaces**? What is a **System of Access**? What consistent practices across different systems and projects?
4. Types of **Attribute Couplings** / parametric couplings: Fitness, Decomposition, Characterization, and Input-Output couplings. What consistent practices across different systems and projects?



- Can a creative, talented team *align* its work?
- What should be *standard* about this, in your customer work?
- What should be *configurable* about this, in your customer work?



Session 2 Homework Discussion

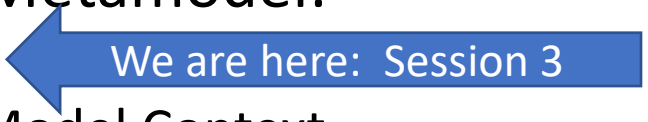
Class 2 homework assignment was:

- Add one or more external Interfaces, with Input-Outputs to the External Actors.
- Add some black box Requirements for the System of Interest, for the modeled Interactions.
- Add some actor Requirements (assumptions) for some external Actors, for the modeled Interactions.
- Decompose the system of interest to logical subsystems and add architectural relationships between them, creating a Logical Architecture model.
- Add at least one Design Component and allocate at least one logical subsystem to it.
- Add at least one attribute to:
  - A modeled Feature;
  - The System of Interest (black box);
  - A Subsystem;
  - A Design Component.
- Add at least one Attribute Coupling of each of the following
  - A Fitness Coupling
  - A Decomposition Coupling
  - A Characterization Coupling

Class 2 homework discussion

(Will appear in Session 3 class day slides.)

Phase I Class Sessions: Microstructure of Models and Patterns (6 Sessions, 3 hours each)

- S*Models, S*Metamodel:
  - Sessions 1-3 

  - In Example Model Context
- Formal mapping of S*Metamodel into Tooling
  - Session 4
- S*Patterns, Configuration: In Example Model Context
  - Session 5
- Introduction to 3rd Generation Pattern Configurator Technology
  - Session 6
- Homework Presentation and Feedback
  - Distributed over above sessions

S*Models, S*Metamodel: Session 3 of 3









- Models of Risk
- Failure Modes
- Failure Impacts
- Counter Requirements
- Generating FMEA versus FTA views
- Homework assignment 3
- Discussion, Q&A.



Models of risk: What do we need to represent?

- Traditional systems engineering example risk analysis representations are well-established, and can be found in:
  - Failure Modes and Effects Analysis (FMEA) or Failure Modes, Effects, and Criticality Analysis (FEMCA).
  - Special cases for risks of designs, risks of production and other processes, risks introduced by human operators (D-FMEA, P-FMEA, A-FMEA).
  - Fault Tree Analysis (FTA).
  - Preliminary Hazard Analysis (PHA).
  - Reliability Centered Maintenance (RCM) analysis.
  - Hazards and Operability Analysis (HAZOP).
  - Safety and Cybersecurity Analysis cases of the above.
- Some are used as part of review of a candidate system design, others to analyze an existing system, as in the case of RCM planning of preventive maintenance, etc.





A typical FMEA table (other columns also apply—probability, RPN, etc.)

DATE:	1/4/93		PAGE:	1	of	1	
PLANT:			SYSTEM:	Cooling Water Chlorination System			
ITEM:	Pressure Check Valve		REFERENCE:				
FAILURE MODE	**CAUSE(S)**	**OPERATIONAL MODE**	**FAILURE EFFECTS**	**FAILURE DETECTION METHOD**	**COMPENSATING PROVISIONS**	**SEVERITY CLASS**	**REMARKS/ ACTIONS**
Too much flow through valve	Both internal pressure valves fail open	Operation	Excessive chlorine flow to Tower Water Basin - high chlorine level in cooling water - potential for excessive corrosion in cooling water system	Rotameter Daily testing of cooling water chemistry	Relief valve on Pressure check valve outlet	III	None
Too little flow through valve	One or both internal pressure valves fail closed	Operation	No/low chlorine flow to Tower Water Basin - low chlorine level in cooling water - potential for excessive biological growth in cooling water system - reduction in heat transfer	Rotameter Daily testing of cooling water chemistry	Automatic temperature controllers at most heat exchangers	IV	None
Chlorine flow to environment	Internal relief valve sticks open Both internal pressure valves fail open and relief valve opens	Operation	Potential low chlorine flow to Tower Water Basin - see above Chlorine released to environment - potential personnel injury due to exposure	Distinctive odor	Pressure check valve located outdoors - unlikely to accumulate significant concentration	III	Action Item: Consider venting relief valve above ground level

Generating an FMEA from a Model: Example Output Is Very Similar to Traditional FMEA

Feature	Effect (Failure Impact)	Severity	Functional Failure (Counter Requirement)	Component	Failure Mode	Probability	Mitigation (Control)
Navigation Feature [GPS-based Location Sensing]	No Confidence in Displayed Position	Serious (4)	The system displays a location that is not accurate to 10 feet.	Vehicle ECM	Erratic ECM	0.0015	Nav Backup Mode: External Nav Module
Navigation Feature [GPS-based Location Sensing]	False Confidence in High Error Displayed Position	Critical (5)	The system displays a location confidence indicator that is not correct.	Vehicle ECM	Erratic ECM	0.0015	None
Navigation Feature [GPS-based Location Sensing]	No Displayed Location	Serious (4)	The system does not display the graphic map presentation.	Panel Display	Fractured Display	0.0003	Nav Backup Mode: External Nav Module

More on pattern-based FMEA in Session 5.

Models of risk: What do we need to represent?

- Most of the risk analysis methods have many years of history of practice before MBSE.
- For that reason, many of them are considered “specialty” skills or disciplines, and are sometimes practiced without being as connected as we’d like to the rest of engineering work:
  - That isolation/specialization can cause “misses” due to disconnection.
  - The related work can also feel very laborious and not very systematic.
- In this session, we will be looking at model-based approaches to generation of such risk analyses:
  - Where the risk models are integrated into the same model that represents other aspects of the managed system.
  - Specialized toolsets for these risk analyses can still be applied—but this approach will more deeply integrate the respective models used.
- These results can still appear visually very much like traditional risk analysis outputs--but generated by model leverage.
- Like all use of models, they are not a full substitute for human judgement—but leveraged to amplify human capabilities.
- A limited number of underlying concepts, translated to model-based form, are at the core of risk analysis . . .



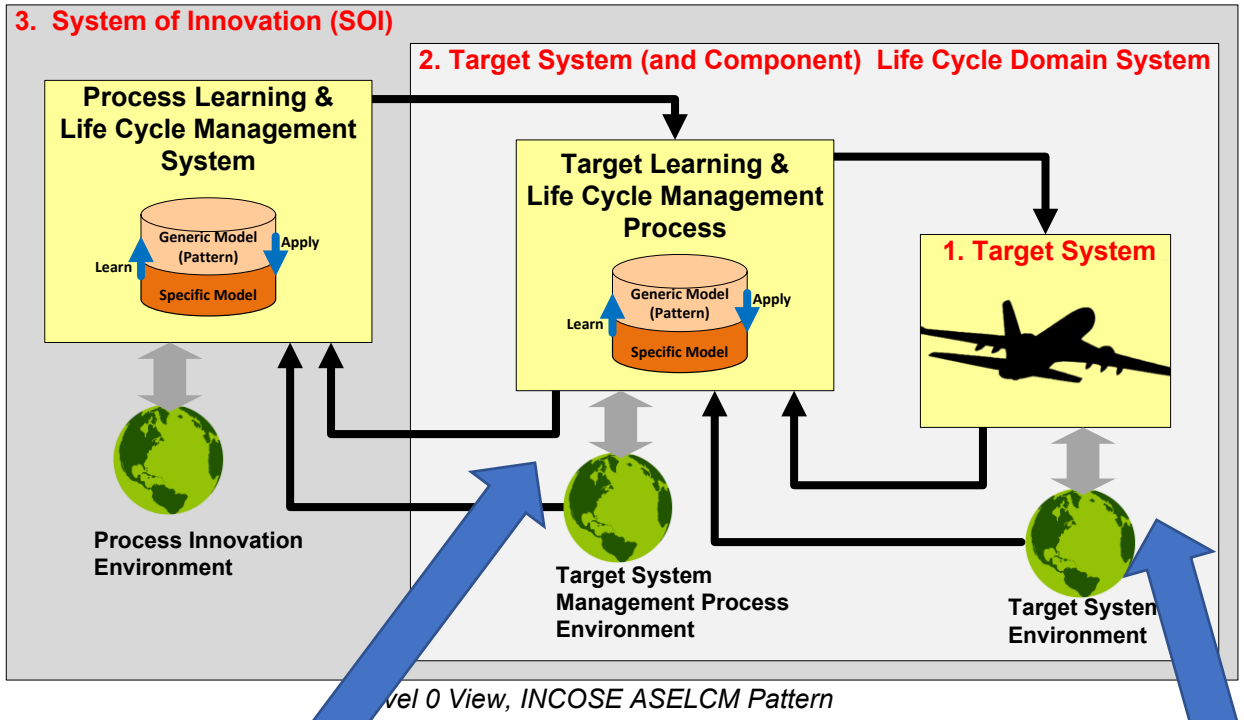


Models of risk: What do we need to represent?

- Most systems engineering relevant definitions of risk include these ideas:
  1. Potential, threat, or hazard of some future harm, injury, death, setback, economic or market loss, impacted assets, loss of advantage, etc.
     - (Sometimes we alternatively model "risk" of a more positive "opportunity" for gains of various kinds--not our current focus in this session.)
  2. One or more causal "modes" which can lead to the negative outcome.
  3. A degree of uncertainty as to that occurrence, whether expressed as likelihood, probability distribution, or otherwise.
  4. Some idea of the severity of impact of such a negative outcome, were it to occur, and what parties would be impacted.
  5. Ability (or inability) to detect whether the negative situation has occurred or is occurring.
  6. Ability (or inability) to mitigate or work around or otherwise tolerate the negative situation, were it to occur.
  7. In some cases, a time evolutionary aspect representing when in time these ideas might apply.
  8. Means of assembling a relatively complete (and potentially large) collection of the above possibilities, and weighing them relative to each other so that attention and resources can be assembled, focused, and allocated to their prevention, mitigation, planning, or other disposition. ("Risk Priority Number" (RPN) scores are common: RPN = Probability x Severity x Detectability.)

Risks in System 1 and System 2

INCOSE ASELCM Ecosystem Pattern:

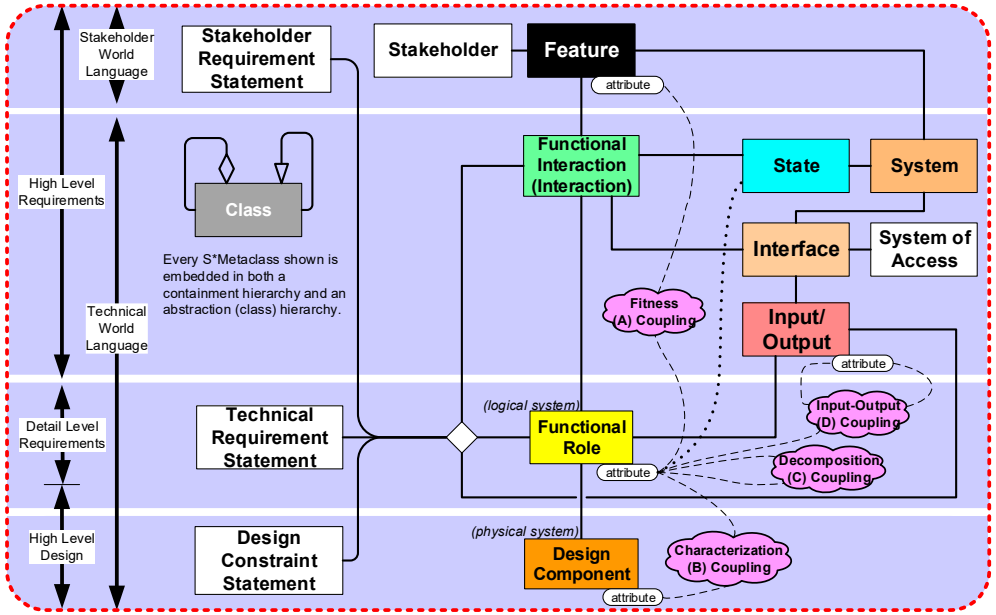


vel 0 View, INCOSE ASELCM Pattern

Life cycle management risks manifest in System 2 and its environment (e.g., Engineering, Production, Distribution, Sustainment, etc.).

Product risks manifest in System 1 and its environment.

S*Metamodel:

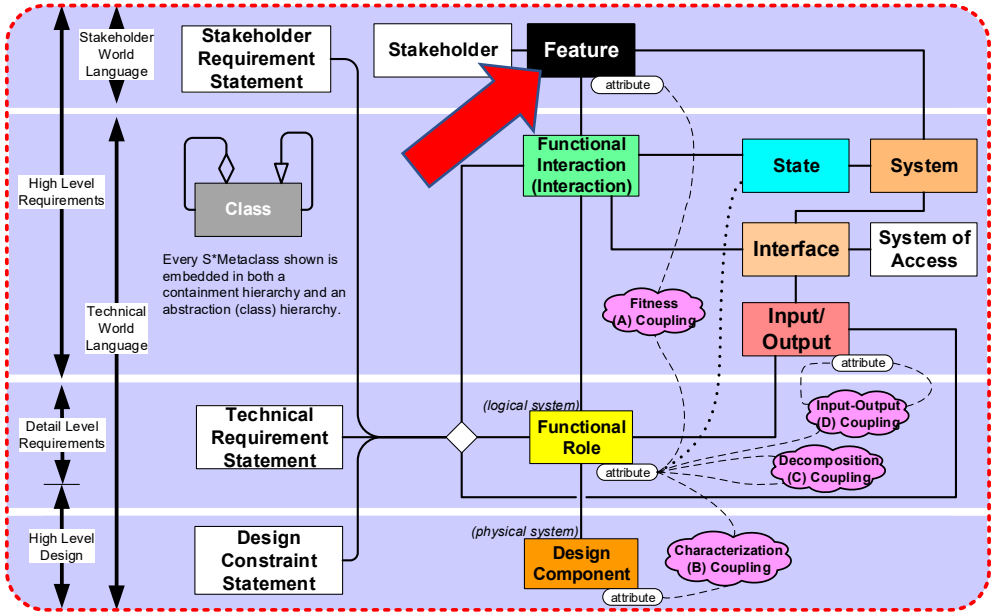


The same S*Metamodel represents modeled risks in both System 1 and System 2—but of course the systems involved have ***very*** different Interfaces, Features, States, Couplings, etc.

Different orders of reasoning apply to different types of risk analyses

- Failure Mode oriented analyses (i.e., FMEA and FMECA) appear to "start with" different possible departures from normal component level behavior (e.g., as in a broken part in a racing car steering system), called a "failure mode", and logically reason to what system level "Effects" would arise as a result (e.g., car goes out of lane and off road, resulting in loss of life and property).
- Fault Tree oriented analyses (i.e., FTA) appear to "start with" a possible harmful system level outcome (e.g., loss of life or health) and reason the other direction, back to all the different component level failure modes that could produce that result.
- The underlying information paths of these two cases is substantially the same, just traced in the opposite direction, "fanning out" while going upstream vs. downstream!
- Single fault assumption: These analyses often ignore multiple simultaneous failure modes—which are less likely but nevertheless a weakness of these analyses.
- Notice that the end-to-end reasoning process, whether traced by person or computer, requires awareness of the design as well as the requirements of a system.

Failure Impacts



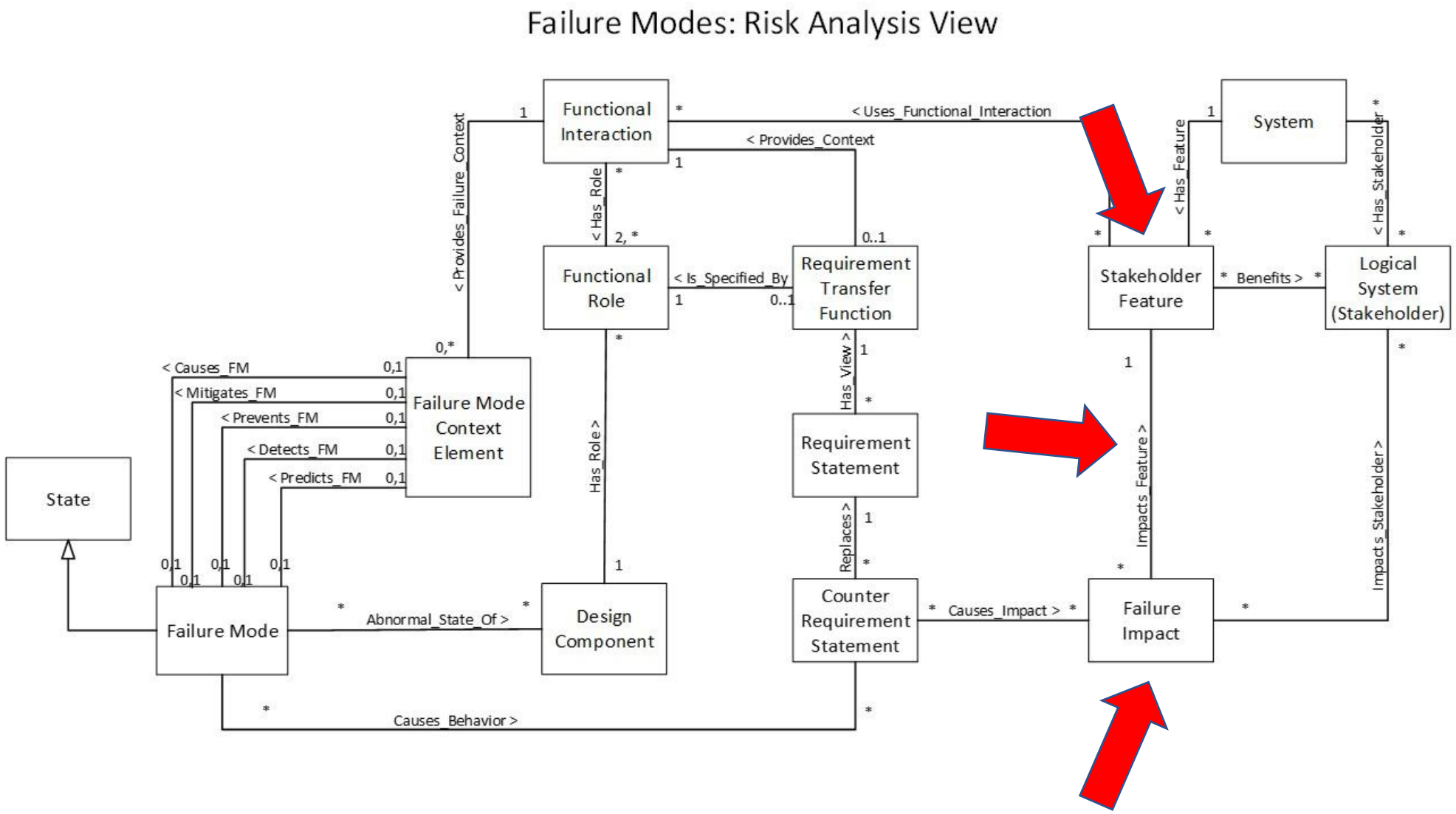
- A pleasant surprise is the discovery that a solid MBSE model of a system's Stakeholder S*Features brings you very close to having a model of all the possible negative Effects (the "E" part of FMEA) that a system may present:
  - Even if we don't yet know the system's design (!), a fully-modeled Feature Space is a direct path to the Failure Effect Space, because . . .
- All risk is risk to Stakeholder Features:
  - If you ever discover a risk that is not represented in the modeled Feature space of a system, then you have just found a missing part of the positive Feature model. (Add it.)
  - So, it turns out that the negative FMEA model part improves the positive part of the model, and the positive part of the model improves the negative FMEA part of the model . . .

Features, Failures, and their Impacts



- Because we have available all modeled system Features, satisfying all system stakeholders, it follows that a system failure is synonymous with not delivering what a Feature promised.
- Each Feature is used to generate one or more Failure Impacts, summarizing the impact of not delivering (at least some aspect of) the Feature's promise to the stakeholder.
- For example:
  - Feature = "The Feature of delivering medication on a dose accurate basis."
  - Failure Impacts of not delivering Feature = "Illness", "Disability", "Death", etc.
  - Severity of Impacts: 3, 4, 5
- As illustrated above, each Failure Impact can also have an associated Impact Severity attribute value, expressing the stakeholder-rated severity of such an Impact.
- To cover all the Stakeholders, Features may include issues important not only to system end users, but also to those who manufacture, distribute, sell, or support the system, as well as shareholders in the profit-making enterprise, etc.
  - We may or may not be interested in Failure Impacts on all these stakeholders, but this offers us the opportunity to explicitly decide, instead of to forget.
  - If a failure analysis is to be limited to certain stakeholder and feature subsets, such as medical harms to patients, then the only features that need to be considered are those that have those impacts on patients.

Basic risk analysis aspects of S*Metamodel



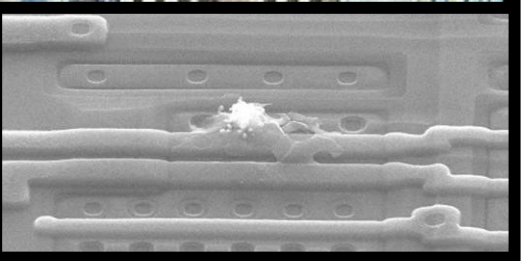


Feature	Effect (Failure Impact)	Severity	Functional Failure (Counter Requirement)	Component	Failure Mode	Probability	Mitigation (Control)
Navigation Feature [GPS-based Location Sensing]	No Confidence in Displayed Position	Serious (4)	The system displays a location that is not accurate to 10 feet.	Vehicle ECM	Erratic ECM	0.0015	Nav Backup Mode: External Nav Module
Navigation Feature [GPS-based Location Sensing]	False Confidence in High Error Displayed Position	Critical (5)	The system displays a location confidence indicator that is not correct.	Vehicle ECM	Erratic ECM	0.0015	None
Navigation Feature [GPS-based Location Sensing]	No Displayed Location	Serious (4)	The system does not display the graphic map presentation.	Panel Display	Fractured Display	0.0003	Nav Backup Mode: External Nav Module

Failure modes

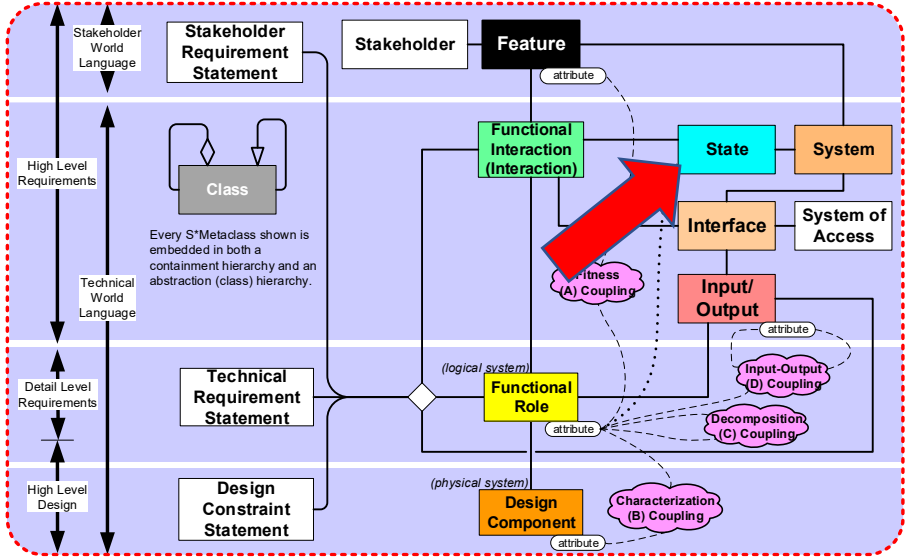












- FMEAs look for departures from a system's normal system level behavior because some component or subsystem is not behaving as that part was designed and verified to "normally" behave:
  - This is not looking for cases in which the design, with all its parts working as specified, will not meet requirements—that would be covered by the more basic design review of whether requirements will be met with all the components working "normally".
- Instead, the FMEA seeks out departures from normal component behavior caused by abnormal modes (states) of the components or subsystems, as in:
  - A worn-out part;
  - A part or subsystem which was already defective when installed (or perhaps harmed in the earlier manufacturing or later maintenance process);
  - A part abused by abnormal use of the product, or unusual conditions such as a lightning strike or power surge.
- These abnormal modes are contingent conditions that are given some modeled probability of occurring for a given rated part, in a given period.

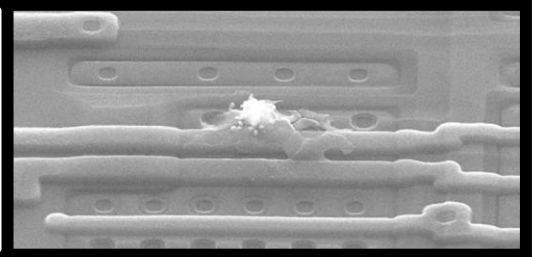
Vehicle Examples:
Component Failure Modes

Physical Entity	Failure Mode
Vehicle ECM	Dead ECM
Vehicle ECM	Network Connector Open
Vehicle ECM	Network Connector Short
Vehicle ECM	Erratic ECM
Battery	Discharged Battery
Battery	Battery Cell Short
Battery	Battery Cell Open
Battery	Battery Leak
Panel Display	Fractured Display
Panel Display	Illuminator Fail
Bluetooth Module	Module Hard Fail
Bluetooth Module	Transmitter Fail
Bluetooth Module	Receiver Fail

Failure modes

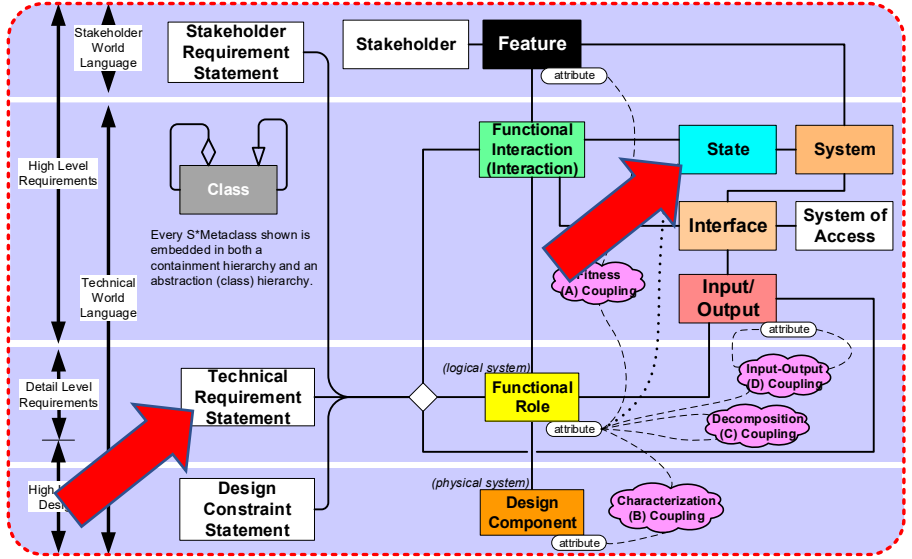










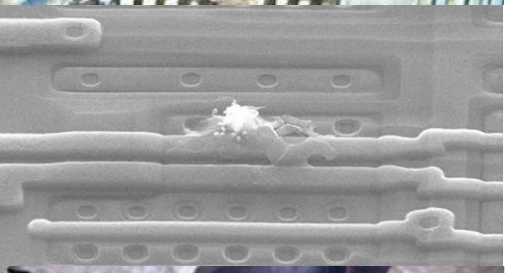


- The unusual condition represented by such a failure model is considered to be a “state” of the component or subsystem.
- When the component or subsystem is in that failure mode state, its behavior departs from its required behavior.
- Therefore, being in a failure mode means that one or more of the component’s or subsystem’s Requirements are not met.
- So, if we have a good complete set of modeled Requirements for the part or subsystem, we can be sure that one or more of those Requirements are specifically violated in a failure mode.

Failure modes

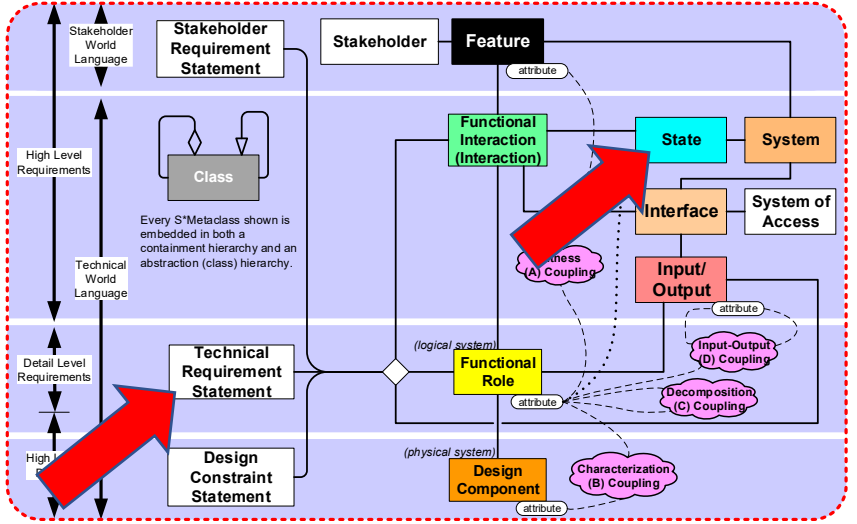












- Since some modeled component Requirement is not met, we are interested in the space of component behaviors which are "bad behaviors", described by "Counter Requirements":

Requirement Statement	Failure Mode	Counter Requirement Statement
"The Gear Train shall convert Input Torque to Output Torque, in a ratio of 7:1."	Gear Tooth Failure	"The Gear Train <<DOES NOT>> convert Input Torque to Output Torque, in a ratio of 7:1."
"The Oscillator shall generate a 3 MHZ Master Clock signal with stability of .01."	Chip Failure	"The Oscillator <<DOES NOT>> generate a 3 MHZ Master Clock signal."
		"The Oscillator <<DOES NOT>> generate a 3 MHZ Master Clock signal with stability of .01."

One Requirement Statement may have more than one Counter Requirement.

Requirements and Counter-Requirements

- Given a good set of Requirements, it is very straightforward to generate Counter-Requirements.
- But: A reasonable concern is that this seems to require having very solid Requirements.
- Yup.
- Isn’t that what we need anyway?
- One of several transitions from pre-MBSE to MBSE.
- Remember the comparison to the VLSI electronics transformation with EDA language at the outer boundary of VLSI system chips.

Basic risk analysis concepts of underlying S*Metamodel

Systematica® Metamodel

Metamodel Version 8.0

05/01/2025

BY S* PATTERNS COMMUNITY
© 2024, 2025, SYSTEM SCIENCES, LLC

2.24 Risk Analysis View

The Risk Analysis View depicts the classes of risk analysis and how they are related to other classes.

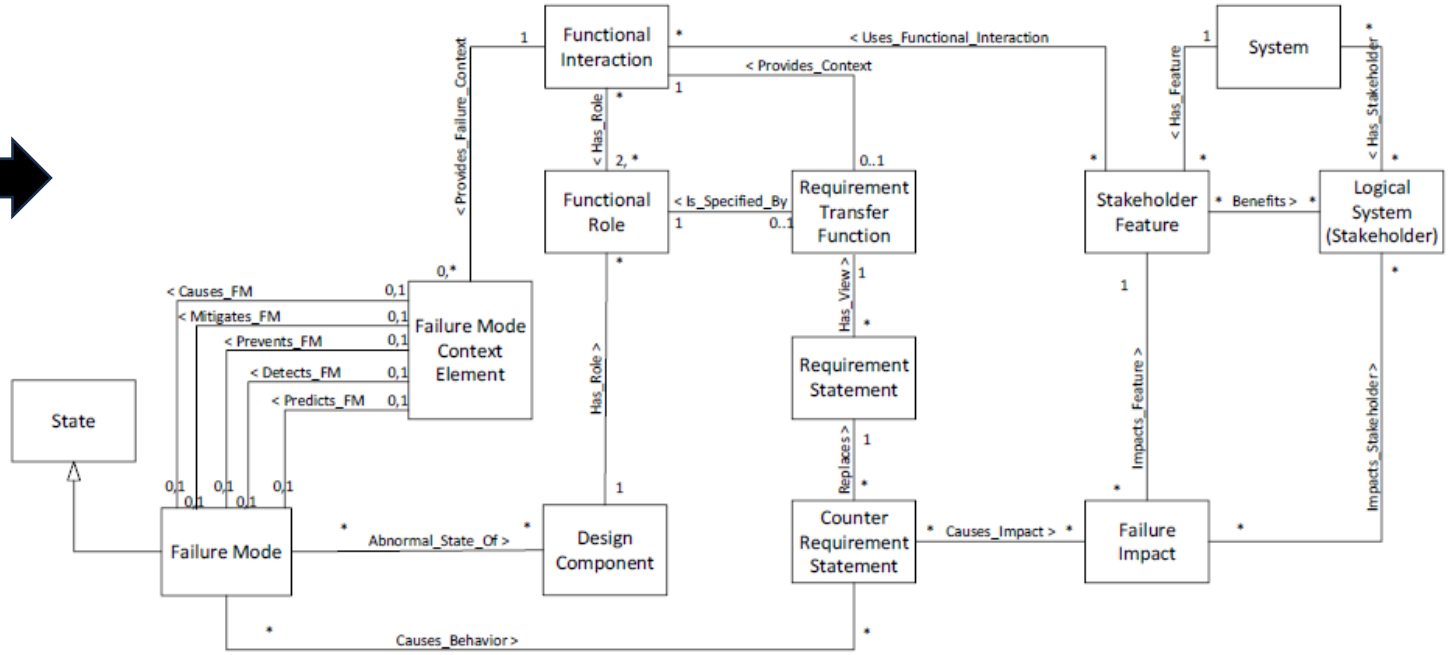
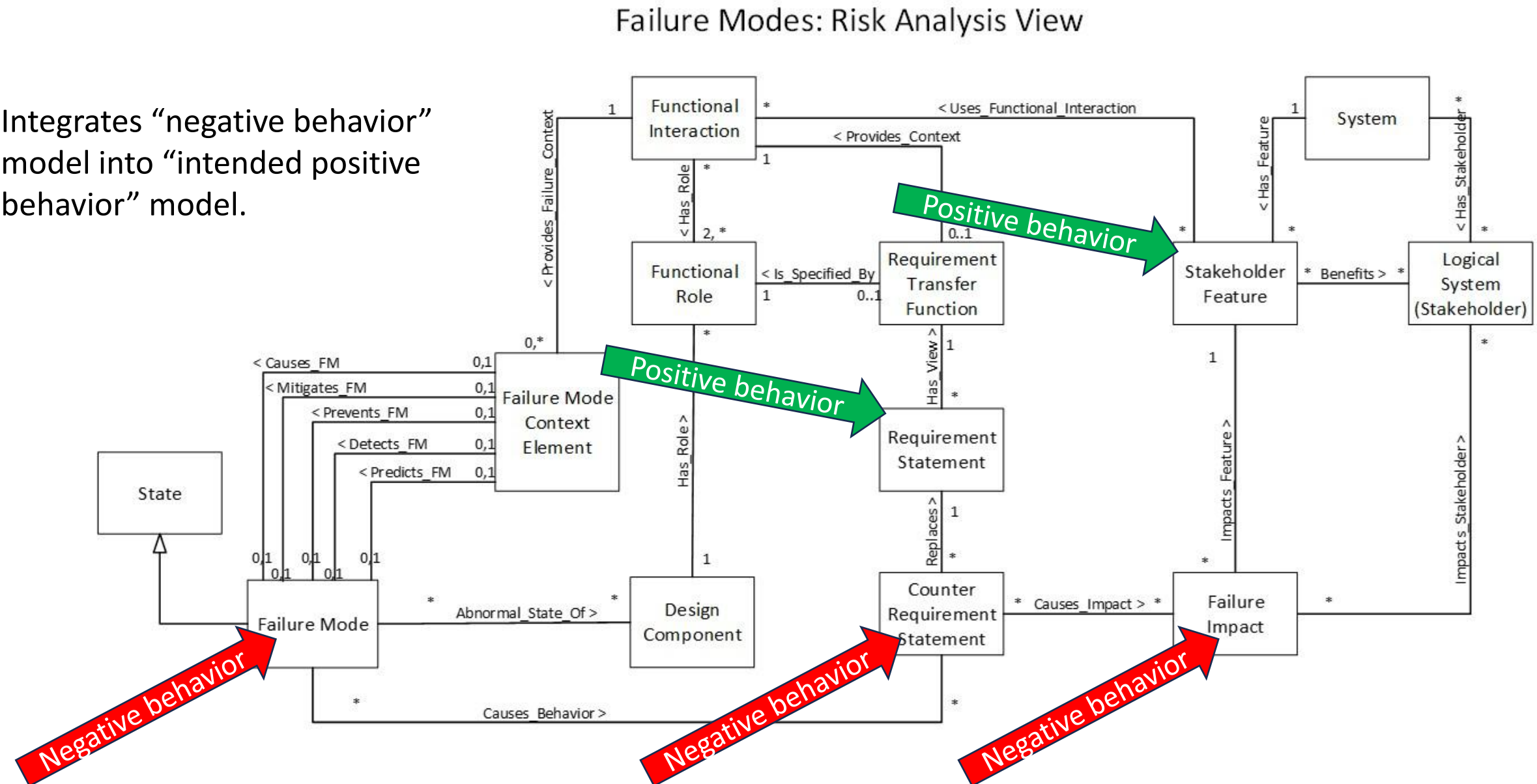


Figure 24: Risk Analysis View

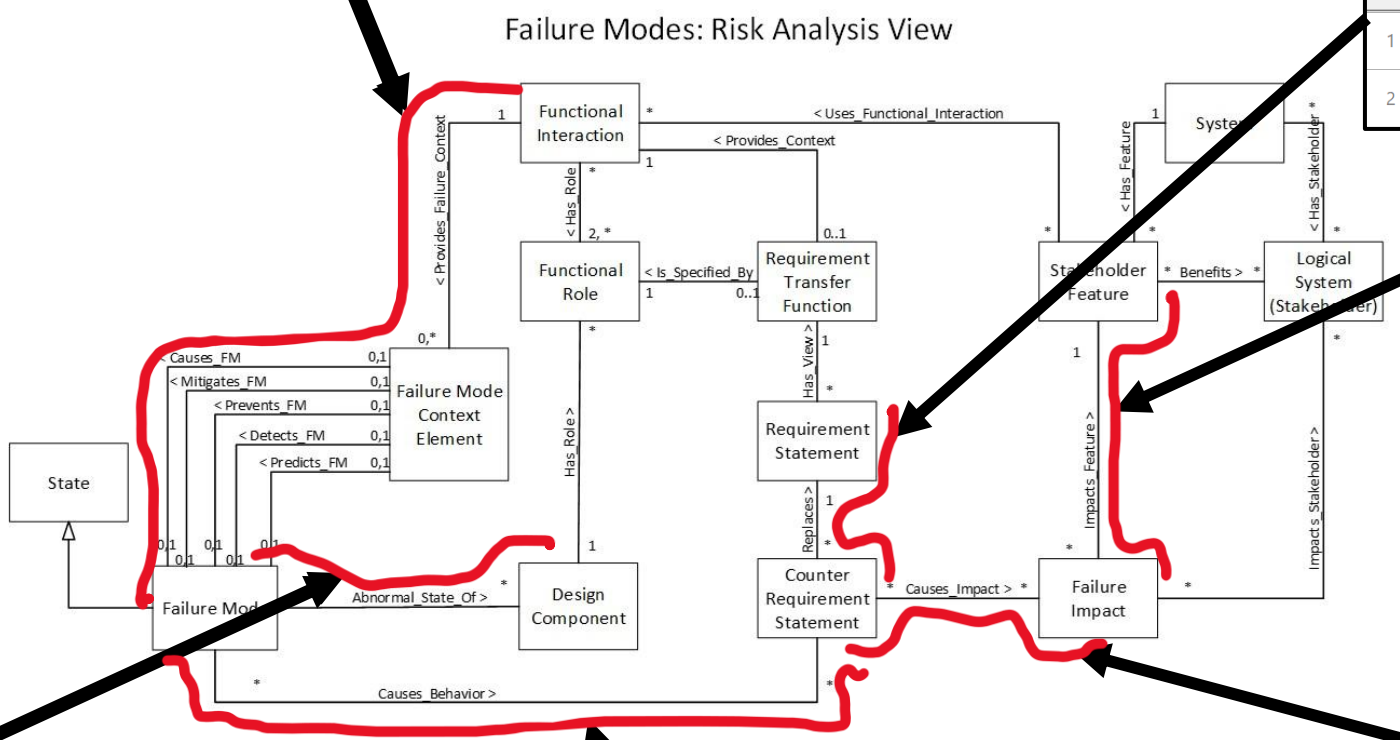
Basic risk analysis concepts of underlying S*Metamodel

Integrates “negative behavior” model into “intended positive behavior” model.



Example: International Power Converter Model

#	Context Element Name	Failure Mode	Interaction	Causes	Mitigates	Prevents	Detects	Predicts
1	IC Overheat	Regulator Failure	Convert Electrical Power	1				
2	Insulation Damage	Internal Electrical Short to Case	Assemble Product	1				



#	Requirement Name	Counter Requirement Name	Counter Requirement Statement
1	REQ 005	CREQ 001	The system presents a shock hazard to users when operated according to its instructions.
2	REQ 006	CREQ 002	The system generates Output Power to attached Electrically Powered Devices which exceeds the [Output Voltage-Power Profile].

#	Feature	Failure Impact	Severity
1	Powered Devices Compatibility	Damage to Powered Device	Serious
2	Safety	Electrical Shock	Severe
3	Reliability and Durability	Loss of Converted Power Output	Serious

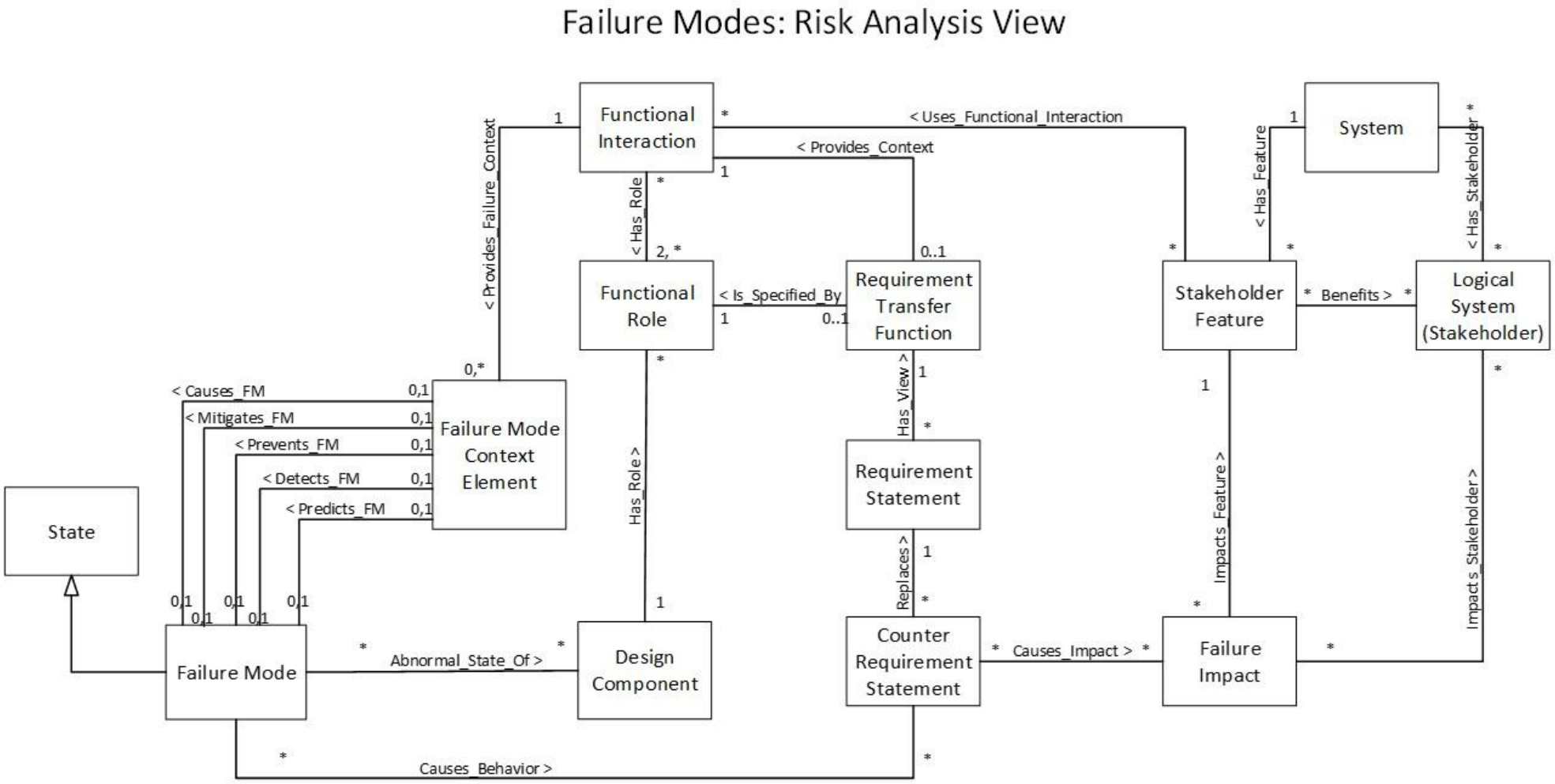
Filling out six model tables populates risk analysis FMEA framework.

#	Probability	Design Component	Failure Mode
1	0.0002	Power Converter Assembly	Regulator Failure
2	0.0003	Power Converter Assembly	Internal Electrical Short to Case

#	Counter Requirement Name	Failure Impact
1	CREQ 001	Electrical Shock
2	CREQ 002	Damage to Powered Device

#	Failure Mode	Counter Requirement Name
1	Regulator Failure	CREQ 002
2	Internal Electrical Short to Case	CREQ 001

Basic risk analysis aspects of S*Metamodel



Notice the direct tie of this model to the Robustness of a system's design: How sensitive are its Feature Impacts to its component Failure modes?

More on this later: Fault-Tolerant Systems

BREAK

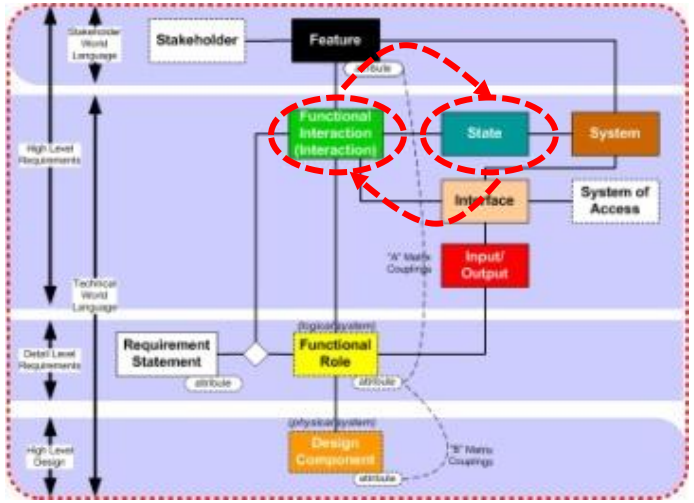
Advanced pattern-based risk analysis

- We will study pattern-based MBSE in Session 5, but for the time being, it should be evident that . . .
- We can improve the completeness and correctness of risk analysis such as FMEA for families and product lines of similar systems, by accumulating reusable pattern libraries of:
  - Design Components, with allocated Functional Roles
  - System Feature Impacts, with Severities
  - Component or Subsystem Counter Requirements
  - Failure Modes, with Probabilities
  - and the *systemic relationships between them*

Advanced pattern-based risk analysis

- Across various product configurations of a system product line, some features may be present or absent, along with required system behaviors and design components, per product line variation.
- How does this play in to model-based risk analysis?
- *The key points are*:
  - Some part failure modes may still be present, but not have consequence in low-featured product configurations;
  - For high-featured product configurations, part failure modes may become significant that were not important in low-feature product configurations;
  - Don't gold-plate components when they are not needed by the advertised Features!
  - Don't use cut-rate components when more is needed by the advertised Features!
- It turns out that the related shifts can be automated through pattern configuration "Joins" we will study further in the Patterns unit . . .

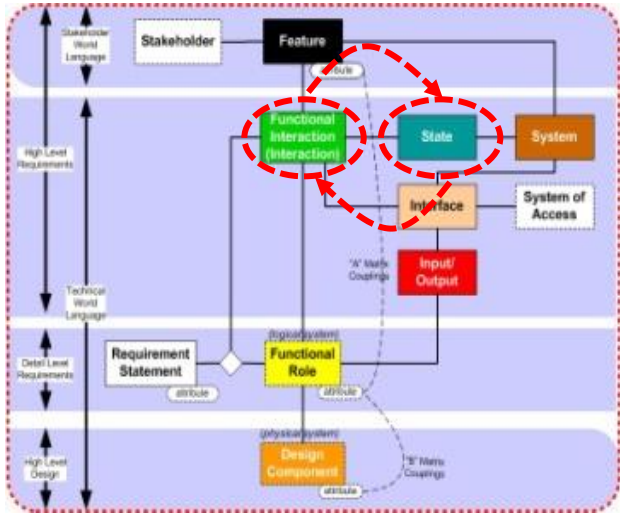
Interaction-State Chains; Causes



- Design Components, States, Interactions, Requirements, and Features information of the Metamodel can be unfolded (split) across normal and abnormal behavior, and across “causality chain” sequences.
- The resulting models add further to the information used to populate a Failure Analysis (e.g., FMEA table).
- In all these cases, the current mode (state) of the system can be viewed as the “immediate” reason that it is behaving a particular way:
  - That behavior is characterized by the interactions the system is currently able to perform (the interactions associated with that state)
  - States imply Interactions imply States imply Interactions . . .

Interaction-State Chains; Causes

- If we ask how the system came to be in its current state, we find that a previous interaction of some sort will have “placed it in the current state”. This leads to the idea that there are “causality chains” that take the form of sequences of alternating interaction, state, interaction, state, etc. For example:
  - Interaction: Turn On the System
  - State: System On
  - Interaction: Request System Menu
  - State: Displaying Menu
- This same idea works for abnormal states:
  - Interaction: Insert Battery
  - State: Battery Inserted Backwards
  - Interaction: Turn On System
  - State: System Inoperative



- In all these cases, the idea of <u>cause</u> can be pursued through earlier parts of the chain:
  - We can say that a later part of the chain is “caused” by the states and interactions of an earlier part of the chain.



Enhanced use of FMEA and risk analysis tools

- Basic and advanced commercial automated tools are available for use in generating FMEA and other forms of failure or risk analysis.
  - –In their most basic use, the analyst manually enters data into relatively fixed forms and generates resulting reports.
  - –In their more advanced form, these tools support customization or configuration of reports, data entry, and some aspects of the underlying information models.
  - –Some also support accumulation and use of re-usable standard categories or other data, and some support integration with other engineering tools, such as requirements management tools.
- The model-based concepts, methodology, and procedures described in this class can be used with a number of these commercial tools, improving their value.
  - –In general, the more powerful and flexible the tool, the more aspects of this methodology may be used.



Enhanced use of FMEA and risk analysis tools

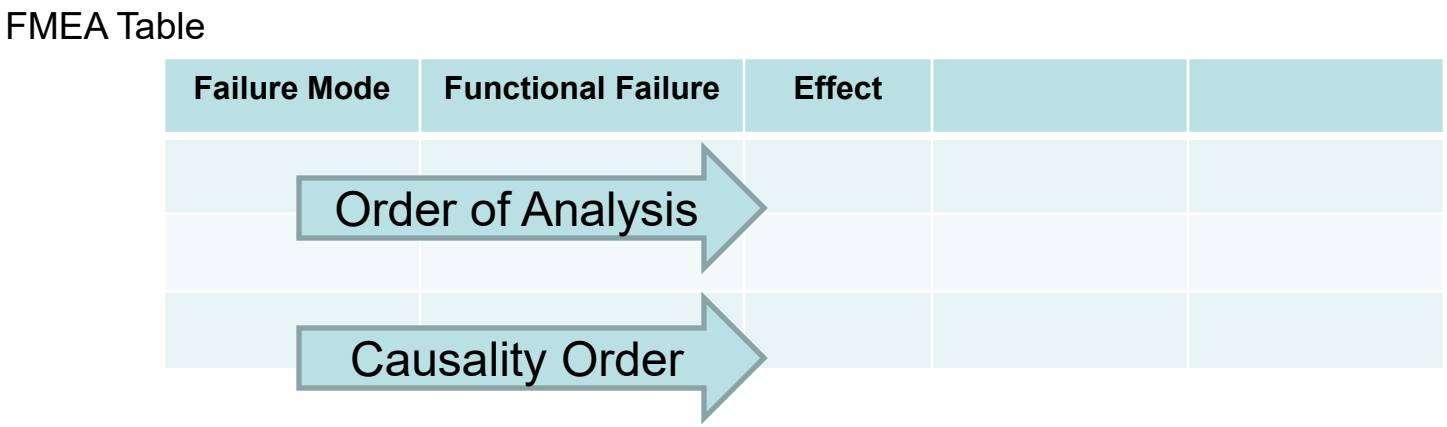
- The simplest, but least beneficial, way to initially do this is to configure the tables and reports of a tool to accept manual entry of data of the type described in this talk.
  - –A more sophisticated approach allows re-use of data from a pattern of requirements, design, and failures (patterns).
  - –Since patterns are relational models, this is more powerful than simply having lists of standard pull-down items.
- This methodology also enhances the ability to integrate an FMEA or failure analysis tool with a Requirements Management tool, by using Counter-Requirements that are associated with the system level Requirements:
  - –This is more powerful than simply having links between data items in two tools.
  - –In fact, if a requirements and design model is available in MBSE form, then tool-based combinatorial algorithms can be used to automatically generate an initial draft FMEA table.
  - –Of course, this does not replace human analysis, but does reduce the drudgery of initial generation, freeing the analyst to do deeper thinking and analysis of the failure data.

More details: A unifying MBSE viewpoint for integrating Risk Analysis

- The MBSE concepts and information can be used to integrate or unify a number of Risk Analysis ideas:
  - Order of occurrence versus order of analysis; checking; FMEA versus Fault Tree
  - Faults vs. Failures; Fault Tolerant Systems; Fail Safe Aspects
  - Subsystem Causing Failure: D-FMEA
  - Peer System Causing Failure: A-FMEA
  - Peer System Causing Failure: P-FMEA
  - D-FMEA, A-FMEA, P-FMEA, and Unified FMEA
  - Risks identified at different times by different parties
- Each of the above are discussed in the following . . .

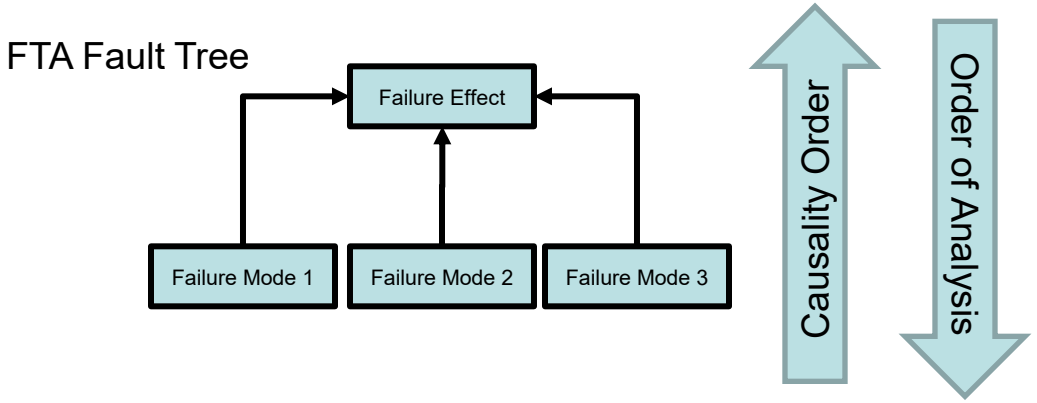
Order of occurrence (causality) versus order of analysis: FMEA versus Fault Tree

- FMEA analysis typically reasons from component failure modes to system level counter-requirements, to the stakeholder impacts (failure effects, such as user injury).
- This traditional analysis thus occurs in the sequence of cause-to-effect, and the methodology described here supports that order of reasoning.
- In a traditional FMEA table, it proceeds more or less from left to right.
- This traditional order of reasoning is why FMEA is typically said to work for analysis of single failure modes but not multiple simultaneous failure modes.



Order of occurrence (causality) versus order of analysis: FMEA versus Fault Tree

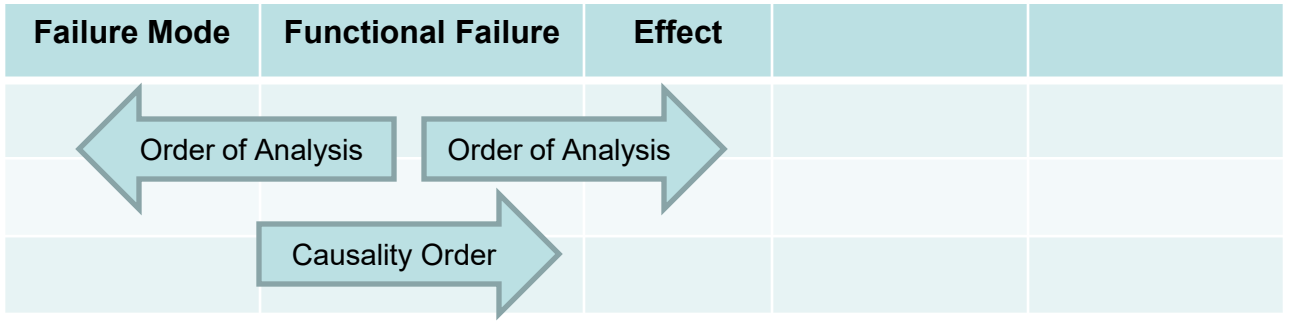
- It can be seen that this methodology also supports the generation of Fault Tree analyses.
  - Whereas an FMEA analysis traditionally begins from each possible component level failure mode and reasons to its effect (typically a one-to-one process generating a row of an FMEA table) . . .
  - A Fault Tree analysis traditionally begins with each effect and reasons backwards to identify each possible component failure mode that might cause it (typically a one-to-many process generating a many-branched fault tree under a single effect).
- Each path of the fault tree is roughly equivalent to a row of the FMEA table.
- The information models described here describe both approaches, differing only by the order in which the data model is filled in during the analysis process.



Order of occurrence (causality) versus order of analysis: FMEA versus Fault Tree

- The use of MBSE failure analysis also allows reasoning in other directions;
- Because it is really about an underlying information model, not an order of reasoning, we can populate that information model in different orders:
- These include backwards reasoning from failure effect to cause (as in a Fault Tree Analysis) . . .
- But also “middle-out reasoning”, from a system counter requirement to both its upstream causes and downstream effects:

FMEA Table



  - This is of major value, as it facilitates completeness checking of the resulting failure analysis table.
  - We can independently check the effects against a complete library of all possible feature-based impacts.
  - We can independently check the middle (the system counter-requirements) against a complete library of all possibilities, based on the listed system requirements.
  - This improves completeness and coherence of the FMEA or other analysis, including its inspectability.

Faults vs. Failures; Fault Tolerant Systems; Fail Safe Aspects

- In the specific language (Anderson and Lee 1981) of fault tolerant systems (which is not always used the same in failure analysis procedures) Faults and Failures are undesirable states or behaviors, but don't mean the same thing.
- A Fault is an abnormal component or subsystem condition (state), which may or may not result in a system level Failure.
- Remembering from above that Failures are not delivering agreed upon stakeholder Features, we can say that a fault tolerant system is a system that does not fail (continues to deliver stakeholder Features) in spite of component or subsystem Faults.
- (That is, it tolerates Faults in its own components, while continuing to deliver external Features.)
- For example, aircraft hydraulic systems typically employ redundancy, so that they can deliver safe flight services (Features) while tolerating a Fault in a hydraulic line.

Faults vs. Failures; Fault Tolerant Systems; Fail Safe Aspects

- In the language of failure mode analysis, the term “Failure Mode” is frequently used to describe an abnormal state of a component or subsystem, even if the overall system was designed to keep delivering all its external services in the presence of that component Failure Mode.
- This is not so inconsistent if you consider that the subsystem or component is not delivering its “external” services, but it can be a little confusing if you don’t expect the term, or are keeping track of formal system decomposition levels.
- Sometimes a system internal fault can present risk of a serious (e.g., life or property threatening) failure behavior by the subject system.
  - In those cases, mitigations are sometimes planned such that, although the system may fail to deliver all of its promised features, it protects from presenting a more serious failure.
- That is, it still fails, but “fails safely”. This is called a fail safe system.

Subsystem Causing Failure: D-FMEA



- In a system, an abnormal state of a component may cause a system level failure.
- We can reason forward from the component state to the system failure it causes, or backward from the component state to its cause.
- For example, the following failure mode is “caused” by the interaction shown:
  - (Interaction) Cause of Failure Mode: Normal Wear
  - Component Failure Mode State: Gear Train Binding/Lash-Up
- Remembering the idea of interaction-state chains, we can see that many such failure mode states can be said to be caused by a previous interaction, whether it is a normal use interaction or some extraordinary damaging interaction.
- If the causal interactions are “normal” behavior by the external systems performing them, then we could say that the failure mode is effectively inherent to the design of the subject system in its normal use.
- Analyzing failures of this kind is typically the subject of D-FMEA (Design Failure Mode Effects Analysis) work.
- Sometimes this leads to a different design to reduce the likelihood of the failure mode occurring, or in other cases to other controls (mitigations) intended to reduce the impact of the failure mode when it occurs.
- In all those cases, it could be said that the role played by the subject system in normal interactions eventually leads to the failure mode of the system’s component.
- However, it is alternatively possible that the system design is not the cause, but rather that the external systems are behaving abnormally.
  - This case is covered in the next two slides—for P-FMEA and A-FMEA analysis.

Peer System Causing Failure: A-FMEA

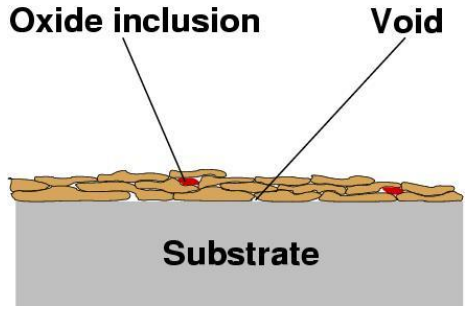


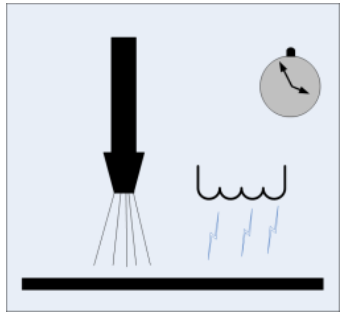
- External systems interacting with the subject system are sometimes called “peer” systems, or “actors”.
  - Unlike the subsystems or components discussed for D-FMEA, they are external to the subject system.
- In an A-FMEA (Application Failure Mode Effects Analysis), attention is focused on the effect of abnormal behavior by external systems (that are typically human “users” of the subject system—or other actors).



- It could be said that the original failure modes in this case are states of the external system. For example:
  - Failure Mode (Pilot State): Attention Overloaded
  - Interaction: Select Target (assume wrong value entered)
  - State (of Weapons System): Awaiting Weapon Release Confirmation
  - Interaction: Confirm Weapon Release
  - State: Delivering Weapon
- As illustrated by the above example, we can have a failure to deliver overall system features even though the subject system meets all of the requirements assigned to it.
- However, it is also possible for an external system to drive the subject system into its own abnormal (e.g., damaged) state, after which it no longer meets requirements assigned to it. For example:
  - Cause of Failure (Interaction): Poor User Training
  - Resulting Failure Mode (State): User Unaware
  - Interaction: User Closes Valve (Over-Tightening)
  - Resulting System Component State: Valve Seal Failure
- Both of these cases are of interest in an A-FMEA—and the second case looks a lot like a D-FMEA after the point of driving the subject system into a bad state.
- Notice that “users” are not the only external systems whose failure modes can damage the subject system’s state.
  - Other mis-behaving systems in the Application Domain may also have to be considered.
- When the external actor that is in an abnormal state is a human being, the MBSE model is about modeling human behavior.
  - This is further discussed in (Schindel 2006).

Peer System Causing Failure: P-FMEA





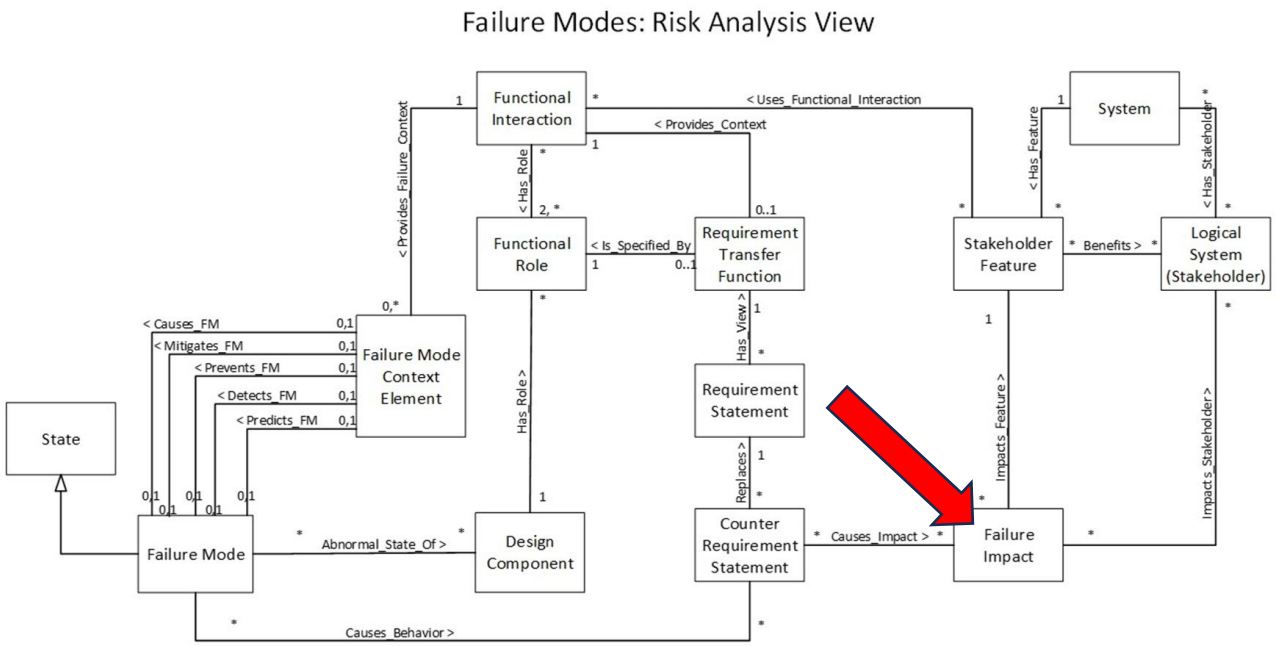
- A special external system traditionally analyzed is the subject system's Manufacturing System.
- This is the subject of a P-FMEA (Process Failure Mode Effects Analysis).
- The nature of a Manufacturing System is to create the subject system, so it may be found that all the P-FMEA failures of interest result in bad product system states. For example:
  - (Interaction) Cause of Failure Mode: Glue Build-Up on Nozzle During Use
  - Component Failure Mode State: Nozzle Obstructed
  - (Interaction) Not enough glue applied
  - Subject System State Part Loose
- There can also be Manufacturing Process failures that fail in the sense of not delivering on all the other manufacturing process systems features, as when manufacturing Yield, Operating Cost, or Safety are impacted by manufacturing faults:
  - Depending on the intended Stakeholder Feature scope of the P-FMEA, these may or may not be of interest to include and analyze.
- Other major processes, such as the commercial Distribution Process, can have faults that create bad states in the subject system. For example:
  - (Interaction) Cause of Failure Mode: Transport Packaged Product
  - Component Failure Mode State: Package Seal Fractured
  - (Interaction) Tolerate Exposure to Contaminants
  - Component Failure Mode State: Food Product Contaminated
- Depending on the intended causal and stakeholder feature scope of the P-FMEA, these other processes may also be considered.

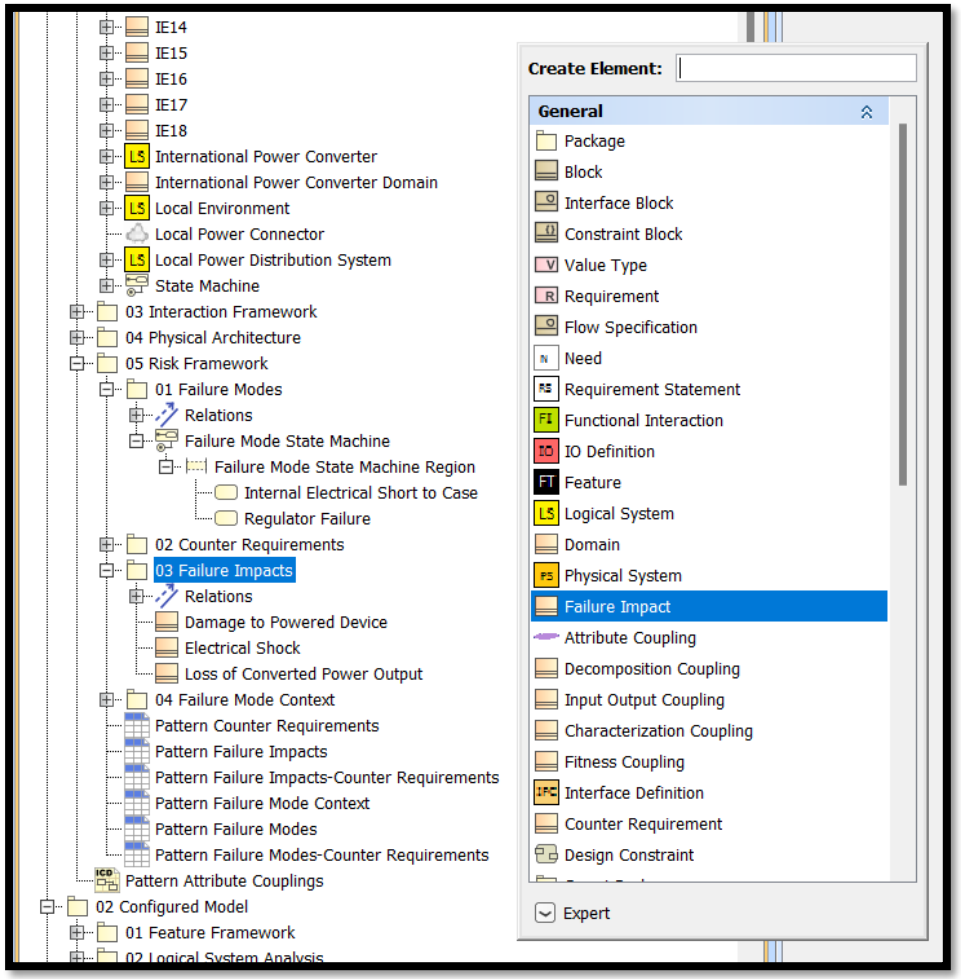
D-FMEA, A-FMEA, P-FMEA, and Unified FMEA

- Although it may be desirable to separate the D-FMEA, P-FMEA, and A-FMEA "reports" for attention by different groups, and to generate and review them using different subject matter experts, it is also desirable to generate them from a consistent underlying information model.
  - For example, all three FMEA types depend on the same system level counter-requirements and feature impacts.
  - If this consistency is used, then it is easier to understand the different FMEAs in a consistent way, and to judge their accuracy and completeness.
- While there may be reasons to differently format or label the tabular "reports" that are generated for these different types of failure analysis, the approach described here at least intends to generate them from a common base of underlying information, and to minimize differences in labeling except where it improves the outcome.

Tool Model Insertion: Failure Analysis-Creating Failure Impacts

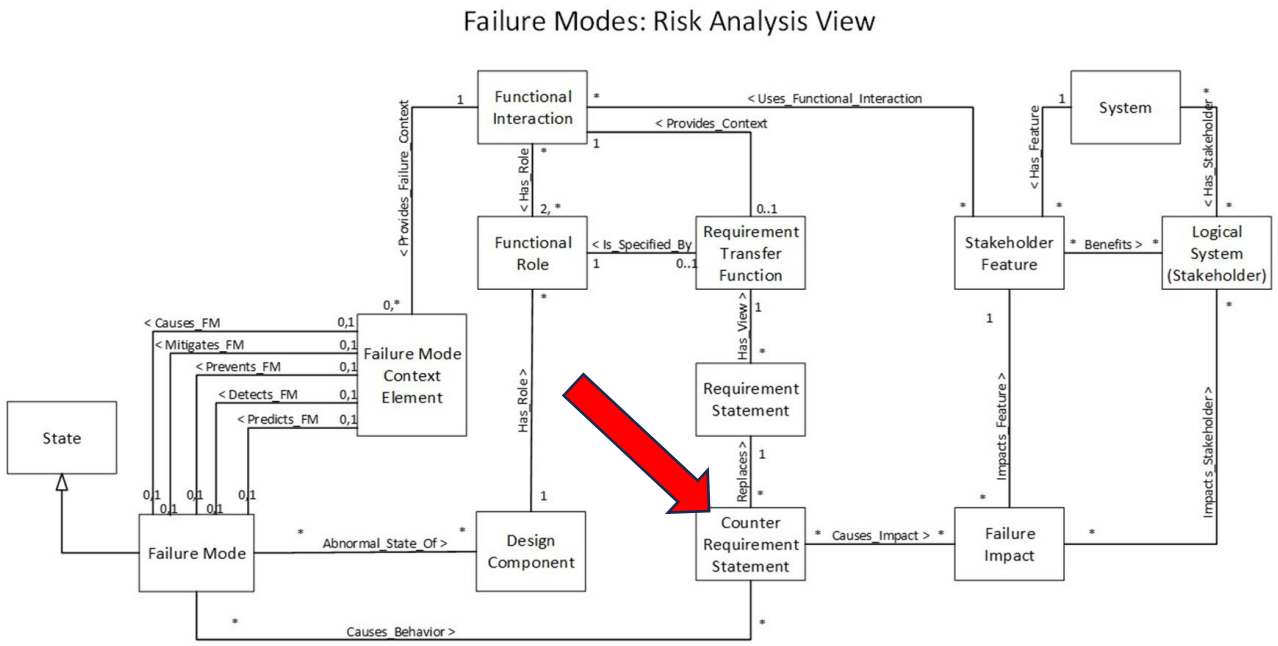
- Pattern Failure Impacts are created in package 01 Local Pattern::05 Risk Framework::03 Failure Impacts
- Use ~Create Element and choose Failure Impact

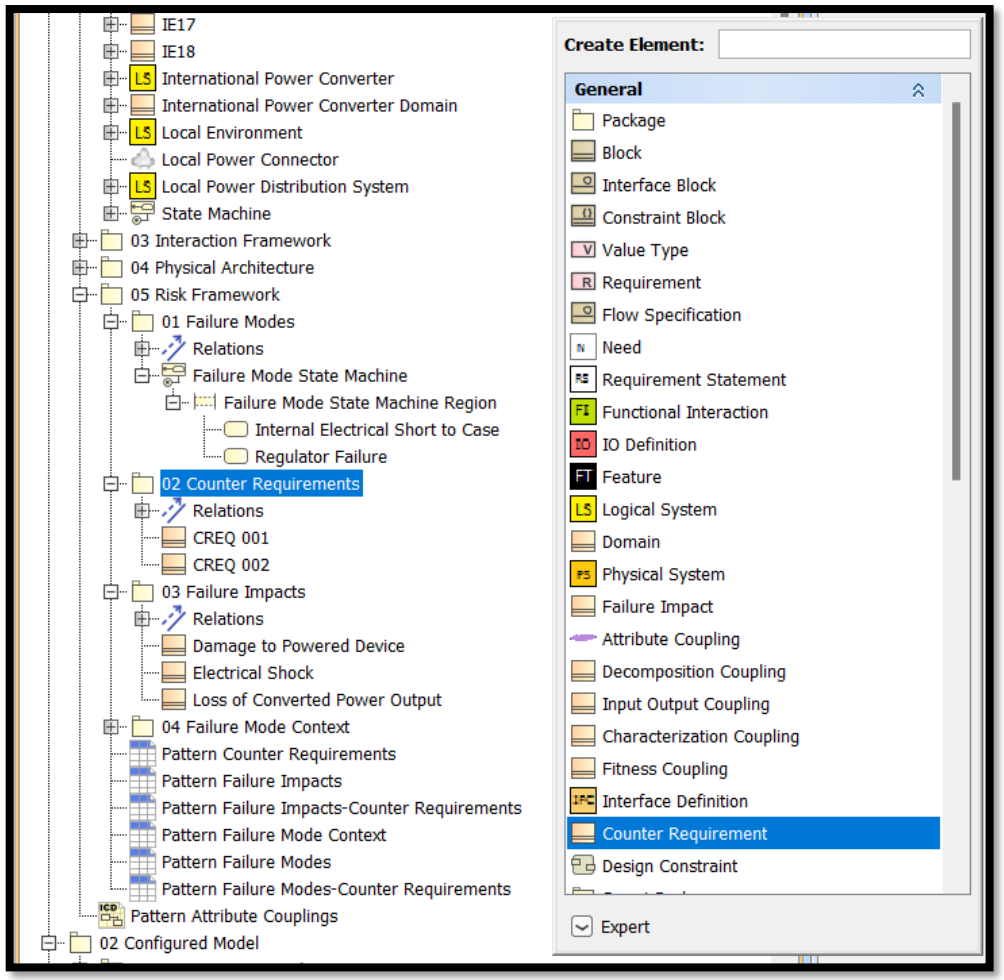




Tool Model Insertion: Failure Analysis-Creating Counter Requirements

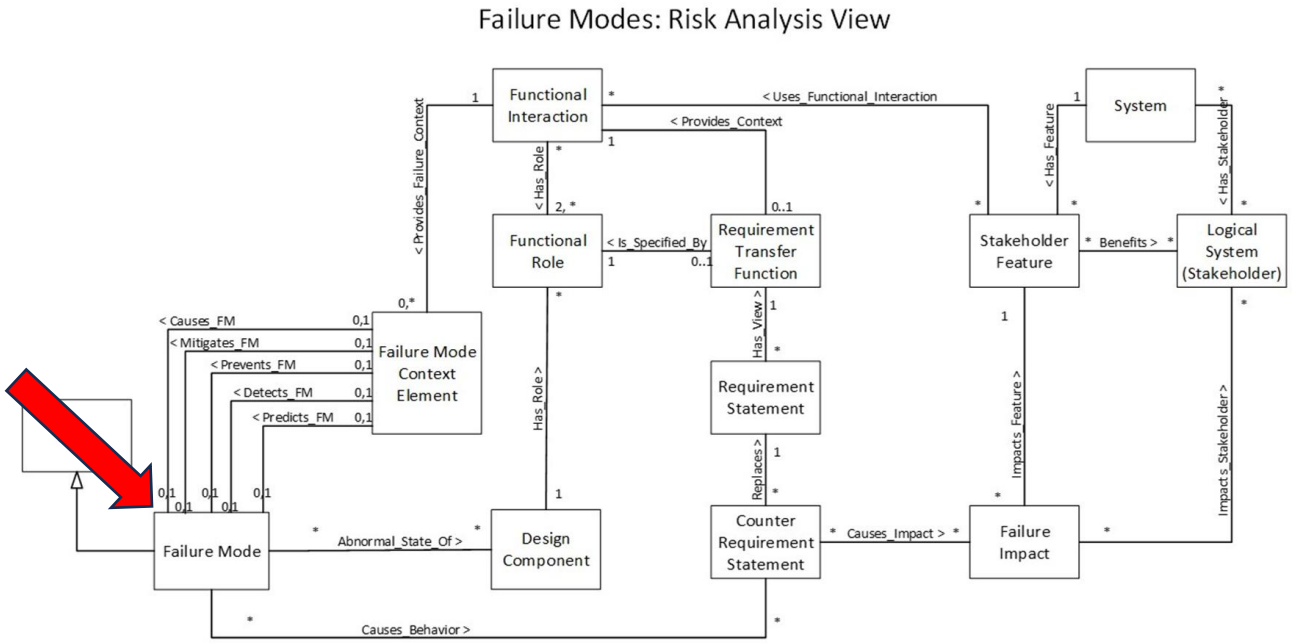
- Pattern Failure Counter Requirements are created in package 01 Local Pattern::05 Risk Framework::02 Counter Requirements
- Use ~Create Element and choose Counter Requirement

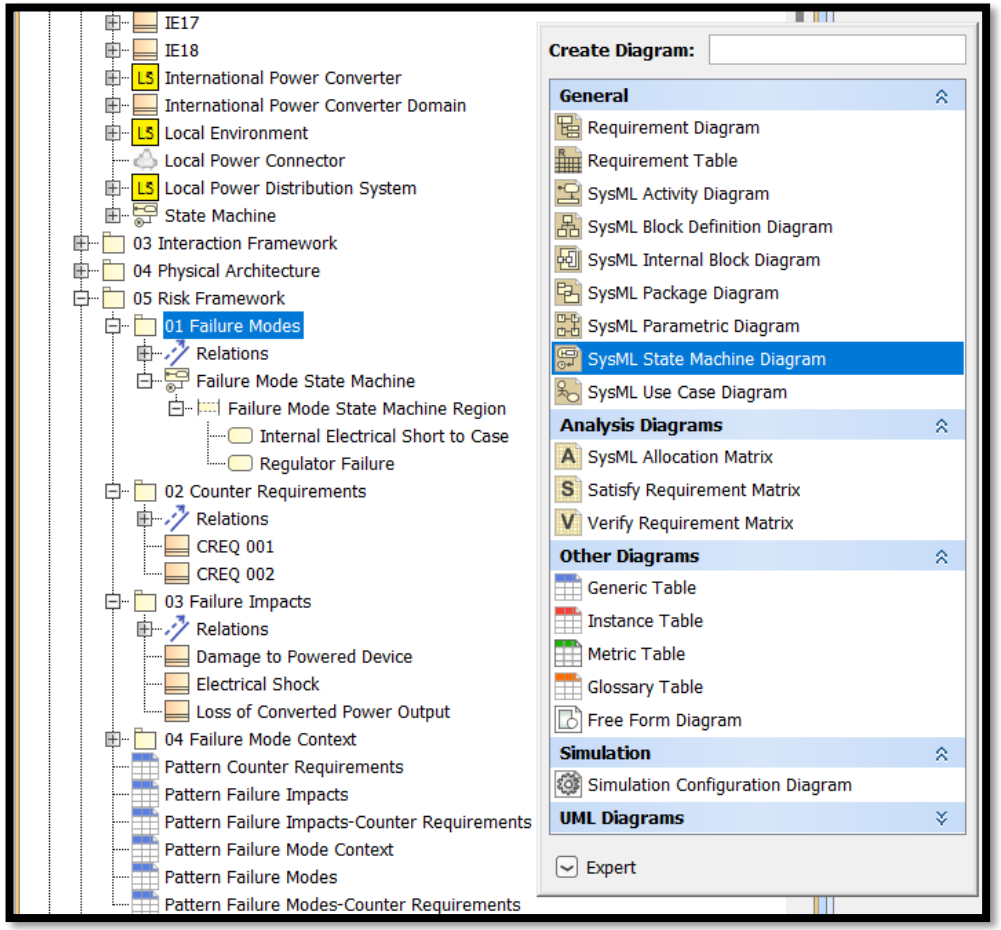




Tool Model Insertion: Failure Analysis-Creating Failure Mode State Machine and Failure Modes

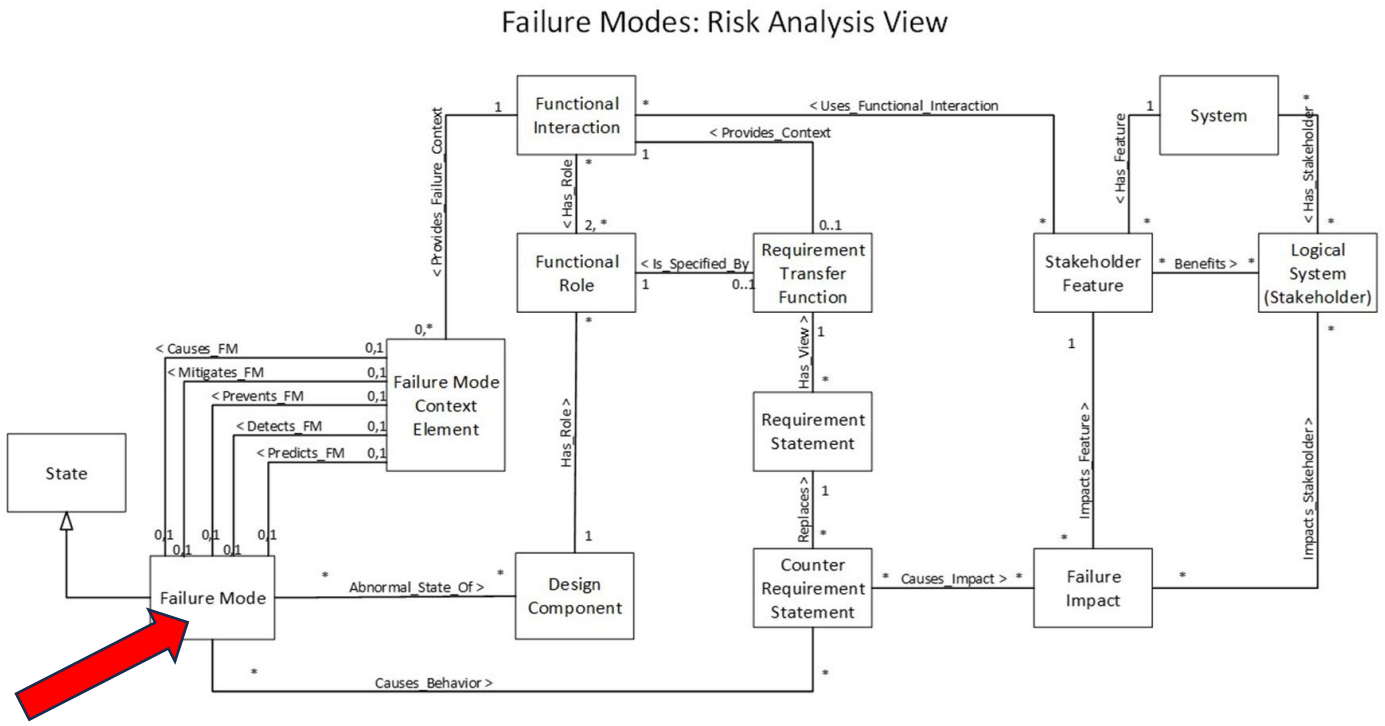
- Pattern Failure Mode State Diagram and Failure Modes are created 01 Local Pattern::05 Risk Framework::01 Failure Modes
- Use ~Create Diagram and choose SysML State Machine Diagram
- Name the Diagram “Failure Mode State Machine”.
- Name the Region “Failure Mode State Machine Region”.

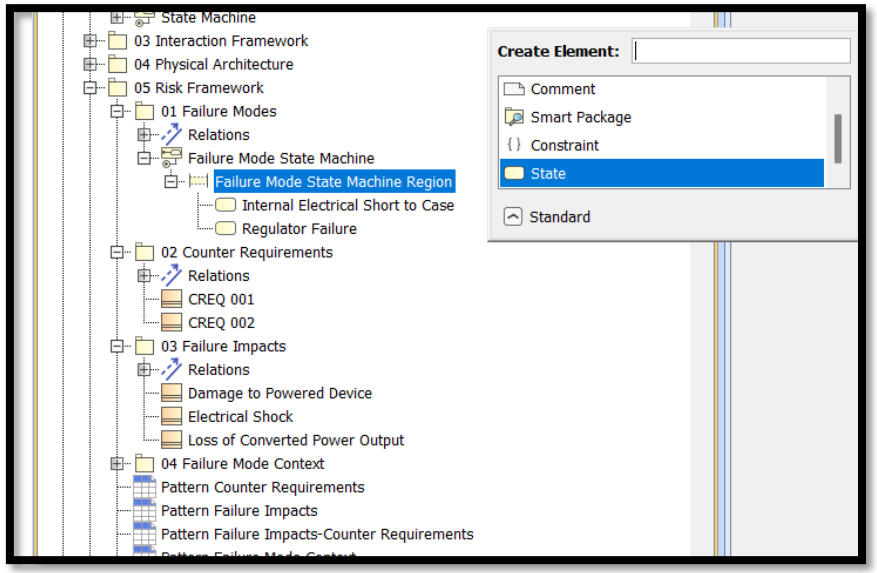


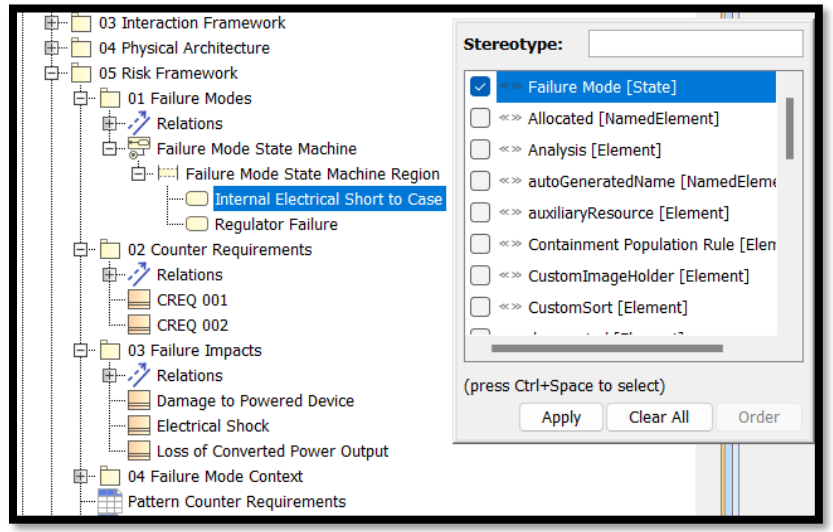


Tool Model Insertion: Failure Analysis-Creating Failure Modes

- Right-click the automatically created Region under the State Model and use ~Create Element and choose State. Make sure the menu is expanded to Expert.
- Right-click the newly created State, select Stereotype, and change it to Failure Mode as shown below.

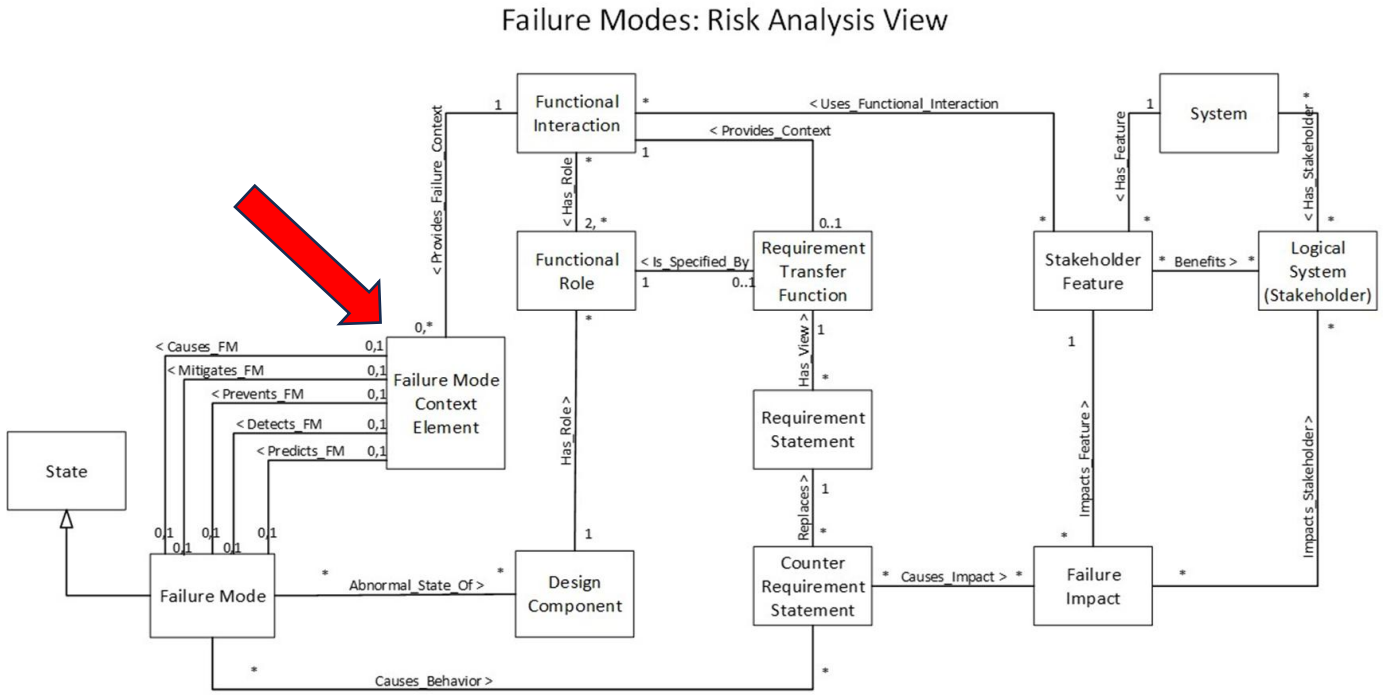


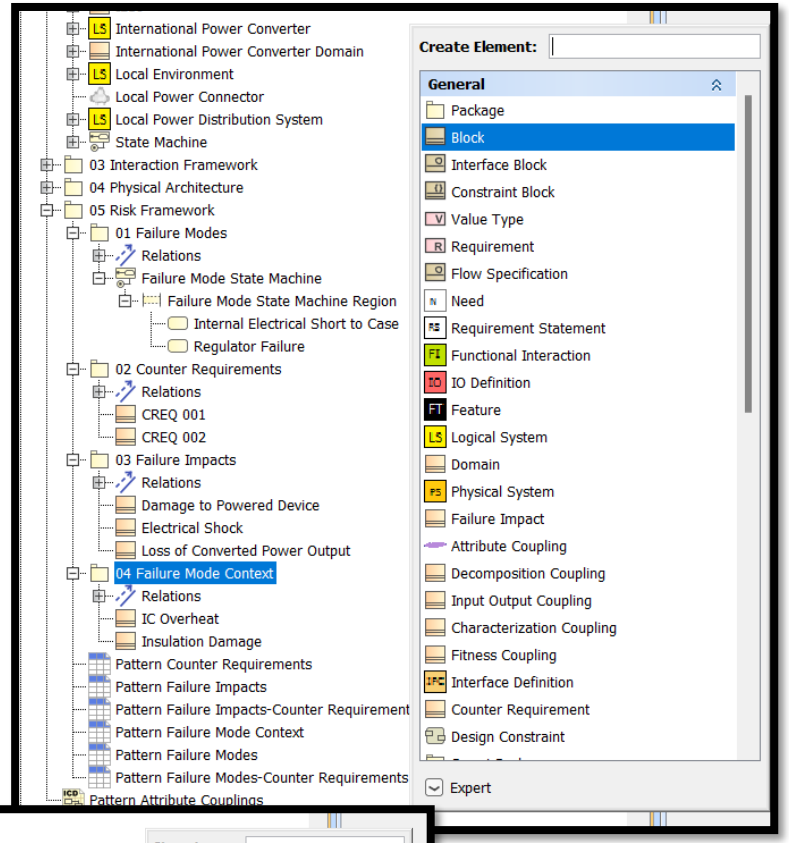


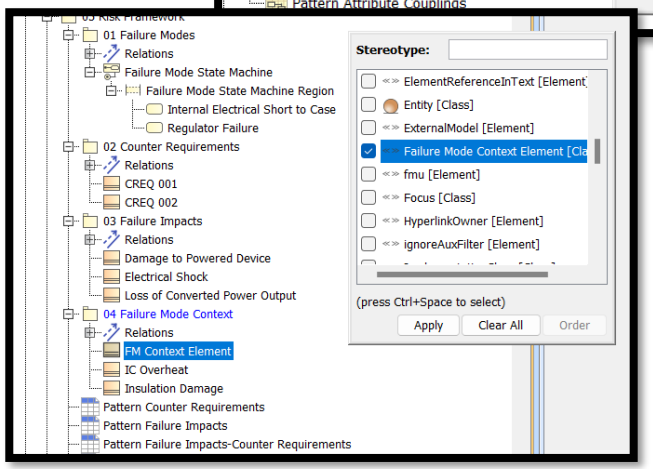


Tool Model Insertion: Failure Analysis-Creating Failure Mode Context

- Pattern Failure Mode Context Elements are created in package 01 Local Pattern::05 Risk Framework::04 Failure Mode Context
- Use ~Create Element and choose Block
- Right-Click on the new element, and change stereotype to Failure Mode Context Element







Discussion, Q&A.

-
-
-
-
-
-



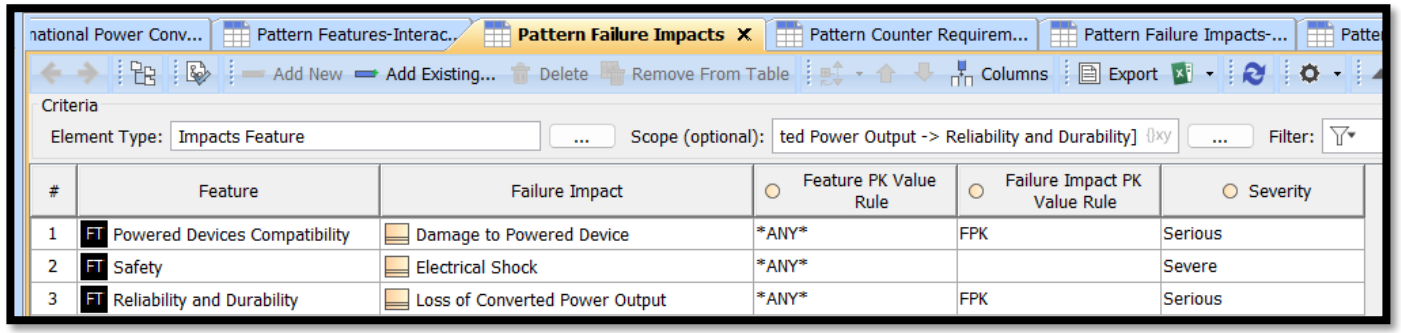
Homework assignment 3

- For the example pattern you have been using for HWs 1 & 2, add metarelationships to the FMEA elements added during the in-class exercise:
- Impacts Feature (Pattern Failure Impacts Table)
- Causes Impact (Pattern Failure Impacts-Counter Requirements Table)
- Replaces (Pattern Counter Requirements Table)
- Causes Behavior (Pattern Failure Modes-Counter Requirements Table)
- Abnormal State Of (Pattern Failure Modes Table)
- Provides Failure Context (Pattern Failure Mode Context Table)
- Provides Failure Mode Context (Pattern Failure Mode Context Table)
- Bonus: Show some RPN scores, based on the mode probability score, severity score, and detectability score.
- See subsequent slides for homework guidance steps.

Metarelationships:
Creating Failure Analysis Relationships—Impacts Feature

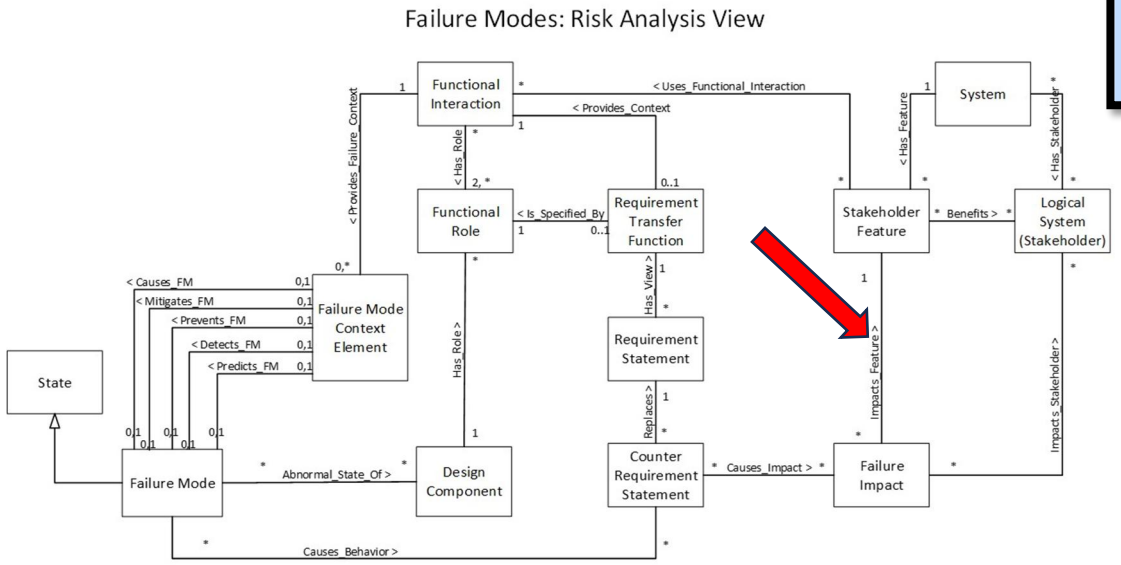
Homework

- Create a Dependency relationship with the Failure Impact as the Source and the Feature as the Target.
- Add a “Impacts Feature” stereotype to the new relationship



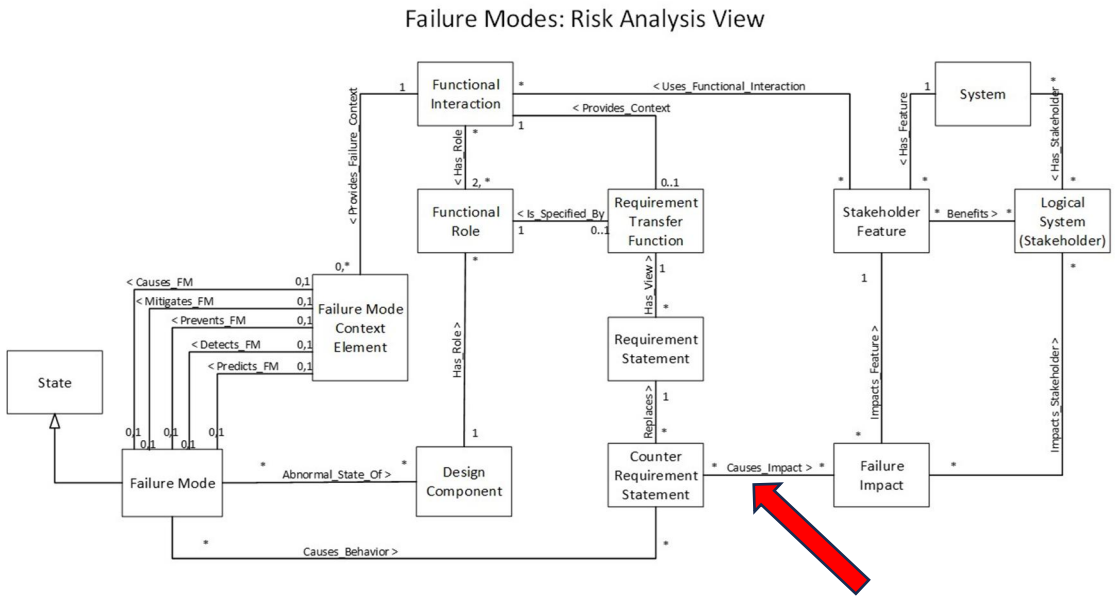
#	Feature	Failure Impact	Feature PK Value Rule	Failure Impact PK Value Rule	Severity
1	Powered Devices Compatibility	Damage to Powered Device	*ANY*	FPK	Serious
2	Safety	Electrical Shock	*ANY*		Severe
3	Reliability and Durability	Loss of Converted Power Output	*ANY*	FPK	Serious

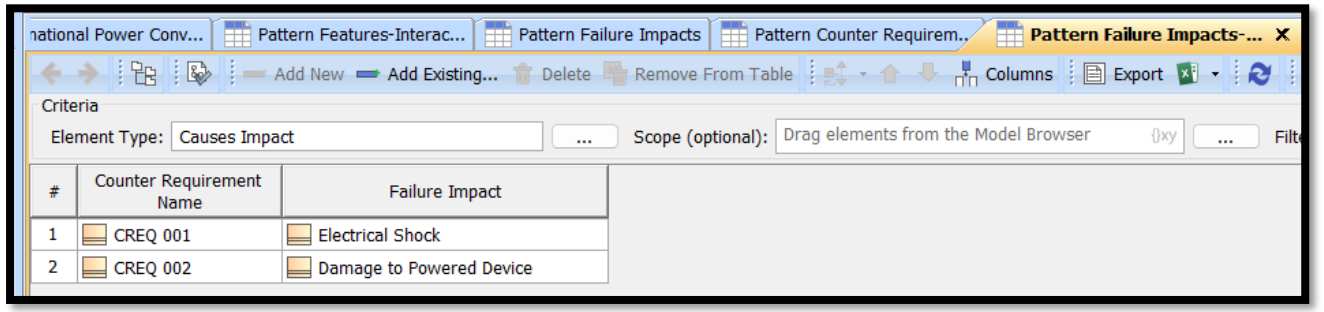
Example Pattern Failure Impacts Table



Metarelationships: **Homework**
Creating Failure Analysis Relationships—Causes Impact

- Create a Dependency relationship with the Counter Requirement as the Source and the Failure Impact as the Target.
- Add a “Causes Impact” stereotype to the new relationship



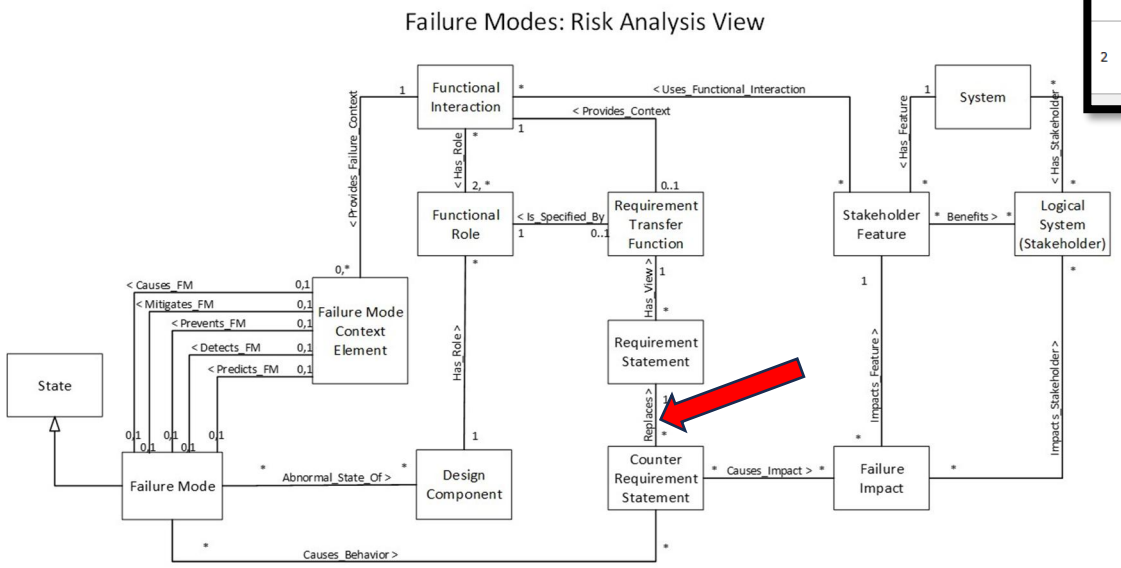


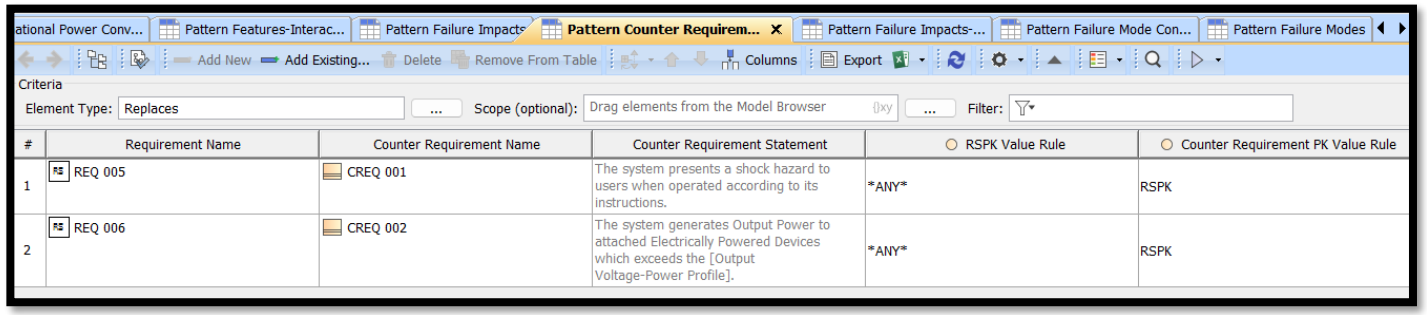
Example Pattern Failure Impacts-Counter Requirements Table

Metarelationships:
Creating Failure Analysis Relationships—Replaces

Homework

- Create a Dependency relationship with the Counter Requirement as the Source and the Requirement as the Target.
- Add a “Replaces” stereotype to the new relationship.





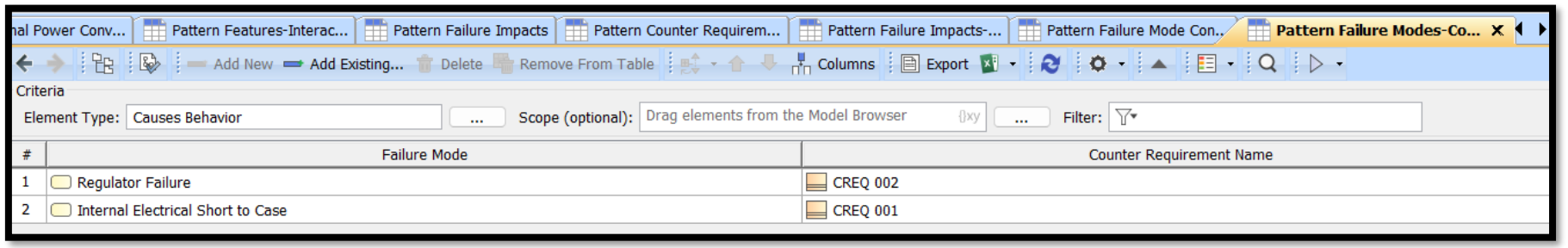
#	Requirement Name	Counter Requirement Name	Counter Requirement Statement	RSPK Value Rule	Counter Requirement PK Value Rule
1	REQ 005	CREQ 001	The system presents a shock hazard to users when operated according to its instructions.	*ANY*	RSPK
2	REQ 006	CREQ 002	The system generates Output Power to attached Electrically Powered Devices which exceeds the [Output Voltage-Power Profile].	*ANY*	RSPK

Example Pattern Counter Requirements Table

Metarelationships:
Creating Failure Analysis Relationships—Causes Behavior

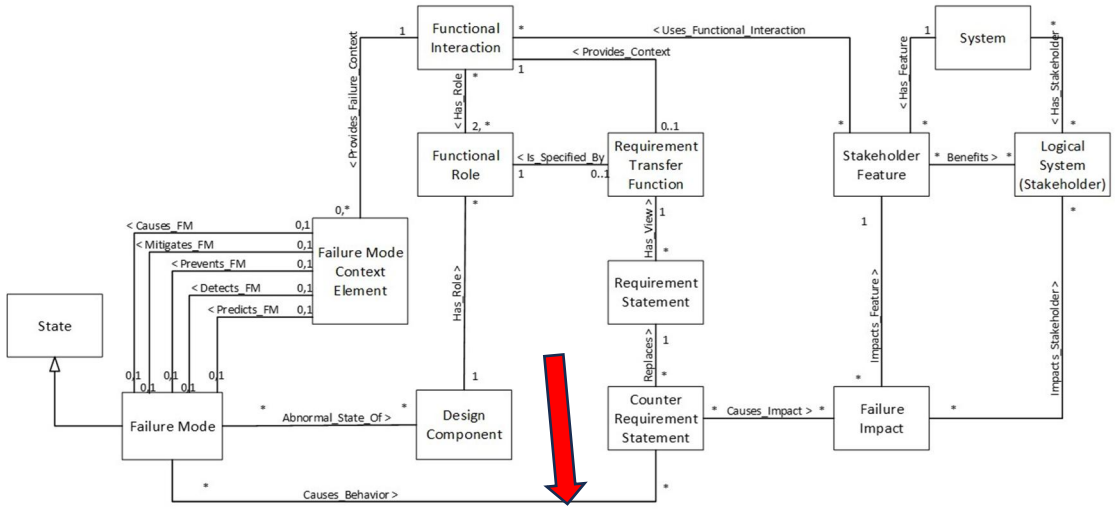
Homework

- Create a Dependency relationship with the Failure Mode as the Source and the Counter Requirement as the Target.
- Add a “Causes Behavior” stereotype to the new relationship.



Example Pattern Failure Modes-Counter Requirements Table

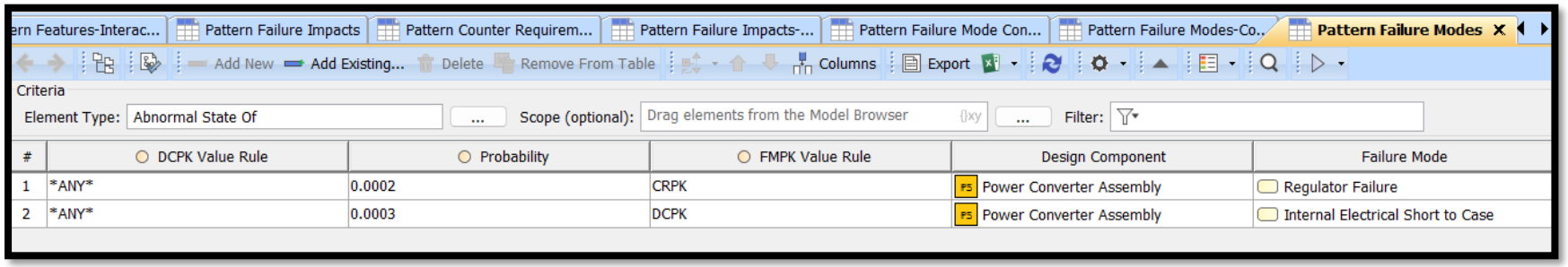
Failure Modes: Risk Analysis View



Metarelationships:
Creating Failure Analysis Relationships—Abnormal State Of

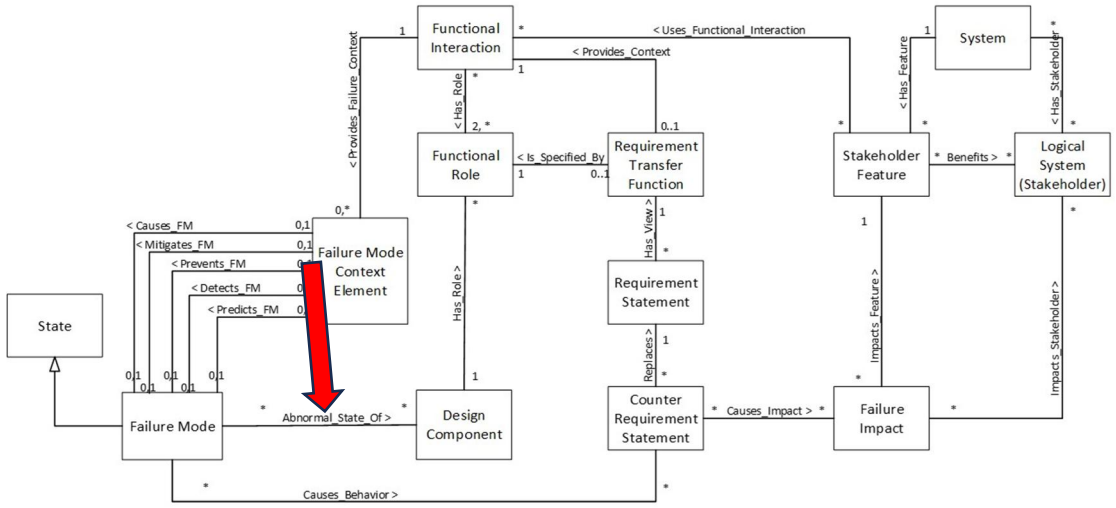
Homework

- Create a Dependency relationship with the Failure Mode as the Source and the Design Component as the Target.
- Add a “Abnormal State Of” stereotype to the new relationship.



Example Pattern Failure Modes Table

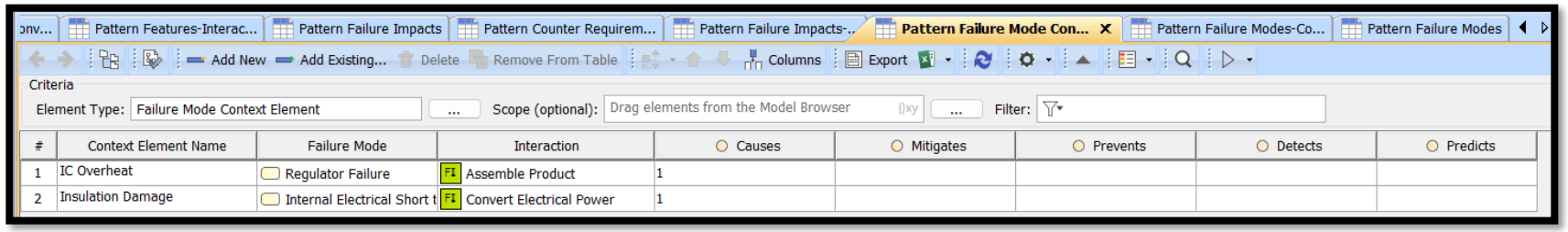
Failure Modes: Risk Analysis View



Metarelationships:
Creating Failure Analysis Relationships—Provides Failure Mode Context

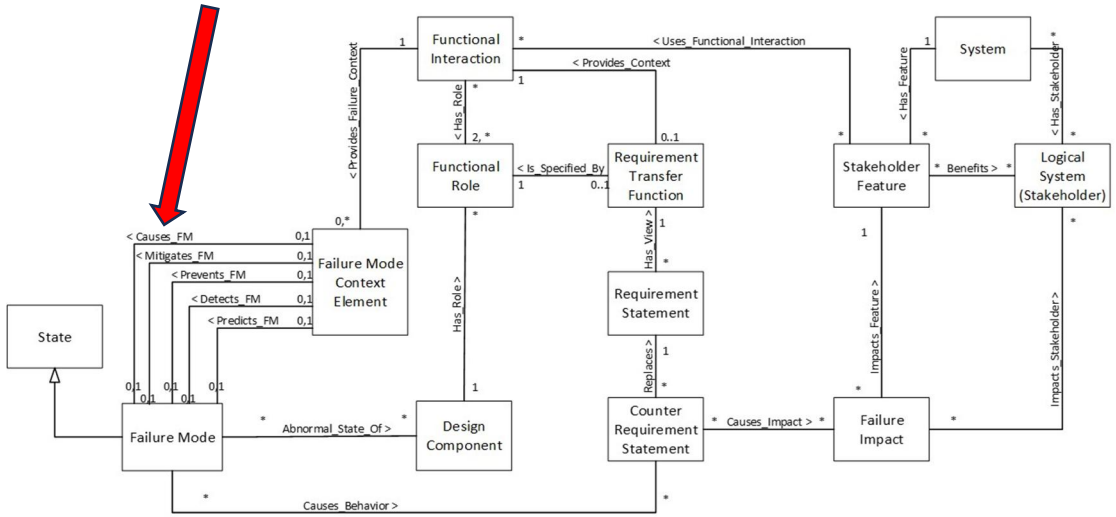
Homework

- Create a Dependency relationship with the Failure Mode Context Element as the Source and the Failure Mode as the Target.
- Add a “Provides Failure Mode Context” stereotype to the new relationship.



Example Pattern Failure Mode Context Table

Failure Modes: Risk Analysis View

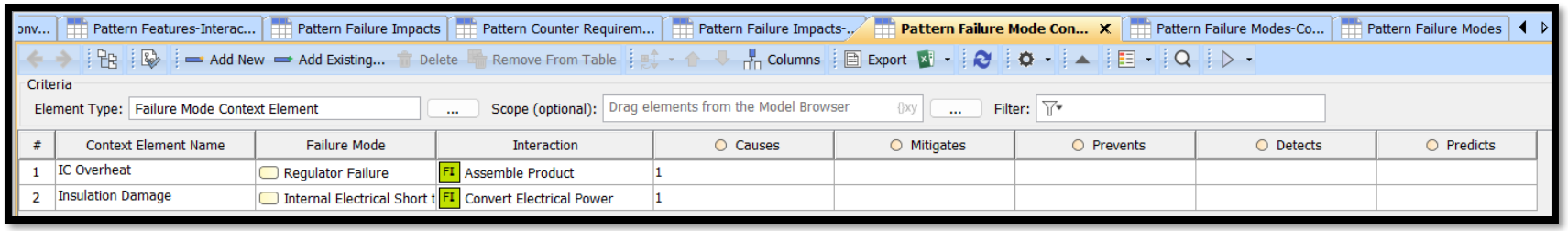


Metarelationships:

Homework

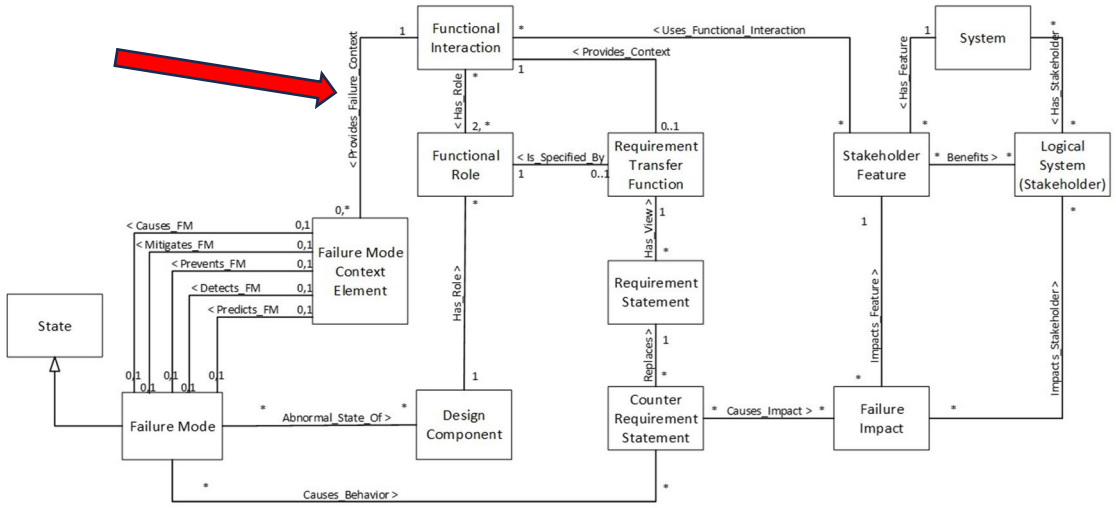
Creating Failure Analysis Relationships—Provides Failure Context

- Create a Dependency relationship with the Failure Mode Context Element as the Source and the Interaction as the Target.
- Add a “Provides Failure Context” stereotype to the new relationship.



Example Pattern Failure Mode Context Table

Failure Modes: Risk Analysis View



Resulting Configured FMEA Table (Shown in Configuration Wizard)

Feature	Failure Impact	FPK Value	Ftr-FPKV	Configured Failure Impact	Counter Requirement Name	Requirement Name	Counter Requirement Statement	Failure Mode	Design Component
Powered Devices Compatibility	Damage to Powered Device	Power Output 1	Powered Devices Compatibility-Power Output 1	Damage to Powered Device [Power Output 1]	CREQ 002	REQ 006	The system generates Output Power to attached Electrically Powered Devices which exceeds the [Output Voltage-Power Profile].	Regulator Failure	Power Converter Assembly
Powered Devices Compatibility	Damage to Powered Device	Power Output 2	Powered Devices Compatibility-Power Output 2	Damage to Powered Device [Power Output 2]	CREQ 002	REQ 006	The system generates Output Power to attached Electrically Powered Devices which exceeds the [Output Voltage-Power Profile].	Regulator Failure	Power Converter Assembly
Powered Devices Compatibility	Damage to Powered Device	Power Output 3	Powered Devices Compatibility-Power Output 3	Damage to Powered Device [Power Output 3]	CREQ 002	REQ 006	The system generates Output Power to attached Electrically Powered Devices which exceeds the [Output Voltage-Power Profile].	Regulator Failure	Power Converter Assembly






Failure Risk Analysis: Insights from Model-Based Systems Engineering



William D. Schindel
ICTT System Sciences
schindel@ictt.com








Presentation for the INCOSE Symposium 2010 Chicago, IL USA 1.3.2

Failure Analysis: Insights from Model-Based Systems Engineering

William D. Schindel
ICTT System Sciences
schindel@ictt.com

Copyright © 2010 by William D. Schindel. Published and used by INCOSE with permission.

Abstract. Processes for system failure analysis (e.g., FMEA) are structured, well-documented, and supported by tools. Nevertheless, we hear complaints that FMEA work feels (1) too labor intensive to encourage engagement, (2) somewhat arbitrary in identifying issues, (3) overly sensitive to the skills and background of the performing team, and (4) not building enough confidence of fully identifying the risks of system failure. In fairness to experts in the process, perhaps such complaints come from those less experienced—but even so, we should care how to describe this process to encourage better technical and experience outcomes. This paper shows how Model-Based Systems Engineering (MBSE) answers these challenges by deeper and novel integration with requirements and design. Just as MBSE powered the requirements discovery process past its earlier, more subjective performance, so also can MBSE accelerate understanding and performance of failure risk analysis--as a discipline deeply connected within the SE process.

What Would We Like to Improve Upon?

Challenges of Traditional Failure Analysis Processes. Processes for system risk and failure identification, analysis, and planning are well-known, documented, and frequently supported by tools. These include Failure Modes and Effects Analysis—FMEA (Dyadem 2002, 2003; ISO/IEC 2006, 2007; US DoD 1980), Fault Tree Analysis—FTA (Hyatt 2003), Reliability Centered Maintenance Planning—RCM (Moubray 1997), Process Hazards Analysis--PHA (Hyatt 2003), and Hazards and Operability Analysis—HAZOP (Hyatt 2003). Those who perform these sometimes voice challenges of these processes, such as the following:

(1) Frequently labor intensive or tedious, adding cost and sometimes discouraging to the energy of those who face the next session;

(2) May overlook certain problems, or feel somewhat arbitrary in identifying issues;

(3) Typically outcome is very sensitive to the skills and background of the performing team;

(4) May not feel systematic in fully identifying the risks of system failure.

These lead us to ask: How can processes for failure identification and analysis be made to feel more systematic and less arbitrary and exhausting? How do we gain assurance we have found all the important failure modes and effects for a system? These and other challenges of traditional systems engineering approaches are being addressed through the use of Model-Based Systems Engineering (MBSE).

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:improving_failure_analysis_using_mbse_v1.3.2.pdf

https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:impact_of_mbse_on_failure_analysis_v1.2.1.pdf

References

1. “System Interactions: Making the Heart of Systems More Visible”, https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:system_interactions--making_the_heart_of_systems_more_visible_v1.2.2.pdf
2. “Formal S*Metamodel”, https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_8_metamodel_v8.0.61.pdf
3. “What Is the Smallest Model of a System?”, https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:what_is_the_smallest_model_of_a_system_v1.4.4.pdf
4. “Maps Versus Itineraries: A Systems Engineering Lesson from Ancient Mariners”, https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:maps_vs._itineraries_v1.3.1.pdf
5. “Requirements Statements Are Transfer Functions”, https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:incose_2005.pdf
6. Guide to Systematica Patterns in Cameo Systems Modeler: https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:guide_to_systematica_patterns_in_cameo_systems_modeler_v1.4.6.pdf

References, continued

7. Anderson, T. and Lee, P. 1981. *Fault tolerance: Principles and practice*. Prentice-Hall (1981) ISBN-10: 0133082547
8. Dyadem. 2002. *Guidelines for failure mode and effects analysis, for automotive, aerospace and general manufacturing industries*, ISBN 0-9731054-1-0. CRC Press, 2002
9. ---. 2003. *Guidelines for failure modes & effects analysis for medical devices*, ISBN 0849319102, Richmond Hill, Ontario, Canada: Dyadem Press, 2003.
10. Estefan, J. 2009. "Survey of model-based systems engineering (MBSE) methodologies", INCOSE-TD-2007-003-01, Rev B
11. Hyatt, N. 2003. *Guidelines for process hazards analysis, hazards identification & risk analysis*. ISBN 0849319099, Richmond Hill, Ontario, Canada: Dyadem Press, 2003.
12. Hybertson, D. 2009. *Model-oriented systems engineering science*. CRC Press, 2009.
13. INCOSE 2010. INCOSE Systems Engineering Handbook, v 3.2.
14. INCOSE 2009. INCOSE Model driven systems design working group we site: http://www.incose.org/practice/techactivities/wg/mdsd/
15. ISO/IEC. 2006. "IEC International standard 60812, analysis techniques for system reliability – Procedure for failure mode and effects analysis (FMEA)", 2006.
16. ---. 2007. ISO/IEC 14971:2007, Medical devices—application of risk management to medical devices. ISO/IEC. 2007.
17. Moubray, J. 1997. *Reliability-centered maintenance*, Second Edition, New York: Industrial Press, Inc., 1997.
18. Powell, M. 2005. "Dealing with uncertainty in systems engineering". INCOSE IS2005 Tutorial F02. July, 2005
19. ---. 2006. "Feelings and physics: emotional, psychological, and other soft human requirements, by model-based systems engineering", *Proceedings of INCOSE 2006 Symposium*, July, 2006.
20. ---. 2010. "Failure Analysis: Insights from Model-Based Systems Engineering", to appear in *Proceedings of INCOSE 2010 Symposium*, July 2010.
21. Schindel, W., and Smith, V. 2002. "Results of applying a families-of-systems approach to systems engineering of product line families", SAE International, Technical Report 2002-01-3086, November, 2002.
22. US DoD 1980. MIL-STD-1629A, "Military standard procedures for performing a failure modes, effects, and criticality analysis", 1980.