Project Overview

Dr. Daniel Hulse

Robust Software Engineering

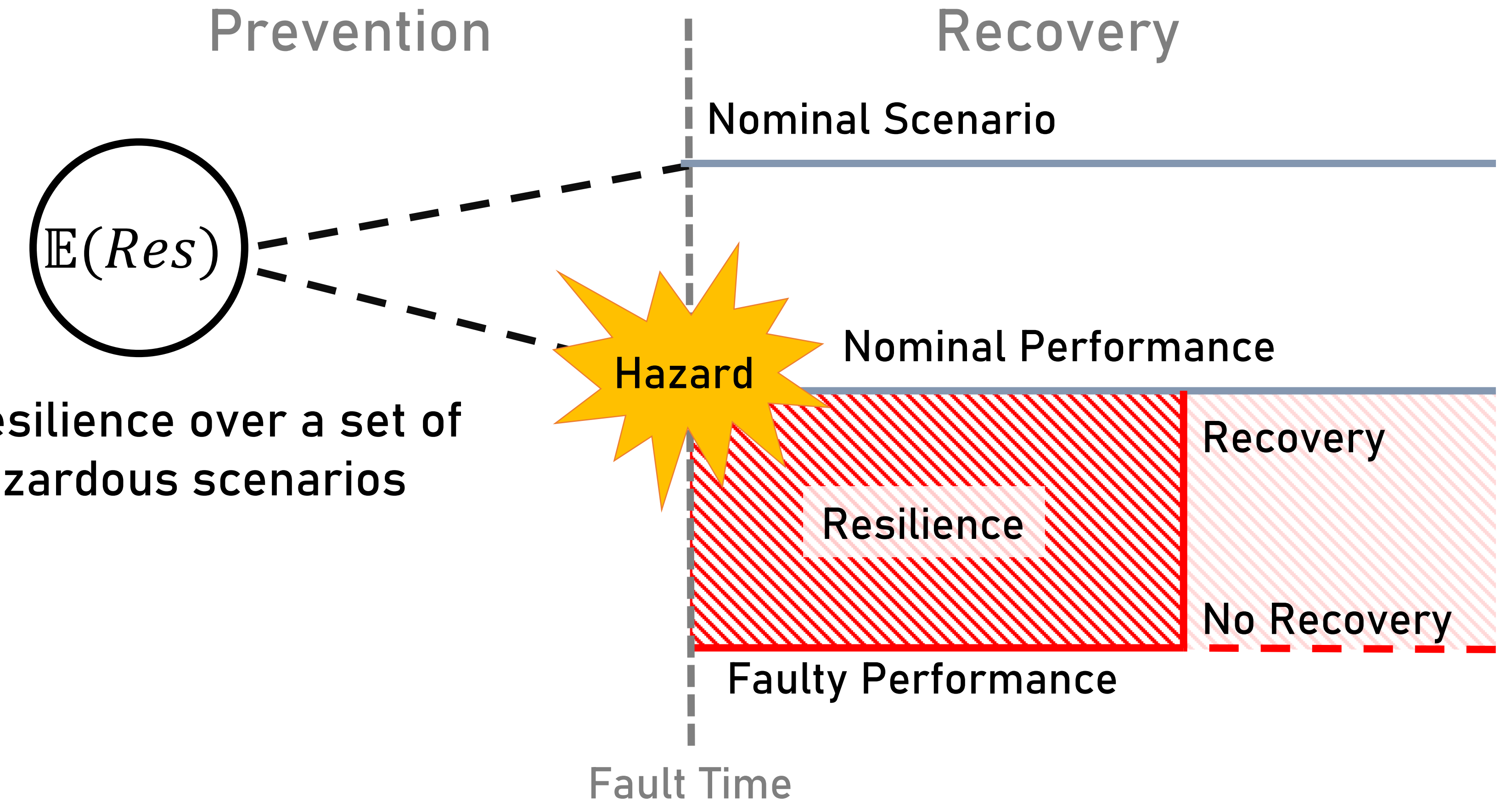
Intelligent Systems Division

NASA Ames Research Center

Presentation Version 1

Current to fmdtools v0.7.5

Goal: Model Early System Resilience



Expected Resilience: Resilience over a set of hazardous scenarios

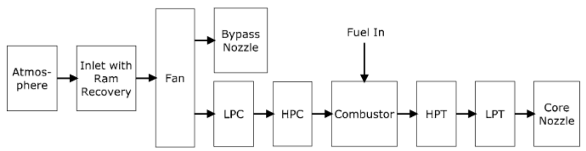
Yodo, N., & Wang, P. (2016).

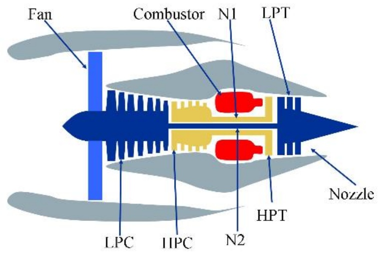
Why model system resilience?

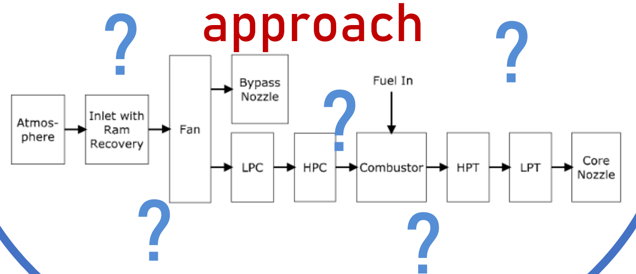
- We want to know how the system will respond to hazardous events, to:
  - Reduce unforeseen maintenance/repairs
  - Ensure system safety
  - Check that system behavior will be robust to unforeseen events
- In a hazardous scenario:
  - Will the system fail to a safe condition?
  - Will it recover?
  - Or, will it wreak havoc on itself, the environment, operators, etc?

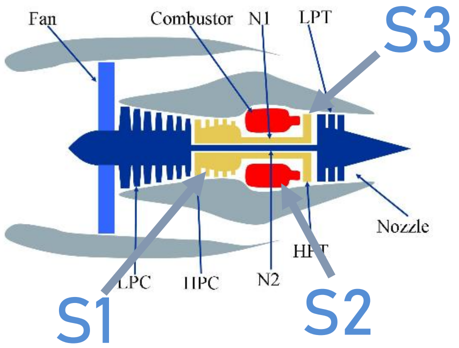
Why include resilience in the design process?

	Concept Design	Embodiment Design	Implementation
Reactive Design	Choose concept	Design system	Retrofit to be resilient
Proactive Design	Establish resilience approach	Design using approach	Verify system resilience









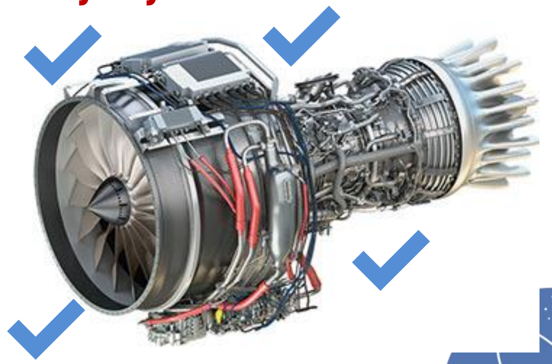
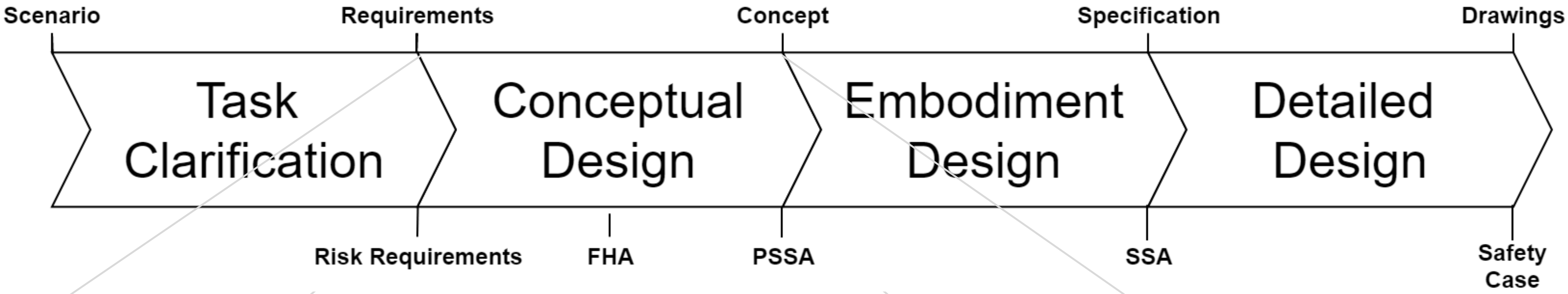


Image 1 & 2 Credit: User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)
Image 3 Credit: https://en.wikipedia.org/wiki/File:General_Electric_Passport.jpg

Risk Assessment in Design

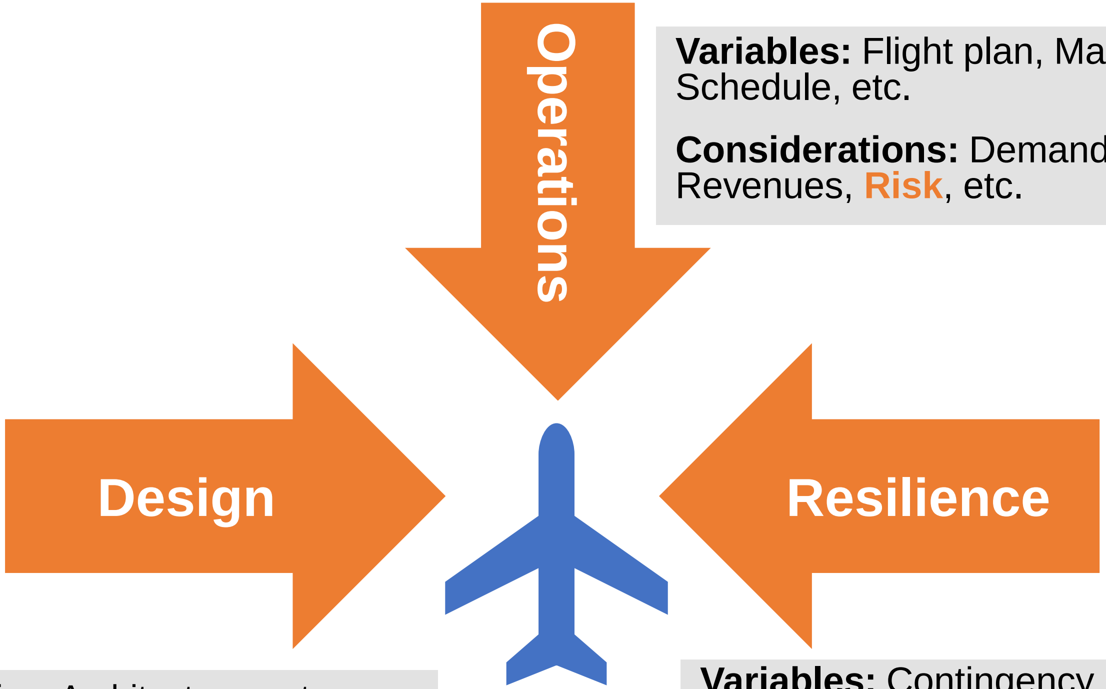


Functional Hazard Assessment:

- Established in SAE ARP4761
- Identifies the high-level hazards in the system structure
- Traditionally a by-hand, expert-driven process
- Issues:
  - Fidelity/Abstraction
  - Not sufficient for **resilience concepts**

Wilkinson, P. J., & Kelly, T. P. (1998). Functional hazard analysis for highly integrated aerospace systems.

Early Design Challenges



Variables: Flight plan, Maintenance Schedule, etc.

Considerations: Demand, Usage, Revenues, **Risk**, etc.

Variables: Sizing, Architectures, etc.

Considerations: Performance, Efficiency, **Component Reliability**, etc.

Variables: Contingency Management, Fault Trim Adjustment, Emergency Procedures, etc.

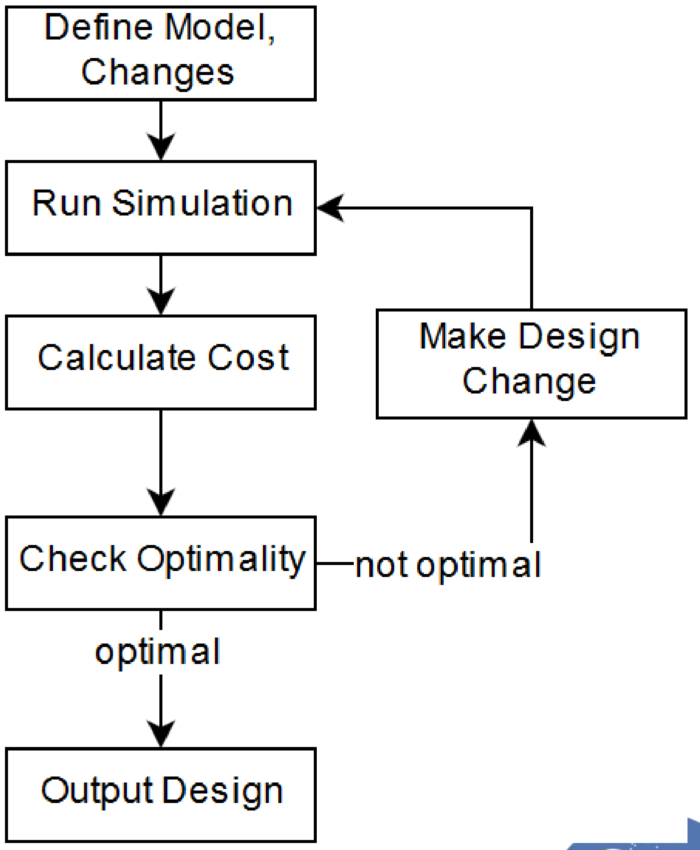
Considerations: **Safety, Damage, Flight Disruption**, etc.

Design, **Operational**, and **Resilience-related** variables and objectives

Need to systematically **manage the complexities** to effectively explore the trade-space

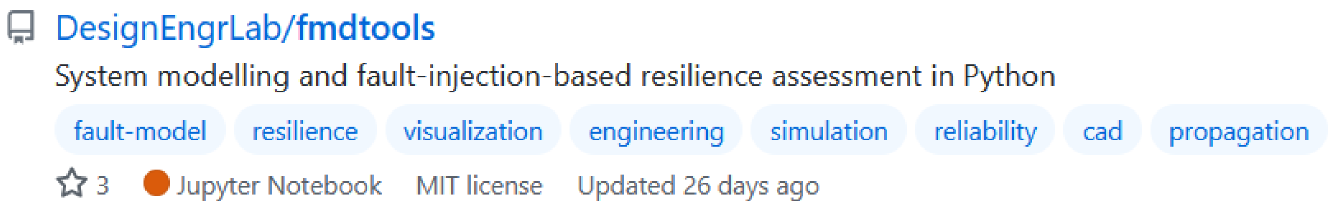
The need for Computational Support

- To enable design space exploration
    - Need computationally-inexpensive high level simulations that can be **parameterized over space of variables**
- To support early resilience incorporation
    - Need ability to simulate large numbers of scenarios to evaluate the **system's inherent robustness** and the **effect of reconfiguration policies**
- To support existing FHA processes
    - From tedious by-hand approach to an **iterative simulation-based approach**



Fmdtools Overview

- Freely available/modifiable/extendable Python module
  - github.com/DesignEngrLab/fmdtools
  - Code-based model definition—define model in script



  - Previously developed in PhD work (v.0.6.2)
  - Development is continuing in RSE group (current version 0.7.5):
    - Lowered computational costs via simulation parallelism and lightweight model formalism
    - Ability to test the system’s response to non-hazard scenarios (stochastic inputs, etc.)

fmdtools

Construct Model
- Function classes
- Model class
- Sample approach

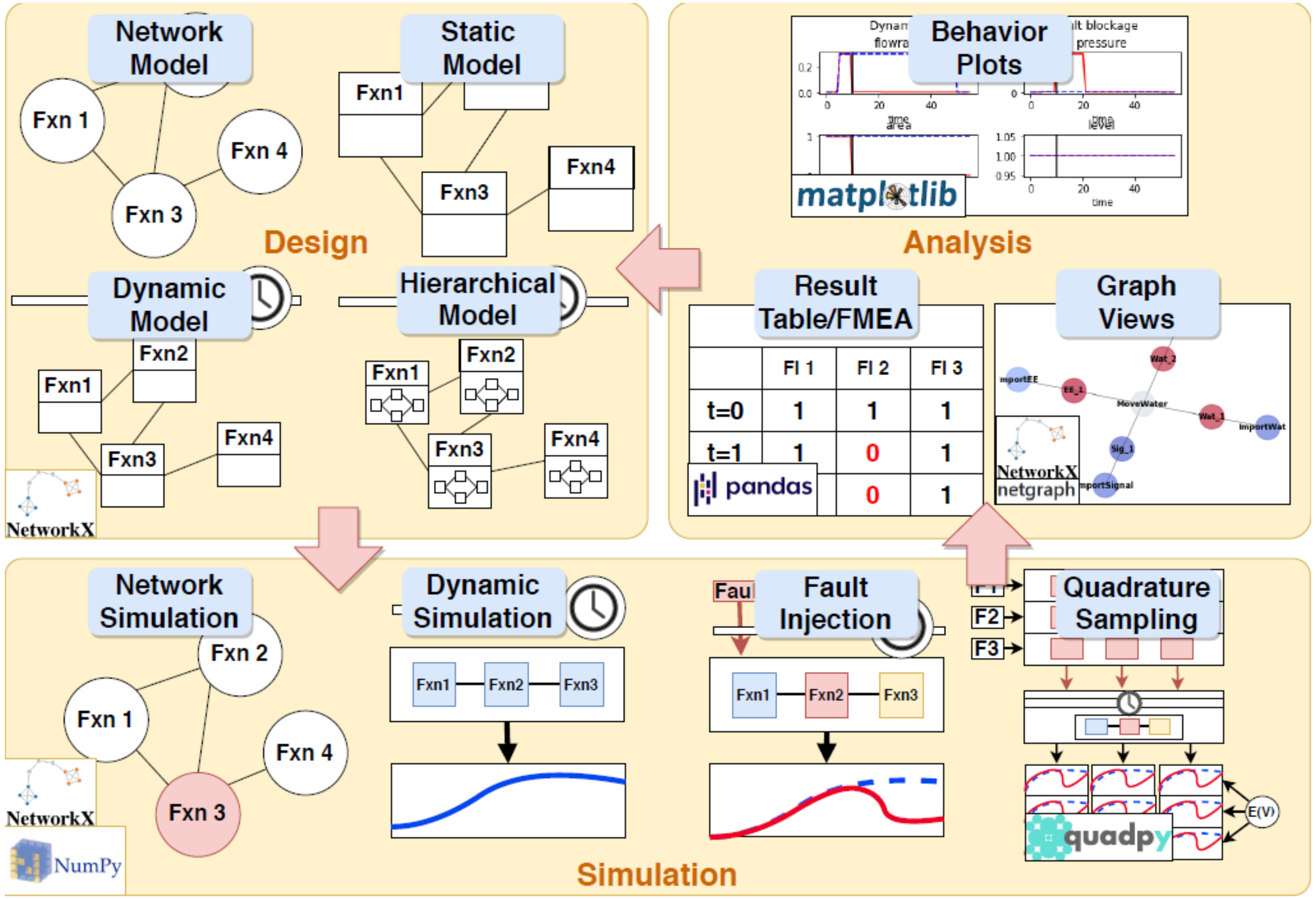
→

Simulate Faults
- Over each fault
- Over each time
- Record result histories

→

Analyze Resilience
- Expected costs
- Behavior over time
- Plots/animations

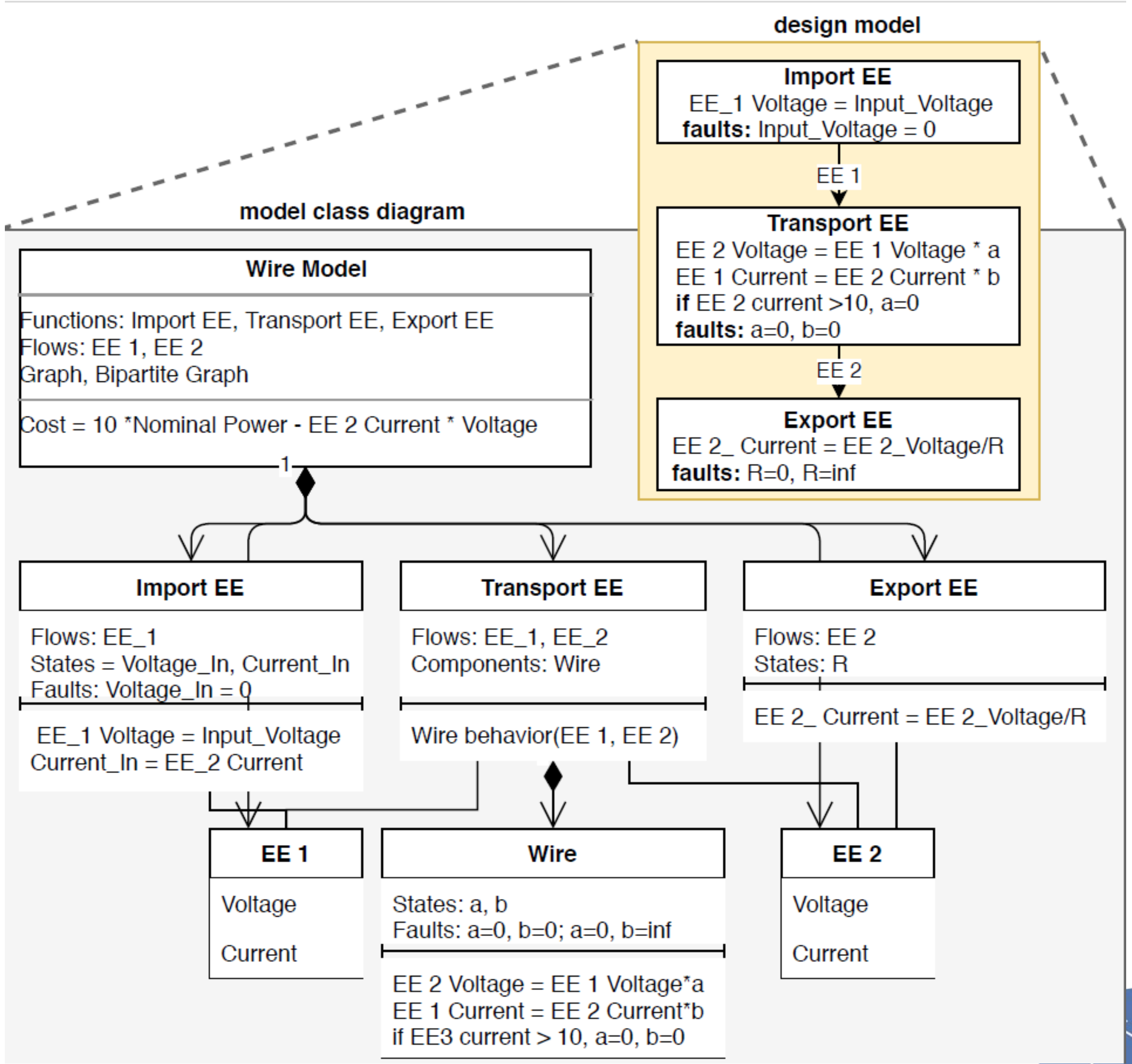
Capabilities



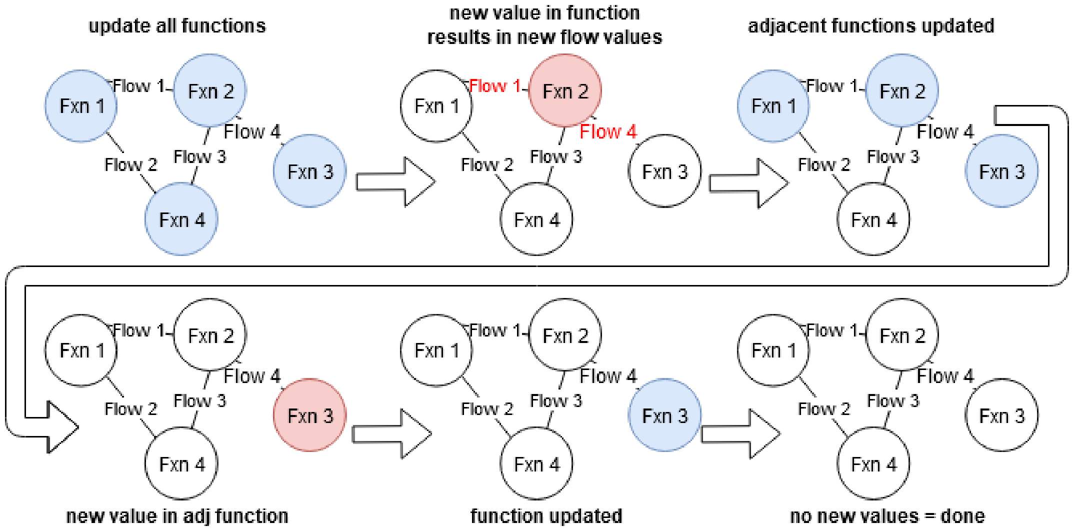
- Constructs (classes) to support differing levels of simulation fidelity
- Appropriate **Simulation and Analysis** methods to match

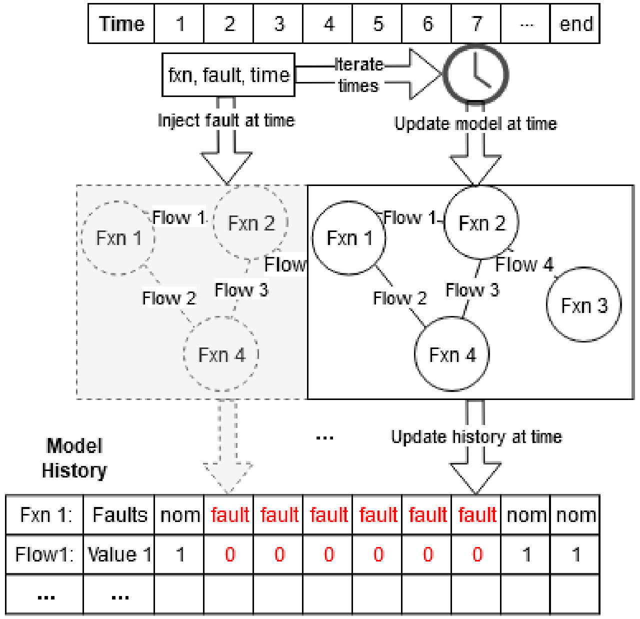
Model structure

- Defined in Model Class
  - Function/Flow relationships
  - Run parameters
  - Classification Method
- Defined in Function Classes
  - Modes, states, parameters
    - (nominal and faulty)
  - Behaviors
    - (dynamic and static)



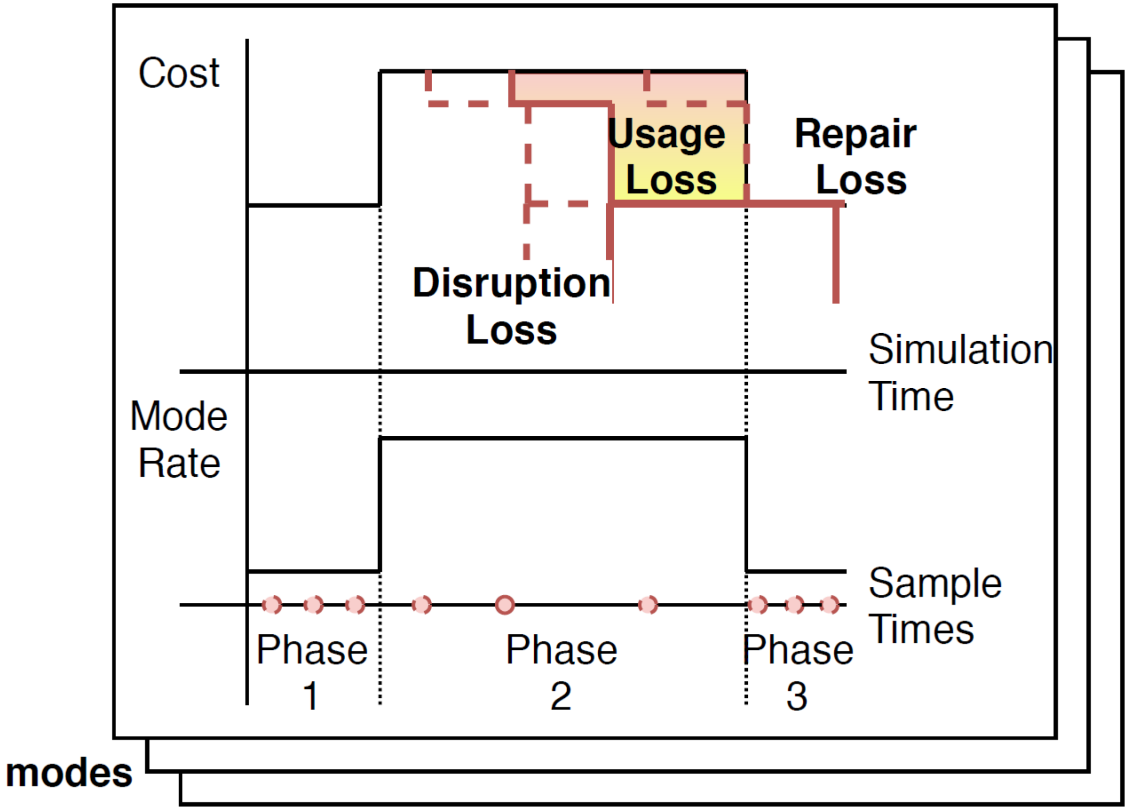
Dynamic and Static Behavior





- Static behavioral propagation
  - Allows faults to propagate through entire model structure
  - Enables timeless (single-timestep) fault propagation models
  - Can be run multiple times per timestep
- Dynamic behavioral propagation
  - Behavior is only run once per timestep
  - Enables faster execution in larger models
  - Captures behavior over time

Fault Sampling approach



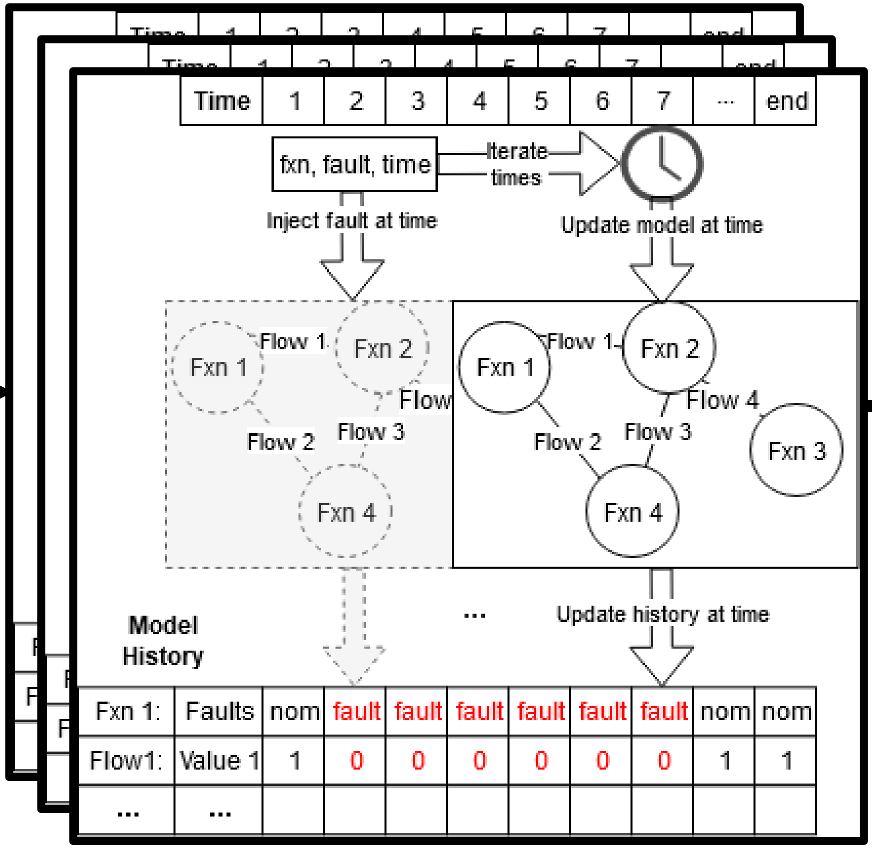
- **Goal:** Quantify expected resilience over a set of fault modes and operational phases
  - What times in the simulation—all? 1-2? Several?
  - What modes—all modes? A selection of modes? Joint fault modes?

Propagation approach

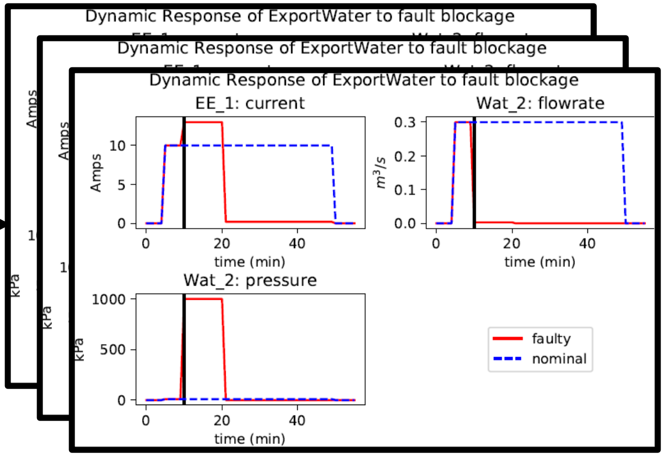
Set of fault scenarios

	Phase 1	Phase 2	Phase 3
Fault 1	1,3,5	20,30,40	
Fault 2		20,30,40	
Fault 3			42,43,44

Injection of each scenario

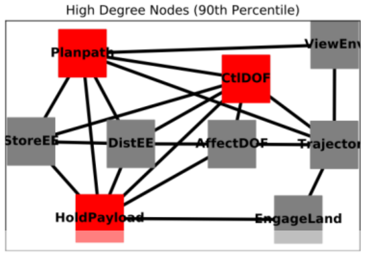


Results for each scenario

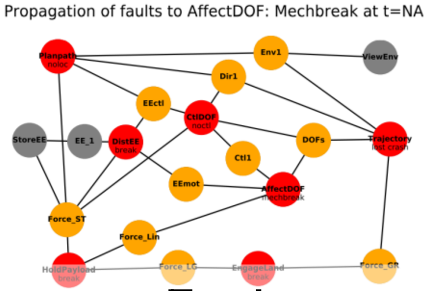


Analyze Resilience

ASPL	Modularity	Robustness Coefficient
1.44	0.12	95.85



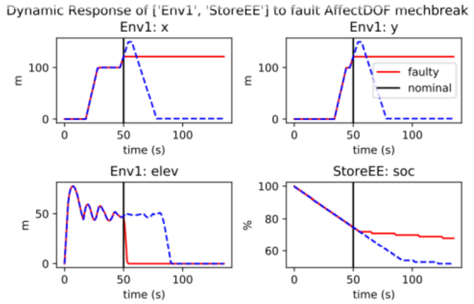
Network Analysis
and
Graph Visualization



Fault
Propagation
Visualization

Fault	Degraded Functions	Degraded Flows	Rate	Cost	Exp. Cost
StoreEE nocharge	StoreEE, DistEE, CtlDOF...	Force_ST, Force_Lin, Force_GR...	1e-05	183300	183300
Planpath degloc	DistEE, CtlDOF, Planpath...	Force_ST, Force_Lin, Force_GR...	8e-06	193000	154400
DistEE short	DistEE, CtlDOF, Planpath...	Force_ST, Force_Lin, Force_GR...	3e-06	186000	55800
AffectDOF ctlbreak	DistEE, AffectDOF, CtlDOF...	Force_ST, Force_Lin, Force_GR...	2e-06	184000	36800
AffectDOF ctlup	DistEE, AffectDOF, CtlDOF..	Force_ST, Force_Lin, Force_GR...	2e-06	183500	36700
DistEE break	DistEE, CtlDOF, Planpath...	Force_ST, Force_Lin, Force_GR...	2e-06	183000	36600
CtlDOF noctl	DistEE, CtlDOF, Planpath...	Force_ST, Force_Lin, Force_GR...	2e-06	183000	36600
AffectDOF short	DistEE, AffectDOF, CtlDOF...	Force_ST, Force_Lin, Force_GR...	1e-06	186200	18620
AffectDOF mechbreak	DistEE, AffectDOF, CtlDOF...	Force_ST, Force_Lin, Force_GR...	1e-06	183500	18350
AffectDOF openc	DistEE, AffectDOF, CtlDOF...	Force_ST, Force_Lin, Force_GR...	1e-06	183200	18320
Planpath noloc	Planpath, Trajectory	Ctl1, DOFs, Dir1	2e-06	60000	12000
CtlDOF degctl	CtlDOF	Force_GR, Force_LG, Ctl1, DOFs	8e-06	10000	8000
AffectDOF propbreak	DistEE, AffectDOF, CtlDOF...	Force_ST, Force_Lin, Force_GR...	3e-07	183200	5496
AffectDOF propstuck	DistEE, AffectDOF, CtlDOF...	Force_ST, Force_Lin, Force_GR...	2e-07	186200	3724
HoldPayload break	DistEE, CtlDOF, Planpath...	Force_ST, Force_Lin, Force_GR...	2e-07	183000	3660
ViewEnv poorview	ViewEnv		2e-06	10000	2000
EngageLand deform	EngageLand		8e-06	1000	800
HoldPayload deform	HoldPayload	Force_ST, Force_Lin	8e-07	10000	800
DistEE degr	DistEE	Force_GR, Force_LG, EEmot...	5e-06	1000	500
EngageLand break	EngageLand		2e-06	1000	200
AffectDOF ctldn	AffectDOF	Force_GR, Force_LG, DOFs	2e-06	500	100
AffectDOF mechfriction	AffectDOF	EE_1, EEmot	5e-07	500	25
AffectDOF propwarp	AffectDOF	Force_GR, Force_LG, DOFs	1e-07	200	2

Static/Dynamic
FMEA/FHA



Nominal/Faulty
Dynamic Resilience
Plots

- Visualizations for faulty behavior
  - What happens in a given hazardous scenario?
  - What is the scope of fault propagation?
- Metrics for summarizing sets of scenarios
  - What functions/flows are most subject to faults?
  - What faults pose the most risk?

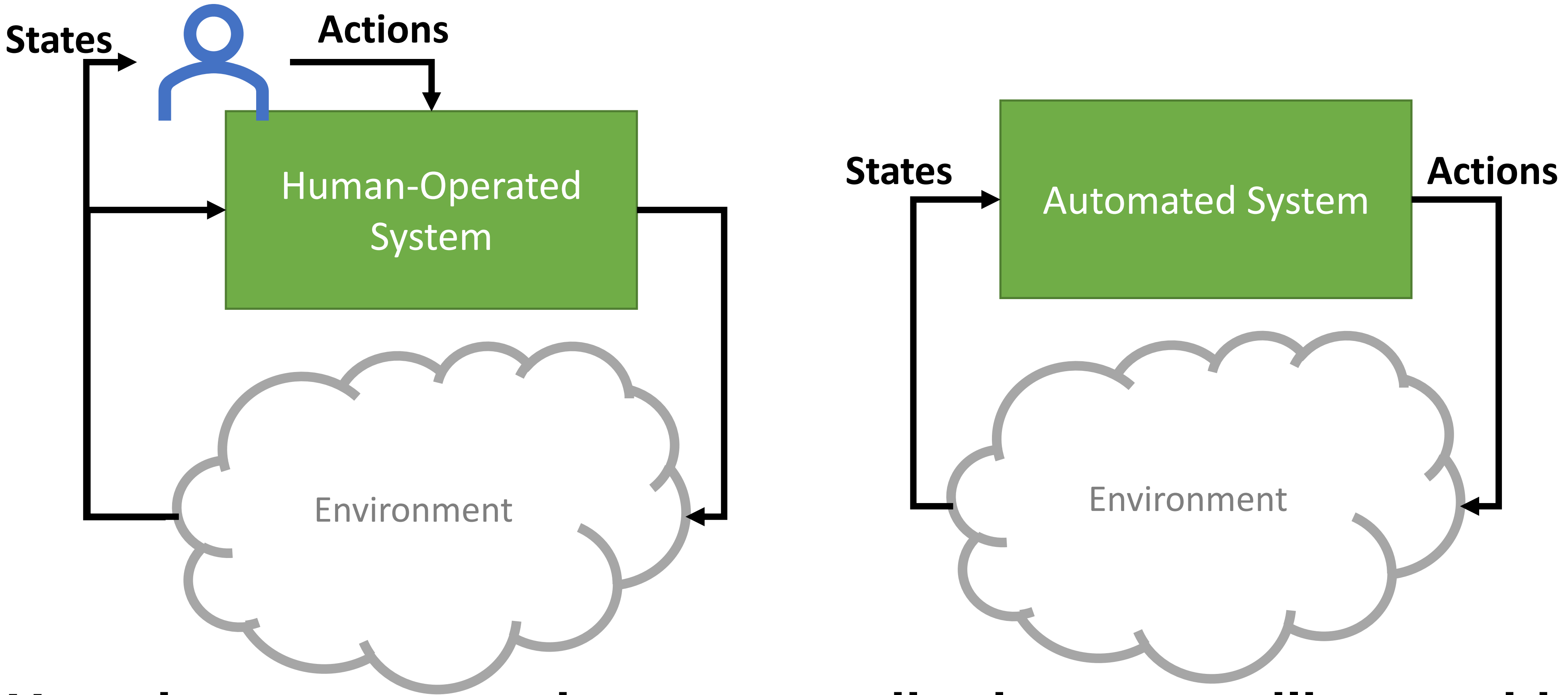
Examples and Use-cases

- Goal: Demonstrate how the fmdtools framework can be used in a real design process
- Previously-developed models (available in github repo)
  - Pump
  - Hold-up tank
  - Drone
  - Pandemic model
  - Electric Power System

Illustrate basic usage of the toolkit, model setup, method syntaxes, etc.

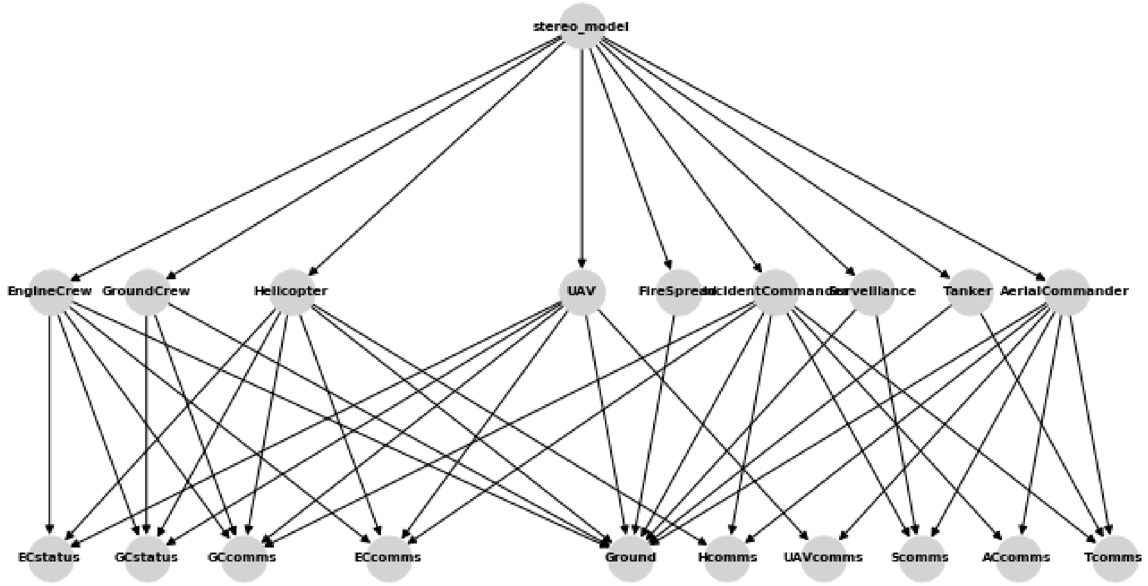
- Ongoing work in advancing the method
  - Smart-Stereo Wildfire Response Model **- modelling complex SoS**
  - Lane-following Rover Model **- modelling human/AI hazards**

Future/Ongoing Research Questions



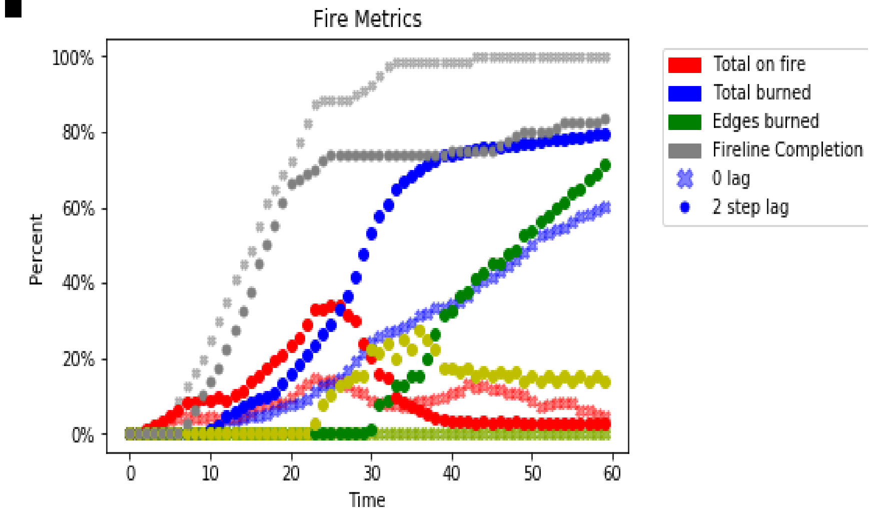
How do we compare human contributions to resilience with AI-enabled automated systems?

SMARt-STEReO model

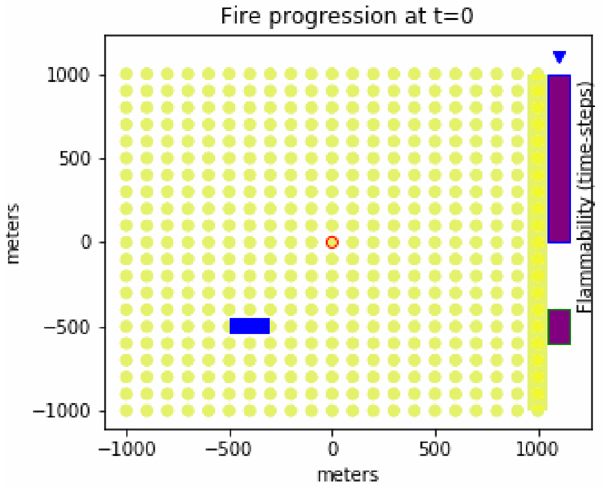


Model of aerial fire response resilience:

- **fire** propagation
- **aerial surveillance** and **firefighting**
- **ground crews** and **fire-line** construction

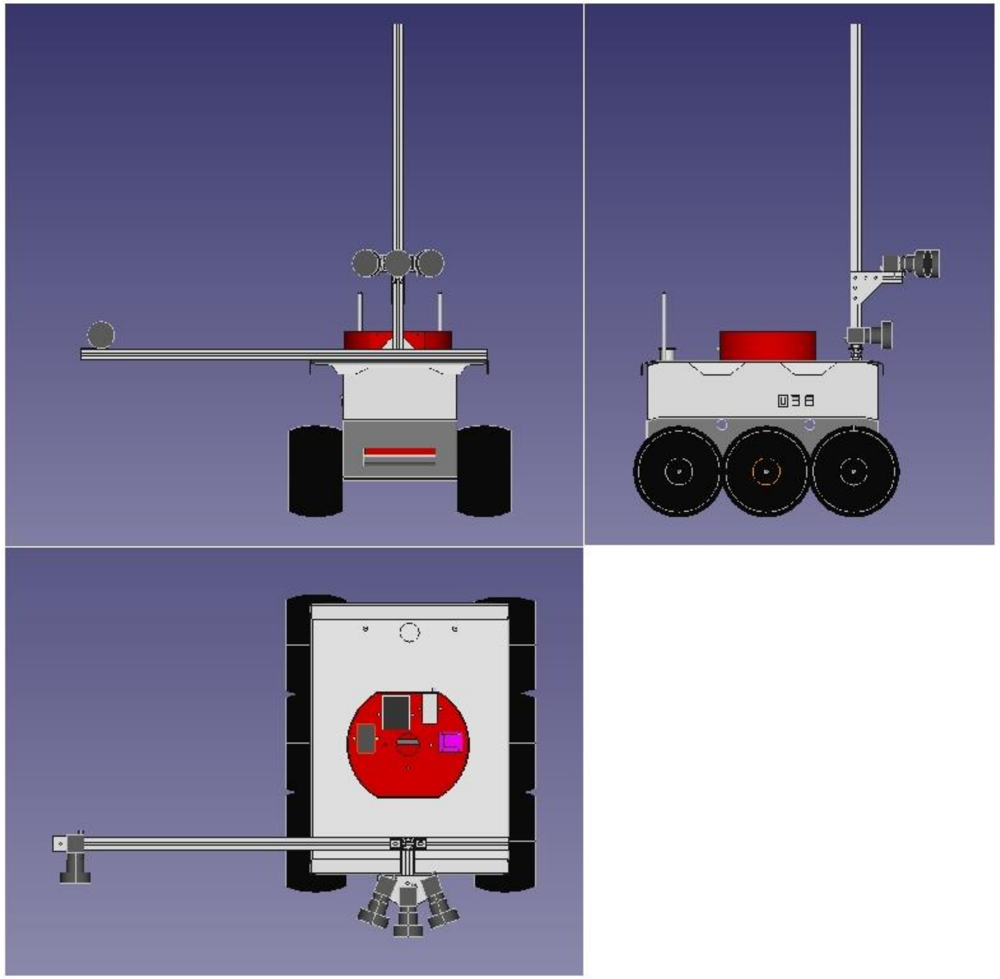


Performance of response over time

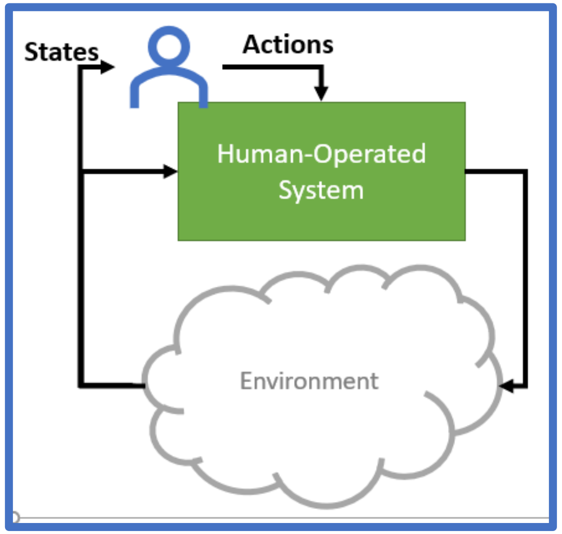


Use-Case: Lane-following Rover

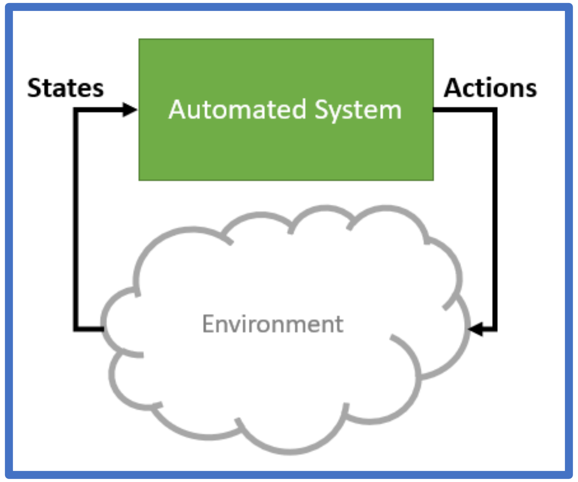
- Test set-up in development in RSE
  - Adaptation of **taxinet** concept
  - Rover follows lanes using AI/ML-based computer vision
  - Used to test research methods
- **Goal** of this model:
  - Understand **safety and performance considerations for AI/ML** in simulation
    - Comparing **AI/ML** with **Human contributions** to resilience
  - **Verify/validate** simulation approach with real system
  - **Workshop** general tools/methods for similar AI/ML problems (resolve gaps)



Expected Results



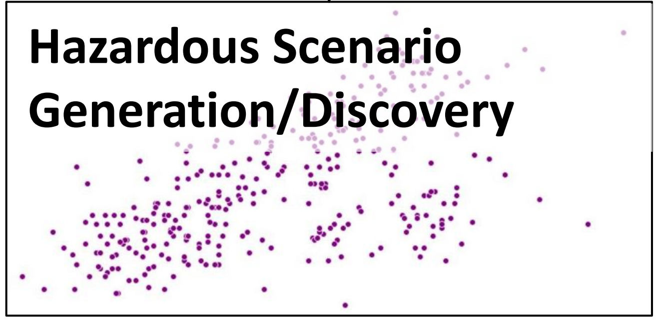
- External and External Hazards
- Human-Induced Hazards
- Human Hazard Controls



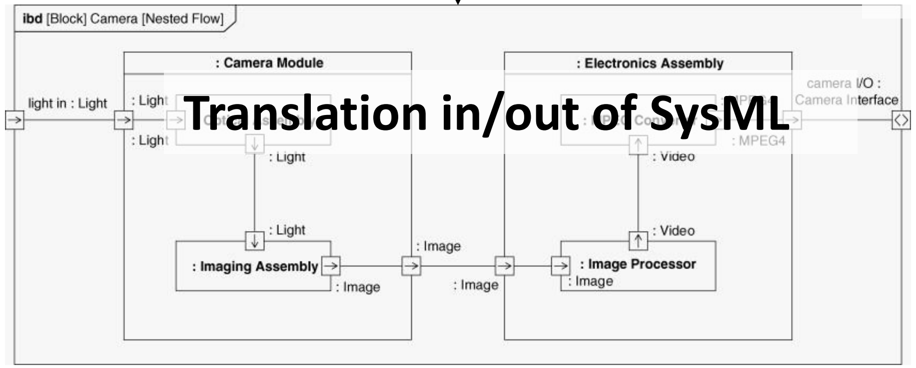
- External and External Hazards
- Human-Induced Hazards
- AI-enabled system hazard control requirements

- Systematic methodology for:
  - identifying and simulating AI & human failure modes
  - identifying and simulating AI & human hazard response
- Then we can understand when to use each, required controls, etc.

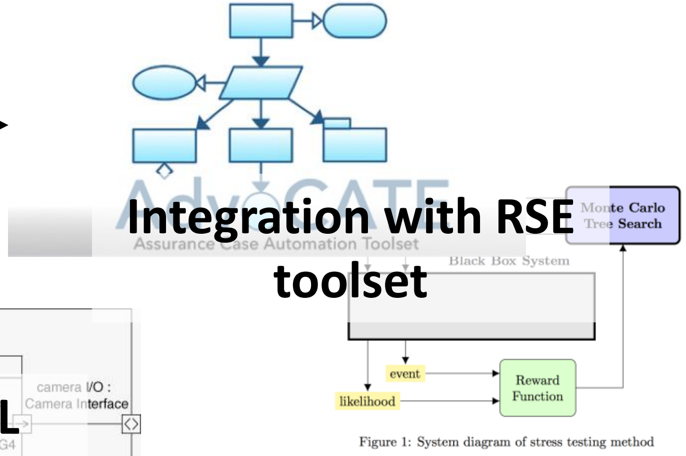
Long-term directions



Hazardous Scenario Generation/Discovery



Translation in/out of SysML



Integration with RSE toolset

Discussion