Concept-of-Operations for Testing AI Systems: Fast-time Emergent Scenario Simulation (FTESS)

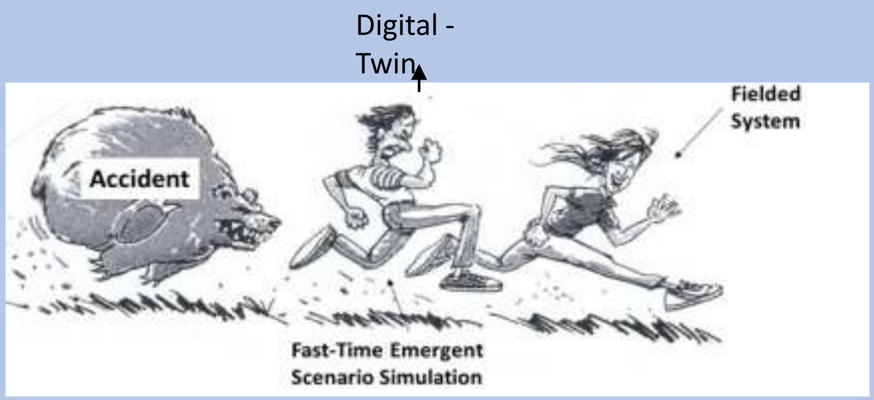
Lance Sherry (lsherry@gmu.edu)

John Shortle, Jie Xu, Jim Baldo, Brett Berlin, C.H. Chen, Ali Raz

Jomana Bashata, Charlie Wang

Current Paradigm

- **Development Phase:**
  - **Limited by imagination to uncover all "unknown-knowns"**
- **Operations Phase:**
  - **Terminate testing on fielding**
  - **Avoid "corner-cases" and pre-cursors**



Proposed Paradigm

- **Development Phase:**
  - **Use Digital Twin to uncover "known-unknows"**
- **Operations Phase:**
  - **Keep testing even after fielding**
  - **Focus on "corner-cases" and pre-cursors**

Table of Contents

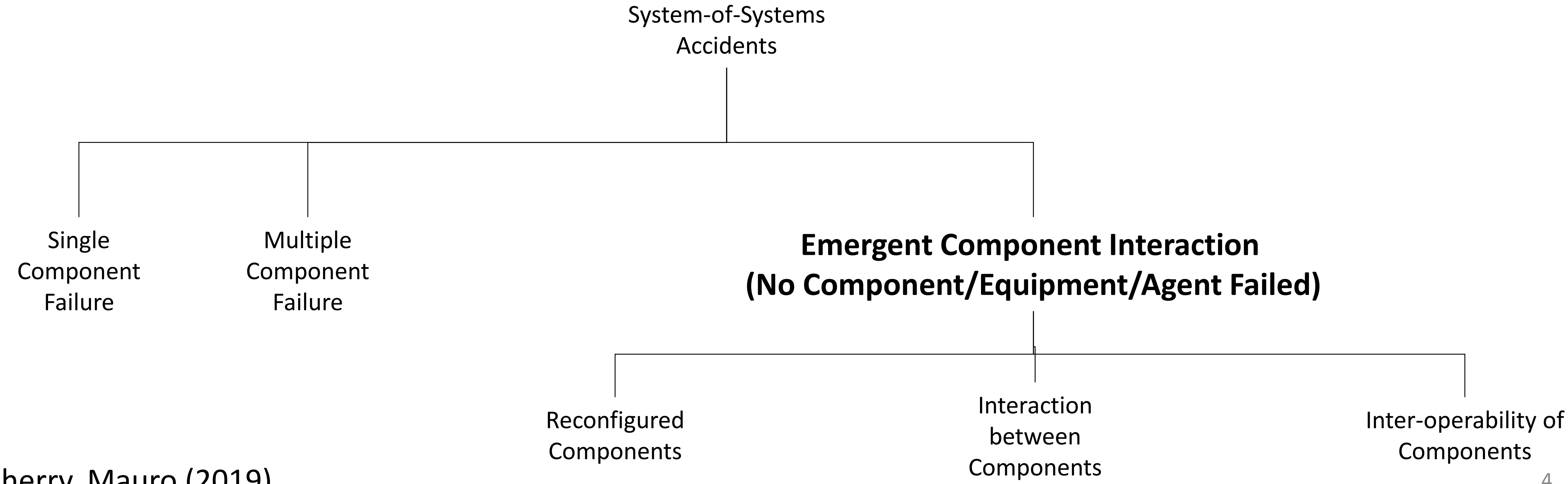
Background – Interaction Accidents

- Accident investigation across multiple domains
- Two types of Accidents
  1. <u>Component failure drives</u> a component/system into hazardous operating regime
     - Taum Sauk Dam (over-topping)
     - Southwest Airlines Flight 1380 (contained engine failure)

  2. No component fails, <u>but unanticipated interaction between components</u> drives a component into hazardous operating regime
     - Munich Airport Runway Excursion/Singapore Airlines 237
     - Three Mile Island Nuclear Accident

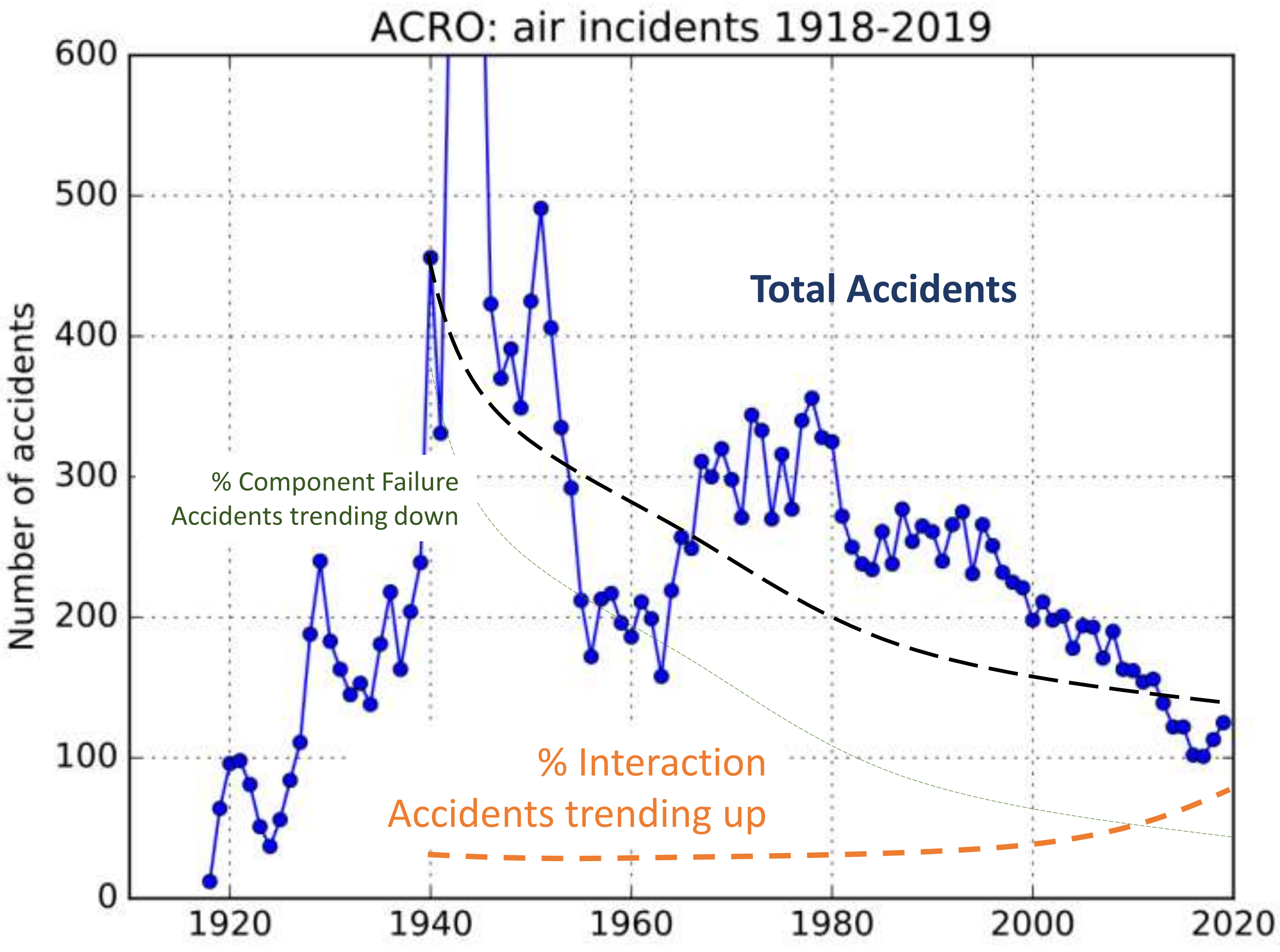
Background – Accident Categories

- Not all accidents/mishaps caused by **component failures**
  - Anatomy of "No-Equipment Failed" Malfunctions (Sherry, Mauro, 2014, 2017a; 2017b, 2018, 2019)



Sherry, Mauro (2019)

Background – Accident Categories



Hypothesis: % Interaction Accidents dominant type of accident

Background – Component Interaction Accidents

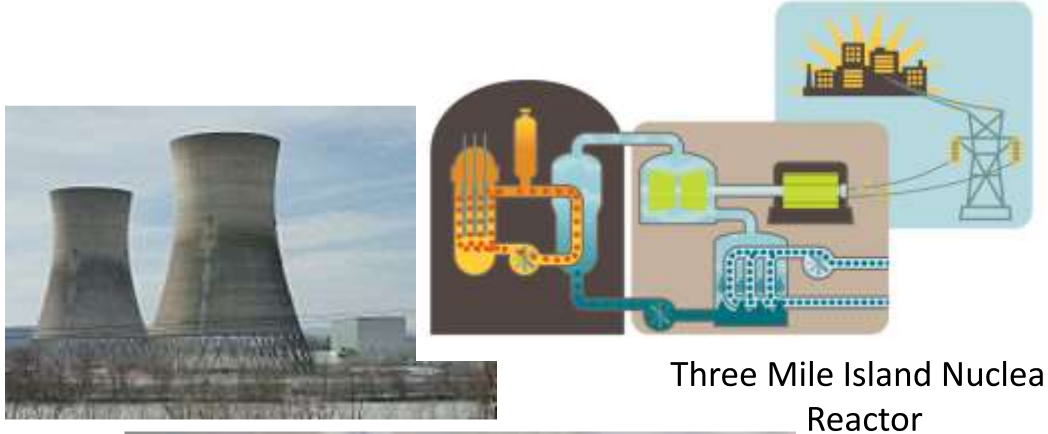
- “Normal Accidents” Perrow (1984)
  - Functional Interaction Complexity Failures/Malfunctions (FICFs) (Sherry et. al., 2014 -20)

- “Normal Accident” Criteria:

1. The System is complex
2. The System is composed of tightly coupled components
3. The System has catastrophic potential when operating in a hazardous operating regime
4. No component fails
5. System (or component migrates into hazardous operatting region

- “Normal Accident” Scenario:

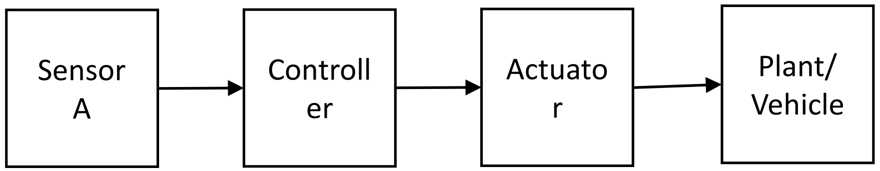
1. System starts the fire
2. System disables the fire extinguisher
3. System provides ambiguous cues (that prevent intervention)



Three Mile Island Nuclear Reactor

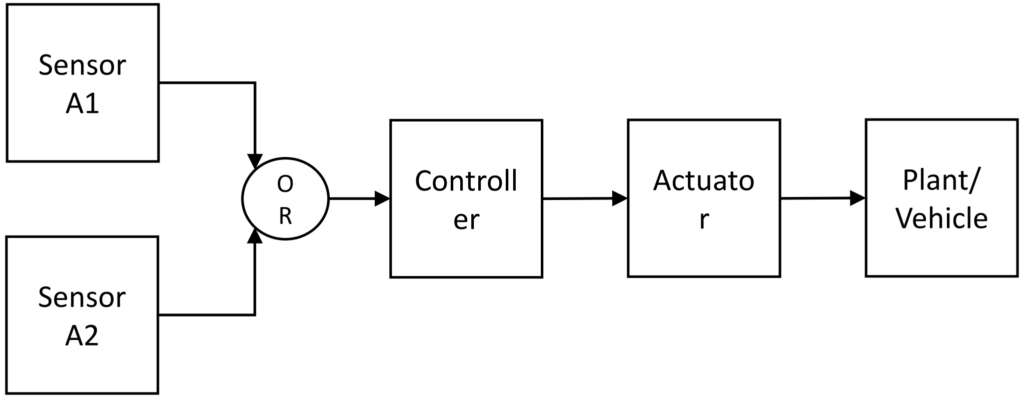
Munich Airport Runway Excursion

Background – Accident Categories



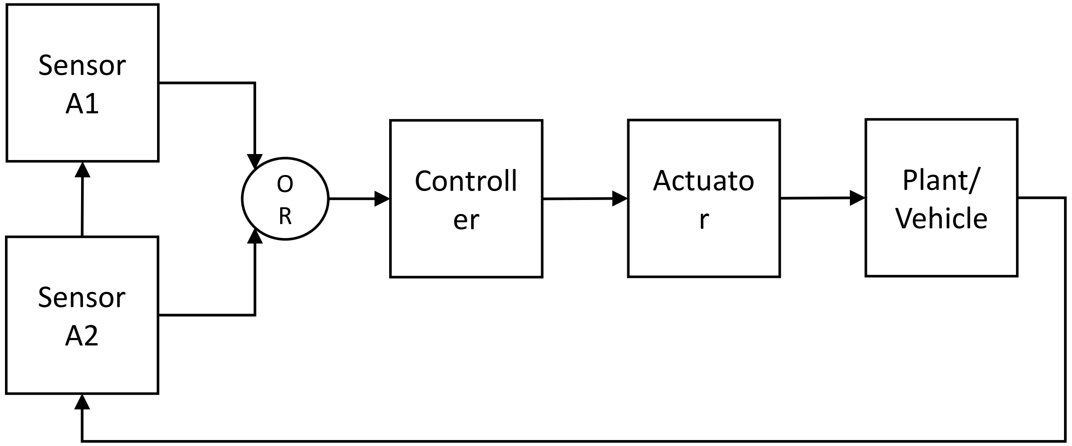
TO / FROM	Sensor A	Controller	Actuator	Plant/Vehicle
Sensor A		Input to		
Controller			Input to	
Actuator				Input to
Plant/Vehicle				

Single Component Failure



TO / FROM	Sensor A1	Sensor A2	Controller	Actuator	Plant/ Vehicle
Sensor A1			Input to		
Sensor A2			Input to		
Controller				Input to	
Actuator					Input to
Plant/ Vehicle					

Multiple Component Failure



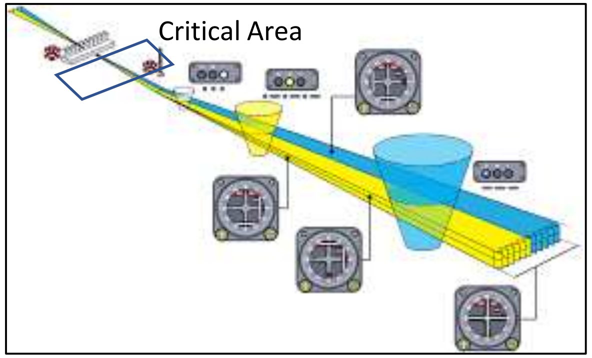
TO / FROM	Sensor A1	Sensor A2	Controller	Actuator	Plant/ Vehicle
Sensor A1			Input to		
Sensor A2	Input 2		Input to		
Controller				Input to	
Actuator					Input to
Plant/ Vehicle		Input to			

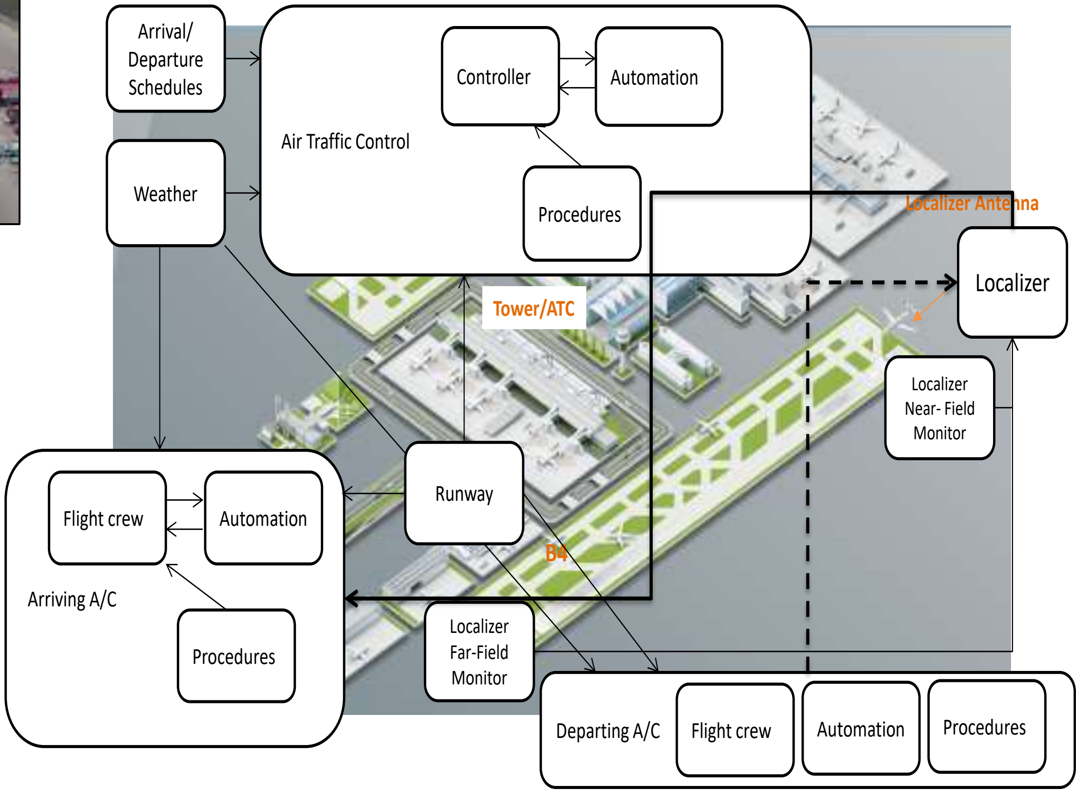
No Component Failure/Component Interaction

Background: Example Interaction Accident:
Munich Airport Runway Excursion/SQ237

<u>System Components:</u>

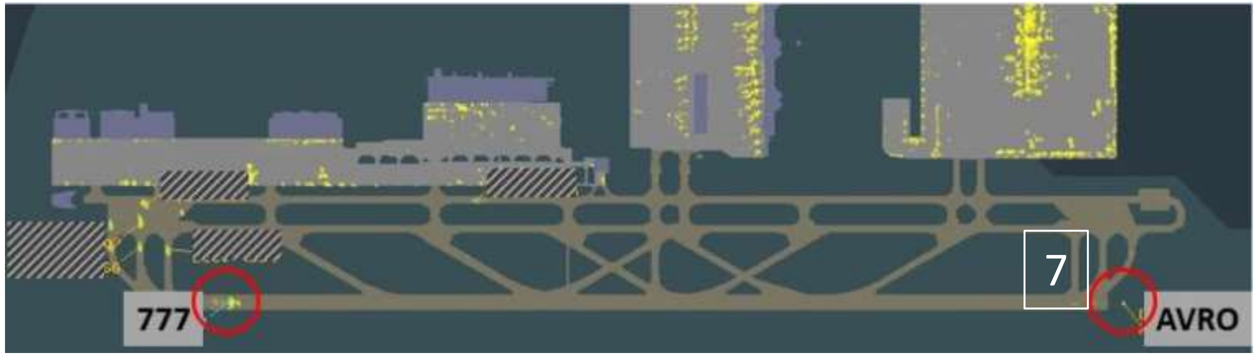
1. Air Traffic Control
    1. Procedures
    2. Automation
    3. Controller
2. Departing Aircraft
    1. Procedures
    2. Automation
    3. Flight crew
3. Arriving Aircraft
    1. Procedures
    2. Automation
    3. Flight crew
4. Airport Arrival/Departure Schedule
5. Weather
6. Runway
7. Localizer
    1. Localizer Near-field Monitor
    2. Localizer Far-field Monitor





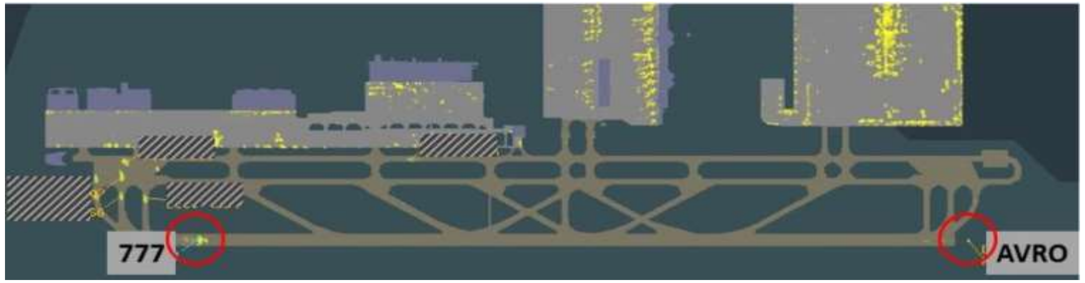
Background: Example Interaction Accident: Munich Airport Runway Excursion/SQ237

1. To accommodate A380, airport moves Localizer antennae away from runway end (changes ILS Critical Area)
2. Low Visibility conditions causes long departure queue at airport
3. Air Traffic Controller, trying to expedite departures, clears Avro for mid-runway takeoff
4. Air Traffic Controller clears SQ237 for approach
5. 777 decides to "practice" CAT III automatic landing
6. Avro takeoff roll to end of runway and lift-off
7. Localizer signal is deflected (due to Avro)
8. 777 Automatic Landing System follows deflected Localizer signal and lands adjacent the runway
9. 777 weight-on-wheels inhibits Go Around button selection by flight-crew to intervene



Background: Example Interaction Accident: Munich Airport Runway Excursion/SQ237

Interaction Between Components		Interaction
ATC Procedures	Singapore Airline Procedures	*Not compatible*
Aircraft Approach type (Cat III)	ATC departure (mid-runway departure)	*No coordination required*
Avro midfield takeoff impacts ILS Critical Area to potentially affect Localizer Signal	777 Localizer deflection monitoring	*Attention Focus for quick response*
Localizer	Localizer monitoring	*Response Time*
Localizer Signal	Flight-crew response time to recognizing deflection	*Response Time*
777 automation inhibit Go Around button	Flight-crew response	*Timing*



Background – Some History

- SDI Software – David Parnas – December 1985
  - Unreliable SW
  - Program verification cannot provide reliable SDI SW
  - SDI SW unattainable
- No Silver Bullet - Essence and Accidents of SW Engineering – Fred Brooks – April 1987
  - “I believe the hard part of building software to be the specification, design, and testing of this conceptual construct, not the labor of representing it and testing the fidelity of the representation.”
    - Complexity
    - Conformity
    - Changeability
    - Invisibility
- Trustworthy AI – Jannette Wing – October 2021
  - SW trustworthiness properties need to be extended beyond reliability, security, privacy, and usability to include properties such as:
    - Probabilistic accuracy under uncertainty
    - Fairness
    - Robustness
    - Accountability
    - Explainability

Conceptual
Provide ASsurance

The requirements of a strategic defense system

In March 1983, President Reagan said, “I call upon the scientific community, who gave us nuclear weapons, to turn their great talents to the cause of mankind and world peace; to give us the means of rendering these nuclear weapons impotent and obsolete.”

To satisfy this request the software must perform the following functions:

—Rapid and reliable warning of attack
—Determination of the source of the attack
—Determination of the likely targets of the attack
—Determination of the missile trajectories
—Coordinated interception of the missiles or warheads during boost, midcourse, and terminal phases, including assignment of responsibility for targets to individual sensors or weapons
—Discrimination between decoys and warheads
—Detailed control of individual weapons
—Evaluation of the effectiveness of each attempt to destroy a target.

Background – Safety Engineering Paradigm

Hazard-free Environment

Hazardous Environment with Environmental Protections

Hazardous Environment with Protective Equipment

Hazardous Environment with Warnings & Alerts

Hazardous Environment with Warnings & Alerts, and Safe Operator Procedures and Training

Background –Traditional Safety, V&V

- Traditional “Safety Analysis” Techniques
  - Hazard Analysis (lists of known hazards)
  - Failure Modes and Effects Analysis (list of component failures)
  - Fault-Tree Analysis (single hazard)
  - Event Trees (single fault)
- Characteristics of Traditional “Safety Analysis” Techniques
  - Assumes one or more ***component failures***
  - Assumes knowns, and known/unknowns (i.e. not unknown/unknowns)
  - Assume component configuration is static (i.e. not configurable)
  - Assume components behavior is deterministic (i.e. not adaptive)
  - Done when system is shipped/goes live (i.e. no continuous evaluation)

Background – Causes of Interaction Accidents

- Accident reports

"… lack of imagination."

"unknown, knowns" (i.e. things we could have known about)

Background – Operational Phase

- Wait for Accidents to happen, then React
- What ever you do, ”Don’t poke the (accident) Bear”
    - Avoid operating on edge, corner cases, pre-cursors
- New technologies AI/ML and increasing complexity of missions are resulting in *more tightly coupled complex functions* → *more opportunities for interaction accidents*

Table of Contents

1. Background
    - System-of-System *Interaction* Accidents (SoSIA)
    - Engineering Methods limitations
2. **Research Objective**
3. Proposed Con-Ops
    - FTEISS (Fast-time Emergent Interaction Scenario Simulation)
4. Functional Architecture
    - FTEISS
5. Demo Applications
6. Conclusion

Research Objective

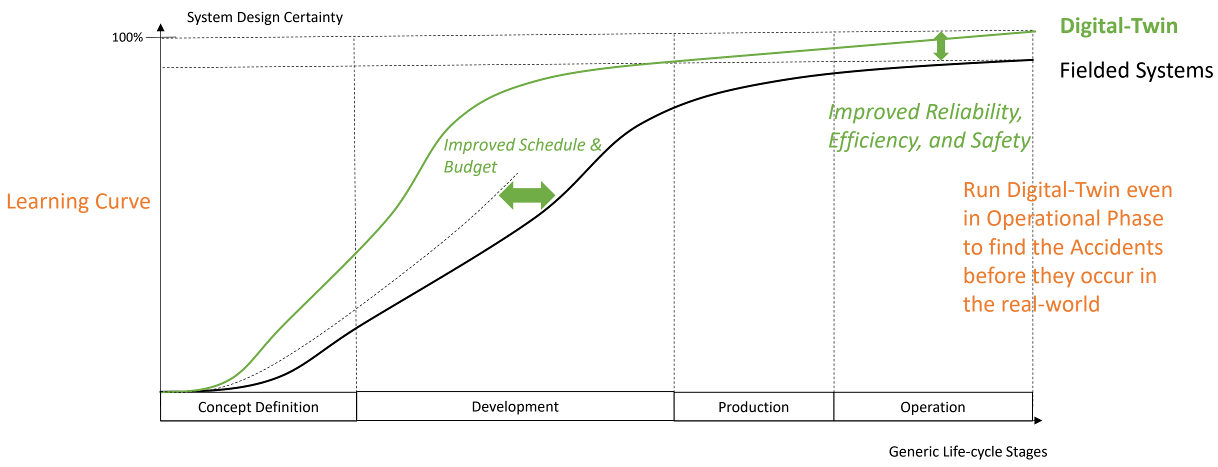
- How does a society "test" these Systems-of-Systems for failures that result from the *Interaction between components*
  - not from component failures

- Components increasingly tightly coupled
- Combinatorial complexity
  - Coupling of components
  - Interactions over time

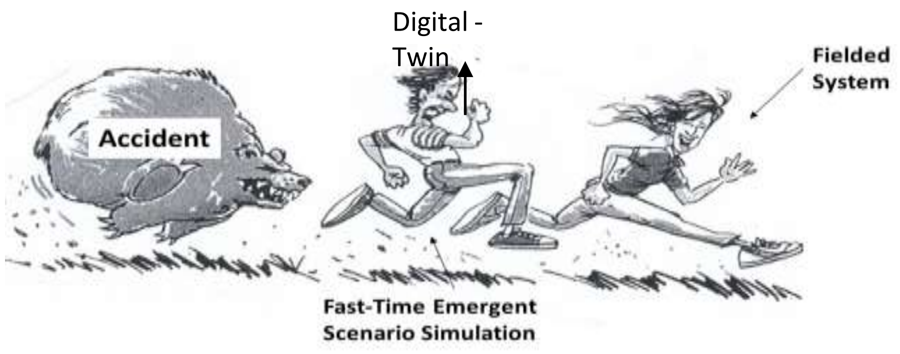
Table of Contents

Con-Ops

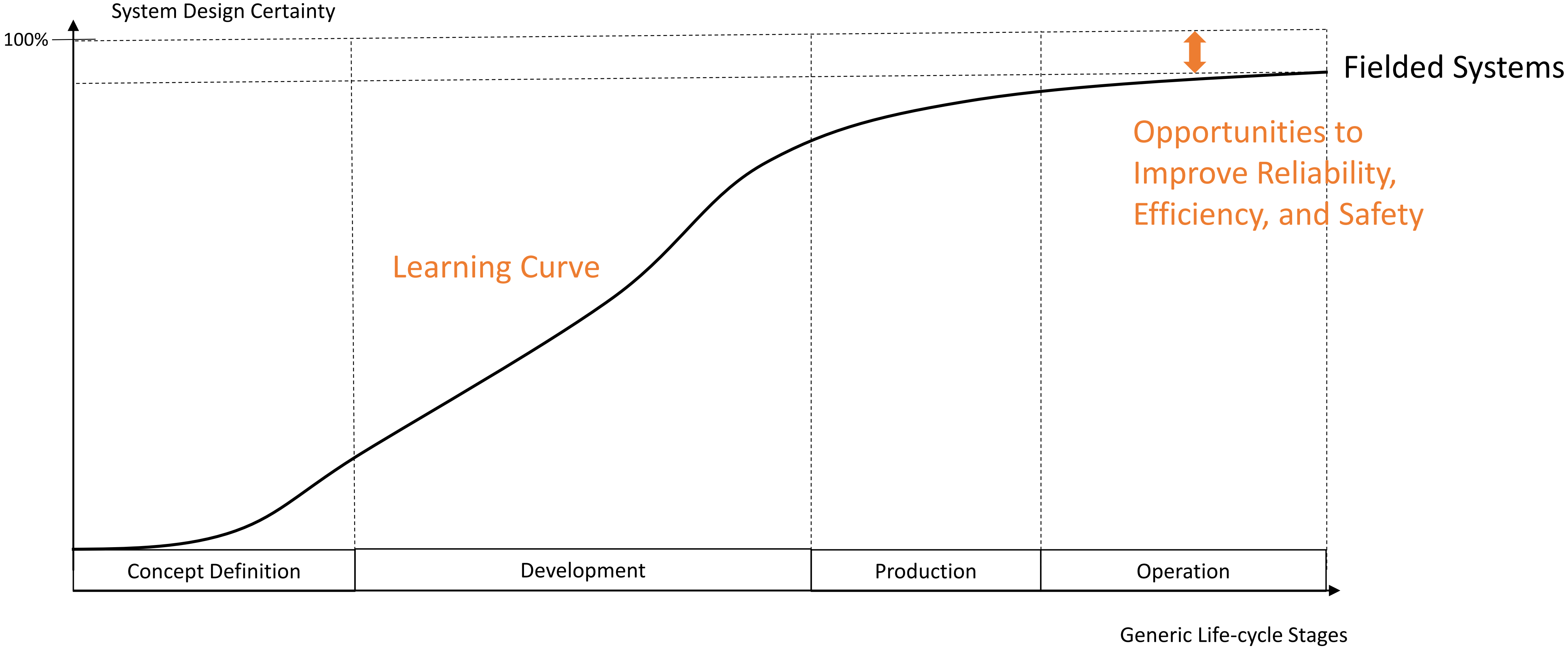
- <u>Development Phase</u>:
  - Use Digital Twin to uncover "known-unknows"



- <u>Operations Phase</u>:
  - Keep testing even after fielding
  - Focus on "corner-cases" and pre-cursors



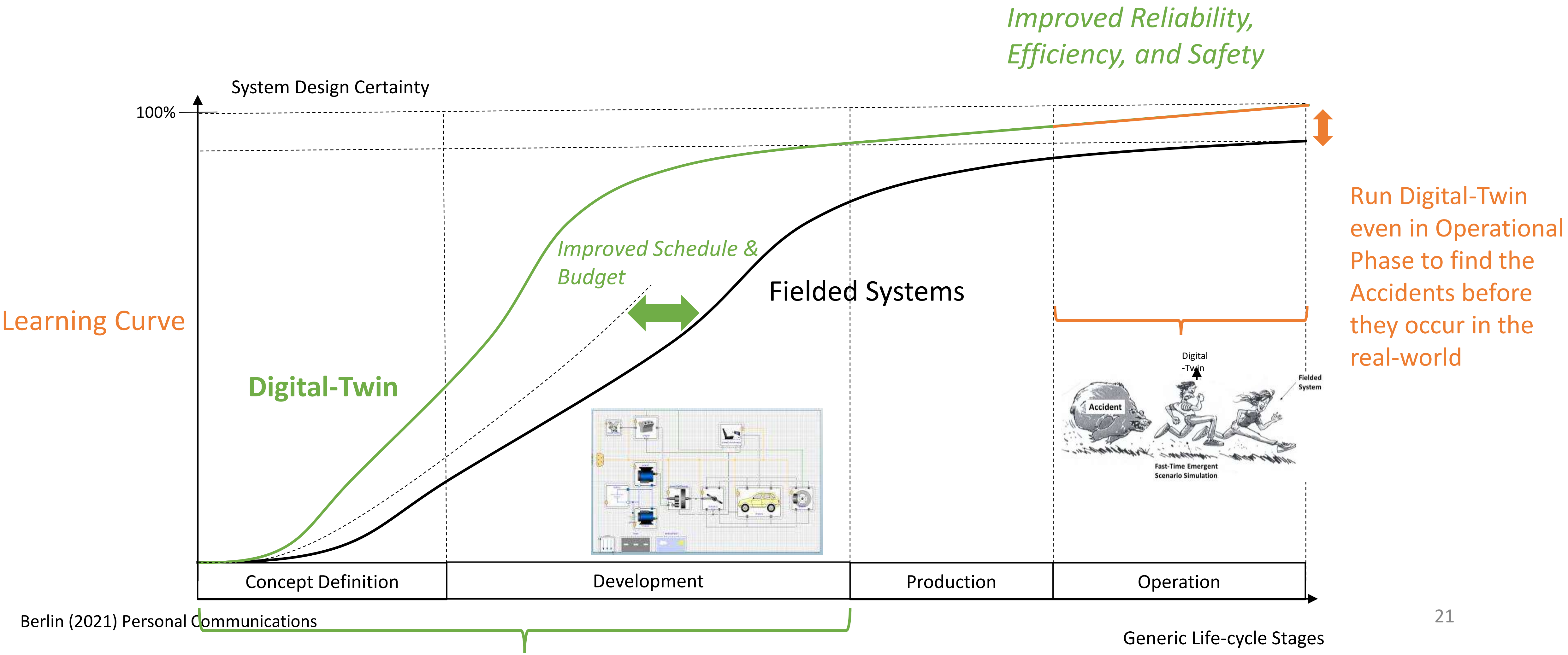
Con-Ops – Digital-Twin = Faster & Longer Learning Curve



Berlin (2021) Personal Communications

Con-Ops – Faster Digital-Twin:

(1) Faster Learning Curve (2) Testing After Fielding

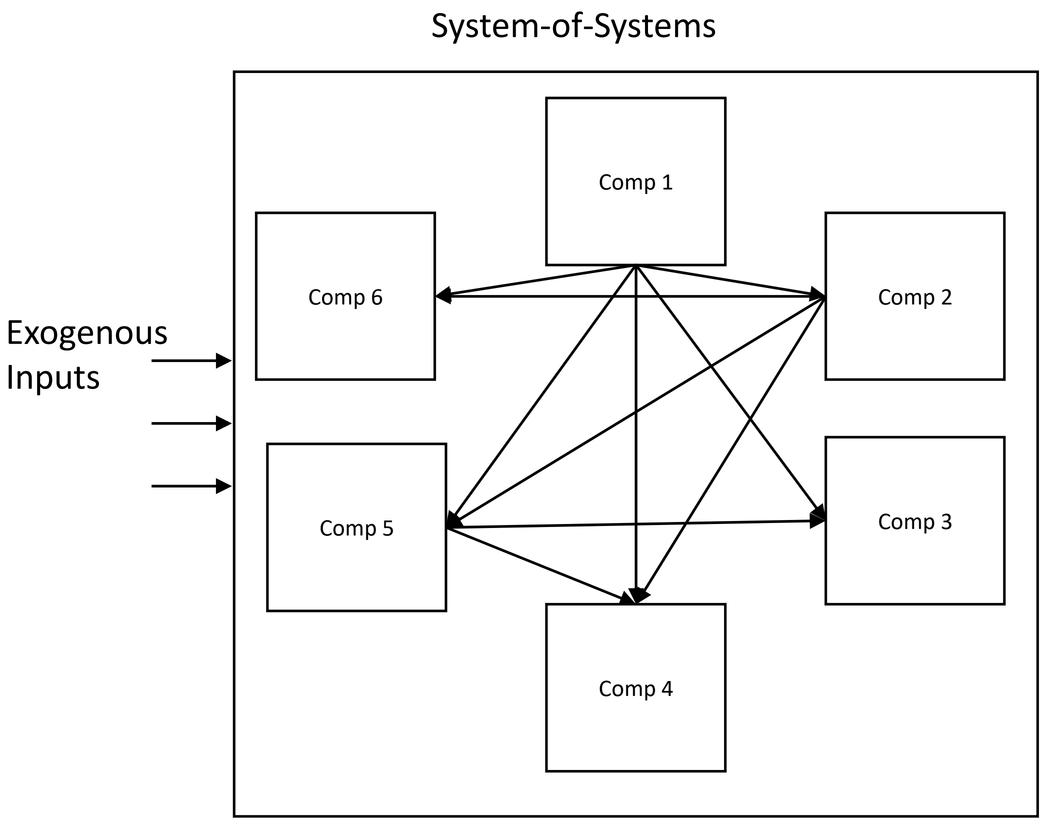


Berlin (2021) Personal Communications

Table of Contents

Terminology – Tightly Coupled System-of-Systems

- System is composed of tightly coupled, interacting components
- Components have complex behavioral complexity
- Components may be ML derived AI
- Components may be adaptive
- System is affected by exogenous inputs from Environment
  - Does not affect environment (i.e. no feedback-loop)

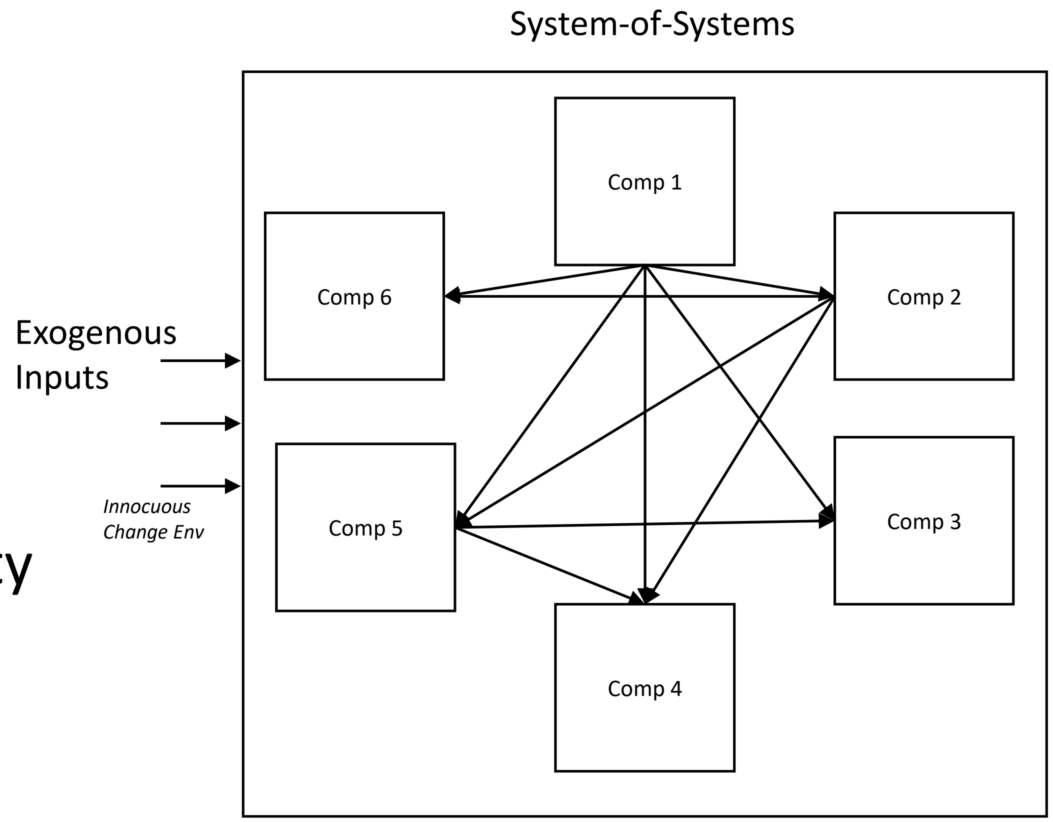


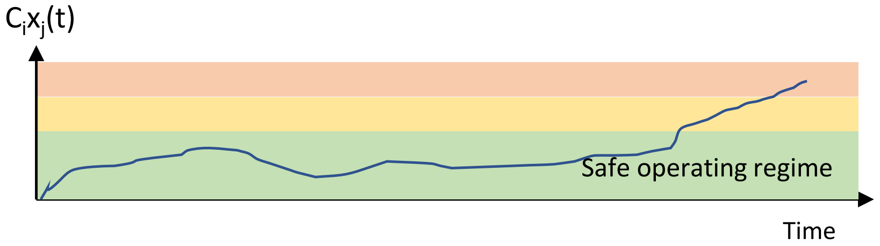
Terminology – S-o-S Performance $C_i x_j(t)$

Overall System performance is measured:
by parameter $C_i x_j(t)$
Where:

- C_i component i
- $x_j(t)$ parameter j as function of time

Note 1: most parameters have operational safety regime not to violate (e.g. max/min speed)





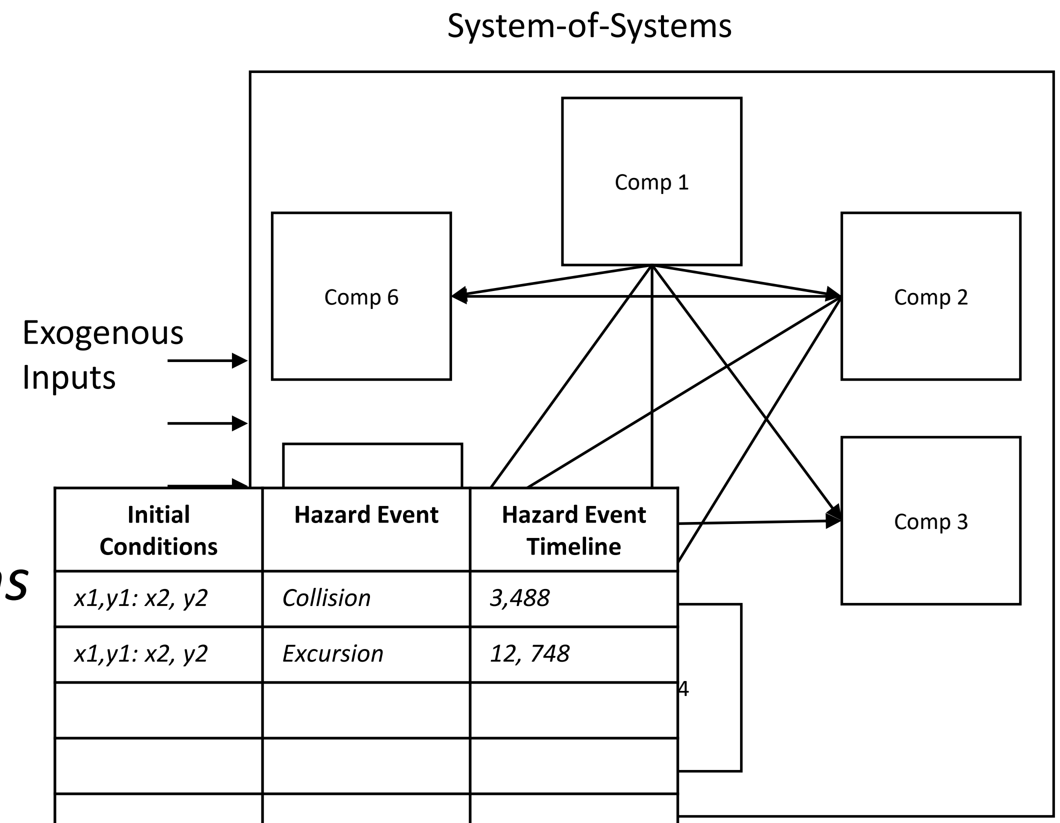
Terminology – Time Threaded Behavior

Overall System performance is measured:
by parameter $C_i x_j(t)$
Where:

- C_i component i
- $x_j(t)$ parameter j as function of time

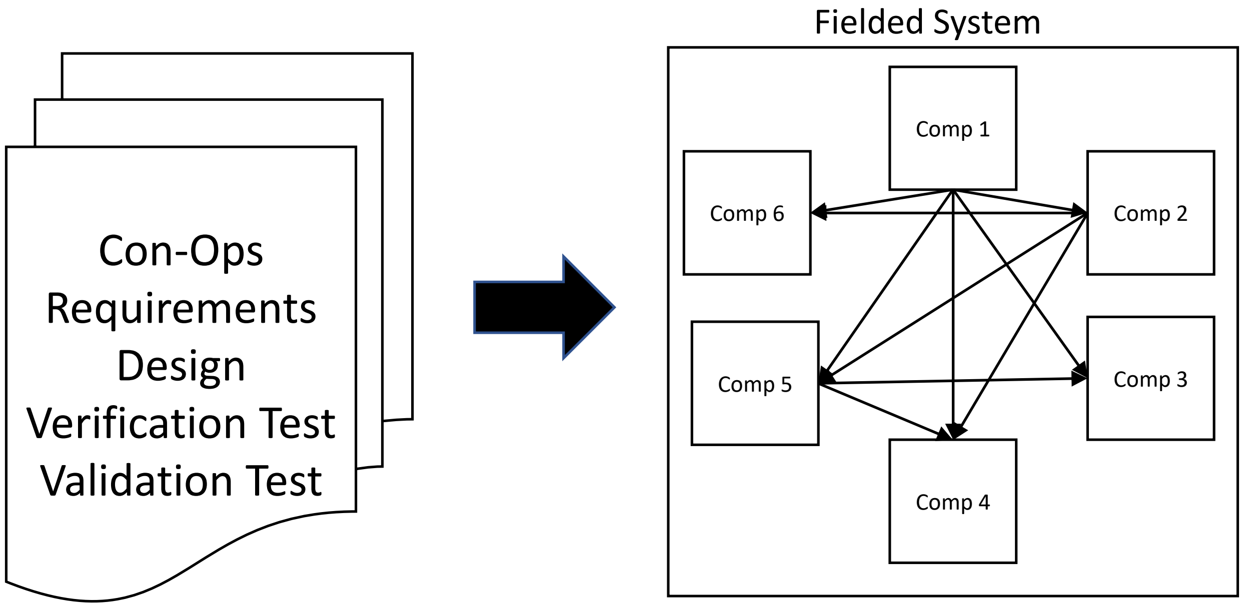
Note 2: Overall system performance is determined *over time* based on *Initial Conditions* $C_i x_j(0)$ and the *resulting interaction over time*

Its not an I/O systems



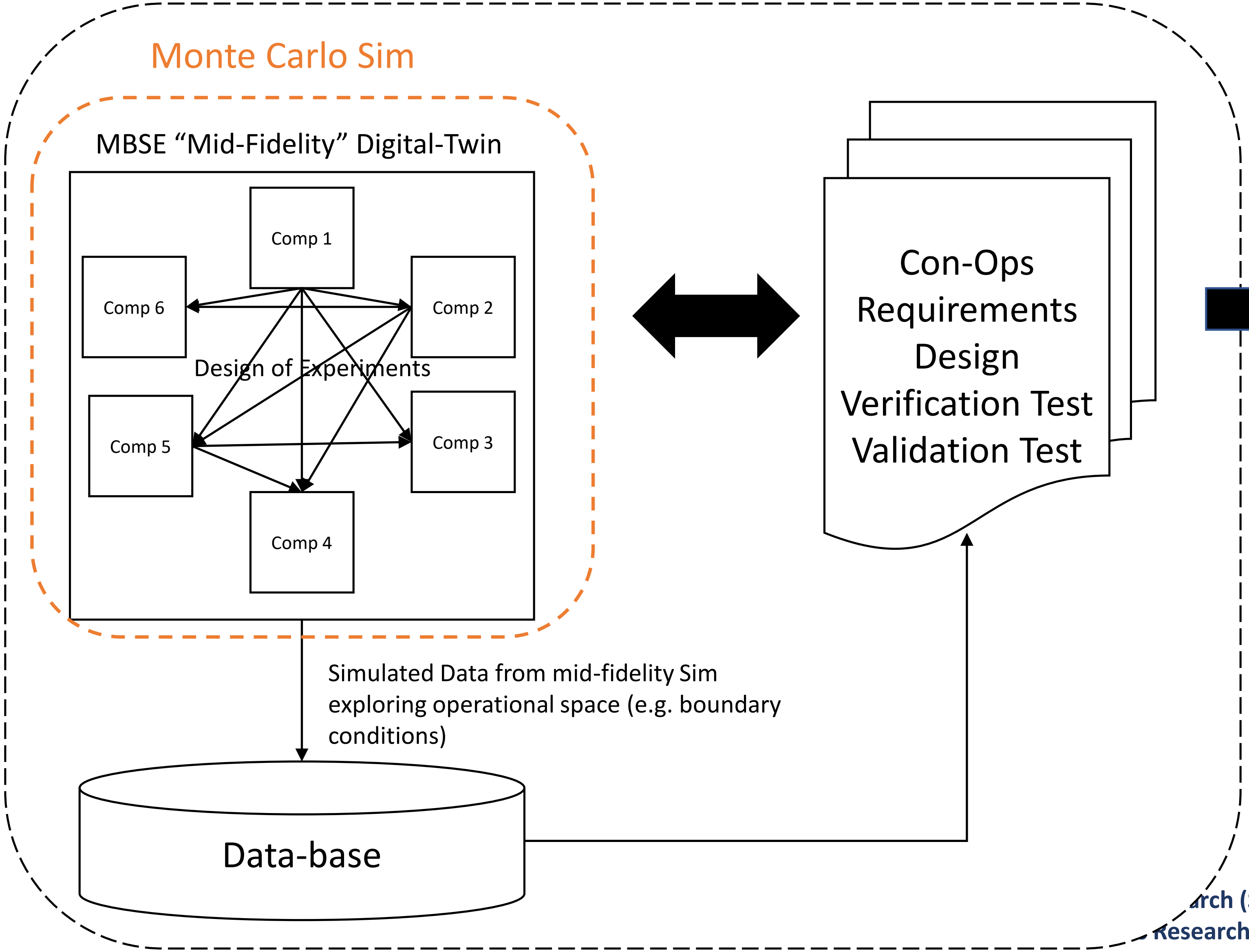
Initial Conditions	**Hazard Event**	**Hazard Event Timeline**
x1,y1: x2, y2	*Collision*	*3,488*
x1,y1: x2, y2	*Excursion*	*12, 748*

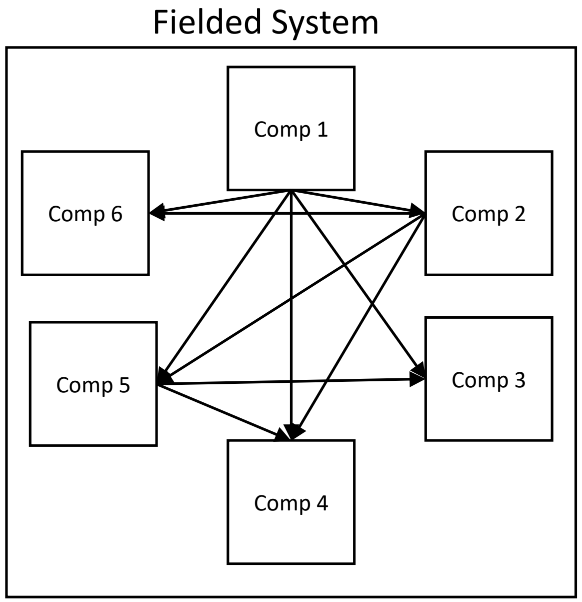
Traditional “Manual/Paper-based” Development



- **Book-keeping complex requirements and test is a challenge**
- **Requirements and Testing only as good as engineering “imagination”**
- **Testing terminates once the system is fielded**

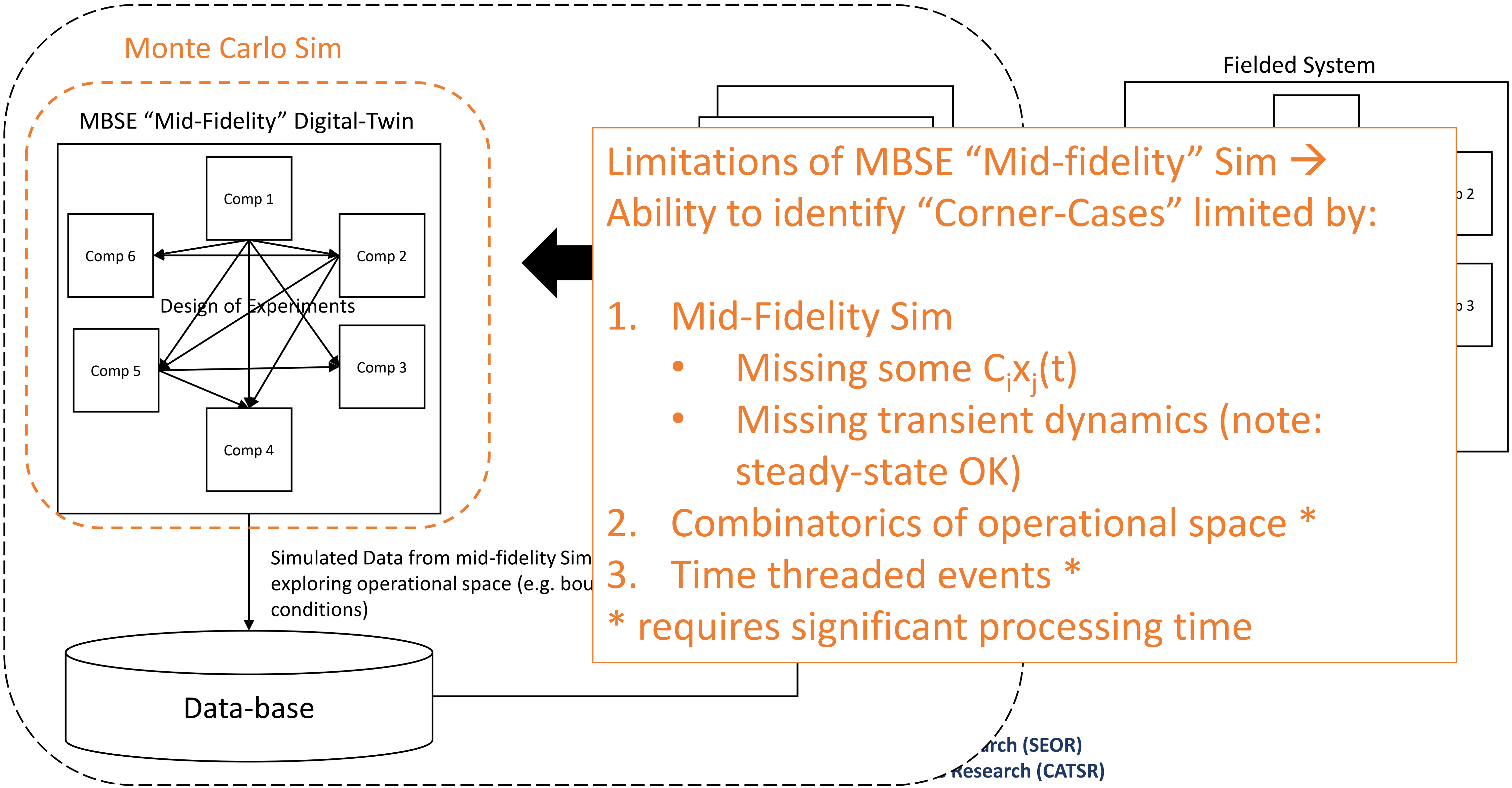
Traditional Digital-Twin System Development





- **Book-keeping complex requirements and test is managed (MBSE)**
- **Requirements and Testing = engineering "imagination" + simulation**
- **Testing terminates once the system is fielded**

Traditional Digital-Twin System Development



Limitations of MBSE "Mid-fidelity" Sim →
Ability to identify "Corner-Cases" limited by:

1. Mid-Fidelity Sim
   - Missing some $C_i x_j(t)$
   - Missing transient dynamics (note: steady-state OK)
2. Combinatorics of operational space *
3. Time threaded events *

* requires significant processing time

Requirements NextGen MBSE Digital Twin

Requirement	Gap	Solution
Sim Fidelity	Higher	Multi-fidelity Model-bias Correction (Xu) Gaussian Random Field (Xu & Chen) Deep Learning (Sokolov)
Combinatorics	Pruning and selection	Design of Experiments (Xu) Monte Carlo Sim
Combinatorics processing time	Importance	Simulation Importance Sampling (Shortle, et.al) Simulation Splitting (Shortle, et.al) Edge/Super Computing (Berlin)
Time threaded events		Simulation Importance Sampling Simulation Splitting Edge Computing

NextGen Digital-Twin System Development

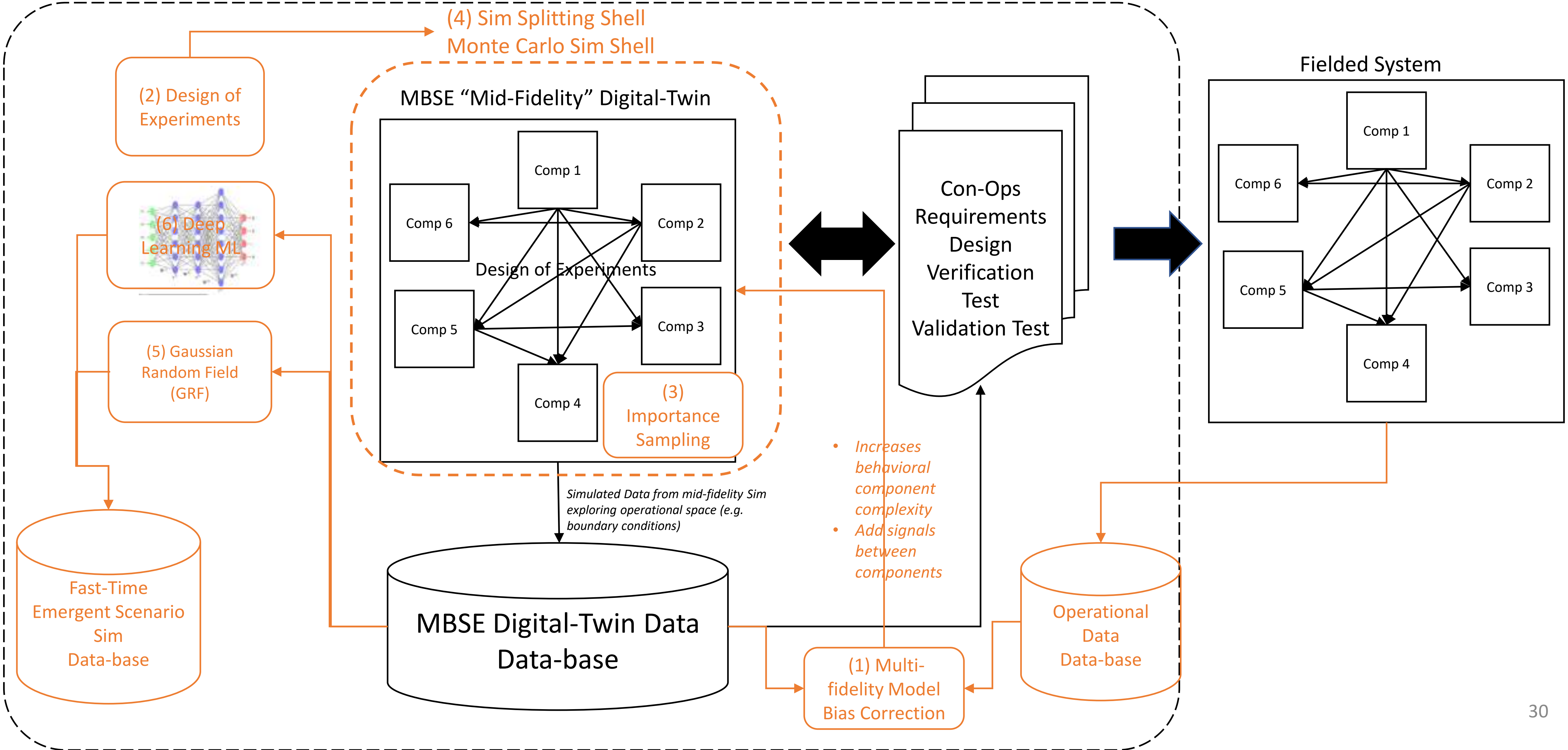
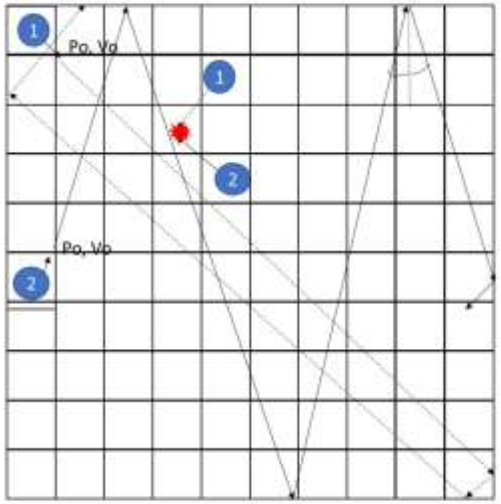
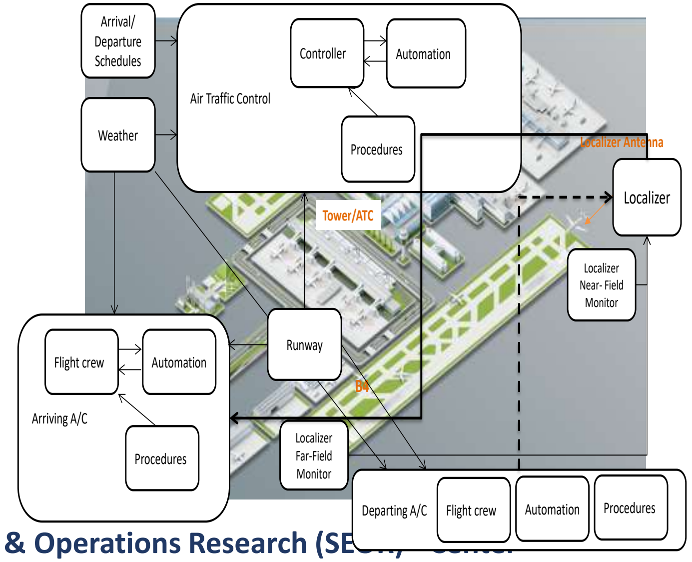


Table of Contents

Demonstration Applications

- Molecules in a Chamber Abstraction
- Munich Runway Excursion/SQ237
- Autonomous Vehicle Guidance and Control
- Airspace Drone Incursion
- …





Initial Conditions	Hazard Event	Hazard Event Timeline
x1,y1: x2, y2	*Collision*	*3,488*

Table of Contents

Summary

- New Paradigm for Accident Prevention = Fast-Time Emergent Scenario Simulation
  - used during development *and* during *deployment/operations*
    - *Find the accidents in the digital-twin (before they occur in operations)*

- Expanding imagination to identify "unknown, knowns" by continuous discovery at the "Edge"

- Learn from the process, not just the results

1. Mid-fidelity Digital-Twin
2. *High performance computing*
3. *Design of Experiments for Monte Carlo Sim*
4. *Importance Sampling*
5. *Splitting*
6. *Multi-fidelity Model Bias Correction*
7. *Gaussian Random Field Space-Filling and/or Deep Learning NN*
8. *Deep Learning Model*

Current Paradigm (after fielding)

Proposed Paradigm (after fielding)

Digital - Twin

lsherry@gmu.edu