Continuous Safety Risk Evaluation By Example

Peter H. Feiler

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM19-1085

Outline

Objective of case study

Low criticality component in context

Repeated quantified safety risk assessment

Related NASA Flight Software Study Results

Objective

To illustrate the importance of including lower Design Assurance Level (DAL) components in a safety risk analysis

To include the pilot's role in the overall system safety analysis

To perform safety risk analysis throughout the product life cycle – even when aircraft are already in operation.

Outline

Objective of case study

Low criticality component in context

Repeated quantified safety risk assessment

Related NASA Flight Software Study Results

Lower Criticality Component in Context

DAL C component in context of DAL A components

- Impact of lower criticality components on overall safety
- Illustrated by DAL C component in context of engines (DAL A)
- Pilot as DAL A backup for DAL C component

What has been captured by the model?

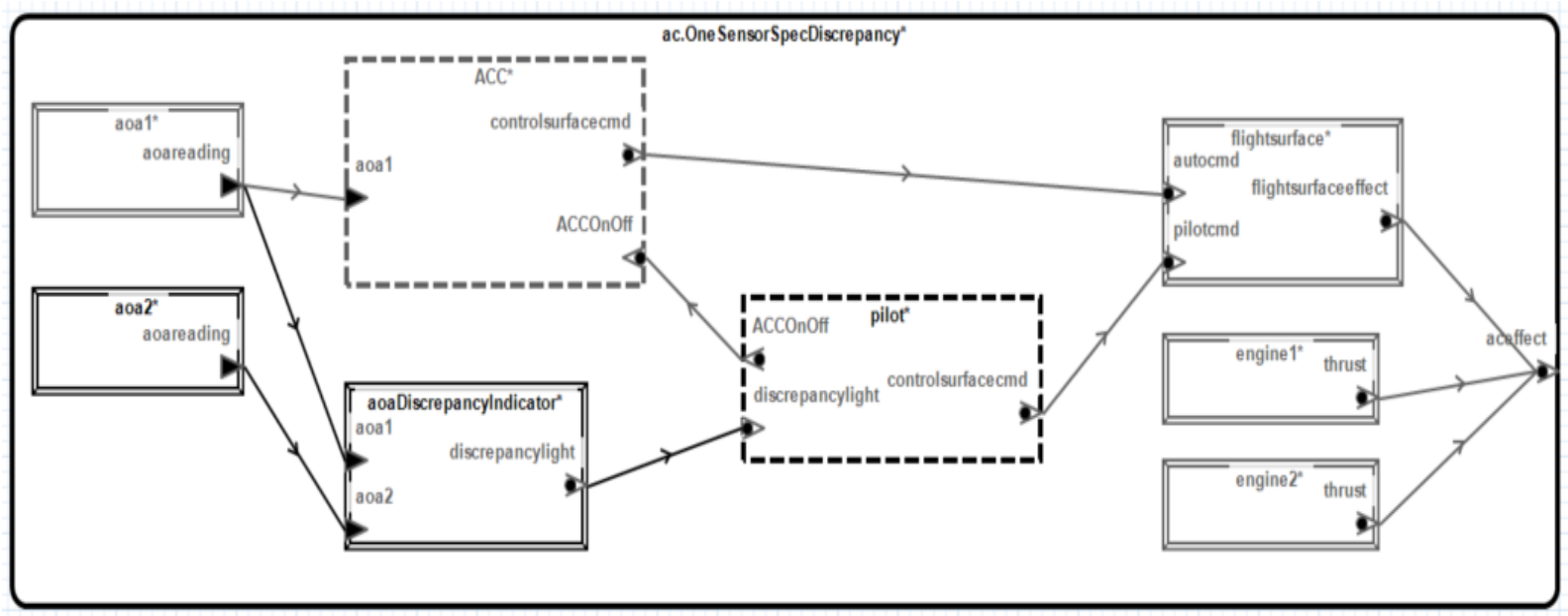
Automated Climb Control (ACC)

Pilot

Angle of Attack (AoA) discrepancy detection

Multiple AoA vendors

Aircraft engines



Use Scenarios

DAL C component in context of DAL A components

- No pilot knowledge of ACC -> Memo about ACC existence
- Non-working AoA discrepancy light -> lack of knowledge of non-functioning ACC
- ACC standby mode indicator in two AoA sensor case
- Pilot/ACC tug of war
- Degrees of pilot knowledge & mistakes: ACC switch off,
- Quality of AoA
  - Omission, Bad value, sym. approx value
  - DAL based spec, known low product quality of one vendor
- One vs. two AoA sensors: purchase option without quantified safety risk buy down
- Unbounded degree of freedom on flight surface control

Component Fault Behavior

AoA Sensor

- Error source for *ServiceOmission* and *BadValue*
- Failure probability of 1.0e-5 (DAL C) and vendor specific values reflecting track record

AoA discrepancy detector

- Component failure and subtle value error misses
- Failure probability: 1.0 or physical component failure

ACC

- ACC switch off/on
- 1 and 2 AoA sensor input configuration
- Discrepancy detection and standby mode in 2 AoA sensor case
- *ContinuousBadControl* error to reflect ACC nor relinquishing control until pilot turns off ACC

Component Fault Behavior - 2

Pilot

- Input from AoA discrepancy detector and ACC standby mode indicator
- Variants for *NoACCKnowledge* and *Nominal*
- Different failure probability (pilot mistakes) for no ACC knowledge, (non)working AoA discrepancy, nominal

Flight surface

- Controlled by ACC or pilot
- Failure probability: mechanical component failure

Engines

- Reflect historical data on engine failures: 1 in 650,000 flight hours

Potential Refinement of AADL Model

Reflect design issues in ACC

- Unbounded degree of freedom on flight surface control due to repeated ACC activation
- Change in degree of freedom of ACC control (0.6 -> 2.5 degrees)
- Change in operational context from high altitude to low altitude
- Time sensitive response time by pilot
- Response time variation due to processor load

Outline

Objective of case study

Low criticality component in context

Repeated quantified safety risk assessment

Related NASA Flight Software Study Results

Quantified Safety Risk Assessment

Different aircraft configurations

Failure probability calculation based on fault tree and minimal cut set

Specification based and vendor specific AoA failure rate

Scenario 1

Single AoA sensor ACC operation retaining control with no pilot ACC knowledge and non-working AoA discrepancy detector

- Aircraft failure probability of 2.1e-5 dominated by ACC/Pilot rate
- ACC/pilot rate is driven by AoA sensor rate

	Error Model Element/Type	Compute...	Specified Proba
'ac.OSSNoACCKnowledge' outgoing 'aceffect'	{Incident}	2.1e-05	
'ac.OSSNoACCKnowledge'	'And' in outgoing propagation condition acincident	2.3e-12	
'engine1' source 'engineFailure'	{Incident} from error source 'engineFailure'		1.5e-06
'engine2' source 'engineFailure'	{Incident} from error source 'engineFailure'		1.5e-06
'flightsurface'		1.3e-12	
'ACC' event 'fail'	{ACCFail} from error event 'fail'		1.3e-06
'pilot' event 'mistakes'	{PilotFlightControlMistake} from error event 'mistakes'		1.0e-06
'flightsurface' event 'FlightSurfaceFailure'	error event 'FlightSurfaceFailure'		2.0e-09
'ACC'		2.1e-05	
'ACC' incoming 'ACCOnOff'	{NoACCTurnOff}	1.0e+00	
'pilot' event 'mistakes'	{NoACCKnowledgeNoTurnoff} from error event 'mistakes'		1.0e+00
'pilot'	'And' in outgoing propagation condition PilotTurnOff	1.0e-03	
'pilot' event 'mistakes'	{AOADiscrepancyUnawareNoTurnoff} from error event 'mistak		1.0e-03
)iscrepancyIndicator' event 'DiscrepancyDetectorF	error event 'DiscrepancyDetectorFailure'		1.0e+00
'ACC'	'Xor' in transition	2.1e-05	
'aoa1' source 'reading'	{BadValue} from error source 'reading'		1.0e-05
'aoa1' source 'reading'	{ServiceOmission} from error source 'reading'		1.1e-05

Scenario 1A

Single AoA sensor ACC operation retaining control with no pilot ACC knowledge and non-working AoA discrepancy detector

- Minimal cut set produces same aircraft failure rate
- Impact of pilot role

	Computed ...	Specified Proba
'ac.OSSNoACCKnowledge' outgoing 'aceffect' {Incident}	2.1e-05	
Cutset1	2.3e-12	
'engine1' source 'engineFailure' {Incident} from error source 'engineFailure'		1.5e-06
'engine2' source 'engineFailure' {Incident} from error source 'engineFailure'		1.5e-06
Cutset2	1.3e-12	
'ACC' event 'fail' {ACCFail} from error event 'fail'		1.3e-06
'pilot' event 'mistakes' {PilotFlightControlMistake} from error event 'mistakes'		1.0e-06
Cutset3	2.0e-09	
'flightsurface' event 'FlightSurfaceFailure' error event 'FlightSurfaceFailure'		2.0e-09
Cutset4	1.0e-05	
'pilot' event 'mistakes' {NoACCKnowledgeNoTurnoff} from error event 'mistakes'		1.0e+00
'aoa1' source 'reading' {BadValue} from error source 'reading'		1.0e-05
Cutset5	1.0e-08	
'pilot' event 'mistakes' {AOADiscrepancyUnawareNoTurnoff} from error event 'mistakes'		1.0e-03
'aoaDiscrepancyIndicator' event 'DiscrepancyDetectorFailure' error event 'DiscrepancyDetectorFailure'		1.0e+00
'aoa1' source 'reading' {BadValue} from error source 'reading'		1.0e-05
Cutset6	1.1e-05	
'pilot' event 'mistakes' {NoACCKnowledgeNoTurnoff} from error event 'mistakes'		1.0e+00
'aoa1' source 'reading' {ServiceOmission} from error source 'reading'		1.1e-05
Cutset7	1.1e-08	
'pilot' event 'mistakes' {AOADiscrepancyUnawareNoTurnoff} from error event 'mistakes'		1.0e-03
'aoaDiscrepancyIndicator' event 'DiscrepancyDetectorFailure' error event 'DiscrepancyDetectorFailure'		1.0e+00
'aoa1' source 'reading' {ServiceOmission} from error source 'reading'		1.1e-05

Scenario 1B

Single AoA sensor ACC operation retaining control with no pilot ACC knowledge and non-working AoA discrepancy detector

- Vendor B AoA sensor rate based on actual replacement data
- Aircraft failure probability of 5.0e-4

	Error Model Element/Type	Computed ...	Specified Prob
'ac.OSVBNoACCKnowledge' outgoing 'aceffect'	{Incident}	5.0e-04	
'ac.OSVBNoACCKnowledge'	'And' in outgoing propagation condition acincident	2.3e-12	
'engine1' source 'engineFailure'	{Incident} from error source 'engineFailure'		1.5e-06
'engine2' source 'engineFailure'	{Incident} from error source 'engineFailure'		1.5e-06
'flightsurface'		1.3e-12	
'ACC' event 'fail'	{ACCFail} from error event 'fail'		1.3e-06
'pilot' event 'mistakes'	{PilotFlightControlMistake} from error event 'mistakes'		1.0e-06
'flightsurface' event 'FlightSurfaceFailure'	error event 'FlightSurfaceFailure'		2.0e-09
'ACC'		5.0e-04	
'ACC' incoming 'ACCOnOff'	{NoACCTurnOff}	1.0e+00	
'pilot' event 'mistakes'	{NoACCKnowledgeNoTurnoff} from error event 'mistakes'		1.0e+00
'pilot'	'And' in outgoing propagation condition PilotTurnOff	1.0e-03	
'pilot' event 'mistakes'	{AOADiscrepancyUnawareNoTurnoff} from error event 'mist		1.0e-03
aDiscrepancyIndicator' event 'DiscrepancyDetectorFail	error event 'DiscrepancyDetectorFailure'		1.0e+00
'ACC'	'Xor' in transition	5.0e-04	
'aoa1' source 'reading'	{BadValue} from error source 'reading'		2.5e-04
'aoa1' source 'reading'	{ServiceOmission} from error source 'reading'		2.5e-04

Scenario 2

Single Sensor ACC Operation Retaining Control with Non-working AoA Discrepancy Detector

- Pilot unaware of AoA discrepancy
- Aircraft failure probability of 2.3e-8
- Vendor B based aircraft failure rate: 5.0e-7

Error Model Element/Type	Computed ...	Specified Proba
'ac.OSSPermanentDiscrepancyFail' outgoing 'aceffect'	{Incident}	2.3e-08	
'ac.OSSPermanentDiscrepancyFail'	'And' in outgoing propagation condition acincident	2.3e-12	
'engine1' source 'engineFailure'	{Incident} from error source 'engineFailure'		1.5e-06
'engine2' source 'engineFailure'	{Incident} from error source 'engineFailure'		1.5e-06
'flightsurface'		1.3e-12	
'ACC' event 'fail'	{ACCFail} from error event 'fail'		1.3e-06
'pilot' event 'mistakes'	{PilotFlightControlMistake} from error event 'mistakes'		1.0e-06
'flightsurface' event 'FlightSurfaceFailure'	error event 'FlightSurfaceFailure'		2.0e-09
'ACC'		2.1e-08	
'ACC'	'Xor' in transition	2.1e-05	
'aoa1' source 'reading'	{BadValue} from error source 'reading'		1.0e-05
'aoa1' source 'reading'	{ServiceOmission} from error source 'reading'		1.1e-05
'pilot'	'Xor' in outgoing propagation condition PilotTurnOff	1.0e-03	
'pilot' event 'mistakes'	{AOADiscrepancyAwareNoTurnoff} from error event 'mistakes'		1.0e-06
'pilot'	'And' in outgoing propagation condition PilotTurnOff	1.0e-03	
'pilot' event 'mistakes'	{AOADiscrepancyUnawareNoTurnoff} from error event 'mistake		1.0e-03
aDiscrepancyIndicator' event 'DiscrepancyDetectorFai	error event 'DiscrepancyDetectorFailure'		1.0e+00

Scenario 3

Working AoA Discrepancy Detector and Single Sensor ACC Operation Retaining Control

- Aircraft failure probability of 2.0e-9
- Flight surface failure affects aircraft failure
  - Redundancy in flight surface usually reduces its failure rate
- Smaller vendor B impact: aircraft failure probability of 2.5e-9

	Error Model Element/Type	Computed ...	Specified Prob
'ac.OneSensorSpec' outgoing 'aceffect'	{Incident}	2.0e-09	
'ac.OneSensorSpec'	'And' in outgoing propagation condition acincident	2.3e-12	
'engine1' source 'engineFailure'	{Incident} from error source 'engineFailure'		1.5e-06
'engine2' source 'engineFailure'	{Incident} from error source 'engineFailure'		1.5e-06
'flightsurface'		1.3e-12	
'ACC' event 'fail'	{ACCFail} from error event 'fail'		1.3e-06
'pilot' event 'mistakes'	{PilotFlightControlMistake} from error event 'mistakes'		1.0e-06
'flightsurface' event 'FlightSurfaceFailure'	error event 'FlightSurfaceFailure'		2.0e-09
'ACC'		2.1e-11	
'ACC'	'Xor' in transition	2.1e-05	
'aoa1' source 'reading'	{BadValue} from error source 'reading'		1.0e-05
'aoa1' source 'reading'	{ServiceOmission} from error source 'reading'		1.1e-05
'pilot'	'Xor' in outgoing propagation condition PilotTurnOff	1.0e-06	
'pilot' event 'mistakes'	{AOADiscrepancyAwareNoTurnoff} from error event 'mistakes'		1.0e-06
'pilot'	'And' in outgoing propagation condition PilotTurnOff	1.0e-09	
'pilot' event 'mistakes'	{AOADiscrepancyUnawareNoTurnoff} from error event 'mistake		1.0e-03
DiscrepancyIndicator' event 'DiscrepancyDetectorFa	error event 'DiscrepancyDetectorFailure'		1.0e-06

Scenario 4

Dual AoA Sensor ACC Operation

- Aircraft failure probability of 2.0e-9 driven by flight surface rate
- ACC/pilot rate 4.2e-17 and vendor B based rate 1.0e-15
- Lower than dual engine failure rate of 4.2e-12

Error Model Element/Type	Computed...	Specified Prob
'ac.TwoSensorSpec' outgoing 'aceffect'	{Incident}	2.0e-09	
'ac.TwoSensorSpec'	'And' in outgoing propagation condition acincident	2.3e-12	
'engine1' source 'engineFailure'	{Incident} from error source 'engineFailure'		1.5e-06
'engine2' source 'engineFailure'	{Incident} from error source 'engineFailure'		1.5e-06
'flightsurface'		4.2e-17	
'flightsurface'	error state 'PilotInCharge'	4.2e-05	
'ACC'	'Xor' in outgoing propagation condition standbynocontrol	2.1e-05	
'aoa1' source 'reading'	{BadValue} from error source 'reading'		1.0e-05
'aoa1' source 'reading'	{ServiceOmission} from error source 'reading'		1.1e-05
'ACC'	'Xor' in outgoing propagation condition standbynocontrol	2.1e-05	
'aoa2' source 'reading'	{BadValue} from error source 'reading'		1.0e-05
'aoa2' source 'reading'	{ServiceOmission} from error source 'reading'		1.1e-05
'pilot'	'And' in outgoing propagation condition PilotControl	1.0e-12	
'pilot' event 'mistakes'	{ACCFailUnawareFlightControlMistake} from error event 'mistakes'		1.0e-03
'ACC' event 'standbyindicatorfail'	error event 'standbyindicatorfail'		1.0e-09
'flightsurface' event 'FlightSurfaceFailure'	error event 'FlightSurfaceFailure'		2.0e-09

Observations on Continuous Quantitative Safety Risk Assessment

Model-based approach allows for continuous and repeated safety risk assessment

- Multiple quantitative safety risk assessment with limited effort to reflect design and management decisions
- Quantified impact of pilot as redundant backup to low criticality component
- Quantified impact of low criticality component on overall system

Outline

Objective of case study

Low criticality component in context

Repeated quantified safety risk assessment

Related NASA Flight Software Study Results

- From https://www.slideserve.com/jacob/nasa-study-flight-software-complexity
- Related to study report https://www.nasa.gov/pdf/418878main_FSWC_Final_Report.pdf

Two Sources of Software Complexity

FSW complexity = Essential complexity + Incidental complexity

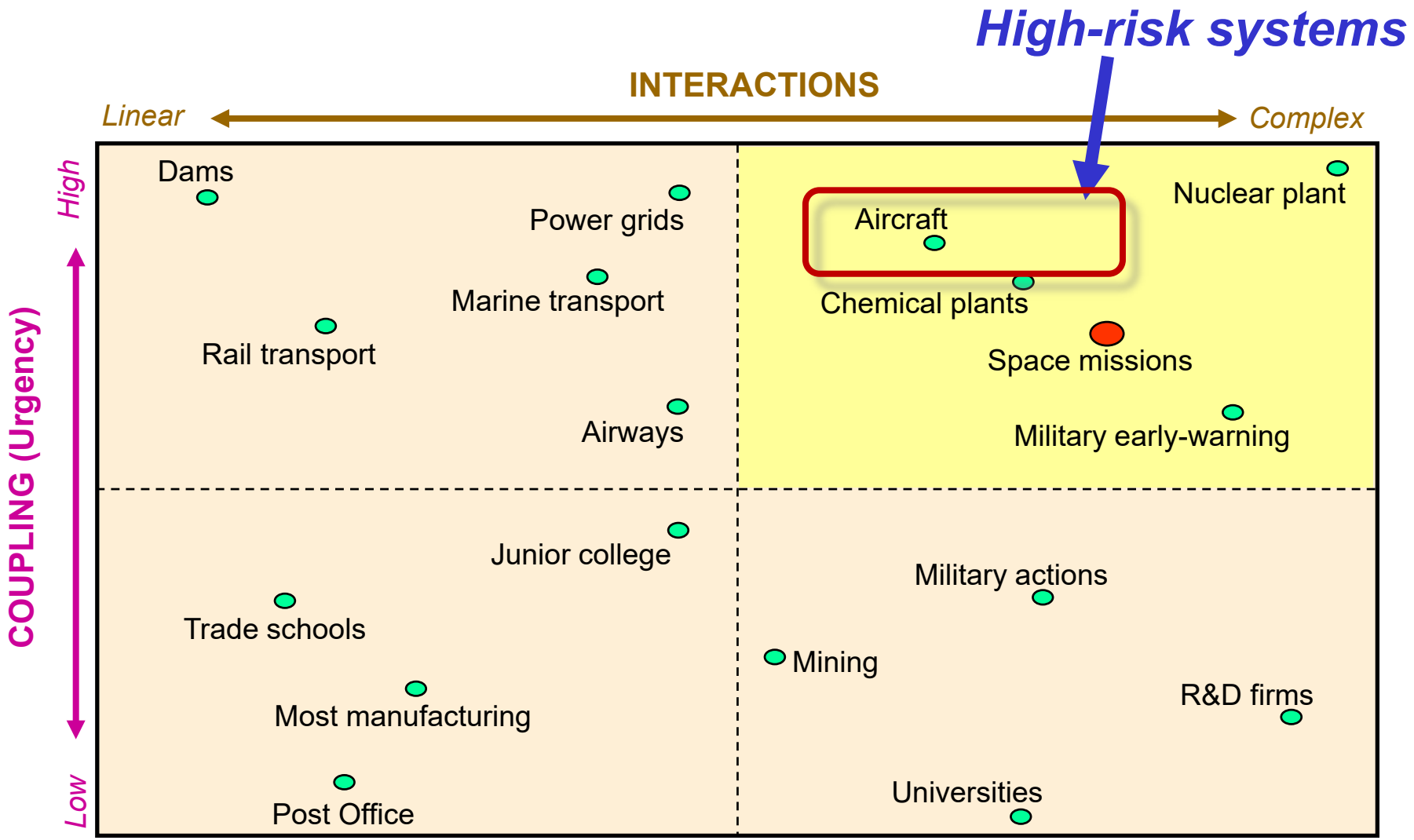
- *Essential complexity* comes from problem domain and mission requirements
- Can reduce it only by descoping
- Can move it (e.g. to ops), but can't remove it

- *Incidental complexity* comes from choices about architecture, design, implementation, including avionics
- Can reduce it by making wise choices

Impact of Low Criticality Failures

In a 1984 book,[4] sociologist Charles Perrow wrote about the causes of failure in highly complex systems, concluding that they were virtually inevitable. Perrow argued that when seemingly unrelated parts of a larger system fail in some unforeseen combination, dependencies can become apparent that could not have been accounted for in the original design. In *safety critical* systems the potential impact of each separate failure is normally studied in detail and remedied by adding backups. But failure *combinations* are rarely studied exhaustively; there are just too many of them and most of them can be argued to have a very low probability of occurrence. A compelling example in Perrow's book is a description of the events leading up to the partial meltdown of the nuclear reactor at Three Mile Island in 1979. The reactor was carefully designed with multiple backups that should have ruled out what happened. Yet a small number of relatively minor failures in different parts of the system (an erroneously closed valve in one place and a stuck valve in another) conspired to defeat the protections and allowed the accident to happen. A risk assessment of the probability of the scenario that unfolded would probably have concluded that it had a vanishingly small chance of occurring.

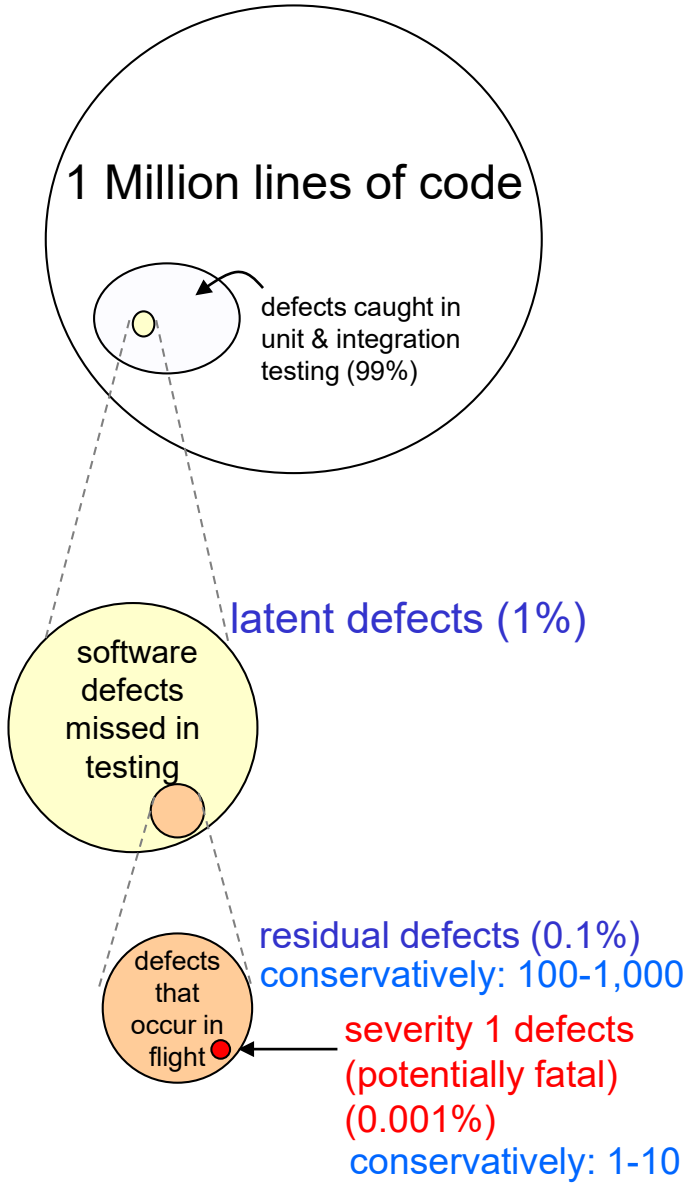
Complex interactions and high coupling raise risk of design defects and operational errors



Source: Charles Perrow, "Normal Accidents: Living with High-Risk Technologies", 1984.

How good are state-of-the-art software testing methods?

- **Most estimates put the number of *residual defects* for a *good* software process at 1 to 10 per KNCSL**
  - A residual software defect is a defect missed in testing, that shows up in mission operations
  - A larger, but unknowable, class of defects is known as *latent software defects* – these are *all* defects present in the code after testing that *could* strike – only some of which reveal themselves as *residual defects* in a given interval of time.
- **Residual defects occur in any severity category**
  - A rule of thumb is to assume that the severity ratios drop off by powers of ten: if we use 3 severity categories with 3 being least and 1 most damaging, then 90% of the residual defects will be category 3, 9% category 2, and 1% category 1 (potentially fatal).
  - A mission with 1 Million lines of flight code, with a low residual defect ratio of 1 per KNCSL, then translates into 900 benign defects, 90 medium, and 9 potentially fatal residual software defects (i.e., these are defects that *will* happen, not those that *could* happen)



Source: Gerard Holzmann, JPL