Evolving Software Trace Links between Requirements and FMEAs
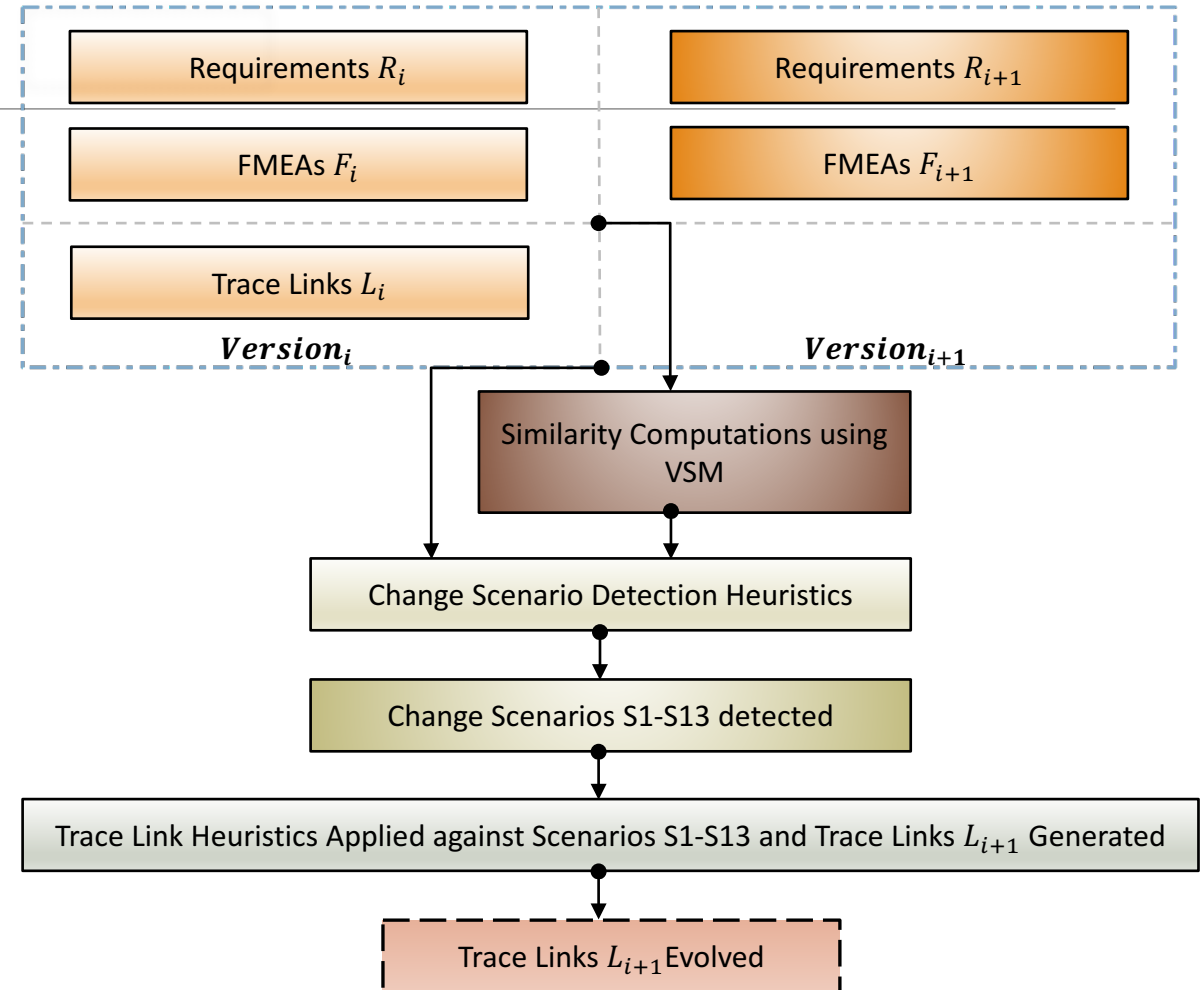
Kuang Wu

Traceability

➢Trace links provide important support for diverse software engineering activities including safety analysis, compliance verification, test-case selection, and impact prediction.

➢However, as the system continually evolves, trace links tend to degrade over time in practice.

➢Evolving trace links is a significant part of software development process in **safety-critical** domains.

➢Artifacts including systems and software level requirements, FMEAs (Failure Modes and Effect Analysis), design decisions, source code, and assumptions.

➢Painful to do it manually.

R32	Drones shall maintain a minimum separation distance of 3 meters between wingtips at all times.	
F2	Violation of minimum separation distance	Drone crashes into another drone.

TLE-RF (Trace Link Evolver for Requirements and FMEAs)

➤ Java

➤ **Vector Space Model (VSM)** was used to calculate similarity scores between requirements and failure modes.

Requirements R_i

Requirements R_{i+1}

FMEAs F_i

FMEAs F_{i+1}

Trace Links L_i

$Version_i$

$Version_{i+1}$

Similarity Computations using VSM

Change Scenario Detection Heuristics

Change Scenarios S1-S13 detected

Trace Link Heuristics Applied against Scenarios S1-S13 and Trace Links L_{i+1} Generated

Trace Links L_{i+1} Evolved

Conclusions

➢ TLE-RF delivers significantly more accurate trace links than those possible when using a standard approach (VSM) to generate links from scratch.

➢ Mona Rahimi, Evolving software trace links between requirements and source code, 2017

➢ Ultimate goal of evolving trace links automatically between every two artifacts in safety-critical domains.