Traceability in Model-Driven Engineering of Safety-Critical Systems

A (Grand?) Challenge?

Richard Paige
paige@cs.york.ac.uk
Department of Computer Science
University of York

Overview

- The message.
- Safety-critical systems engineering.
- Why certification is (largely) a traceability problem.
- Using MDE to build safety critical systems.
  - What do we need?
- Conclusions.

The Message

- There is substantial interest in applying MDE (and not only modelling) in the safety-critical systems engineering field.
- However, *certification* is paramount.
- Traceability is one of the (if not *the*) key notions underpinning certification.

This community can make a real contribution to enabling MDE for safety-critical systems.

Traceability

Identification	**Representation / Description**
Maintenance	**Usage**

Safety-Critical Systems Engineering

- Usually long-lived (embedded) systems.
- Often developed over long periods of time (20-30 years, in some cases).
- Traditionally developed following accepted docu-heavy processes.
  - Emphasis on verification and validation.
- Majority of such systems must be certified prior to their deployment.

Certification

- Development is overseen and assessed by an independent body.
  - e.g., the CAA or an independent safety auditor.
- Developers must present evidence that completed system meets its requirements.
- Numerous standards and guidance exist.
  - e.g., DO-178B for avionics software.
- Process include systems engineering as well as a safety lifecycle.

Safety Lifecycle

1. Identify potential system hazards.
2. Risk assessment.
3. Derive safety requirements.
4. Identify potential designs and refine safety requirements.
5. Develop system.
6. Produce evidence that implementation adheres to design, and safety requirements have been met.
   - Evidence often in form of *safety case.*

Traceability and Safety?

- So what's the connection?
- Most safety standards require traceability:
  - between process phases, design artefacts, implementation artefacts, and safety evidence.
- Traceability exists to enable certification.
- Consider DO-178B.
  - Software Considerations in Airborne Systems and Equipment Certification.
  - Consists of a number of process objectives & guidelines.

DO-178B Table A-3

	Objective		Applicability by SW Level				Output		Control Category by SW level			
	Description	Ref.	A	B	C	D	Description	Ref.	A	B	C	D
1	Software high-level requirements comply with system requirements.	6.3.1a	●	●	○	○	Software Verification Results	11.14	②	②	②	②
2	High-level requirements are accurate and consistent.	6.3.1b	●	●	○	○	Software Verification Results	11.14	②	②	②	②
3	High-level requirements are compatible with target computer.	6.3.1c	○	○			Software Verification Results	11.14	②	②		
4	High-level requirements are verifiable.	6.3.1d	○	○	○		Software Verification Results	11.14	②	②	②	
5	High-level requirements conform to standards.	6.3.1e	○	○	○		Software Verification Results	11.14	②	②	②	
6	High-level requirements are traceable to system requirements.	6.3.1f	○	○	○	○	Software Verification Results	11.14	②	②	②	②
7	Algorithms are accurate.	6.3.1g	●	●	○		Software Verification Results	11.14	②	②	②	

LEGEND:

●	The objective should be satisfied with independence.
○	The objective should be satisfied.
Blank	Satisfaction of objective is at applicant's discretion.
①	Data satisfies the objectives of Control Category 1 (CC1).
②	Data satisfies the objectives of Control Category 2 (CC2).

Summary

- DO-178B objectives explicitly or implicitly require trace-links to be established.
  - Between artefacts, process phases, evidence.
- Of different kinds:
  - Coverage
  - Conformance
  - Satisfaction
  - Implementation
  - Strategic

Using MDE to build SCS

- Should we even try?
  - Is MDE fundamentally at-odds with, e.g, DO-178B?
- What might MDE contribute?
  - M2M transformations can be used to satisfy some A-3 objectives.
  - M2T transformations can deliver evidence to satisfy some A-5 objectives.
- However, all of these operations must be able to expose traceability info explicitly.

A Challenge: Table A-10

"Communicating understanding to the certifying authority."

- Basically, we need to convince an ISA that safety requirements are met.
  - Our evidence is trace-links!
  - How is our evidence represented?
  - What guarantees do we have that our tools don't introduce errors?

What do we need?

- Standard modelling approaches.
  - UML, profiles… (a baby step to DSLs)
  - Not because they are ideal, but because they are more likely to be understood by an ISA.
- Standard ways of representing evidence in a form acceptable to an ISA.
  - Partly depends on who your ISA is.
  - Partly depends on reviewing approaches.

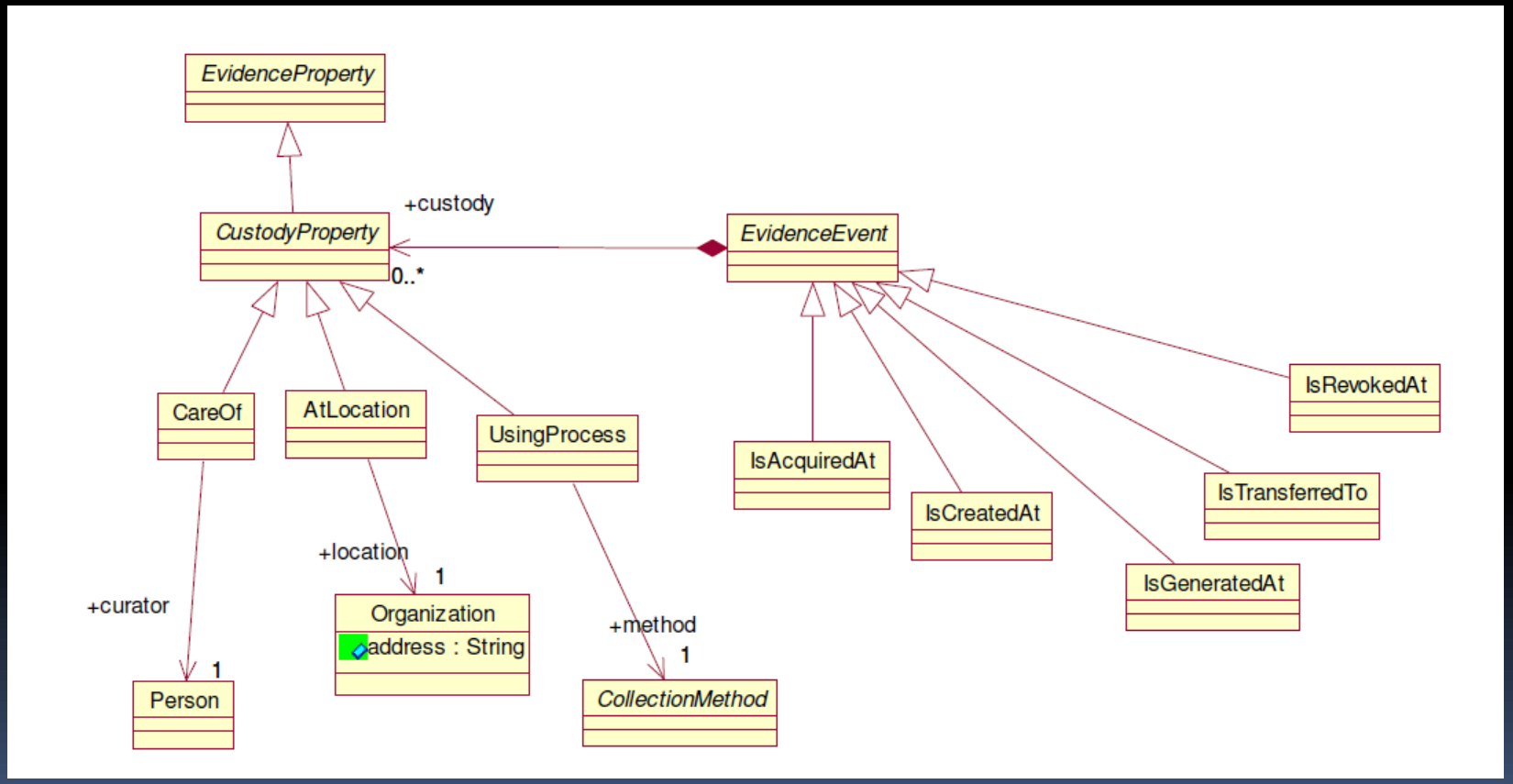
Standardised Evidence

- The OMG Software Assurance Evidence metamodel (SAEM) is a first step towards this.
- It is used to represent facts about software artefacts, developers, process and compliance controls.
- Contributes to an overall assurance case, which could be presented to an ISA.

Properties

- A key part of the Evidence Metamodel is properties:
  - These effectively encode trace-links!
- Provenance (who created, who approved, who owns)
- Custody (where)
- Timing (when)

EvidenceEvent Diagram



Challenge:Evidence Metamodel

- Tools are needed that produce evidence that conforms to it.
  - Existing Traceability tools could help to support this.
- Evidence models need to be connected to an Assurance Case for delivery to an ISA.
- Transformations from existing languages used for safety/assurance cases (e.g., GSN) need to be built, targetting this.

What else is needed?

- Moving forwards…
- Traceability and transformation tools must be qualified.
- Ultimately, a substitution argument for relevant safety standards is needed.
  - i.e., that the evidence produced by applying MDE is at least as convincing as the typical processes followed for building safety critical systems.
  - We have done this for formal methods, but not yet for MDE.

Additionally…

- We need flexibility in how trace-links are established.
- A top-down (req -> design -> code) process isn't always followed.
  - Especially as more iterative and incremental approaches become used.
- Trace establishment through applying model management operations and through manual instantiation.
  - And at arbitrary times, e.g., post-facto.

Conclusions

- MDE is applicable to safety-critical systems engineering.
- But is it acceptable?
  - Engineers need standards and well supported tools.
  - The end-goal is to produce a system that is certifiable.
  - The tools and standards must reflect this goal, and must provide *evidence* in acceptable forms.

What next?

- There is an opportunity for the MDE traceability community to address this problem.
  - Trace-links are evidence that enables certification.
- Explore how to make this evidence standards-compliant, and how it can be connected to certification arguments.
- Qualify your tools.
- Deploy them in safety-critical engineering projects.

?