Managing Assurance Cases in Model Based Software Systems

Sahar Kokaly
McMaster University and University of Toronto, Canada

kokalys@mcmaster.ca

Supervised by: Tom Maibaum (McMaster University) and Marsha Chechik (University of Toronto)

ICSE'17 Doctoral Symposium May 23, 2017
The “thesis” of my thesis

- The state of practice in compliance management can be made more effective using model management.
The “thesis” of my thesis

- The state of practice in **compliance management** can be made more effective using model management.
Robotics

U.S. Wants Makers of Driverless Cars to Prove They Are Safe

The auto industry is beginning to get some clarity on the rules of the road for autonomous cars.

by Will Knight  September 20, 2016

The U.S. government has issued its first rules for automated vehicles. They include a 15-point set of “safety assessment” guidelines for self-driving systems. These cover issues such as cybersecurity, black-box recordings to aid crash investigations, and potential ethical conundrums on the road.

Software failures were behind 24 percent of all the medical device recalls in 2011, according to data from the U.S. Food and Drug Administration, which said it is gearing up its labs to spend more time analyzing the quality and security of software-based medical instruments and equipment.

by Paul Roberts

June 26, 2012, 8:08 pm
Regulations and standards, e.g., ISO 26262 (Functional Safety in Road Vehicles) to demonstrate compliance.
Safety Cases
(a special kind of assurance case)

- A Safety Case is an argument which demonstrates that each of the safety goals has been met, by eventually linking them to evidence in the system.

- Evidence can come in many forms: e.g., test results, analyses, model checking results, expert opinion, etc.
Example: Power Sliding Door

SG1: Avoid activating the actuator while the vehicle speed is greater than 15 km/h

S1: Decompose by AND refinement

B1: The VS ECU sends the accurate vehicle speed information to the AC ECU
  C1.1: ASIL C

B2: The AC ECU does not power the actuator if the vehicle speed is greater than 15 km/h
  C1.2: ASIL B (C)

B3: The VS ECU sends accurate vehicle speed information to the Redundant Switch
  C1.3: ASIL C

B4: The Redundant Switch is in an open state if the vehicle speed is greater than 35 km/h
  C1.4: ASIL A (C)

B5: The actuator is activated only when powered by the AC ECU and the Redundant Switch is closed
  C1.5: ASIL C

B6: Sufficient independence of the AC ECU and the Redundant Switch is shown
  C1.6: ASIL C

Sn1: Software Verification Plan (9.5.1)
Sn2: Software Verification Plan (9.5.1)
Sn3: Software Verification Plan (9.5.1)
Sn4: Software Verification Plan (9.5.1)
Sn5: Software Verification Plan (9.5.1)
Sn6: Domain Expert Judgement
The “thesis” of my thesis

- The state of practice in compliance management can be made more effective using model management.
Model Based Software Systems

- Models as first class citizens

- Model Driven Engineering “MDE” and the ultimate goal of automatic code generation

- Advantages:
  - Reduce accidental complexity by working at a higher level of abstraction
  - Minimize development cost
Example: Power Sliding Door System

Power Sliding Door System

PSD: CD

<table>
<thead>
<tr>
<th>Driver Switch</th>
<th>VS ECU</th>
<th>Redundant Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>requestDoorOpen()</td>
<td>getSpeed(sensed_speed)</td>
<td>request Speed()</td>
</tr>
<tr>
<td>requestDoorClose()</td>
<td>sensed_speed: Real</td>
<td>closed: Boolean</td>
</tr>
</tbody>
</table>

AC ECU
requestSpeed()
sensed_speed: Real

Actuator
openDoor()
closeDoor()
powered: Boolean
activated: Boolean

Door
State: (open, closed)

CommunicatesWith

controls

powers

par

R: CD-SD

PSD: SD

vs_ecu: VS ECU
ac_ecu: AC ECU
a: Actuator
dr: Driver Switch
s: Redundant Switch

s.requestSpeed()

[if s.sensed_speed <= 15] s.closed else s.open

ac_ecu.requestSpeed()

[if ac_ecu.sensed_speed <= 15 and a.powered and s.closed] a.activated = True,
a.openDoor()

ac_ecu.requestSpeed()

[if ac_ecu.sensed_speed <= 15 and a.powered and s.closed] a.activated = True,
a.closeDoor()

dr.requestDoorOpen()

dr.requestDoorClose
The Model Management Toolbox

- slice
- merge
- lift
- match
- diff
- (bidirectional) model transformation

+ MegaModel Management Operators (map, filter, reduce) and lifted operators
The “thesis” of my thesis

- The state of practice in compliance management can be made more effective using model management.
let’s consider the following compliance management scenario...
**Problem:** Can we aid the safety engineer in constructing a safety case for an evolved system by reusing the components of the original safety case as much and as soundly as possible, thus reducing the overall revision cost incurred?

**Necessary step:** Impact assessment to identify how changes in the system affect the safety case.
Model Based Safety Case Impact Assessment due to System Changes

Model Checking

PSD: CD
PSD: SD

Test Results

Model Based Safety Case Impact Assessment Algorithm

Delta (change)

Model Slicers

[MODELS’16]: original approach
[SafeComp’17]: improved approach, safety case slicer, cost-savings analysis

Annotated Safety Case
Research Objectives

- RO1: Assurance Case Modeling.
- RO2: Modeling the Compliance Ecosystem.
- RO3: Assurance Case Operators.
- RO4: Model Management Workflows for Compliance Scenarios.
- RO5: Application in the Automotive Domain.
- RO6: Tool Building.
- RO7: Validation.
Research Objectives

- RO1: Assurance Case Modeling.
- RO2: Modeling the Compliance Ecosystem.
- RO3: Assurance Case Operators.
- RO4: Model Management Workflows for Compliance Scenarios.
- RO5: Application in the Automotive Domain.
- RO6: Tool Building.
- RO7: Validation.
Research Contributions

- RO1: Assurance Case Modeling → [SafeComp’17]
- RO2: Modeling the Compliance Ecosystem → [MiSE’16]
- RO3: Assurance Case Operators → [SafeComp’17]
- RO4: Model Management Workflows for Compliance Scenarios → [MiSE’16, MoDELS’16, SafeComp’17]
- RO5: Application in the Automotive Domain → partially in [SafeComp’17]
- RO6: Tool Building → [planned]
- RO7: Validation → [planned]
Related Work: Using Modeling for Compliance

- Compliance Management Frameworks
  - e.g., [Hamou-Lhadj], [DLVara], [Habli2008]

- Algorithms and Operators for Compliance
  - e.g., [Nejati], [Ghanavati]

- Modeling Standards and Assurance Cases
  - e.g., [Kelly2004], [Luo] [Panesar-Walawege], [Ghanavati] [Bandur]

- Model-based Approaches for Compliance
  - e.g., [Habli2010], [Gallina]

- Safety Case Construction and Maintenance
  - e.g., [Kell2001], [Li], [Jaradat]
Our Work:

- Uses model management to construct workflows that address compliance scenarios especially for automotive systems.

- Uses specific operators defined in related work to construct larger and more impactful workflows.

- Out of scope: we do not focus on safety case construction, instead, we assume presence of a safety case and focus on constructing model-based workflows for scenarios such as safety case impact assessment and reuse.
Next Steps: Tool Support

• RO6: Tool Support
  • Q1: How do we best provide tool support for our approaches?
  • Q2: How can the compliance management workflows we propose be implemented on top of an existing model management tool? If so, what type of adaptation is needed?
  • Q3: What are the missing components that will allow us to work with and manage compliance ecosystems which are heterogeneous in nature?
Next Steps: Validation

- RO7: Validation
  - Q1: Given the tool support, how do we validate the correctness of the approaches presented as model management workflows?
  - Q2: How do we evaluate their effectiveness with respect to efficiency, cost savings and applicability (e.g., how they fit into a real-world software development and safety assessment process)?
The “thesis” of my thesis: The state of practice in compliance management can be made more effective using model management.

So, what is the “impact”?

Industrial:
- reduce cost of managing assurance cases as model-based systems evolve (e.g., composition, incrementality, product lines)

Academic:
- many open problems stated in “Model Management for Regulatory Compliance: a Position Paper” [MiSE’16]
- MMINT: Model Management INTeractive tool https://github.com/adisandro/MMINT/
Managing Assurance Cases in Model Based Software Systems

Sahar Kokaly
McMaster University and University of Toronto, Canada
kokalys@mcmaster.ca

Supervised by: Tom Maibaum (McMaster University) and Marsha Chechik (University of Toronto)

ICSE'17 Doctoral Symposium May 23, 2017

Questions?


Key Contributions

1. Model Management for Regulatory Compliance [MiSE’16]

2. A Model Management Approach for Assurance Case Reuse Due to System Evolution [MoDELS’16, SafeComp’17]


4. Tool Support and Application in the Automotive Domain [SafeComp’17 and planned SoSym’17]
Other Contributions


5. “Heterogenous Megamodel Slicing for Model Evolution.” Rick Salay, Sahar Kokaly, Marsha Chechik, Tom Maibaum [ME’16]