Managing Assurance Cases in Model Based Software Systems

Sahar Kokaly
McMaster Centre for Software Certification, McMaster University, Canada
Department of Computer Science, University of Toronto, Canada
kokalys@mcmaster.ca

***Abstract*—Software has emerged as a significant part of many domains, including financial service platforms, social networks and vehicle control. Standards organizations have responded to this by creating regulations to address issues such as safety and privacy. In this context, compliance of software with standards has emerged as a key issue. For software development organizations, compliance is a complex and costly goal to achieve and is often accomplished by producing so-called *assurance cases*, which demonstrate that the system indeed satisfies the property imposed by a standard (e.g., safety, privacy, security). As systems and standards undergo evolution for a variety of reasons, maintaining assurance cases multiplies the effort. In this work, we propose to exploit the connection between the field of model management and the problem of compliance management and propose methods that use model management techniques to address compliance scenarios such as assurance case evolution and reuse. For validation, we ground our approaches on the automotive domain and the ISO 26262 standard for functional safety of road vehicles.**

***Keywords*-co-evolution; impact assessment; reuse; model management; regulatory compliance; assurance cases.**

I. Problem Statement

Problem. From vehicle control to social networks to financial services, software has become a part of much of life's activities. In order to protect the best interests of citizens, responsbile organizations (e.g., International Organization for Standardization) have responded to this trend by creating standards to address issues such as safety, security and privacy. In this context, compliance of software to standards and regulations has emerged as a key issue. For organizations, compliance is a complex and costly goal to achieve. They may have to comply with multiple standards due to multiple jurisdictions or to address different aspects of the software which may overlap or conflict. The evolution of standards must be tracked and changes assessed. Evidence for claims of compliance must be collected and managed. Finally, maintaining families of related software products further multiplies the effort. Increasingly, models and model-driven engineering are being used as a means to facilitate communication and collaboration between the stakeholders in the compliance value chain and, further, to introduce automation into regulatory compliance tasks. A complexity problem also exists with the proliferation of software models in model-based software development, and the field of Model Management (MM) has emerged to address this challenge. MM focuses on a high-level view in which entire models and their relationships (i.e., mappings between models) can be manipulated using specialized operators to achieve useful outcomes. We propose to exploit this connection between model driven engineering and regulatory compliance, and explore how to use MM techniques to address software compliance management issues.

Hypothesis. The state of practice in compliance management can be made more effective using model management.

II. Preliminaries

A. Model Management

Some of the model management operators that have been studied include *match* [1], *diff* [1], *lift* [2], *slice* [3] and *merge* [4].

For example, the *slice* operator accepts a model and a *slicing criterion* and extracts the subset of the model satisfying the criterion. Model slicing is a way to manage model complexity by focusing on a relevant submodel of a given model. The *merge* operator accepts two models and a relationship expressing the overlap between them and produces a model that combines the content of the models respecting the overlap. Model merge must address the issue of conflicts that could occur when the content is combined.

To help work with collections of models and their relationships, model management uses a special type of model called a *megamodel* [5] whose elements represent models and links between elements represent relationships between the models. In [6], we presented operators for megamodel management, namely, *filter*, *map*, and *reduce*, and a slicing approach for heterogenous megamodels in [7] which is useful for assessing the impact of changes in parts of a megamodel on the rest of it due to evolution.

B. Assurance Cases

Quality standards mandate the creation of quality-specific requirements and assurance cases. For example, ISO 26262 describes how safety requirements, levels of specification and a safety case (a particular kind of assurance case concerned with safety properties) for these must be produced to certify the safety of a vehicle.

An *assurance case* is an artifact that shows how important claims about the system (e.g., requirement satisfaction) can be argued for, ultimately from evidence obtained about the system such as test results, expert opinion, etc. Several approaches to modeling assurances cases have been proposed and are

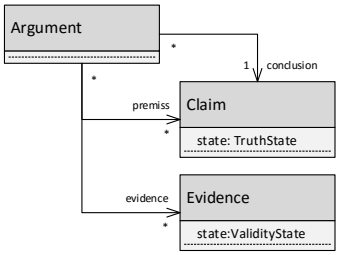

Fig. 1. Generic assurance case metamodel.

discussed in Sec.VI. All of these approaches agree that an assurance case must contain three core concepts: claims, arguments and evidence. A basic assurance case metamodel can be seen in Fig. 1.

III. Approach

The goal of this work is to demonstrate the usefulness of applying model management techniques in the area of regulatory compliance. This work is not a direct application of model management to classical meta-data problems, particularly because of challenges introduced when applying it in the context of regulatory compliance. Such challenges appear due to three factors: 1. The extent to which natural language is used in expressing the standards and the assurance cases. 2. The human-in-the-loop factor and reliance on expert opinion. 3. The assurance artifacts that need to be managed when applying the various model management operators.

We plan to achieve this goal by answering the following set of research questions:

RQ1: Assurance Case Modeling. How do we best develop an assurance case metamodel that captures the basic assurance artifacts and relationships and dependencies between them? What are the assumptions and constraints on this metamodel?

RQ2: Modeling the Compliance Ecosystem. How do we best model the compliance ecosystem including system design, standards, system development processes and assurance cases? What types of relationships (e.g., mappings, refinements, etc.) exist between these entities?

RQ3: Assurance Case Operators. What type of traditional model management operators (e.g., match, merge, slice, etc.) can be used in the context of assurance cases? How can these operators be adapted to work with assurance cases, taking into account the elements of an assurance case (claims, arguments, evidence), dependencies between them, and inference rules used to relate them?

RQ4: Model Management Workflows for Compliance Scenarios. What are some compliance management scenarios where model management workflows can be implemented to address them? What are these model management workflows and can they be made generic? If they are semi-automated and require human intervention (expert opinion) to complete, how can we best involve experts in completing them?

RQ5: Application in the Automotive Domain. What are the kinds of processes, standards and constraints we have to work with in order to apply our solutions in the automotive domain? How do we adapt our assurace case metamodel, operators and workflows to work in this context?

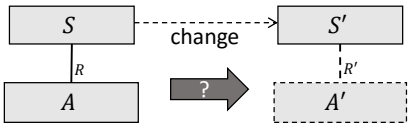

Fig. 2. Assurance case evolution scenario.

RQ6: Tool Building. How do we best provide tool support for our approaches? How can the compliance management workflows we propose be implemented on top of an exisitng model management tool? If so, what type of adapatation is needed? What are the missing components that will allow us to work with and manage compliance ecosystems which are heterogenous in nature?

RQ7: Validation. Given the tool support, how do we validate the effectiveness of the approaches presented as model management workflows in **RQ4**? How do we evaluate them with respect to efficiency, cost savings and usability?

IV. Contributions

In this section, we discuss the current state of our research in terms of completed and expected contributions.

A. Model Management for Regulatory Compliance

There is evidence that the model management approach to compliance management may fill the gaps left by other research on this topic. A recent survey [8] found that a key area where support is needed is in managing the impact of regulations. They identified four factors: (1) frequent changes to regulations; (2) legislation weaknesses; (3) inconsistencies; and, (4) overlap in regulations. They also identified a disproportionate lack of research in these areas as compared to other issues in compliance management. Although they hypothesized that this may be due to the fact that these factors may be in the legal (as opposed to IT) realm, all but factor (2) correspond to general model management problems. For example, (1) corresponds to the change propagation due to model evolution, (3) to addressing inconsistencies, which is a standard step within a model merge operation, and (4) to identifying overlaps, which is the outcome of the model match operation. Taking advantage of the above, we demonstarted in [9] our ideas for using model management in the context of compliance. This helped partially answer **RQ2**, as we presented a general model of compliance which encapsulates the various artifacts and relationships between them. This model can be further improved to include additional artifacts and relations. The contributions in [9] also partially addressed **RQ4** by identifying compliance management scenarios that can be mapped to model management solutions.

B. A Model Management Approach for Assurance Case Reuse Due to System Evolution

In [10], we focused on one of the scenarios we presented in [9]– assurance case reuse due to system evolution – and developed it in detail. Fig. 2 illustrates the scenario at a high level. Assume that S describes the specification for the software in a vehicle. In addition, a type of assurance case A, called a *safety* case, has been developed complying with the

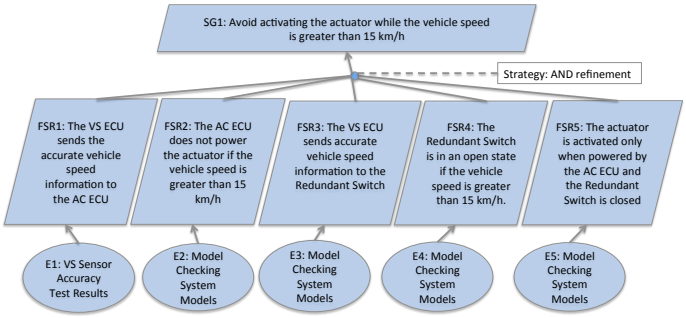

Fig. 3. Goal tree for original system.

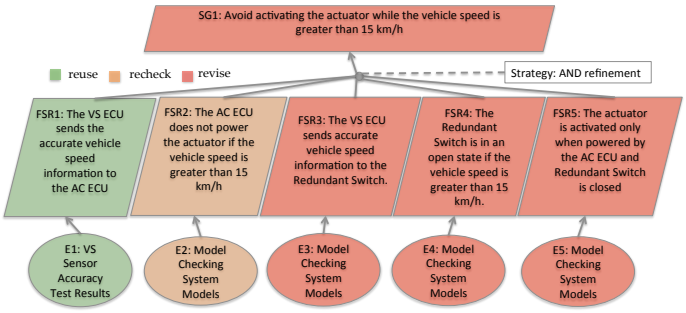

Fig. 4. Goal tree after running the impact assessment algorithm.

ISO 26262 vehicle functional safety standard [11]. Safety case A contains perhaps thousands of safety claims about different components of the vehicle, as well as arguments and evidence to support these claims. Now if S is evolved to S' – for example, as a result of a new requirement or a bug fix – a corresponding safety case A' for S' must be developed. Due to the complexity and effort required to develop a safety case, there is strong incentive to reuse as much of A as possible in the creation of A'. We address this problem using a model management strategy by developing an assurance case impact assessment algorithm. The algorithm does the following: given a model of the system and an assurance case linked to it (e.g., the assurance case of a power sliding door system shown in Fig. 3), and a known change to the system design, the algorithm uses a series of model management operations (e.g., slice, merge, trace) to assess the impact of these changes on the assurance case. It also uses the dependencies between the elements of the assurance case to propogate this impact accross the assurance case itself. The algorithm eventually outputs an annotated version of the assurance case similar to that in Fig. 4, which highlights the elements of the assurance case that can be reused (unaffected by changes), should be rechecked (indirectly affected by the changes), or need to be revised (directly affected by the changes).

In doing so, we have addressed part of **RQ4** by providing a model management workflow that performs impact assessment on an assurance case due to system evolution – one possible compliance management scenario. We would like to consider other scenarios and provide model management solutions for them, as well as understand how to best involve human expertise in completing them.

C. Assurance Case Model Management

In order to fully realize our approach in [10], we need to define a complete assurance case metamodel that captures the entities and dependencies in a way that allows us to reason soundly about them. This will help address **RQ1**. We also need to specify both slice and merge operators for this metamodel. These will allow us to manage assurance cases in a model management fashion under various compliance scenarios and will help address **RQ3**.

D. Application in the Automotive Domain

To help understand the validity of our approaches, we will ground them in the automotive domain, and work with the ISO 26262 standard for functional safety of road vehicles. To do so, certain aspects of the standard (e.g., the notions of ASIL integrity levels and Work Products) have to be captured in our assurance case metamodel. This will allow us to trace the assurance case to both the standard and the system design, which will enable a more effective assurance case impact assessment under different change scenarios. In doing so, we will address **RQ5** and parts of **RQ7**, especially those related to assessing the applicability of our approach.

V. Plan for Evaluation

For the purpose of addressing **RQ6**, we are currently extending the MMINT [12] framework to include models of assurance cases and compliance relationships. We are first adding the capability to work with heterogenous megamodels (which represent the compliance ecosystem), and operators that can be applied to them. We are also implementing a workflow language that allows us to combine model management operators to achieve certain scenarios of interest. Next, we will incorportate assurance cases in the tool by adding the assurance case metamodel as a type and implementing its type-specific model management operators (e.g., slice and merge). We will also implement the model management workflows we propose for compliance management scenarios using our workflow language. Once our framework is complete, we plan to conduct a complete case study from the automotive domain with our industrial partner General Motors. The case study will help in the validation of our approach and addressing the remainder of **RQ7**. We will consider various change scenarios (adding, removing and modifying elements) to assess *applicability* and different size models to assess *scalability*. We plan to assess *usability* by involving an assurance engineer and understanding how their input can be used for completing the semi-automated methods that require expert opinion. Finally, *effectiveness* of our approach can be linked to cost savings, both when assessing compliance (by adding automation) and when re-assessing compliance under incrementality and evolution (by enabling reuse of assurance artifacts).

VI. Related Work

We are unaware of any existing research specifically on applying model management techniques to compliance management problems; however, there is substantial research that

is related to using model-based approaches to aid in compliance. We consider this research to be complementary to our objectives and review it here.

Compliance Management Frameworks. Compliance management frameworks have been proposed. [13] sketches a proposal for comprehensive compliance management in software organizations. Although this is at a higher-level of abstraction than our proposal, the authors discuss the need for tool support for working on large, possibly overlapping or conflicting compliance documents – this is clearly within the scope of our proposal. Researchers have proposed generic metamodels that can be used to structure any safety standard as well as the related safety case information in a project [14], [15]. An advantage of such an approach would be the possibility of providing generic tooling to address the compliance to any safety standard. Thus, we consider these proposals to be complementary to our objectives and will investigate their feasibility as the basis for model management operators.

Algorithms and Operators for Compliance. Specific algorithms for different aspects of compliance management have been proposed. For example, Nejati et al. [3] have developed a model slicing algorithm for extracting parts of a design that are relevant to a given safety requirement, and the authors of [16] propose an algorithm for comparing multiple regulations. We view these strands of research as complementary to our proposal as they can form the basis for the implementation of model management operators.

Modeling Standards and Assurance Casess. Standards and regulations can be expressed as models. For example, [17] shows that the ISO 26262 standard for functional safety of road vehicles can be represented by a combination of a structure model, conceptual model, process model while [18] proposes a conceptual model of IEC 61508. We consider this work as a prerequisite for applying model management techniques to the compliance problem. For assurance cases, a variety of methods have been proposed for modeling them. Goal models and requirements models are used in [19]. [20] presents a formal approach for safety argumentation using KAOS goal models and applies it to a Complex UAV System. The GSN notation [21] has also been proposed as a modeling notation for assurance cases. Our work builds on all of these ideas and assumes that an assurance case can be modelled in a variety of ways as long as it presents the core components – claims, arguments and evidence.

VII. Conclusion

This PhD work presents the problem of assurance case management in a regulatory compliance context and proposes the use of model management techniques to support it. We have discussed work done so far on identifying problems in the regulatory compliance domain that are amenable to model management solutions, and presented our work done on one of these problems, namely that of managing assurance case reuse due to system evolution. We discussed next steps and presented a plan for evalution.

Acknowledgments

This PhD is funded by Automotive Partnership Canada and NSERC and is supervised by Tom Maibaum (McMaster University) and Marsha Chechik (University of Toronto). Thanks to my supervisors, my colleagues Rick Salay and Valentin Cassano and our industry partners Joseph D'Ambrosio and Ramesh S (General Motors) for their valuable input.

References

[1] P. A. Bernstein, "Applying Model Management to Classical Meta Data Problems," in *Proc. of CIDR'03*, vol. 2003, 2003, pp. 209–220.

[2] R. Salay, M. Famelis, J. Rubin, A. Di Sandro, and M. Chechik, "Lifting Model Transformations to Product Lines," in *Proc. of ICSE'14*. ACM, 2014, pp. 117–128.

[3] S. Nejati, M. Sabetzadeh, D. Falessi, L. Briand, and T. Coq, "A SysML-based Approach to Traceability Management and Design Slicing in Support of Safety Certification: Framework, Tool Support, and Case Studies," *Information and Software Technology*, vol. 54, no. 6, pp. 569–590, 2012.

[4] G. Brunet, M. Chechik, S. Easterbrook, S. Nejati, N. Niu, and M. Sabetzadeh, "A Manifesto for Model Merging," in *Proc. of GAMMA@ICSE'06*. ACM, 2006, pp. 5–12.

[5] Z. Diskin, S. Kokaly, and T. Maibaum, "Mapping-Aware Megamodeling: Design Patterns and Laws," in *Proc. of SLE'13*. Springer, 2013, pp. 322–343.

[6] R. Salay, S. Kokaly, A. Di Sandro, and M. Chechik, "Enriching Megamodel Management with Collection-Based Operators," in *Proc. of MODELS'15*. IEEE, 2015, pp. 236–245.

[7] R. Salay, S. Kokaly, M. Chechik, and T. Maibaum, "Heterogeneous megamodel slicing for model evolution," in *Proceedings of the Models and Evolution Workshop*, 2016.

[8] N. S. Abdullah, S. Sadiq, and M. Indulska, "Emerging Challenges in Information Systems Research for Regulatory Compliance Management," in *Proc. of CAiSE'10*. Springer, 2010, pp. 251–265.

[9] S. Kokaly, R. Salay, M. Sabetzadeh, M. Chechik, and T. Maibaum, "Model management for regulatory compliance: a position paper," in *Proc. of MiSE'16*. ACM, 2016, pp. 74–80.

[10] S. Kokaly, R. Salay, V. Cassano, T. Maibaum, and M. Chechik, "A model management approach for assurance case reuse due to system evolution," in *Proc. of MODELS'16*. ACM, 2016, pp. 196–206.

[11] *ISO 2626: Road Vehicles – Functional Safety*, International Organization for Standardization, 2011, 1st version.

[12] A. Di Sandro, R. Salay, M. Famelis, S. Kokaly, and M. Chechik, "MMINT: A Graphical Tool for Interactive Model Management." in *Proc. of MODELS'15 (demo track)*, 2015.

[13] A. Hamou-Lhadj and A. Hamou-Lhadj, "Towards a Compliance Support Framework for Global Software Companies," in *Proc. of SEA'07*, 2007, pp. 182–192.

[14] J. L. de la Vara and R. K. Panesar-Walawege, "Safetymet: A Metamodel for Safety Standards," in *Proc. of MODELS'13*. Springer, 2013, pp. 69–86.

[15] I. Habli and T. Kelly, "A Model-Driven Approach to Assuring Process Reliability," in *Proc. of ISSRE'08*. IEEE, 2008, pp. 7–16.

[16] S. Ghanavati, A. Rifaut, E. Dubois, and D. Amyot, "Goal-Oriented Compliance with Multiple Regulations," in *Proc. of RE'14*. IEEE, 2014, pp. 73–82.

[17] Y. Luo, M. van den Brand, L. Engelen, J. Favaro, M. Klabbers, and G. Sartori, "Extracting Models from ISO 26262 for Reusable Safety Assurance," in *Proc. of ICSR'13*. Springer, 2013, pp. 192–207.

[18] R. K. Panesar-Walawege, M. Sabetzadeh, and L. Briand, "Supporting the verification of compliance to safety standards via model-driven engineering: Approach, tool-support and empirical validation," *Information and Software Technology*, vol. 55, no. 5, pp. 836–864, 2013.

[19] S. Ghanavati, D. Amyot, and L. Peyton, "A Systematic Review of Goal-Oriented Requirements Management Frameworks for Business Process Compliance," in *Proc. of RELAW'11*. IEEE, 2011, pp. 25–34.

[20] J. Brunel and J. Cazin, "Formal Verification of a Safety Argumentation and Application to a Complex UAV System," in *Prof. of SAFECOMP Workshops*. Springer, 2012, pp. 307–318.

[21] T. Kelly and R. Weaver, "The Goal Structuring Notation – A Safety Argument Notation," in *Proc. of DSN'04*, 2004.