Towards a Framework for Safe and Secure Adaptive Collaborative Systems

Aida Čaušević
Mälardalen University
Västerås, Sweden
aida.causevic@mdh.se

Alessandro V. Papadopoulos
Mälardalen University
Västerås, Sweden
alessandro.papadopoulos@mdh.se

Marjan Sirjani
Mälardalen University
Västerås, Sweden
marjan.sirjani@mdh.se

***Abstract*—Real-time adaptive systems are complex systems capable to adapt their behavior to changing conditions in the environment, and/or internal state changes. Highly dynamic and possibly unpredictable environments, and uncertain operating conditions call for new paradigms of software design, and run-time adaptation mechanisms, to overcome the lack of knowledge at design time. Main application areas include vehicles or robots that need to collaborate to achieve a common task, e.g., minimize fuel consumption, moving objects at a construction site, or performing a set of operations in a factory. Moreover, these vehicles or robots need to interact and possibly collaborate with humans in a safe way, e.g., avoiding accidents or collisions, and prevent hazardous situations that may harm humans and/or machines. This paper proposes a framework for developing safe and secure adaptive collaborative systems, with run-time guarantees. To enable this, our focus is on requirement engineering and safety assurance techniques to capture the specific safety and security properties for the collaborative system, and to provide an assurance case guaranteeing that the system is sufficiently safe. Moreover, the paper proposes an architecture and behavioral models to analyze the requirements at run-time. Finally, we design a suitable deployment platform to perform the run-time analysis and planning while guaranteeing the real-time constraints.**

I. Introduction

The complexity and diversity of today's systems is rapidly growing. Moreover, there is a high requirement on their ability to adapt to changes in the environment and to be capable to collaborate with other systems. Given the complex interplay and unpredictable environments they possibly operate in, the behavior of such systems cannot be fully predicted and analyzed at design-time. Current state-of-practice in system architecture, software development and safety and cyber-security assurance is challenged by this development, to the point where development of new products that could simplify our lives and provide solutions to key problems in our society is hampered.

Let us consider a cooperative adaptive cruise control that aims to achieve safe driving that avoids vehicle accidents and traffic jams. This can be achieved by exchanging road traffic information (e.g., traffic flow, traffic density, velocity variation, etc.) among neighboring vehicles. Several critical issues should be addressed in such systems, including the difficulty to control the inter-vehicle distances among the neighboring vehicles, ensure stability in the traffic flow, etc. We assume a platoon-based cooperative driving, and in such a scenario we need to determine the adaptive platoon length and the velocity of vehicles. These decisions are made based on traffic information collected from global and local clouds [1]. One of the main concerns is assuring safety and cyber-security in the control system while adaptation takes the place. Let us assume that a report about an accident has arrived. In that case, the platoon has to either change the planned route to avoid slow traffic due to the accident or change the route completely. No matter which decision will be chosen, the adaptation has to follow strict safety requirements, including to avoid any collision. Getting prompt information in this case is of highest importance. Similarly, in a case of a cyber-attack to the platoon communication system, one has to be able to activate a countermeasure (mitigation) as soon as it is discovered, and check whether the platoon is still sufficiently safe.

In our approach the main goal is to develop run-time behavioral models for collaborative adaptive distributed systems, analysis techniques for continuous safety and cyber-security assurances, with real-time guarantees for the assumptions made in the model. To enable this, we need to design behavioral models, and techniques to analyze and check the safety and cyber-security requirements at both at design-, and run-time. The analysis of such models will be executed in a suitable cloud-based platform capable of providing real-time guarantees. In particular, safety and cyber-security requirements need to be identified and analyzed in continuously evolving systems, where new functionalities can be added over time (e.g., to extend the capabilities of robots or a vehicle), or basic functionalities of a vehicle are modified (e.g., releasing a new firmware, deploying a new control system, or updating the software with cyber-security patches). Such requirements need to be analyzed through suitable behavioral models, not only at design-time, but also at run-time to possibly intervene and take corrective actions to deal with unprecedented events. Whereas the analysis of behavioral models requires a large computational power, as well as global knowledge of the overall system of subsystems, timely decisions are needed to guarantee that corrective actions can preserve a safe collaboration. Therefore, we focus also on how to best deploy such applications on a time-predictive fog/cloud platform, while providing guarantees on the completion time of the analysis, and the communication to the agents involved with the adaptation.

The paper is organised as follows. In Section II we introduce necessary background and concepts used in this paper. Section III describes the details of the proposed approach. Finally in Section IV we provide some concluding remarks including plans for the future work.

II. Background

A. Formal modeling and adaptive systems

In [2], the authors analyze fundamental challenges for the development of Validation and Verification (V&V) methods and techniques that provide certifiable trust in self-adaptive and self-managing systems. Moreover they present a proposal for including V&V operations explicitly in feedback loops for ensuring the achievement of software self-adaptation goals, by defining viability zones, that is the set of states where the systems requirements and desired properties (i.e., adaptation goals) are satisfied. A survey on the use of formal methods in self-adaptive systems is presented in [3]. They claim that formal methods in the field of self-adaptation are mainly used for modeling and reasoning (read discussing) about properties of interest. However, the full power of formal methods which comes with automation and tools, in particular for model checking and theorem proving, are currently under-exploited.

Ghezzi et al. [4] propose to use compositional reasoning of a modularized system through an assume-guarantee reasoning, so if the effect of a change is encapsulated within the boundary of a module, then redoing the verification of other modules will not be necessary. In [5], Timed Automata is used to model the managed system and the environment in the knowledge part of the feedback loop. A survey on using models at run-time to address assurance for self-adaptive systems is provided in [6], while [2] propose a way to integrate run-time V&V tasks in the elements of the feedback loop.

The existing proposed verification techniques are mainly at the level of transition systems or automata. We believe that using actors in building the architecture and the model@run-time will give us the opportunity to use compositional verification and reduction techniques that can exploit the structure of the model. Our aim is to build upon our experience [7], [8], but we will need to extend the existing approaches to work in a dynamic setting and with an acceptable response time such that they can be used at run-time.

B. Safety and cyber-security engineering

tIn the domain of hazard analysis and risk assessment of safety and safety-relevant cyber-security, there has been a significant amount of new approaches proposed, mostly driven by needs in the automotive domain. In most cases, these approaches are built on already existing approaches, such as HARA [9] and STRIDE [10], resulting in a Security-Aware Hazard and Risk Analysis (SAHARA) [11]; Failure Modes, Vulnerabilities and Effect Analysis (FMVEA) [12] based on an approach from the safety domain (FMEA); a method called STPA - Sec [13], grounded on the already existing top-down safety hazard analysis method System-Theoretic Process Analysis (STPA); etc. In most cases these approaches provide a common reasoning about safety and cyber-security [14], while some have a possibility to reuse previously acquired results and redo the analysis in case a new threat or vulnerability is identified.

The applicability of these approaches to more complex and dynamic systems in a continuous manner is questionable. Moreover, most of these approaches focus on safety and cyber-security in separate processes. Our approach is novel as it will enable joint safety and cyber-security analysis, independent of the domain, with feedback within the analysis process. Our existing work include some initial ideas on how one can approach this issues [15], [16], but this has to be further developed to account for collaborative adaptive systems and evaluated on real-world industrial applications.

The notion of a dynamic assurance case has already been proposed in the safety domain [17], [18], e.g., the introduction of a set of rules for updates and a set of monitors for establishing a link between the system and a confidence structure within the safety case. Moreover, a method to generate reusable safety case argument-fragments that include supporting evidence related to compositional safety analysis already exists [19], [20]. The generation is performed from safety contracts that carry safety-relevant behavior of components in assume/guarantee fashion supported by evidence.

We believe that each mentioned technique carries a piece of the puzzle that can contribute to building a common safety and safety-relevant cyber-security assurance model. Although they need to be adapted, further developed and complemented with other relevant solutions to handle a dynamic cyber-security assurance case applicable in the domain of adaptive systems. Moreover, we will provide constructs for (semi-)automatic assurance case adaptation.

C. Cloud computing

There are several research groups trying to address problems related to real-time cloud. In particular, in the last years, researchers from the real-time system community have tried to extend some of the classical results of scheduling theory to cloud systems. In [21], the authors highlight the main challenges related to the design and implementation of real-time cloud solutions. The main approaches that have been developed are IRMOS, and RT-Xen.

IRMOS [22], is a deadline-based real-time scheduler for the Linux kernel that provides scheduling guarantees to individual Virtual Machines (VMs) scheduled on the same system, processor and core. It provides temporal isolation among multiple software components, such as entire VMs. The IRMOS scheduler can be extended also to other technologies, such as LXC containers as well as JVM instances. One of the main advantages of the IRMOS scheduler is that it can temporally isolate the execution of a VM from additional cloud management workload that is controlled by the cloud provider, such as the monitoring system, or the migration of VMs from one physical machine to another.

RT-Xen [23], is a scheduler that bridges the gap between real-time scheduling theory and the Xen platform by schedul-

ing VMs using fixed-priority server algorithms designed based on real-time scheduling theory. The real-time VM scheduler in the hypervisor and the schedulers in the guest operating systems form a scheduling hierarchy whose real-time schedulability can be formally analyzed using existing hierarchical real-time scheduling theory. The most recent version, RT-Xen 2.0, is a real-time multi-core scheduler with a rich set of configurable features including global and partitioned schedulers, static and dynamic priority schemes, and different server algorithms [24].

RT-Xen and IRMOS have the same common goal to enable predictable execution and real-time performance in virtualized environments. Both investigate the use of scheduling algorithms rigorously designed based on real-time scheduling theory to provide CPU scheduling guarantees for VMs, and both advocate the use of hierarchical scheduling techniques to assess the schedulability of real-time workloads running within the VMs. However, none of them focuses on the dynamic allocation of the amount of resources that can be assigned in order to speed-up the execution time in order to meet the deadline. Our approach will investigate how to combine these techniques, with scaling techniques, such as [25], [26] in order to meet possible deadlines. Moreover, novel techniques for hard real-time scheduling of server systems have been recently designed [27], and could be exploited to extend the real-time guarantees and the flexibility of previously developed approaches.

The expression "fog architecture" refers to a distributed system composed of millions of edge, near-user devices that, as a whole, have a huge cumulative computational power [28], that could complement the cloud whenever communication constraints cannot meet the required deadlines. Modern cyber-physical systems such as vehicular networks, smart cities and advanced metering infrastructures are examples of fog architectures that are pervasive, and that collect data that, once analyzed properly, can enhance our daily lives. The overall idea of fog is to push the intelligence towards the physical objects, and to have them connected to fog nodes with a much more predictable and reliable connection. In such a configuration, researchers started devising solutions that can push for meeting real-time requirements also for cloud services and applications [29], [30], [31].

III. Concept and approach

As formulated in Section I the problem we are tackling in this paper is complex and broad and therefore a synergy of different sub-disciplines within computer science are combined in order to provide a suitable approach, namely safety and cyber-security engineering, model-checking and cloud computing. Fig. 1 depicts the details of the approach being proposed in this paper. The approach is divided in three subprojects (SPs), where APAC (see Section III-A) is the main link between other two SPs by providing architecture, modeling, analysis and planing methodology exploited by other two SPs.

The main contribution of this part lies in various models addressing adaptive features of the system, formal analysis, verification techniques and tools to verify that safety and cyber-security requirements are valid for a given system configuration. Run-time verification techniques cater for automatic (re)configuration of systems during adaptation, being scalable and able to guarantee the dependability of service in normal and adaptation phase.

CASSA (see Section III-B) puts focus on providing relevant safety and cyber-security requirements fed into both the analysis models defined within SP1, and the time-predictive methods defined in RTCloud (see Section III-C). Moreover, in order to cope with the dynamic and adaptive nature of such systems, CASSA contributes by extending hazard analysis and risk assessment techniques, as well as proposing methods for (semi-)automatic adaptation of safety and cyber-security assurance cases in such systems. The knowledge gathered with respect to safety relevant cyber-security and its mitigation strategies in CASSA is the starting point for including cyber-security aspects to detect and prevent attacks. RTCloud focuses on providing timing guarantees on the execution of real-time applications in the cloud.

A. Actor-based Platform for Adaptive Collaborative Systems (APAC)

In our approach we aim at developing an architecture that can capture different features of heterogeneous components, dynamic configuration and open environments of collaborative systems. The envisioned architecture of the system is based on the MAPE-K model of IBM (Monitor - Analysis - Plan - Execute - together with a Knowledgebase).

Our goal is to use various models to address adaptive features of the system. Formal analysis and verification techniques and tools will enable to verify that safety and cyber-security requirements are valid for a given system configuration. We envision the need to contribute to run-time verification techniques in order to build the analysis component, and to support automatic (re)configuration of systems during adaptation. These analysis techniques should be scalable and able to guarantee the dependability of service in a normal and adaptation phase. For the analysis and planning components, we also need performance evaluation, optimization, and decision-making policies.

Integrating the MAPE-K feedback loop with decentralized agent inspired approaches has been one of the challenges in the research community [32]. Actor model is among the pioneering approaches to address concurrent and distributed applications. The actor language has been originally introduced by Hewitt [33] as an agent-based language for programming secure distributed systems. Later on, it has been developed as a concurrent object-based language by Agha [34], and its formal semantics has been provided by Talcott [35]. The first formal verification tool, and the theory for compositional verification of an imperative actor-based language, Rebeca, is developed by Sirjani et al. [36]. With their loosely coupled units of concurrency, asynchronous message passing, and event-driven computation, actors are natural candidates to model a highly dynamic distributed system. The so-called isolation of actors

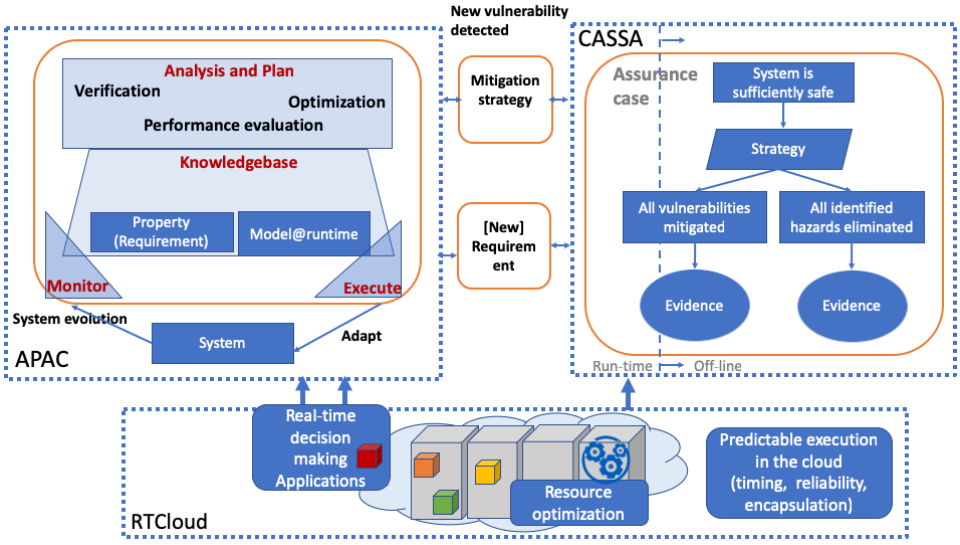

Fig. 1: A visual description of the proposed approach

helps in establishing efficient modular and compositional analysis and verification theories and techniques [37].

Within this part of the approach the following needs to be addressed, and it will be built based on the approach presented in [38]:

- **Building an architecture for run-time adaptation.** We have to design a hierarchical architecture to deal with safety and cyber-security in a dynamically changing environment, with a globally distributed and locally centralized setting.
- **Building run-time models.** We need to keep run-time models, so called models@run-time, as light-weight abstract reflections of the system, to be able to perform efficient and effective analysis, including formal verification and optimization, whenever necessary.
- **Developing run-time analysis and formal verification methods.** Build formal verification and analysis techniques for adaptive systems by focusing on change, to come up with more efficient techniques instead of heavy design-time techniques.
- **Developing run-time planning and optimization techniques.** Build (re-)planning and (re-)optimization techniques at run-time in order to handle possible changes and uncertain environment in a continuous way.

B. Continuous and Adaptive Safety and Security Assurance (CASSA)

Safety and cyber-security engineering have for a long time been regarded as two separate disciplines, which has resulted in separate cultures, regulations, standards and practices. Given the facts gathered in the literature [13], [39], [40], we can state that the need of joint safety and cyber-security work are increasingly understood and accepted, but that the state-of-the-practice has not reached the same level of maturity. There is still a huge gap between safety and cyber-security practices in the industry, due to separate standards, assessment and assurance processes, and authorities. To add on these already existing issues, today's systems are being built to connect to public or semi-public networks, are able to communicate with other systems, e.g., in the context of Internet-of-Things (IoT), involve multiple stakeholders, have dynamic system reconfigurations, and operate in increasingly unpredictable environments. In such complex systems, assuring safety and cyber-security in a continuous and adaptive manner is a major challenge, not the least due to the increasing number of attack surfaces resulting from the increased connectivity.

In CASSA our main focus is on bringing safety and cyber-security together by developing methods and approaches to enable continuous safety and cyber-security assurance in the domain of safety-critical adaptive systems, where systems are expected to address societal challenges. Systems relying on high level of adaptation will only be allowed in the "real world" when demonstrated to be safe and secure enough.

Having all previously described in mind, the following aspects needs to be taken care of:

- **Providing a set of mitigation strategies focusing on a complex adaptive environment.** To enable extensions of safety work towards cyber-security, we need to have knowledge about countermeasures that are effective in preventing exploitation of vulnerabilities that might lead to already identified or completely new hazards.
- **Extending the hazard analysis and risk assessment mandated by safety standards to include both safety and cyber-security**. Providing an approach that will consider cyber-security while reasoning about safety is highly important. Since the systems are increasingly interconnected, dynamic and adaptive, with such an approach we will be able to propose adequate measures suitable for common safety and cyber-security considerations.
- **Building a common safety and cyber-security assur-**

ance approach that will cater for a joint safety and cyber-security assurance case in complex adaptive systems. We see as an opportunity to investigate a possibility to use and extend on a pattern-based approach, similar to [41]. In this way, we will be able to establish and in a consistent way reuse existing safety and cyber-security assurance cases, enable knowledge preservation and traceability leading to guaranteeing that the system is sufficiently safe. We aim at evaluating different ways of expressing assurance cases, including Goal Structuring Notation (GSN) [42], as it is a structured notation, which provides graphically differentiated basic safety assurance element types, such as goals, evidence, strategy, etc., as well as description of all connection types between these elements.

- **Provide a solution on (semi-)automatic safety and cyber-security assurance case adaptation**. In our work we assume that the system evolution or vulnerability detection occurs at run-time. It is therefore important to provide techniques for the corresponding adaptation of the assurance case, possibly including both the argument structure and the evidence on which it relies. Ideally, this technique should be fully automatic, but since this is not possible in the general case, a more realistic ambition is a semi-automatic approach in which some steps are automatic and other are manual. Still, for some system adaptations, automatic adaptation of the assurance case could be possible.

C. Real-time cloud and fog computing (RTCloud)

Several industrial applications require time-predictable performance for guaranteeing a correct operation, including safety requirements. In a complex scenario, where multiple heterogeneous agents are involved, such as robots, cars or vehicles in a construction site, coordination and cooperation become a primary asset for maximizing the performance of the overall system, while at the same time minimizing the operational cost. This often results in a large and complex optimization problem that requires both a global knowledge of the status of the considered system, and large computational and storage capacity.

Whereas current cloud computing solutions can cope with such kind of problems, they fall short of time predictability. As a result, safety, and operational performance maybe compromised. Fog computing has been proposed as a computational model for extending the cloud closer to the objects to be controlled, and providing better timing performance. In our approach we aim at designing novel solutions for providing real-time guarantees on applications run in the fog and in the cloud, for safety-critical applications. To enable such an approach the following needs to be in place:

- **Define a model for safety critical applications to be run in the cloud**. Our aim is that the model uses and extends classical real-time systems scheduling theory, including resource models and virtualization technologies.
- **Development of real-time system scheduling techniques for real-time cloud applications**. Real-time applications must be scheduled according to the safety-critical requirements in virtualized environments.
- **End-to-end analysis of real-time cloud applications**. The fundamental properties of real-time applications have to be guaranteed at the level where the data is generated, and where the results of the computation is needed. For example, for a control application the time between the sensing and the actuation must be limited. Including such an aspect in classical real-time analysis techniques, is not trivial and requires several extensions to account for possible uncertainties introduced by the network connection, as well as communication delays and resource limitations.
- **Application placement**. Completely offloading applications to the cloud might be not feasible when strict timing requirements are needed. Therefore, intermediate solutions for providing real-time guarantees can be obtained by using new computation paradigms, such as fog computing, that provides less computational power than cloud computing, but has more reliable and predictable performance, as well as lower delays. Deciding where to place the real-time applications is therefore a key problem to provide predictable performance.
- **Resource optimization of real-time cloud applications**. The run-time execution of safety critical applications should be optimized through an efficient usage of the available resources, e.g., CPUs, memory, bandwidth, etc. Independently of where the application is executed, the amount of resources should be optimized, and the amount of reserved resources for the application are typically higher than the ones that are utilized at run-time, therefore overbooking mechanisms can be sought [43]. Such methodology, however, is conflicting with the predictability and the real-time system properties of safety critical applications, that require completely reserved resources. Therefore it is of highest importance ti analyze trade-offs and opportunities of such approaches.
- **Prototype implementation and evaluation**. The most promising solutions will be prototyped in collaboration with the industrial partners, in order to evaluate their performance in a realistic setting.

IV. Conclusions

In this paper we describe an approach that will enable development of run-time behavioral models for collaborative adaptive distributed systems, analysis techniques for continuous safety and cyber-security assurance, with real-time guarantees for the assumptions made in the model. The problem we are aiming to tackle with such an approach is complex and for its realisation it is needed to include three complementary disciplines within computer science, namely formal methods, safety and cyber-security engineering and cloud computing. Our initial intention is to test and apply proposed methods in industrial robotics, transportation and vehicular domain. However, our ultimate goal is to provide a methodology

applicable in several other domains including health and smart cities.

ACKNOWLEDGMENT

This work is performed within the following projects: the SAFSEC-CPS project funded by The Knowledge Foundation, the FiC and Serendipity projects funded by the Swedish Foundation for Strategic Research.

REFERENCES

[1] B.-J. Chang, Y.-L. Tsai, and Y.-H. Liang, "Platoon-based cooperative adaptive cruise control for achieving active safe driving through mobile vehicular cloud computing," *Wireless Personal Communications*, vol. 97, no. 4, pp. 5455–5481, 2017.

[2] G. Tamura, N. Villegas, H. Müller, J. P. Sousa, B. Becker *et al.*, "Towards Practical Runtime Verification and Validation of Self-Adaptive Software Systems," in *Software Engineering for Self-Adaptive Systems II*, ser. LNCS, R. de Lemos *et al.*, Eds., 2013, vol. 7475, pp. 108–132.

[3] D. Weyns, M. U. Iftikhar, D. G. de la Iglesia, and T. Ahmad, "A survey of formal methods in self-adaptive systems," in *C3S2E '12*, 2012.

[4] L. Baresi and C. Ghezzi, "A journey through SMScom: self-managing situational computing," *Computer Science - R&D*, 2013.

[5] M. U. Iftikhar and D. Weyns, "ActivFORMS: Active Formal Models for Self-Adaptation," in *9th Int. Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2014, pp. 125–134.

[6] B. Cheng, K. Eder, M. Gogolla *et al.*, "Using models at runtime to address assurance for self-adaptive systems," in *Models@run.time*, ser. LNCS, N. Bencomo *et al.*, Eds., 2014, vol. 8378, pp. 101–136.

[7] M. Sirjani and M. M. Jaghoori, "Ten years of analyzing actors: Rebeca experience," in *Formal Modeling: Actors, Open Systems, Biological Systems*, 2011, pp. 20–56.

[8] M. Sirjani and E. Khamespanah, "On time actors," in *Theory and Practice of Formal Methods*, ser. LNCS, vol. 9660, 2016, pp. 373–392.

[9] Int. Organization for Standardization, "Iso 26262 road vehicles functional safety part 1–10," 2011.

[10] Microsoft Corporation, "The stride threat model," 2005. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/commerce-server/ee823878(v=cs.20)

[11] G. Macher, A. Höller, H. Sporer, E. Armengaud, and C. Kreiner, *A Combined Safety-Hazards and Security-Threat Analysis Method for Automotive Systems*, pp. 237–250.

[12] C. Schmittner, T. Gruber, P. Puschner, and E. Schoitsch, *Security Application of Failure Mode and Effect Analysis (FMEA)*, 2014.

[13] W. Young and N. Leveson, "Systems thinking for safety and security," in *29th Annual Computer Security Applications Conf.*, ser. ACSAC, 2013.

[14] E. Lisova, I. Šljivo, and A. Čaušević, "Safety and security co-analyses: A systematic literature review," *IEEE Systems Journal*, pp. 1–12, 2018.

[15] A. Šurković, D. Hani'c, E. Lisova, A. Čaušević, K. Lundqvist, D. Wenslandt, and C. Falk, "Incorporating attacks modeling into safety process," in *SAFECOMP, ASUURE Workshop*, 2018.

[16] E. Lisova, A. Čaušević, K. Hänninen, H. Thane, and H. Hansson, "A systematic way to incorporate security in safety analysis," in *IEEE/IFIP Int. Conf. on Dependable Systems and Networks Workshops*, 2018.

[17] E. Denney, G. Pai, and I. Habli, "Dynamic safety cases for through-life safety assurance," in *IEEE/ACM IEEE Int. Conf. on Software Engineering*, vol. 2, 2015, pp. 587–590.

[18] R. Calinescu, S. Gerasimou, I. Habli, M. U. Iftikhar, T. Kelly, and D. Weyns, "Engineering trustworthy self-adaptive software with dynamic assurance cases," *CoRR*, 2017.

[19] I. Šljivo, B. Gallina, J. Carlson, H. Hansson, and S. Puri, "A method to generate reusable safety case argument-fragments from compositional safety analysis," *Journal of Systems and Software: Special Issue on Software Reuse*, vol. 131, pp. 570–590, 2016.

[20] I. Šljivo, B. Gallina, J. Carlson, and H. Hansson, "Configuration-aware contracts," in *4th Int. Workshop on Assurance Cases for Software-intensive Systems*, vol. 9923, 2016, pp. 43–54.

[21] M. García-Valls, T. Cucinotta, and C. Lu, "Challenges in real-time virtualization and predictable cloud computing," *Journal of Systems Architecture*, vol. 60, no. 9, pp. 726 – 740, 2014.

[22] T. Cucinotta, F. Checconi, G. Kousiouris, K. Konstanteli, S. Gogouvitis, D. Kyriazis, T. Varvarigou, A. Mazzetti, Z. Zlatev, J. Papay, M. Boniface, S. Berger, D. Lamp, T. Voith, and M. Stein, "Virtualised e-learning on the irmos real-time cloud," *Service Oriented Computing and Applications*, vol. 6, no. 2, pp. 151–166, 2012.

[23] S. Xi, J. Wilson, C. Lu, and C. Gill, "Rt-xen: Towards real-time hypervisor scheduling in xen," in *2011 Ninth ACM Int. Conf. on Embedded Software (EMSOFT)*, 2011, pp. 39–48.

[24] S. Xi, C. Li, C. Lu, and C. Gill, "Prioritizing local inter-domain communication in xen," in *IEEE/ACM 21st Int. Symposium on Quality of Service (IWQoS)*, 2013, pp. 1–10.

[25] E. B. Lakew, A. V. Papadopoulos, M. Maggio, C. Klein, and E. Elmroth, "Kpi-agnostic control for fine-grained vertical elasticity," in *17th IEEE/ACM Symposium on Cluster, Cloud and Grid Computing*, 2017.

[26] A. Ilyushkin, A. Ali-Eldin, N. Herbst, A. Bauer, A. V. Papadopoulos, D. Epema, and A. Iosup, "An experimental performance evaluation of autoscalers for complex workflows," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, 2018.

[27] A. V. Papadopoulos, M. Maggio, A. Leva, and E. Bini, "Hard real-time guarantees in feedback-based resource reservations," *Real-Time Systems*, vol. 51, no. 3, pp. 221–246, 2015.

[28] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *WS on Mobile Cloud Computing*, 2012.

[29] W. Tärneberg, A. V. Papadopoulos, A. Mehta, J. Tordsson, and M. Kihl, "Distributed approach to the holistic resource management of a mobile cloud network," in *IEEE Int. Conf. on Fog and Edge Computing*, 2017.

[30] I. Lujic, V. D. Maio, and I. Brandic, "Efficient edge storage management based on near real-time forecasts," in *IEEE Int. Conf. on Fog and Edge Computing (ICFEC)*, 2017, pp. 21–30.

[31] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards qos-aware fog service placement," in *2017 IEEE 1st Int. Conf. on Fog and Edge Computing (ICFEC)*, 2017, pp. 89–96.

[32] J. Andersson, R. de Lemos, S. Malek, and D. Weyns, "Modeling dimensions of self-adaptive software systems," in *Software Engineering for Self-Adaptive Systems*, 2009, pp. 27–47.

[33] C. Hewitt, "Description and theoretical analysis (using schemata) of PLANNER: A language for proving theorems and manipulating models in a robot," MIT Artificial Intelligence Technical Report, 1972.

[34] G. Agha, *Actors: A Model of Concurrent Computation in Distributed Systems*, 1990.

[35] C. Talcott, "Composable semantic models for actor theories," *Higher-Order and Symbolic Computation*, vol. 11, no. 3, pp. 281–343, 1998.

[36] M. Sirjani, A. Movaghar, A. Shali, and F. de Boer, "Modeling and verification of reactive systems using Rebeca," *Fundamenta Informatica*, 2004.

[37] M. M. Jaghoori, F. de Boer, D. Longuet, T. Chothia, and M. Sirjani, "Compositional schedulability analysis of real-time actor-based systems," *Acta Informatica*, 2016.

[38] M. Bagheri, M. Sirjani, E. Khamespanah, N. Khakpour, I. Akkaya, A. Movaghar, and E. A. Lee, "Coordinated actor model of self-adaptive track-based traffic control systems," *Journal of Systems and Software*, vol. 143, pp. 116–139, 2018.

[39] C. Schmittner, Z. Ma, E. Schoitsch, and T. Gruber, "A Case Study of FMVEA and CHASSIS As Safety and Security Co-Analysis Method for Automotive Cyber-physical Systems," in *ACM Workshop on Cyber-Physical System Security*, 2015.

[40] S. Kriaa, L. Pièrre-Cambacédès, M. Bouissou, and Y. Halgand, "A survey of approaches combining safety and security for industrial control systems," *Reliability Engineering and System Safety*, 2015.

[41] T. P. Kelly and J. A. McDermid, "Safety case construction and reuse using patterns," in *Safe Comp 97*, P. Daniel, Ed., 1997, pp. 55–69.

[42] T. Kelly and R. Weaver, "The goal structuring notation–a safety argument notation," in *dependable systems and networks 2004 workshop on assurance cases*. Citeseer, 2004.

[43] L. Tomás and J. Tordsson, "Improving cloud infrastructure utilization through overbooking," in *ACM Cloud and Autonomic Computing Conf.*, ser. CAC '13, 2013, pp. 5:1–5:10.