

ISSN 2186-7437

# NII Shonan Meeting Report

No. 2016-8

## Controlled Adaptation of Self-adaptive Systems (CASaS)

Paola Inverardi  
Nicolas D'Ippolito  
Kenji Tei

April 25–28, 2016



National Institute of Informatics  
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

# Controlled Adaptation of Self-adaptive Systems (CASaS)

Organizers:

Paola Inverardi (University of L'Aquila)  
Nicolas D'Ippolito (Universidad de Buenos Aires)  
Kenji Tei (National Institute of Informatics)

April 25–28, 2016

Self-adaptive systems are required to adapt its behaviour in the face of changes in their environment and goals. Such a requirement is achieved by developing self-adaptive systems as closed-loop systems following a Monitor Analyse Plan Execute (MAPE) scheme. MAPE loops are a mechanism that allows systems to monitor their state and produce changes aiming to guarantee that the goals are met. In order to achieve the desired goals, self-adaptive systems must combine a number of MAPE loops with several responsibilities and abstraction levels.

Higher-level goals require decision-level mechanisms to produce a plan in terms of the actions to be performed. Several mechanisms can be adopted for automatically generating decision-level plans (e.g. event-based controller synthesis) providing guarantees on the satisfaction of hard goals, and improvements on soft goals. Lower-level goals, on the other hand, require control mechanisms that sense the state of the environment and react to it thousands of times per second. Standard solutions to this problem are based on classical control theory techniques such as discrete-time control.

Indeed, systems requirements cannot be achieved by isolated MAPE loops, on the contrary, a combination of coordinated low-level and decision-level MAPE loops must be considered. Hence, multiple layers of control combining low latency discrete-time controllers with decision-level event-based controllers are a sensible approach to guarantee satisfaction of sophisticated goals in self-adaptive systems.

However, self-adaptive systems community has neglected the use of control theory as a central tool to guarantee hard goals while aiming to better satisfy soft goals. Very recently, software engineering community has started to study the application of control theory and the formal guarantees it provides in the context of software engineering. Proof of this is the Dagstuhl seminar Control Theory meets Software Engineering that has been held in 2014. The seminar covered diverse possible applications of control theory in software engineering. The Shonan meeting we propose, on the other hand, focuses more on formal guarantees that can be provided in self-adaptive system via the use of the broad area of control theory (e.g. event-based controller synthesis, discrete-time control).

In this meeting, we plan to involve a group of very active researchers in key areas such as Self-Adaptive Systems, Control theory, Game theory, Software Engineering, and Requirements Engineering, setting an adequate environment to discuss current and future applications and possibilities of control theory as a mechanism to provide formal guarantees on self-adaptive systems (e.g. convergence, safety, stability). We expect that having a number of participants from a wide variety of research areas will bring to light the benefits of incorporating the application capabilities and formal framework provided by control theory to self-adaptive systems.

We envisage the meeting to be organised in two main parts. First, there is a part for “stating common grounds” where participants will give talks stating their perspectives on their fields and key aspects where they visualise relations between their research area and other participants fields. Then, there is a second part for “discussions and work” where the most relevant topics selected by the participants will be discussed in work groups. Finally we plan to summarise the workshop findings and propose future actions.

We expect the meeting to encourage participants from different areas and communities to interact and actively search for common interests that may result in possible collaborations and joint work.

# Schedule

## Day 1st: 25th April

- 8:30-9:00: Opening
- 9:00-10:20: Session 1
  - Betty Cheng: A Search-Based Approach to Exploring Uncertainty for Self-Adaptive Systems
  - Gregor Engels: Model-Integrating Development of Software Systems: A Flexible Component-Based Approach
  - Pooyan Jamshidi: Fuzzy Self-Learning Controllers for Elasticity Management in Dynamic Cloud Architectures
  - Gabriel Tamura, Norha Villegas, and Hausi Muller: Lessons to Learn from Control Theory in Self-Adaptive Software Systems
- 10:20-10:50: Break
- 10:50-11:50: Session 2
  - Jeff Kramer: Architectures for Adaptation
  - David Garlan: Human-in-the-Loop Control Synthesis and Other Challenges for Controlled Adaptation
  - Shihong Huang: Incorporate Human Mental State in Self-Adaptive Systems
- 12:00-14:00: Lunch
- 14:00-15:20: Session 3
  - Zhenjiang Hu: Self-adaptation and bidirectional programming: towards a happy marriage
  - Romina Spalazzese: Towards Supporting the Internet of Things through Model-Driven Engineering
  - Shin Nakajima: Lessons Learned from a Self-adaptive Web Application
  - Thomas Vogel and Holger Giese: Toward Self-Adaptive Software Employing Model Predictive Control
- 15:20-15:50: Break
- 15:50-16:50: Session 4
  - Alan Colman: The illusion of control
  - Javier Camara Moreno: Managing Trade-Offs in Offline Synthesis of Adaptation Behavior
  - Nicolas D'Ippolito: FOND Planning as Reactive Control

## Day 2nd: 26th April

- 9:00-10:20: Session 5
  - Antonio Filieri: Lightweight Adaptive Filters for Learning Software Behaviors @Runtime
  - Marco Autili: Automatic Synthesis of Adaptable and Evolving Choreographies
  - Alessandro Papadopoulos: Control of Self-Adaptive Software in Presence of Uncertainty
  - Sebastin Uchitel: Dynamic Controller Update
- 10:20-10:50: Break
- 10:50-11:50: Session 6
  - Sasinee Pruekprasert: Quantitative Supervisory Control Game for Discrete Event Systems
  - Nir Piterman: Considering Time in Discrete Control
  - Martina Maggio: Automated Software Control: closing the loop around software systems with formal guarantees and limited prior knowledge
- 12:00-14:00: Lunch
- 14:00-15:20: Session 7
  - Filip Krikava: Towards the use of runtime models for infrastructure management and self-adaptation
  - Patrizio Pelliccione: Leveraging Collective run-time Adaptation for UAV-based systems
  - Liliana Pasquale: Topology Aware Adaptive Security of Cyber-Physical Systems
  - Kenji Tei: Qualitative Analysis for Adaptation Strategies
- 15:20-15:50: Break
- 15:50-16:50: Session 8
  - Paola Inverardi: Analysis for self-adaptation
  - Giovanni Quattrocchi: Advanced self-adaptation of cloud applications with containerization and control theory
  - Toshimitsu Ushio: Nonblocking Supervisory Control of Mealy Automata with Nondeterministic Output Functions

### **Day 3rd: 27th April**

- 9:00-9:30: Topic selection and team building
- 9:30-11:30: Break out discussion
- 11:30-13:00: Lunch
- 13:30-19:00: Excursion : Kamakura tour
- 19:00-21:30: Banquet in Kamakura

### **Day 4th: 28th April**

- 9:00-10:00: Presentation preparation
- 10:00-11:30: Presentation and discussion
- 11:30-11:45: Closing

## Overview of Talks

### A Search-Based Approach to Exploring Uncertainty for Self-Adaptive Systems

Betty Cheng, Michigan State University

A controller-based self-adaptive system (SAS) can monitor itself and its environment to assess requirements satisfaction at run time. However, unexpected combinations of system configurations and environmental parameters may prevent an SAS from satisfying its key functional and performance objectives. It is also impossible for a human to exhaustively enumerate all possible conditions that the SAS will experience throughout its lifetime. We overview Loki, a novelty search-based approach for discovering environmental and system conditions that can lead to requirements violations and hidden, unwanted behaviors in an SAS.

### Model-Integrating Development of Software Systems

Gregor Engels, Universitt Paderborn

Software systems need to be modified continuously to suit new customer requirements. To support this endeavour, systems need to be flexible. They should be modifiable and changeable with reasonable effort and be able to adapt themselves permanently. A promising way to develop flexible software systems is to use models as part of the system that are analyzed, modified and executed at runtime.

Building such model-integrating systems is a challenging task. On the one hand, the design of Domain-Specific Modeling Languages (DSMLs) needs to be considered explicitly during the software development process. On the other hand, the systems need to be developed in a modular way by composing the system from building blocks.

Model-Driven (MDD) and Component-Based Development (CBD) are two established orthogonal approaches that can tackle the mentioned challenges. MDD is based on the use of models and modeling languages as first-class entities to systematically engineer software systems. CBD enables the engineering of modular systems by facilitating a divide-and-conquer approach with reuse. However, combining and aligning the individual principles from both approaches is an open research problem.

In the talk, we describe Model-Integrating Development (MID), an engineering approach that enables the systematic development of component-based, model-integrating software. MID combines principles from MDD and CBD and is based on the central assumption that models and code shall be treated equally as first-class entities of software throughout its lifecycle. In particular, MID leverages the added flexibility that comes with models at run-time, i.e., models are an integral part of software.

The feasibility of the proposed solution concept is rationalized based on a reference implementation that provides the basis for a thoroughly described and critically discussed feasibility study: a dynamic access control product line.

## A Flexible Component-Based Approach

Pooyan Jamshidi, Imperial College London

Cloud controllers support the operation and quality management of dynamic cloud architectures by automatically scaling the compute resources to meet performance guarantees and minimize resource costs. Existing cloud controllers often resort to scaling strategies that are codified as a set of architecture adaptation rules. However, for a cloud provider, deployed application architectures are black-boxes, making it difficult at design time to define optimal or preemptive adaptation rules. Thus, the burden of taking adaptation decisions often is delegated to the cloud application. We propose the dynamic learning of adaptation rules for deployed application architectures in the cloud. We introduce FQL4KE, a self-learning fuzzy controller that learns and modifies fuzzy rules at runtime. FQL4KE is an extension of our previous cloud controller RobustT2Scale with an online learning capability. The benefit is that we do not have to rely solely on precise design-time knowledge (as it was the case in RobustT2Scale), instead FQL4KE acquires knowledge by interacting with the environment as a reinforcement learning agent. The experimental results for cloud applications on Azure and OpenStack demonstrate that FQL4KE outperforms RobustT2Scale and the native Azure and OpenStack auto-scaling.

## Fuzzy Self-Learning Controllers for Elasticity Management in Dynamic Cloud Architectures

Gabriel Tamura<sup>(1)</sup> and Norha Villegas<sup>(1)</sup> and Hausi Muller<sup>(2)</sup>, <sup>(1)</sup>Icesi University <sup>(2)</sup>University of Victoria

Self-adaptive software (SAS) perform dynamic adjustments to their own structure or behavior autonomously, to maintain desired properties in response to monitored changes in the systems' operational contexts. Control theory provides verifiable feedback models to realize this kind of autonomous control, for a broad class of industrial systems for which precise models can be defined.

Even though recent MAPE-K models for SAS systems, along with variants such as the hierarchical ACRA, address a broader range of control tasks, they do not provide the inherent assurance mechanisms that control theory does, as they do not explicitly identify and establish the properties that reliable controllers should have. These properties, in general, result not from the abstract models, but from the specifics of control strategies, which are precisely what these models fail to analyze. We show that, even for systems too complex for direct application of classical control theory, the abstractions of control theory provide design guidance that identifies important control characteristics and raises critical design issues about the details of the strategy that determines the controllability of the resulting systems. This in turn enables careful reasoning about whether the control characteristics are in fact achieved.

In this presentation we analyze the correspondences between feedback control and the MAPE-K loop elements, and examine several control strategies illustrated with examples from both domains, classical control theory and SAS, and show how the issues addressed by these strategies can and should be considered for the design and assurance of self-adaptive software systems. From this



examination we distill challenges for developing principles that may serve as the basis of a control theory for the assurance of self-adaptive software systems.

## **Architectures for Adaptation**

Jeff Kramer, Imperial College London

Self-adaptive systems should cope with unforeseen changes in the environment and/or the system goals at runtime. A self-adaptive software architecture must therefore support runtime change of the system configuration (i.e., the systems components, their bindings and operational parameters) and behaviour update (i.e., component orchestration). Although controlling configuration and behaviour at runtime has been discussed and applied to architectural adaptation, architectures for self-adaptive systems often compound these two aspects reducing the potential for adaptability. We propose a reference architecture, MORPH, that allows for coordinated yet transparent and independent adaptation of system configuration and behaviour, supported by a knowledge repository for logging and reasoning.

## **Human-in-the-Loop Control Synthesis and Other Challenges for Controlled Adaptation**

David Garlan, Carnegie Mellon University

One of the driving concerns for adaptive systems is to eliminate or reduce costly and error-prone human involvement in the dynamic monitoring and repair of systems. However, in many cases fully automatic adaptation - without any user in the loop - may not be possible or desirable. For instance, achieving certain system goals may require a human to carry out a physical operation. In other cases, a system may be able to achieve a goal fully automatically, but the task could be done even better if a user were involved. This leads to an important problem: how and when to involve a user in achieving system adaptation goals? In this talk we describe an approach, based on stochastic multi-player games, that synthesizes human-in-the-loop control strategies. Key to the approach is a simple but effective model of humans that permits quantitative reasoning about human participation. We offer this problem and its solution as one of several emerging challenges for controlled adaptation, and outline several more that are critical to progress in the field.

## **Incorporate Human Mental State in Self-Adaptive Systems**

Shihong Huang, Florida Atlantic University

The increasing complexity and dynamism of software intensive systems require that the systems must be able to adapt themselves at runtime to react and deal with the uncertainties and unpredictable nature of the environments in which they operate. The uncertainty may be due to changes in the operational environment, variability of resources, new user needs, etc. Human behavior also contributes large amounts of uncertainty. For example, smart environments are impoverished in large measure because the system has so little knowledge about

what the user wants, is thinking, is confused about, likes, or dislikes. As a result it has been very difficult to build effective systems that anticipate and react to users needs. On the other hand, research has shown that including human participation (i.e., human-in-the-loop adaptation) can lead to more productivity and effectiveness. This points to one of the missing links, and an important research question: how to anticipate and react to humans mental states i.e., their thoughts and intention.

The advancement in the field of Brain Computer Interfaces (BCI) and the availability of more general accessible gadgets to non-neuroscientist users, have made it possible to measure a humans mental states by using neural input (brain waves). Having direct neural inputs to systems is the natural next step to enable software systems to feel and anticipate users.

In this talk, we describe an approach of using neural input as measurement of humans mental states. The goal is to use quantitative measurements of mental states to predict and evaluate the effectiveness of a human participant in a human-in-the-loop system, allowing for efficient decision-making about when and whether they should be included in system adaptation. This talk also intends to share thoughts more generally on ways the software engineering community can move beyond overt human behavior in order to directly connect brain neural input to computer systems. Through the unique features of neural inputs (e.g., signally intentions, desired actions, covert attention, thoughts, memories, covert emotions, etc.), smart software systems will someday feel and anticipate users intention and therefore react self-adaptively.

## **Self-adaptation and bidirectional programming: towards a happy marriage**

Zhenjiang Hu, National Institute of Informatics

In this talk, I would like to show that bidirectional programming can provide a powerful mechanism to modularize self-adaptive software. This mechanism would be very useful not only for reusing a self-adaptive software for different target systems, but also for maintaining separation of concerns when developing complex self-adaptive software.

## **Towards Supporting the Internet of Things through Model-Driven Engineering**

Romina Spalazzese, Malmo University

Nowadays connectivity and technology are more and more ubiquitous and affordable and a trend is to connect everything that can benefit from being connected, both from the digital and physical world. The Internet of Thing (IoT) has a great potential to improve our way of living and working through a seamless and highly dynamic cooperation among heterogeneous Things.

However, properly dealing with the design, development, deployment and runtime management of IoT applications means to provide solutions for a multitude of challenges. Among them there are for instance: supporting complexity and heterogeneity management, supporting collaborative development,

maximising reusability of design artefacts, and providing self-adaptation of IoT systems.

We are investigating these issues from different perspectives and we devise Model-Driven Engineering (MDE) as a key-enabler for solving such challenges and supporting the lifecycle of IoT systems, from their design to runtime management. More specifically, (i) we devise a Model-Driven Engineering Framework supporting the modeling of Things and self-adaptation of Emergent Configurations of connected systems in the IoT; and (ii) show how MDE in general, and our framework in particular, can help in tackling the above mentioned challenges by providing the Smart Street Lights case as concrete case.

## **Lessons Learned from a Self-adaptive Web Application**

Shin Nakajima, National Institute of Informatics

Open systems such as Web applications are dynamically adaptive to allow desired levels of flexibility. We conducted a series of studies on some issues with a PHP-based adaptive Web application system. First, we present an abstract, declarative framework and relate it to the adaptive Web application architecture, which takes a model-based adaptation approach. The formalism can be a basis for understanding the distinctive roles of the model information and runtime mechanism. Second, the adaptivity may be achieved by replacing constituent software components at runtime, and a new method is required to ensure that the replacement or substitution is safe. We propose a formal framework of the safe substitution for self-adaptive systems and present a policy-based, system integrity verification method. Third, runtime monitoring and checking usually has a large impact on the runtime execution performance. The overhead of checking is unavoidable for fatal situations. For less severe anomalies, the performance penalty may not be acceptable if their occurrence rate is almost negligible. We propose a new method of estimating the occurrence rate of such rare events at runtime with the importance sampling method. The estimated result gives us a hint whether we incorporate the runtime checking method into the system.

## **Toward Self-Adaptive Software Employing Model Predictive Control**

Thomas Vogel and Holger Giese, HassoPlattnerInstitut (University of Potsdam)

Basic control schemes that are proposed for self-adaptive software are often inherently linked to the offline analysis of the open and closed-loop models and they do not fit well to the characteristics of software. In contrast, model predictive control offers an promising direction to reason about the impact of control actions at runtime, which fit better to the characteristics of software. This better match is witnessed by related efforts to build strategies maximizing the expected reward for agents. Consequently, we suggest employing model predictive control enriched with online system identification as an orientation to engineer self-adaptive software with dynamic architectures.

In this presentation we will outline the high-level idea of model predictive control enriched with online system identification and related ideas in the agent

community. Then, we outline the development and assurance steps to systematically develop self-adaptive software with dynamic architectures, which are implied by the proposed orientation. We will clarify which capabilities are required for realizing each of these steps, which results already exist in this direction, and where further research is needed. Finally, we will broaden our considerations and also discuss the decentralized case. In this case, we have multiple adaptation engines each operating as a model predictive controller. We will discuss the additional problems and challenges that result from the decentralization.

## **The illusion of control**

Alan Colman, Swinburne University of Technology

Based on the findings of a large systematic survey of research into the application of control theory to software systems, I will discuss some of the trends, challenges and inherent limitations of these approaches. I will ponder the extent to which the tantalizing vision of 'systems theory' might lead us ignore some of these limitations.

## **Managing Trade-Offs in Offline Synthesis of Adaptation Behavior**

Javier Camara Moreno, Carnegie Mellon University

Self-adaptation improves the resilience of software-intensive systems by endowing them with the ability to adapt their structure and behavior to run-time changes (e.g., in workload and resource availability). Many of these approaches reason about the best way of adapting by synthesizing adaptation plans online via planning or model checking tools. This method enables the exploration of a rich solution space, but optimal solutions and other formal guarantees (e.g., constraint satisfaction) are computationally costly and can result in long planning times during which changes may invalidate plans.

An alternative to online planning that eliminates the burden of run-time overhead involves employing methodologies grounded in control theory that target tunable variables to produce offline control strategies for adaptive systems with formal guarantees. However, most control-theory based approaches are not flexible enough to explicitly consider trade-offs among multiple concerns as a first order element (e.g., to capture situations in which reducing performance is acceptable if the security of the system is improved). In this talk, I will outline an approach for offline synthesis of adaptation behavior that can explicitly consider trade-offs among multiple concerns, and explore potential synergies with control theory-based approaches.

## **FOND Planning as Reactive Control**

Nicolas D'Ippolito, Universidad de Buenos Aires

In this talk I'll present the relation between automated planning and controller synthesis. I'll present an efficient approach to translate FOND planning problems to controller synthesis problems that falls in the same computational complexity class (i.e. EXPTIME).

## **Lightweight Adaptive Filters for Learning Software Behaviors @Runtime**

Antonio Filieri, Imperial College London

Adaptive software is required to withstand unpredicted and continuously changing usage behaviors and execution environments. Effective adaptation decisions require learning current information about the system behavior, usually via a combination of monitoring and learning. Learning requires coping with uncertainty, to distinguish signal from noise, and detecting changes promptly and robustly, to react timely without being skewed by outliers. These conflicting requirements are further complicated at runtime by the limited availability of time and computational resources, challenging most of the statistical learning methods appeared in literature.

In this talk, I will discuss the use of adaptive filters for learning numerical qualities of a running systems from monitoring data. A particular focus will be given to learning transition probabilities for Markov models, including a comparison with established statistical learning techniques.

## **Automatic Synthesis of Adaptable and Evolving Choreographies**

Marco Autili, University of L'Aquila

In the near future we will be surrounded by a virtually infinite number of software applications that provide services in the digital space. This situation promotes reuse-based software production through composition of existing software services distributed over the Internet. Choreographies are a form of decentralized composition that models the external interaction of the participant services by specifying peer-to-peer message exchanges from a global perspective. When mismatching third-party services are to be composed, obtaining the distributed coordination and adaptation logic required to suitably realize a choreography is a non-trivial and error prone task. Automatic support is then needed. Nowadays, very few approaches address the problem of actually realizing choreographies in an automatic way. In this talk, a method for the automatic synthesis of adaptable and evolving choreographies is presented. Coordination software entities are synthesized in order to proxify and control the participant services' interaction. When interposed among the services, coordination entities enforce the collaboration prescribed by the choreography specification. The ability to evolve the coordination logic in a modular way enable choreography evolution in response to possible changes. I illustrate the method at work on a use case in the Urban Traffic Coordination domain. The use case is being developed as part of CHOReVOLUTION, a 3 years EU H2020 project on automated synthesis of choreographies for the Future Internet.

## **Control of Self-Adaptive Software in Presence of Uncertainty**

Alessandro Papadopoulos, Lund University

The increasing complexity in different areas of computing systems calls for

novel inter-disciplinary approaches, able to cope with dynamic and uncertain environments. Different decision-making mechanisms are considered in the literature, but the robustness of the proposed techniques are proven only on a limited number of cases, without providing general guarantees. The aim of this talk is to present: 1) an overview of simple control techniques that can be applied to relevant problems in computing systems, 2) a performance assessment methodology that gives probabilistic guarantees of the proposed techniques in presence of uncertainty. These concepts will be explained via a running example on load-balancing techniques for cloud computing.

## Dynamic Controller Update

Sebastin Uchitel, Universidad de Buenos Aires

In this talk I will discuss the problem of dynamically updating the controller of a reactive system. Dynamic controller update is essential for providing continuous system operation in the face of environment and requirements change. Given a system being controlled to satisfy a specification (environment assumptions, requirements and interface) how should the controller be changed to ensure the satisfaction of a new specification? How should the transition from one specification to the other be handled? I will discuss the notion correctness for dynamic update and how its correctness criteria can be specified. Furthermore, I will present a technique for automatically computing a controller that handles the transition from the old to the new specification, assuring that the system will reach a state in which such a transition can correctly occur. I will show how, using controller synthesis, a controller that guarantees both progress towards update and safe update can be constructed. I will also discuss various case studies we have experimented with.

## Quantitative Supervisory Control Game for Discrete Event Systems

Sasinee Pruekprasert, Osaka University

We formulate an optimal supervisory control problem for quantitative non-terminating discrete event systems (DESSs) modeled by finite weighted automata. The control performance of a supervisor is evaluated by the worst-case limit-average weight of the infinite sequences generated by the supervised DES. An optimal supervisor is a supervisor that avoids deadlocks and maximizes the control performance. We propose a game theoretical design method for an optimal supervisor using a two-player turn-based mean-payoff game automaton. As the first player, the objective of the supervisor is to maximize the worst-case limit-average weight of the generated sequences; as the second player, the DES aims to minimize it. We show that an optimal supervisor can be computed from an optimal strategy (of the first player) for this game. Then, we propose an algorithm to compute an  $f$ -minimally restrictive optimal supervisor, which is a finite-memory optimal supervisor that enables as many sequences as possible and can be represented by an optimal strategy for a finite version of the two-player game.

## **Considering Time in Discrete Control**

Nir Piterman, Leicester University

In recent years researches in the robotics community have been using discrete synthesis techniques to produce discrete controllers that are then combined with motion- and device-controllers to produce hybrid robot controllers. The combination of the discrete controller into a hybrid controller leads to discrepancies between the arising and desired behavior. In this talk we report on extending the modeling approach, and following it the synthesis algorithm, in order to take into account the duration of time of actions in the ensuing controller.

## **Automated Software Control: closing the loop around software systems with formal guarantees and limited prior knowledge**

Martina Maggio, Lund University

Modern software should satisfy multiple concerns simultaneously: it should provide predictable performance, be robust to failures, handle peak loads and deal seamlessly with unexpected conditions and changes in the execution environment. For this to happen, the design of software applications should take into account the possibility of executing changes at runtime and providing formal guarantees on the software's behavior. Control theory has been identified as one of the possible design drivers for runtime adaptation, but the adoption of this discipline's principles often requires additional knowledge to be processed by a specialist. This talk presents a way to overcome this limitation: an automated methodology that try to extract the necessary information from experimental data and design a control system for runtime adaptation. The talk will cover the work done in this area, starting from systems that achieve a single goal, and moving on to systems that achieve one goal at a time, creating a chain of controllers. Finally, a generalization of the approach is described, showing the first automated strategy that takes into account multiple concerns without separating them into multiple control strategies. Avoiding the separation allows one to tackle a vaster class of problems, with respect to the previous research and to provide a wider set of guarantees, at the expense of centralization.

## **Towards the use of runtime models for infrastructure management and self-adaptation**

Filip Krikava, Czech Technical University

The emergence of tools allowing to treat infrastructure configurations programmatically have revolutionized the way computing resources and software systems are managed. However, these tools keep lacking an explicit model representation of underlying resources making it difficult to introspect, verify or reconfigure the system in response to external events.

In this presentation, I will outline a novel approach that treats system infrastructure as explicit runtime models. A key benefit of using such models@run.time representation is that it provides a uniform semantic foundation

for resources monitoring and reconfiguration including non-functional properties (e.g. reconfiguration timescale). Adopting models at runtime allows one to integrate different aspects of system management, such as resource monitoring and subsequent verification into an unified view which would otherwise have to be done manually. This in turn simplifies development of self-adaptation without requiring engineers and researchers to cope with low-level system complexities.

## **Leveraging Collective run-time Adaptation for UAV-based systems**

Patrizio Pelliccione, Chalmers University of Technology

UAV-based systems are both mission-critical and safety-critical since UAVs collaborate with humans and other devices to accomplish defined missions. UAV-based systems are composed of a team of drones, various devices (like movable cameras, sensors), and human agents. Environments in which UAV-based systems operate are often unpredictable and unknown; in other words a model of the environment containing for instance obstacles, no fly zones, wind and weather conditions might be available, however, we cannot assume that such model will be always both correct and complete. In this paper, we describe a novel approach for managing the run-time adaptation of UAV-based systems. Our approach is based on a generic collective adaptation engine, which addresses collective adaptation problems in a decentralized fashion, operates at run-time, and enables the addition of new entities at any time. Moreover, our approach provides a way to dynamically understand which parts of the system should be selected to solve an adaptation issue. The approach has been empirically evaluated in the context of a private company surveillance scenario.

## **Topology Aware Adaptive Security of Cyber-Physical Systems**

Liliana Pasquale, Lero

Cyber-Physical Systems can be harmed through both cyber-enabled or physically-enabled attacks, particularly ones that exploit the often ignored interplay between the cyber and physical spaces characterizing a system operating environment. Awareness of the topology of the operating environment of systems as well as its dynamics can support adaptive security more effectively.

In this talk I propose the use of Bigraphical Reactive Systems to represent the topology of cyber and physical spaces. I describe how to use this representation to reason about the consequences of the evolution of topological configurations on the satisfaction of security requirements. I also illustrate a planning technique to identify an adaptation strategy to be used at runtime, to circumvent, prevent, or mitigate security requirements violations previously identified. Finally, I briefly discuss future work, which will address assurances about when and how it is better to adapt.



## **Qualitative Analysis for Adaptation Strategies**

Kenji Tei, National Institute of Informatics

Software systems are required not only to continue working but also to maintain its qualities at desired levels, even in the environment where continuously changes. Making a system as self-adaptive, by adding an adaptation strategy that checks current state of the system obtained by probes and gauges, and conducts tactics meeting their conditions, is a promising approach. Research question addressed here is how does the system efficiently determine whether a strategy is hopeless in current situation?. Even if a strategy initially worked well, it may become impossible to lead the system to satisfy quality requirements due to changes in the environment. Such a hopeless strategy should be detected as soon as possible and be swapped with hopeful one in order to achieve required levels of quality. In this presentation, I will show techniques to validate hopeless adaptation strategies at runtime. In essence, our approach is computationally and informationally cheap. The techniques consists of (1) qualitative modeling of adaptation strategies and managing system and (2) model checking over these models to identify hopeless situations of the strategies. To support the techniques, we devised a procedure to create qualitative models, and developed tools for editing, transforming, and verifying the models.

## **Analysis for self-adaptation**

Paola Inverardi, University of L'Aquila

In this talk i will discuss the role that (non-functional) analysis can play in developing self-adaptive context aware systems. I will discuss the general approach, show an application on the synthesis of self-adapting connectors and discuss current and future work.

## **Advanced self-adaptation of cloud applications with containerization and control theory**

Giovanni Quattrocchi, Politecnico di Milano

Nowadays many web applications are deployed and executed in the Cloud. Different techniques have been presented to automate and improve the management of these applications: academia mainly focused on self-adaptation and dynamic resource allocation at the infrastructure layer; on the other hand, industry addressed portability, agile development, and maintenance leading to the DevOps movement. In this context containerization is an emerging technique of virtualization where many processes (containers) can run independently on the same machine sharing a host operating system. Because containers do not contain an entire OS, they are more lightweight and faster to boot than virtual machines. In addition, containers can be also run within a virtual machine thus enabling finer-grained resource management.

This presentation introduces a MAPE-based solution that exploits both this granularity and a control theory based planner. The planner consists of a discrete-time feedback controller tailored to the structure assumed for the

command-to-metrics dynamics to be governed, plus a linearisation block also constructed based on the same assumed dynamics. The planner is also endowed with an internal saturation management (antiwindup) mechanism. We also advocate that a modern self-adaptive system for web application should not only manage the infrastructure layer but also the platform (middleware) and the application layers. We show how containers can help us to provide multi-layer adaptation keeping a generic and technology-agnostic planner. The evaluation of our approach has been conducted by using two cloud applications: we first compared our control theory planner with the state of the art using only virtual machines, then we compared our planner with and without container management.

## **Nonblocking Supervisory Control of Mealy Automata with Nondeterministic Output Functions**

Toshimitsu Ushio, Osaka University

Supervisory control is a logical control method for discrete event systems modeled by automata. We review fundamental results of the supervisory control such as controllability and observability. Then, we consider a discrete event system with perturbed outputs modeled by Mealy automata with nondeterministic output functions. We introduce an anti-permissive supervisor and show necessary and sufficient conditions for the existence of a strongly nonblocking anti-permissive supervisor for the Mealy automata.

## List of Participants

- Paola Inverardi, University of L'Aquila, Italy
- Nicolas D'Ippolito, Universidad de Buenos Aires, Argentina
- Kenji Tei, National Institute of Informatics, Japan
- Antonio Filieri, Imperial College London, UK
- Martina Maggio, Lund University, Sweden
- Nir Piterman, Leicester University, UK
- Marco Autili, University of L'Aquila, Italy
- Sebastin Uchitel, Universidad de Buenos Aires, Argentina
- Jeff Kramer, Imperial College London, UK
- Gabriel Tamura, Icesi University, Colombia
- Norha Villegas, Icesi University, Colombia
- Hausi A. Muller, University of Victoria, Canada
- Gregor Engels, Universitt Paderborn, Germany
- Thomas Vogel, University of Potsdam, Germany
- Holger Giese, HassoPlattnerInstitut(University of Potsdam), Germany
- Giovanni Quattrocchi, Politecnico di Milano, Italy
- Alan Colman, Swinburne University of Technology, Australia
- Zhenjiang Hu, National Institute of Informatics, Japan
- Toshimitsu Ushio, Osaka University, Japan
- David Garlan, Carnegie Mellon University, USA
- Filip Krikava, Czech Technical University, Czech Republic
- Alessandro Papadopoulos, Lund University, Sweden
- Shin Nakajima, National Institute of Informatics, Japan
- Patrizio Pelliccione, Chalmers University of Technology, Sweden
- Romina Spalazzese, Malmo University, Sweden
- Javier Camara Moreno, Carnegie Mellon University, USA
- Betty Cheng, Michigan State University, USA
- Shihong Huang, Florida Atlantic University, USA
- Liliana Pasquale, Lero, Ireland
- Sasinee Pruekprasert, Osaka University, Japan
- Pooyan Jamshidi, Imperial College London, UK