Analysis of Autonomous Mobile Collectives in Complex Physical Environments

Gleirscher, Mario; Haxthausen, Anne Elisabeth; Leucker, Martin ; Linker, Sven

Published in:
Dagstuhl Seminar Proceedings

Link to article, DOI:
10.4230/DagRep.9.10.95

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

Report from Dagstuhl Seminar 19432

Analysis of Autonomous Mobile Collectives in Complex Physical Environments

Edited by

Mario Gleirscher[1], Anne E. Haxthausen[2], Martin Leucker[3], and Sven Linker[4]

1 **University of York, GB,** mario.gleirscher@york.ac.uk
2 **Technical University of Denmark – Lyngby, DK,** aeha@dtu.dk
3 **Universität Lübeck, DE,** leucker@isp.uni-luebeck.de
4 **University of Liverpool, GB,** s.linker@liverpool.ac.uk

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19432 "Analysis of Autonomous Mobile Collectives in Complex Physical Environments". Our working hypothesis for this seminar was that for systems of such complexity and criticality, the trustworthy certification and the successful operation in society will strongly benefit from the coordinated application of several rigorous engineering methods and formal analysis techniques. In this context, we discussed the state-of-the-art based on the working example of a Smart Farm. Our aim was to understand the practical challenges and the capabilities and limitations of recent formal modelling and analysis techniques when tackling these challenges, and to initiate a special research community on the verification of autonomous collectives.

Seminar October 20–23, 2019 – http://www.dagstuhl.de/19432
2012 ACM Subject Classification Computer systems organization → Robotic autonomy, Computing methodologies → Model development and analysis, Theory of computation → Logic and verification, Theory of computation → Timed and hybrid models
Keywords and phrases autonomous collectives, control engineering, formal verification, hybrid systems, uncertainty and risk
Digital Object Identifier 10.4230/DagRep.9.10.95
Edited in cooperation with Frederik Forchhammer Foldager

1 Executive Summary

Mario Gleirscher (University of York, GB)
Anne E. Haxthausen (Technical University of Denmark – Lyngby, DK)
Martin Leucker (Universität Lübeck, DE)
Sven Linker (University of Liverpool, GB)

Motivation

Autonomous vehicles (AVs) are facing strong proof obligations. Individual AVs can be part of a *collective* (e.g. a platoon of utility vehicles on a farm field, a truck convoy on a highway, a convoy of passenger vehicles on urban road, an in-door aerial platoon, a railway convoy) and act within a heterogeneous environment of other collectives, for example, pedestrians, bicyclists, and motorcyclists. Multiple AVs might have to correctly and reliably negotiate

Analysis of Autonomous Mobile Collectives in Complex Physical Environments, *Dagstuhl Reports*, Vol. 9, Issue 10, pp. 95–116
Editors: Mario Gleirscher, Anne E. Haxthausen, Martin Leucker, and Sven Linker
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

their order of passing a crossing or reliably and robustly arrange in a certain work layout on agricultural land. Individuals and collectives in such environments, whether controlled in a centralised or distributed way, are subjected to change, uncertainty, and defects. Moreover, *complex environments* typically deny a comprehensive segregation of physical space and, hence, involve interactions with entities out of control (e.g. human-controlled machines, pedestrians, animals) and mostly also out of sight of an individual machine's (short-range) sensors.

Objective

This seminar was centred around an application challenge, the **Smart Farm**. Participants were encouraged to discuss how their research addresses typical **engineering tasks** (ETs; upper layer in Fig. 1) to be accomplished for the given challenge or for similar challenges. These tasks include

1. the identification, modelling, and analysis of operational situations in complex environments
2. real-time coordination, composition, and reconfiguration of machine collectives with a focus on (i) interaction with human-operated systems, humans, animals, infrastructure and (ii) situation-specific centralised or distributed control regimes
3. the determination of strongest safety and performance guarantees with a focus on (i) the estimation of upper resilience bounds of machine collectives and lower reliability bounds of individual machines and (ii) the determination of strongest guarantees under partial state knowledge, with minimal infrastructural support, and under reduced controllability.

In the discussions of how the ETs can be accomplished best, we also aimed at investigating **abstractions of defects and uncertainties**, for example:

- controller, communication, and infrastructure failures (e.g. erroneous vehicle-to-X connection and communication, deficient road infrastructure),
- undesired interference or disturbance of autonomous operation (e.g. malicious and unintended misuse; controller, communication, and infrastructure attacks),
- practical sensor uncertainties, actuator perturbations, and partial state knowledge.

Defects and uncertainties are crucial for constructing *realistic models* of the behavioural spectrum of mobile collectives and yet abstract enough to perform *practical reasoning*. Likewise, such models allow the necessary freedom to express ideal and actual behaviour, independent of whether such behaviour is desirable. This freedom can involve the use of non-deterministic models. In any case, a *(property) specification* would label some of the observable behaviours as *desirable*, some as *undesirable*, others somewhere in between (cf. quantitative verification). The more complete and precise such a specification, the better the distinction between correct, undesirable, and other classes of behaviours of a collective.

Our **overall objective** with this seminar was to gain a common understanding of acceptable safety and performance of autonomous mobile collectives in presence of defects and other uncertainties typically occurring in complex open environments. The **overarching approach** of all seminar contributions was the **formal analysis and verification of behavioural correctness** under these assumptions (lower layer in Fig. 1) by using techniques such as, e.g. theorem proving, model checking, run-time verification, and model-based testing.

Our **central assumption for this seminar** was that the given application challenge or any similar challenges render individual methods for the analysis and verification of such systems insufficient. For example, in control-theoretic models such collectives are modelled by differential equations. Interaction within and among collectives and with their

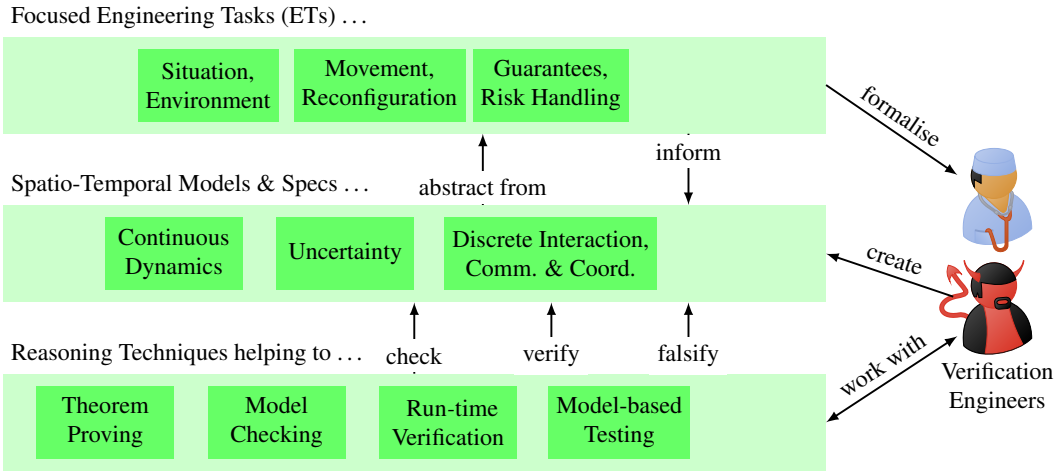

Figure 1 Topic structure of the seminar.

environment, governing these equations, cannot be easily encoded. Approaches that express such interactions well, however, typically struggle with the detailed description of the physical laws the AVs need to adhere to. Hence, for ensuring correct behaviour in such a setting, layered abstractions, corresponding models, and **specialised reasoning techniques have to be combined**.

Organisation

Before the seminar, we provided each participant with material about the *application challenge* (see Section 4.1) together with list of *engineering tasks* and *research questions*. We encouraged the participants to apply their approach, if available, to at least one of the ETs of the application challenge and to answer at least one of the research questions. Alternatively, participants were invited to present any research and practical experiences related to the seminar topic and the challenge. Everyone was given the opportunity to give a full-length talk. Table 1 shows the seminar structure, the talks, and further sessions. After the welcome session, participants introduced themselves to the group. The rest of the seminar was organised into *talk sessions* and *break-out sessions*.

Talks

In the talk sessions, we investigated several **research questions** from different angles. We had talks about (1) industry challenges, (2) the analysis and verification of properties of individual autonomous vehicles (two sessions), (3) the analysis and verification of properties of autonomous collectives, and (4) the modelling of uncertainty for the (quantitative) property verification of critical autonomous systems. Nine talks dealt with an **introduction of a specific verification approach** suitable for tackling an aspect of the application challenge, including a summary of the state-of-the-art of this approach. Four talks were about **industrial examples** of a nature similar to the Smart Farm, highlighting technical challenges, encountered issues, and perceived practical obstacles. Five talks focused on the **application of a particular approach to a particular aspect of the Smart Farm**, addressing some of the research questions.

In the following, we list the main questions and the participants whose talks highlighted a particular aspect of the corresponding question. For more details, see the list of talk abstracts below.

Table 1 Seminar schedule.

	Monday	Tuesday	Wednesday
9:00	Introductions	***Industry Challenges*** J. Brauer: *Verification of Autonomous Transport Systems - Some Industrial Prospects*	*Break-out session*
9:30		S. Fröschle: *Trustworthy identity and key management for mobile systems in transportation*	
10:00–10:30	break	break	break
10:30	***Individual Properties*** P.G. Larsen/F. Foldager: *A Journey Towards a Fleet of Autonomous Robots for Agricultural Field Operations*	***Uncertainty Modelling*** K.G. Larsen: *Synthesis of Safe, Optimal and Small Strategies for Advanced Driver Assistance using UPPAAL Stratego*	*Break-out and discussion*
10:50	J.B. Jeannin: *Collision avoidance and path replanning of individual farm robots*	D. Parker: *Probabilistic model checking for safety and performance guarantees*	*Closing discussion*
11:10	A. Fantechi: *Safety aspects of autonomous systems*	R. Calinescu: *Stochastic modelling underpinning the engineering of trustworthy autonomous systems*	
11:30	P.C. Ölveczky: *Formal modeling and analysis of real-time systems using Real-Time Maude*	M. Gleirscher: *Risk Structures*	
12:15–13:30	lunch	lunch	lunch
13:30	***Collective Properties*** M. Waga: *Optimization of the watering schedule by run-time and design-time analysis*	***Individual Properties*** C. Heinzemann: *Context Analysis and Requirements Derivation with SCODE*	
13:50	É. André: *White-box and black-box quantitative verification of timing properties*	S. Bogomolov: *Trusted Autonomous Systems: Verification Meets Falsification*	
14:10	P. Ribeiro: *Modelling and Verification using RoboChart*	S. Mitsch: *Modular Verification of Cyber-Physical Systems in KeYmaeraX*	
14:30	(spare)	(spare)	
15:00–15:30	break	break	
15:30 16:00 16:30 17:00	*Break-out session* *Discussion of results*	*Break-out session* *Discussion of results*	
18:00	dinner	dinner	

1. How can each ET be solved? How can we achieve safety in presence of distribution, mobility, and uncertainty? Which mechanisms fit best to ensure safety in the *application challenge*?
   Frederik Foldager and Peter Gorm Larsen
2. How do we model the systems and verify *safety and progress* properties? Can we always find acceptable PARETO optima over safety and performance, at traffic level, at the level of a collective, and for individual machines?
   Étienne André, Sergiy Bogomolov, Kim Larsen, David Parker
3. How can we *exploit the structure* of practical AVs and collectives to craft specific verification techniques (e.g. prevent state space explosion, identify fundamental theorems)?
   Stefan Mitsch, Pedro Ribeiro
4. Which benefits do we gain from *integrating* design-time verification, model-based testing, and run-time verification?
   Mario Gleirscher, Masaki Waga
5. How can verification techniques be incorporated into the *development process* of AVs?
   Jörg Brauer, Radu Calinescu, Alessandro Fantechi, Peter Csaba Ölvecky

6. Which *complications* arise from the verification of AVs and how can we mitigate the impact of these complications, particularly, during practical verification?
   Sibylle Fröschle, Christian Heinzemann

Break-Out Sessions

To stimulate interaction, we created *break-out groups* on each seminar day and on the following topics: challenges of verifying autonomous collectives, the challenge of uncertainty (using, e.g. quantitative verification, parametric model checking), abstractions of space & uncertainty, the impact of IT security issues on AV safety, and safe platooning. Additionally, several smaller groups (sometimes consisting of only two participants) met to discuss combinations and extensions of the topics they presented in their respective talks.

One break-out group focused on creating a big picture of the **challenges of verifying autonomous mobile collectives** in the Smart Farm. The identified problems include

- estimation of behavioural properties (e.g. exact arrival times of agents, dead-lock freedom of the plan), real-time interleaving of sensing and control, and finding the "sweet spot" between precision and performance when used at run-time,
- model checking at scale, when to use online or offline analysis for verification and synthesis (e.g. synthesis of distributed safety controllers for automatic repair/fallback),
- useful architectural abstractions, compositionality, and refinement (e.g. how to safely partition the tasks of a mission between system components or whole robots?),
- security of communication and robustness of control to communication glitches (e.g. how to integrate a jamming model into overall system verification?),
- languages/models for dealing with system failures (e.g. how to cope with failures of individual autonomous vehicles in the context of a collective?) and component failures (e.g. how to safely integrate machine learning into autonomous systems?), and
- safety in the presence of uncertainty (e.g. how to quantify uncertainty?, how to deal with uncertainty in parameters and in the structure of the system and the environment?).

Another group investigated **the challenge of uncertainty in modelling**, discussing how uncertainty (e.g. due to partial observability) can be dealt with in automated verification and how techniques such as quantitative verification can be used to solve verification problems with uncertainties in the considered parameters. Depending on the Smart Farm aspect to be tackled, state-of-the-art approaches include the use of interval abstractions for parameters, the calculation of confidence intervals for verification results, and the use of counterexample-guided abstraction refinement.

The break-out session on **space and uncertainty** stretched over all three days, and was concerned with the possible ways to specify spatial aspects, as well as how to incorporate uncertainty into such specifications. Our discussion proceeded on different topics. We discussed, which types of sensors allow robotic systems to gain spatial knowledge, and what levels of uncertainty can be expected. Based on this, we examined whether several layers of space are necessary and beneficial to specify both the systems and their desired properties (e.g., a discrete layer for planning high-level actions and a continuous layer, on which more local properties are ensured by controllers, as for example obstacle avoidance). Furthermore, we compared the different types of uncertainty, the level of spatial layers they occur on, and their impact on systems in the Smart Farm. This included a discussion of how much knowledge needs to be globally available, and what can be kept locally at the level of each individual entity. We realised that while the modelling scenario allowed for different levels of space and uncertainty, it was not easy and straightforward to identify necessary and

interesting spatial properties to analyse. Hence, we agreed that the case study needs to allow for more degrees of freedom (e.g., different routes to reach physical targets, to permit several alternative plans).

The session about **IT security of farm collectives** focused on the aspect of communication security. First, the group identified the typical communication requirements between the actors of a smart farm such as: between a robot and a supervisory control (perhaps including a drone), between two robots that carry out a task on the same field (e.g. to carry out the task cooperatively or for collision avoidance), between a sensor and a control centre (e.g. for watering). Altogether, it became clear that the operation of a smart farm critically depends on the secure and timely communication between the various actors. It is also clear that in the setting of the smart farm the actors must communicate over wireless channels. Hence, the usual threats against communication over an open medium apply, e.g. message spoofing and manipulation, eavesdropping and jamming. On the one hand, this requires us to employ appropriate security protocols and key management, which can guarantee origin and message authenticity as well as confidentiality. On the other hand, this requires further measures against availability attacks such as jamming. The group focused on the threat of jamming. While jamming cannot be prevented in an open system the general idea was to take a 'detect and mitigate' approach. For example, jamming can be detected by the absence of regular 'heartbeat' signals and by combination with visual channels. Mitigation strategies involve raising an alarm and removing the jamming device in a timely fashion while ensuring the system is not overly susceptible to false positives and denial-of-service attacks. Neither detection nor mitigation seemed trivial when discussed in detail. On the positive side, the verification methods and tools presented at the seminar could be used to evaluate possible strategies, and perhaps, even to synthesise them. Later on the group joined the break-out group on platooning, where communication is particularly critical.

In the break-out session on **safe platooning on the farm**, we discussed

1. the handling of *planned events* being part of the normal operation of a platoon (e.g. several farm vehicles, lorries and harvesters, form a platoon including leader election; a lorry wants to join or leave a harvesting platoon; a platoon with two consecutive lorries needs to be rearranged; a lorry decides to leave the platoon) and
2. the detection of *critical (not necessarily undesired) events* to be dealt with or to recover from during normal operation (e.g. a foreign vehicle, a farmer's car, enters the platoon area; communication error because of a jamming attack or a hardware failure disturbs the platoon controller; the current leader looses trustworthiness, e.g. because of being hacked, by deviating from the common goal of the platoon; farm workers enter the working area of the platoon).

Our discussions lead to a deeper understanding of the intricacies, both from the perspectives of different verification approaches and from the viewpoint of certification obligations. The results of our discussion are suitable for the identification of *formal properties* to be used as proof obligations in certification activities as well as the modelling of so-called protocol automata *describing the inter- and intra-modal behaviour* required to handle some of the mentioned events. Such models can then serve as a basis for hazard and risk assessment activities as well as for safety verification.

Outcomes and Conclusions

Our expectations for this first seminar were modest. We wanted to learn from each others' perspectives, to discuss available approaches, and to identify the hardest and most relevant **open challenges**.

Our discussions opened **paths to an integration and application of the presented theories and models** (middle layer in Fig. 1), particularly, continuous models (e.g. timed and hybrid automata), uncertainty models (e.g. Markov chains, probabilistic automata), communication and coordination models (e.g. timed process algebra). We investigated the use of such models in the context of various reasoning techniques (e.g.theorem proving, model checking, run-time verification, model-based testing). These discussions lay a basis for the *derivation of guidelines* on how the approaches, when applied to systems such as the Smart Farm, can be combined and/or enhanced to tackle the identified problems *in practical contexts subject to certification efforts*.

The attendees were from various fields such as formal verification, testing, certification, mechanical and control engineering, and embedded IT security, working at universities, in industry-oriented research institutes, or directly in industry. In this setting, we were able to **share experiences and insights from various application domains** (e.g. smart farming, smart energy systems, train/railway systems, automotive and transportation), to discuss issues of the Smart Farm scenario, and to examine potential research directions. Particularly, we observe that commonalities among the used approaches give rise to an *integrated and more versatile approach*. Our participants from industry receive the opportunity to convert any of these insights into lasting process improvements in their safety-critical domains. We expect our findings to be *relevant to regulatory authorities* in these domains.

In overall, we believe this seminar was an important step to **foster collaboration** of researchers and practitioners experienced with the different models and reasoning techniques, and to **initiate a research community** focusing on autonomous collectives of similar or even higher complexity than the Smart Farm. To that end, we are planning **further meetings** of the seminar's participants in the near future, to allow for further refinement of the models, and combinations of the methods presented. Additionally, we will further improve and **extend the modelling scenario**, so that a particular combination of specification and verification approaches can be explored in more detail. Eventually, we intend to collect our findings possibly in a special issue of a suitable journal.

Funding and Acknowledgements. Sven Linker was supported by the Engineering and Physical Sciences Research Council programme grant EP/N007565/1 (S4: Science of Sensor Systems Software). Mario Gleirscher was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the Grant no. 381212925. We are grateful to Sibylle Fröschle for summarising the results of the IT security session. Further thanks go to Frederik Foldager for collecting and compiling the abstracts. We would like to spend sincere gratitude to all participants for their contributions and for their support and active engagement in making this seminar an insightful experience.

2 Table of Contents

Executive Summary
Mario Gleirscher, Anne E. Haxthausen, Martin Leucker, and Sven Linker 95

Overview of Talks

White-box and black-box quantitative verification of timing properties
Étienne André 103

Trusted Autonomous Systems: Verification Meets Falsification
Sergiy Bogomolov 104

Verification of Autonomous Transport Systems – Some Industrial Prospects
Jörg Brauer 104

Stochastic modelling underpinning the engineering of trustworthy autonomous systems
Radu Calinescu 104

Safety aspects of autonomous systems
Alessandro Fantechi 105

A Journey Towards a Fleet of Autonomous Robots for Agricultural Field Operations
Frederik Foldager and Peter Gorm Larsen 105

Trustworthy Identity and Key Management for Mobile Systems in Transportation
Sibylle Fröschle 106

Risk Structures: Specification Templates for Controller Synthesis
Mario Gleirscher 106

Context Analysis and Requirements Derivation with SCODE
Christian Heinzemann 106

Synthesis of Safe, Optimal and Compact Strategies using UPPAAL Stratego
Kim Guldstrand Larsen 107

Modular Verification of Cyber-Physical Systems in KeYmaera X
Stefan Mitsch 107

Probabilistic Model Checking for Safety and Performance Guarantees
David Parker 108

Modelling and Verification using RoboChart
Pedro Ribeiro 109

Optimization of the watering schedule by run-time and design-time analysis
Masaki Waga 109

Formal modeling and analysis of real-time systems using Real-Time Maude
Peter Csaba Ölveczky 110

Open problems

Specification of the Application Challenge
Mario Gleirscher, Anne E. Haxthausen, Martin Leucker, and Sven Linker 110

Participants 116

3 Overview of Talks

3.1 White-box and black-box quantitative verification of timing properties

Étienne André (University of Paris North, FR)

Joint work of Masaki Waga, Étienne André, Ichiro Hasuo
Main reference Masaki Waga, Étienne André, Ichiro Hasuo: "Symbolic Monitoring Against Specifications Parametric in Time and Data", in Proc. of the Computer Aided Verification – 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I, Lecture Notes in Computer Science, Vol. 11561, pp. 520–539, Springer, 2019.
URL https://doi.org/10.1007/978-3-030-25540-4_30

In this talk, I will envision two parts: on a white box model, i.e., on a formal model of (part of) the system, I will propose to use parametric timed model checking techniques to formally evaluate the correctness of (some of) the timing aspects, but also to evaluate their robustness, i.e., the effect of infinitesimal variations on the system correctness. That is, how critical can be some timing parameters, such as *del_t* or *gps_t*, to the system correctness? The formalism used will be parametric timed automata [1].

Then, on a black box model (obtained by either concrete execution or, more likely, on simulation using tools such as Simulink), I will propose efficient run-time verification techniques to *monitor* the system behavior, again taking into consideration the timing aspects and their robustness. On the one hand, on a "shorter-time scale", the absence of collisions, but also the *robust* absence of collisions (i.e., situations of "near collisions") should be monitored. On the other hand, on a "longer-time scale", the absence of rotten ripes, and their robust counterpart ("near-rotten" situations) should be monitored. The ultimate goal is to not only perform a Boolean monitoring, but to detect problematic timeframes, and to provide them with a quantitative measure of the property. This implies to be able to write specifications in some quantitative formalism sufficiently expressive to allow to detect such failure, together with some robustness values. The formalism used could be (variants of) *parametric timed data automata*, a formalism recently proposed with Ichiro Hasuo and Masaki Waga [2].

References

1 Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. Parametric real-time reasoning. In Rao Kosaraju, David S. Johnson, and Alok Aggarwal (eds.), STOC'93, ACM, pages 592–601, 1993. DOI: 10.1145/167088.167242

2 Masaki Waga, Étienne André and Ichiro Hasuo. Symbolic monitoring against specifications parametric in time and data. In Işil Dillig and Serdar Tasiran (eds.), CAV'19, Springer LNCS 11561, pages 520-539, July 2019. DOI: 10.1007/978-3-030-25540-4_30

3.2 Trusted Autonomous Systems: Verification Meets Falsification

Sergiy Bogomolov (Newcastle University, GB)

Joint work of Sergiy Bogomolov, Goran Frehse, Amit Gurung, Dongxu Li, Georg Martius, Rajarshi Ray
Main reference Sergiy Bogomolov, Goran Frehse, Amit Gurung, Dongxu Li, Georg Martius, Rajarshi Ray: "Falsification of hybrid systems using symbolic reachability and trajectory splicing", in Proc. of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2019, Montreal, QC, Canada, April 16-18, 2019, pp. 1–10, ACM, 2019.
URL https://doi.org/10.1145/3302504.3311813

Falsification algorithms for hybrid systems aim at finding trajectories that violate a given safety property. This is a challenging problem, and the practical applicability of current falsification algorithms still suffers from their high time complexity. In contrast to falsification, verification algorithms aim at providing guarantees that no such trajectories exist. Recent symbolic reachability techniques are capable of efficiently computing linear constraints that enclose all trajectories of the system with reasonable precision. In this talk, we present an approach which leverages the power of symbolic reachability algorithms to improve the scalability of falsification techniques. Recent approaches to falsification reduce the problem to a nonlinear optimization problem. We propose to reduce the search space of the optimization problem by adding linear state constraints computed by a reachability algorithm. We showcase the efficiency of our approach on a number of standard hybrid systems benchmarks demonstrating the performance increase in speed and the number of falsifiable instances.

3.3 Verification of Autonomous Transport Systems – Some Industrial Prospects

Jörg Brauer (Verified Systems International GmbH – Bremen, DE)

Coming from industry, most of our projects are to some extent based on development standards such as the RTCA DO-178 for avionics systems, which have not really been set up with adaptive or autonomous systems in mind. In this talk, we focus on some aspects of how safety certification and autonomy do not really match up, and what we can do about it.

3.4 Stochastic modelling underpinning the engineering of trustworthy autonomous systems

Radu Calinescu (University of York, GB)

Stochastic modelling is a powerful tool for establishing performance, dependability and other key properties of systems and processes during design, verification and at run-time. However, the usefulness of this tool depends on the accuracy of the models being analysed, on the efficiency of the analysis, and on the ability to find models corresponding to effective system and process architectures and configurations. This talk will describe how recent approaches to stochastic model learning, analysis and synthesis address major challenges posed by these prerequisites, extending the applicability of stochastic modelling to autonomous systems.

3.5 Safety aspects of autonomous systems

Alessandro Fantechi (University of Florence, IT)

The talk will review the currently considered/implemented techniques and policies for safety enforcement of autonomous railway vehicles, with the aim to derive a more general conceptual model encompassing the principles upon which safety of autonomous vehicles is assessed

Notions of uncertainty over positioning and speed metering of autonomous vehicles are also inherited from what is currently investigated in the railway domain, and generalised to the three-dimensional case.

The sketched concepts are then instantiated on the provided benchmark, as a contribution to develop an analytic safety assessment process.

3.6 A Journey Towards a Fleet of Autonomous Robots for Agricultural Field Operations

Frederik Foldager (Aarhus University, DK) and Peter Gorm Larsen (Aarhus University, DK)

In this presentation, we provide an overview of the collaboration between a proactive Danish SME called Agrointelli and Aarhus University to make the vision of a fleet of autonomous robots for arable farming a reality. The work surrounds a full-scale robot called Robotti which is now sold commercially. The journey includes both a series of different joint research projects involving many other institutions as well as considerations of commercial and business development. We will give an introduction to how we have modelled the soil-machine interaction using the Discrete Element Method on a component level, as well as explaining the models that have been made both of the dynamics of the robot, its complex physical environment, in particular in relation to different soil-types and the model of the different levels of the discrete event controllers on a systems level. Many of these have been combined using a technology called co-simulation which also includes capabilities for exploring alternative designs in a virtual setting as well as connecting it to 3D visualization engines. Some of these models are naturally commercially sensitive but we are also able to share a purely public version of these multi-models. Our current research involves supporting this with a digital twin capability in a real-time fashion and scaling up to a fleet of robots operating in collaboration with humans. We expect to close the presentation with some research challenges that we currently see as the most prominent ones.

3.7 Trustworthy Identity and Key Management for Mobile Systems in Transportation

Sibylle Fröschle (OFFIS – Oldenburg, DE)

In this talk I will talk about the importance and challenges of trustworthy identity and key management for mobile autonomous systems, and illustrate this by examples from the automotive, aerospace, and maritime domain. I will then present current research on how to answer these challenges, including how to obtain verifiable security and resilience guarantees on the system-of-systems layer. Finally, I will report on practical experiences within the working group "Identity management and security" of the Maritime Connectivity Platform (MCP).

3.8 Risk Structures: Specification Templates for Controller Synthesis

Mario Gleirscher (University of York, GB)

Main reference Mario Gleirscher: "Run-Time Risk Mitigation in Automated Vehicles: A Model for Studying Preparatory Steps", in Proc. of the Proceedings First Workshop on Formal Verification of Autonomous Vehicles, FVAV@iFM 2017, Turin, Italy, 19th September 2017, EPTCS, Vol. 257, pp. 75–90, 2017.
URL http://dx.doi.org/10.4204/EPTCS.257.8

To achieve desirable safety, autonomous systems will have to detect, predict, and reduce risk by incorporating risk models and risk handling mechanisms that enhance their mission controllers. Complex environments and the missing fallback to human operators pose tough challenges to the engineering of risk handlers, particularly, to the hazard analysis and risk modelling leading to such handlers. This talk will discuss research on an algebraic framework for risk modelling and analysis. It will also be highlighted how one can use a specific risk model to derive proof obligations for mission controllers with safety mechanisms.

3.9 Context Analysis and Requirements Derivation with SCODE

Christian Heinzemann (Robert Bosch GmbH – Stuttgart, DE)

Autonomous systems, particular autonomous driving systems, need to cope with complex environments and are subject to a multitude of influences that have an impact on the necessary behavior of a system. To this end, key questions are what constitutes a correct behavior in a given situation and how to derive a complete-as-possible set of requirements for an autonomous system in a given environment (or context)? In my talk, I will outline an approach based on essential analysis (also known as morphological analysis) for capturing influence factors from a system's context and for deriving a set of top-level requirements (or modes of operation) that denote an expected system reaction to a specific combination of external influences. The approach guarantees that the derived top-level requirements (or modes of operation) are consistent and complete with respect to the known and specified influences.

3.10 Synthesis of Safe, Optimal and Compact Strategies using UPPAAL Stratego

Kim Guldstrand Larsen (Aalborg University, DK)

In this talk I gave an overview of the UPPAAL tool suite with outset in the Smart Farming Benchmark of the seminar. The classical version of UPPAAL allows for a Timed Automata model of the timed behaviour of robots capturing their movement on the road as well as entering and leaving collection point and field. In particular, timing properties may be verified here given best and worst case timing information.

A refinement of the timed automata model interpret delays stochastically giving rise to Stochastic Timed Automata. Here expected and probabilistic threshold properties may be settled using the statical model checking engine of UPPAAL SMC.

In the setting of two robots, we model their joint behaviour as a (product) Timed Game. This allows for synthesis of most permissive safety controllers, where crashes between robots is guaranteed to be avoided.

Finally, we add stochastic components for weather prediction and hybrid components in terms of differential equations describing the growth of crops in the field. Given this overall model – a stochastic hybrid game – we use the reinforcement learning method of UPPAAL Stratego to obtain a near optimal sub-strategy of the no-crash safety strategy.

3.11 Modular Verification of Cyber-Physical Systems in KeYmaera X

Stefan Mitsch (Carnegie Mellon University – Pittsburgh, US)

Joint work of Stefan Mitsch, Andre Platzer, Brandon Bohrer, Yong Kiam Tan, Nathan Fulton, Andreas Müller, Wieland Schwinger, Werner Retschitzegger, Jan-David Quesel, Marcus Völp, Magnus O. Myreen

Cyber-physical systems (CPS) combine cyber aspects such as communication and computer control with physical aspects such as motion in space; they have many important applications, e.g., in robotics, aerospace, and automotive domains, but require careful designs to meet stringent safety demands. Formal verification techniques justify such safety properties but need to handle mathematical models of CPSs called hybrid systems, i.e., those that combine the discrete dynamics of stepwise controller computations with the continuous dynamics of their differential equations. Modularity principles for the design and formal verification of cyber-physical systems are especially beneficial when a system consists of many cooperating entities that together must satisfy some safety criteria. This talk discusses how differential dynamic logic (dL) for hybrid systems can be used to model and verify CPS in a modular fashion. Its theorem prover KeYmaera X provides compositional verification techniques for hybrid systems, which not only handle nonlinear systems but also use invariants to reduce the verification of larger systems to subsystems. For very large models, component-based modeling can be used to split large models into multiple component models with local responsibilities to further reduce modeling complexity.

References

1 Brandon Bohrer, Yong Kiam Tan, Stefan Mitsch, Magnus O. Myreen, and André Platzer. VeriPhy: Verified controller executables from verified cyber-physical system models. In Dan Grossman, editor, *PLDI*, pages 617–630. ACM, 2018.
2 Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer. KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In Amy Felty and Aart Middeldorp, editors, *CADE*, volume 9195 of *LNCS*, pages 527–538, Berlin, 2015. Springer.
3 Nathan Fulton and André Platzer. Safe reinforcement learning via formal methods: Toward safe control through proof and learning. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *AAAI*. AAAI Press, 2018.
4 Andreas Müller, Stefan Mitsch, Werner Retschitzegger, Wieland Schwinger, and André Platzer. Tactical contract composition for hybrid system component verification. *STTT*, 20(6):615–643, 2018. Special issue for selected papers from FASE'17.
5 Andreas Müller, Stefan Mitsch, Wieland Schwinger, and André Platzer. A component-based hybrid systems verification and implementation tool in keymaera X (tool demonstration). In Roger D. Chamberlain, Walid Taha, and Martin Törngren, editors, *Cyber Physical Systems. Model-Based Design – 8th International Workshop, CyPhy 2018, and 14th International Workshop, WESE 2018, Turin, Italy, October 4-5, 2018, Revised Selected Papers*, volume 11615 of *LNCS*, pages 91–110. Springer, 2018.
6 Stefan Mitsch and André Platzer. ModelPlex: Verified runtime validation of verified cyber-physical system models. *Form. Methods Syst. Des.*, 49(1-2):33–74, 2016. Special issue of selected papers from RV'14.
7 André Platzer. A complete uniform substitution calculus for differential dynamic logic. *J. Autom. Reas.*, 59(2):219–265, 2017.

3.12 Probabilistic Model Checking for Safety and Performance Guarantees

David Parker (University of Birmingham, GB)

This talk gives an overview of the state of the art in probabilistic model checking, with a particular focus on the theme of the seminar: formally analysing collections of autonomous robots. I will describe some recent related applications of these techniques, including synthesising autonomous mobile robot plans with probabilistic guarantees and verifying adaptive mission plans for unmanned underwater vehicles. Motivated by the application challenge for the seminar, I will also summarise some recent directions on verification for partially observable models, stochastic games and multi-robot systems.

3.13 Modelling and Verification using RoboChart

Pedro Ribeiro (University of York, GB)

Joint work of Pesro Ribeiro, James Baxter, Ana Cavalcanti, Madiel Conserva, Simon Foster, Wei Li, Alvaro Miyazawa, Pedro Ribeiro, Augusto Sampaio, Jon Timmis, Jim Woodcock
Main reference Alvaro Miyazawa, Pedro Ribeiro, Wei Li, Ana Cavalcanti, Jon Timmis, Jim Woodcock: "RoboChart: modelling and verification of the functional behaviour of robotic applications", Software and Systems Modeling, Vol. 18(5), pp. 3097–3149, 2019.
URL http://dx.doi.org/10.1007/s10270-018-00710-z

Designing robotic systems can be very challenging, yet controllers are often specified using informal notations with development driven primarily by simulations and physical experiments, without clear relation to abstract models of requirements. Our goal is to support roboticists in writing models and applying modern verification techniques using a language familiar to them. To that end, we consider RoboChart, a domain-specific modelling language based on UML, but with a restricted set of constructs to enable a simplified formal semantics and automated reasoning. It supports the specification of reactive, timed and probabilistic behaviours. We illustrate how RoboChart can be used to specify the behaviour of individual robots in the context of the smart farm. We pursue an analysis of the collective using a discrete model of the environment and the model-checker FDR.

References

1 A. Miyazawa, P. Ribeiro, W. Li, A. Cavalcanti, J. Timmis, and J. C. P. Woodcock. Robochart: modelling and verification of the functional behaviour of robotic applications. *Software & Systems Modeling*, 18(5):3097–3149, Oct 2019.
2 A. Miyazawa, P. Ribeiro, W. Li, A. Cavalcanti, and J. Timmis. Automatic property checking of robotic applications. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3869–3876, Sep. 2017.
3 A. Cavalcanti, A. Sampaio, A. Miyazawa, P. Ribeiro, M. Conserva Filho, A. Didier, W. Li, and J. Timmis. Verified simulation for robotics. *Science of Computer Programming*, 174:1 – 37, 2019.

3.14 Optimization of the watering schedule by run-time and design-time analysis

Masaki Waga (National Institute of Informatics – Tokyo, JP)

By design-time verification of a real-time model (e.g., timed automata or time Petri Nets), we can verify if there are any potential deadline misses. However, to confirm the verified deadline is reasonable, we have to model the environment, or we have to exploit some empirical knowledge (e.g., previous environmental data). In this talk, I will talk about a data-driven approach to confirm the deadline in the watering by robots. Typically, we modeled the watering by the robots and the change of the water level of the fields, and show how to obtain the safe set of the watering intervals by symbolic monitoring, which is one of the run-time verification methods. As an example of the watering strategy, we also show that a simple round-robin strategy can be modeled by a parametric timed automaton and the worst-case watering interval can be obtained e.g., by IMITATOR.

3.15 Formal modeling and analysis of real-time systems using Real-Time Maude

Peter Csaba Ölveczky (University of Oslo, NO)

Real-Time Maude is a tool that extends the rewriting-logic-based Maude system to support the executable formal modeling and analysis of real-time systems. Real-Time Maude is characterized by its general and expressive, yet intuitive, specification formalism, and offers a spectrum of formal analysis methods, including: rewriting for simulation purposes, search for reachability analysis, and both untimed and metric temporal logic model checking. Real-Time Maude is particularly suitable for specifying real-time systems in an object-oriented style, and its flexible formalism makes it easy to model different forms of communication.

This modeling flexibility, and the usefulness of Real-Time Maude for both simulation and model checking, has been demonstrated on many advanced state-of-the-art applications, including both distributed protocols of different kinds and industrial embedded systems. Furthermore, Real-Time Maude's expressiveness has also been exploited for defining the formal semantics of MDE languages for real-time/embedded systems, including Ptolemy discrete-event models, a subset of the avionics modeling standard AADL, and a timed extension of the MOMENT2 model transformation framework. Real-Time Maude thereby provides formal model checking capabilities for these languages for free, and such analysis has been integrated into the tool environment of a number of modeling languages.

This talk gives a high-level overview of Real-Time Maude and some of its applications. The talk also briefly discusses what features of Real-Time Maude and associated Maude-based tools are suitable for certain aspects of the smart farm case study (e.g., object orientation to model robots, the ability to define complex data types and functions to model, e.g., areas and collision courses, and so on) and for which aspects of the case study the tool environment seems less suitable (e.g., complex continuous behaviors).

4 Open problems

4.1 Specification of the Application Challenge

Mario Gleirscher (University of York, GB), Anne E. Haxthausen (Technical University of Denmark – Lyngby, DK), Martin Leucker (Universität Lübeck, DE), and Sven Linker (University of Liverpool, GB)

The following material was provided to and used by the seminar participants to present their approach in the context of common application domain.

4.1.1 Purpose of this Specification

In the following, we describe a scenario, where several autonomous robots solve a common task. The intention behind this description is to provide a framework for the discussion within the seminar. To that end, we invite you to model parts of the scenario with formalisms of your choice. However, we **do not expect** that you model the whole scenario, but encourage

you to pick the parts that you are interested in. Furthermore, even if your formalisation for certain aspects is not complete, we appreciate comments on whether this is due to your choice of formalism, or for other reasons.

The *main goal of this exercise* is to identify common ground between different formalisations and approaches, and how they could be used in combination to enhance modelling and analysis of such scenarios. In other words,

1. when similar aspects of this challenge have been modelled by different seminar participants, we expect to discuss the *differences as well as advantages and disadvantages* of each approach, and
2. when complementary aspects have been modelled, we expect a discussion of *how these models are related* and together contribute to the assurance of the overall plant.

4.1.2 The Challenge

The scenario we consider is an instance of *smart farming*. A local farm consists of several fields and green houses, where fruit and vegetables are grown. The farm and the fields are connected by public roads, which may (and will) be used by the general public, as well as the agricultural machines.

Each field is covered by sensors detecting the moisture levels of the ground. The farm employs several different autonomous robots: On the one hand, we have *worker robots*, which are used both for maintenance, that is to repair other robots, as well as for plant care, that is to cut, water, and fertilise the plants. On the other hand, we have *transportation robots*, which harvest, collect the harvested plants and transport them to delivery stations. Robots of each category can be used for all of the tasks within the category. For example, any worker robot can water plants, or be used to cut the plants. For worker robots, the farm uses both flying robots, as well as robots driving on the ground, while all transportation robots are ground-based.

We assume that there is *no central controlling element*, and that the robots do not have the full knowledge about everything in the environment.

However, the farm still employs humans who maintain the machines, and who may take over some of the responsibilities (e.g., harvesting fields or cutting plants). Hence, the robots need to take the *presence of humans* into account, and need to adapt their behaviour accordingly. In particular, this means it is always possible that *manually operated machines* (for maintenance, plant care, harvesting or transportation, or simply other traffic) may be present in the farm and/or on the roads, as well as humans outside of any vehicles.

Goals

- Ensure safety of all entities involved, in particular working personnel and general public using connecting streets
  - Low-level safety: obstacle avoidance, collision avoidance
  - High-level safety: exclusive access to working areas
  - Avoidance of other hazards
- Optimise yield of farm and reduce potential losses during fertilisation, watering harvest and transportation

Table 2 Information on the actors in the smart farm.

Entity	Purpose	Number	Information Type
Field	Grow vegetables (salad, potatoes, turnips) or grains (wheat, rye)	4	Global
Green House	Grow vegetables or fruit (bell peppers, tomatoes, cucumbers, peaches)	2	Global
Worker Robot (Flying/Ground)	Plant crops, water and fertilise fields, repair other robots	3/2	Local
Harvester/ Transporter	Harvest plants and transport goods between farms/greenhouses and delivery station	3	Local

4.1.3 A cutout of typical activities in the smart farm

Example use case

1. Field X is *empty*
2. Robot A drives to X and plants potatoes
3. Robot B waters X
4. Robot B applies fertiliser to X
5. Field X is now in state *growing*, while steps 3 and 4 may be repeated
6. When field X (or rather the sensors on field X) sends message that plants are ripe (state *harvest*): Robot C comes to harvest potatoes
7. Robot C delivers the potatoes to the farm collection point

Example of an emergency scenario

1. Robot D detects utility vehicle on its path
2. Robot D avoids crash by replanning path

Further example of an emergency scenario

1. Robot E crashes into road/field-side ditch and gets immobile or collides with an object and gets damaged
2. Maintenance service is notified
3. Unoccupied worker takes care or issue will be delegated to supervisory control

4.1.4 Actors in the smart farm

Table 2 contains all different actor types of the smart farm. The first and second column contains the name and purpose of each category of actors, while the third column contains the number of single entities in each category. The final column denotes, whether the information about entities in this category is available to all other entities (global information), or only within each single entity (local information).

4.1.5 Layout of the smart farm

The fields and green houses are all connected to the collecting point on public streets. However, while the green houses can be approached separately and independent from each other, the fields share a common road for the approach. That is, the layout can be imagined as in Fig. 2: C denotes the collecting point, the rectangles marked by G are green houses and the rectangles marked with F are the fields. The lines in between indicate the road structure.

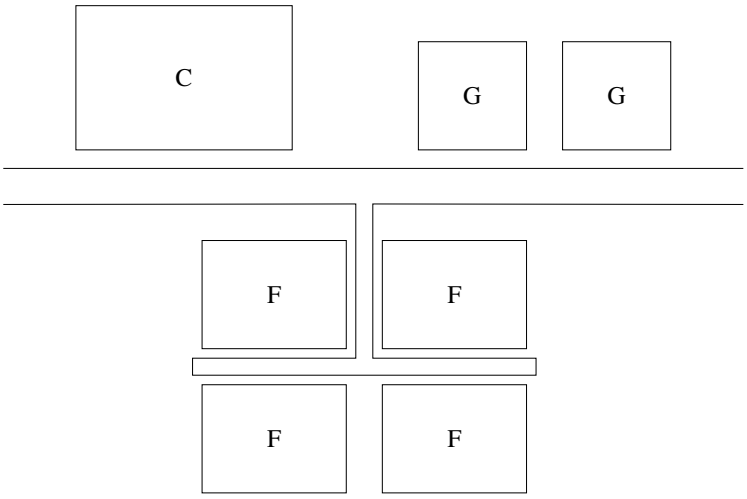

Figure 2 Layout of the smart farm.

4.1.6 Modelling Parameters according to Abstraction Level

For a more structured discussion, we distinguish several levels of details for this system, the environment and the hazards, which refer to the level of detail for the physicality of the system. The different levels are

1. Discrete
2. Real-Time
3. Physical

The first level contains the purely discrete aspects of the system components. That is, communication channels, structure and data, as well as possible (discrete) states of each autonomous entity. The second level incorporates real-time aspects of the behaviour, for example durations and time bounds. The third and final level includes more physical laws, for example in the form of differential equations. All of these models may include probabilistic aspects, or, in the case of real-time and physical models, limits on how exact durations and time bounds can be satisfied.

Generally, we assume that suitable sensors provide information about the different entities, and that this information may be shared via suitable channels (message passing, ...) For simplicity, we assume that this information is always correct, if not stated otherwise.

In order to focus and integrate the modelling approaches during the seminar, we strongly encourage you to use the following modelling parameters that are supposed to represent the variables of the Smart Farm state space. However, if you need to change these parameters, please be transparent about this in your model and its presentation.

Parameters and Parameter Types for Discrete Modelling:

- Map (areas/road segments):
  - state: *occupied, empty*

  You can assume that there is an attributed map available (to all vehicles) with geometry data (precision .5 meters). Depending on the activity and on a per-vehicle basis, SLAM[1] might be used to update volatile attributes of the area in the mapping information (local

[1] Simultaneous localisation and mapping

to a vehicle). Markers with high precision (+/-10cm) at convenient but practical places of the map can also be used for mapping and positioning.

- Resource (Field/Greenhouse):
  - contents: *peppers, salad, turnips, potatoes, wheat, rye, empty*
  - state: *harvest, growing, empty*
  - water level: *low, good*
  - fertiliser level: *low, good*
  - Invariants: $state\ empty \implies (contents\ empty \wedge water\ level\ good \wedge fertiliser\ level\ good)$
- Worker Robots:
  - cargo_type: *water, fertiliser*
  - movement: *ground, flying*
  - cargo: *full, empty*
  - or alternatively cargo: *(finite) set of values in [0,1], where 0 means empty, 1 means full*
- Harvester/Transporter:
  - state: *harvesting, transport to drop off*
  - cargo: *full, empty*
  - alternatively cargo: *(finite) set of values in [0,1], where 0 means empty, 1 means full*

Parameters and Parameter Types for (Distributed) Real-Time Modelling:

- Resource
  - The only real-time aspects for the resources would be the duration plants need to grow. However, since the time-scale of these durations is very different from communication and other aspects, we refrain from any further specification of this aspect.
- Communication
  - message delay: del_t seconds from sending to reception
  - localisation messages may have different delays:
    - global positioning (GPS, precision $\pm 2m$): gps_t
    - local positioning (with respect to finite set of fixed markers, precision $\pm .01m$): loc_t
- Worker Robots
  - filling up the cargo bay from empty to full: $care\_fill\_t$
- Harvester/Transporter
  - filling up the cargo bay from empty to full: $trans\_fill\_t$
- Relations between parameters: $del\_t < care\_fill\_t < trans\_fill\_t$

Parameters and Parameter Types for Continuous/Physical Modelling:

- Vehicles
  - speed
  - position
  - maxaccel
  - maxdecel
  - maxspeed: 30 kph
  - $1m \leq length \leq 5m$
  - $300kg \leq weight \leq 5000kg$
- Human traffic on public streets (bicycles and cars)
  - $15kph < speed < 60kph$
- Human traffic on farm streets (bicycles and pedestrians)
  - $3kph < speed < 20kph$

Failure probabilities

- Resources
  - Rotting goods: *.02/h*
- Ground Based Vehicle
  - Failure rate: .05/h
- Flying Vehicle
  - Failure rate: .1/h
- Message loss
  - *p_m_loss*
- Probabilities of humans (on bicycles or in cars) on public streets
  - *p_h_public*
- Probabilities of humans (pedestrians, or on bicycles) on farm streets:
  - *p_h_farm*

Feel free to refine these uncertainties (e.g. probability of vehicles on roads) by introducing further parameters.

4.1.7 Properties

The following properties of the Smart Farm control scheme refine the goal of the seminar.

Safety Constraints (depending on environment: Public street, street between fields, ...)

- Public road
  - avoids vehicles from general public (cars driving through, bicyclists, pedestrians)
  - avoids colliding with other utility vehicles
- Rural road
  - avoids working personnel (trained, but may still make errors)
  - avoids colliding with other utility vehicles
- Fields
  - avoids colliding with other utility vehicles

These constraints should depict the variety of collision situations to be encountered in the Smart Farm. It is of course possible to cover these constraints with a generalised constraint of the form: "Avoid collision with any moving vehicle or person or any static object in the Smart Farm."

Productivity Requirements (Liveness, Progress)

- Resources
  - Harvest ripe goods timely (alternatively: plants shall not rot on the fields/in the green houses)

Participants

- Étienne André
University of Paris North, FR
- Andreas Bauer
KUKA Systems GmbH – Augsburg, DE
- Sergiy Bogomolov
Newcastle University, GB
- Jörg Brauer
Verified Systems International GmbH – Bremen, DE
- Radu Calinescu
University of York, GB
- Alessandro Fantechi
University of Florence, IT
- Frederik Foldager
Aarhus University, DK
- Sibylle Fröschle
OFFIS – Oldenburg, DE
- Mario Gleirscher
University of York, GB
- Anne E. Haxthausen
Technical University of Denmark – Lyngby, DK
- Christian Heinzemann
Robert Bosch GmbH – Stuttgart, DE
- Jean-Baptiste Jeannin
University of Michigan – Ann Arbor, US
- Kim Guldstrand Larsen
Aalborg University, DK
- Peter Gorm Larsen
Aarhus University, DK
- Martin Leucker
Universität Lübeck, DE
- Sven Linker
University of Liverpool, GB
- Stefan Mitsch
Carnegie Mellon University – Pittsburgh, US
- Laura Nenzi
University of Trieste, IT
- Peter Csaba ölveczky
University of Oslo, NO
- David Parker
University of Birmingham, GB
- Pedro Ribeiro
University of York, GB
- Masaki Waga
National Institute of Informatics – Tokyo, JP