Report from Dagstuhl Seminar 19112

Engineering Reliable Multiagent Systems

Edited by

Jürgen Dix[1], Brian Logan[2], and Michael Winikoff*[3]

1 **Technische Universität Clausthal, DE,** dix@tu-clausthal.de
2 **University of Nottingham, GB,** brian.logan@nottingham.ac.uk
3 **Victoria University of Wellington, NZ,** michael.winikoff@vuw.ac.nz

Abstract

This report documents the program and outcomes of Dagstuhl Seminar 19112 "Engineering Reliable Multiagent Systems". The aim of this seminar was to bring together researchers from various scientific disciplines, such as software engineering of autonomous systems, software verification, and relevant subareas of AI, such as ethics and machine learning, to discuss the emerging topic of the *reliability* of (multi-)agent systems and autonomous systems in particular. The ultimate aim of the seminar was to establish a new research agenda for engineering reliable autonomous systems.

Seminar March 10–15, 2019 – http://www.dagstuhl.de/19112

2012 ACM Subject Classification Computing methodologies → Multi-agent systems, General and reference → Reliability

Keywords and phrases agent-oriented programming, multi agent systems, reliability, software and verification methodologies

Digital Object Identifier 10.4230/DagRep.9.3.52

Edited in cooperation with Tobias Ahlbrecht

1 Executive Summary

Jürgen Dix (Technische Universität Clausthal, DE, dix@tu-clausthal.de)
Brian Logan (University of Nottingham, GB, brian.logan@nottingham.ac.uk)
Michael Winikoff (Victoria University of Wellington, NZ, michael.winikoff@vuw.ac.nz)

The multi-disciplinary workshop on Reliable Multiagent Systems attracted 26 leading international scholars from different research fields, incuding theoretical computer science, engineering multiagent systems, machine learning and ethics in artificial intelligence.

This seminar can be seen as a first step to establish a new research agenda for engineering reliable autonomous systems: clarifying the problem, its properties, and their implications for solutions.

In order to move towards a true cross-community research agenda for addressing the overarching challenge of engineering reliable autonomous systems we have chosen a slightly different organization than usual: the seminar was comprised of (short) talks (days 1 and 2), and extensive discussions and dedicated group work (days 3-5).

* Michael was still at University of Otago, NZ, when he prepared and attended the seminar.

The first two days were opened by two longer (45 minutes each) tutorials, followed by short "teaser talks" ($10 + 5$ minutes) related to the main topic of *reliable multiagent systems*. Almost all participants gave their view of the topic and highlighted possible contributions. The talks were meant to be less "conference-style", and more inspiring and thought-provoking.

At the end of the second day, we established four working groups to dive deeper into the following questions:

1. What **(detailed) process** can be used to identify properties that a particular reliable autonomous system or MAS needs to satisfy?
2. How can we engineer reliable autonomous systems that include **learning**?
3. How can we engineer reliable autonomous systems that include **human-machine interaction** (including human-software teamwork)?
4. How can we engineer reliable autonomous systems comprising **multiple agents** (considering teamwork, collaboration, competitiveness, swarms, ...)?

The groups met on Wednesday and Thursday for extensive discussions and reported back intermediate results in plenary sessions. Participants were encouraged to move between groups to enrich them with their expertise. The seminar concluded on Friday morning with a general discussion where all groups presented their results.

We summarise below the key results from the four discussion groups.

Identifying properties: This group discussed the challenge of identifying requirement properties to be verified. It focused in particular on autonomous systems that replace humans in domains that are subject to regulation, since these are most likely to require and benefit from formal verification.

The group articulated the following arguments:

- That autonomous systems be viewed in terms of three layers: a continuous control layer at the bottom, a "regulatory" layer in the middle, and an "ethical" layer at the top. The distinction between the regulatory and ethical layers are that the former deals with the expected normal behaviour (e.g. following the standard rules of the domain), whereas the latter deals with reasoning in situations where the rules need to be broken. For example, breaking a road rule given appropriate justification.
- That for these sorts of systems we can derive verification properties by considering the licencing that is used for humans and how human skills and capabilities are assessed, as well as relevant human capabilities that are assumed, and relevant laws and regulations. The group sketched a high-level process for identifying requirement properties by considering these factors.

The group considered a range of domains, for each one showing how these points would apply.

These ideas were developed into a draft paper during the workshop, and work on this paper has continued subsequently.

Learning in reliable autonomous system: The second group was concerned with methods for engineering reliable autonomous systems that involve learning.

The notion of sufficient reliability varies from domain to domain. For example, in planning of telecommunication networks there are simulators that are trusted to be a good model of reality. Hence the simulation rules could be used for formal verification. In other domains, such as autonomous driving, there is no established trusted model of reality.

Assuming a formal model exists and safety properties can be formulated with temporal logic, there are still remaining challenges: complex models with a large state space and hybrid continuous and discrete behavior can make formal verification intractable, especially when the learned policies are equally complex. On the other hand learning methods (e.g. reinforcement learning) often "discover" key strategies that do not depend on all details of the system. The group discussed ideas for abstracting/discretizing transition systems based on learned policies.

Human-machine interaction in reliable autonomous systems: The third group focused on how to engineer reliable human-agent interaction.
For that, the first step was to carve out what it means for human-machine communication to be reliable. Values and norms are definitely involved. Drawing from human communication, being truthful, up-to-date with relevant knowledge and honouring commitments are major parts. Another important building block is transparency: is it always clear, which values are in play, what the agent's purpose is, or what happens with the collected data? The desired result would be reliable outcomes, e.g. through reliably following a protocol, effective communication and getting to a shared understanding. A number of tools and methods to achieve this were identified: stakeholder analysis, plan patterns/activity diagrams, interaction design patterns, appropriate human training, and explainability (i.e. explainable AI) were among the most prominent engineering solutions. This group also concluded their discussions early and distributed themselves among the other groups after that.

Multiple agents in reliable autonomous systems: The fourth group focussed on the challenges of ensuring reliability in systems comprising multiple, possibly heterogeneous, agents interacting in complex ways.
A number of issues emerged from the discussion, including what does it mean for a multiagent system to be "collectively reliable", and what is the relationship between the reliability or otherwise of individual agents and the coordination mechanisms through which they interact, and the collective reliability of the system of a whole. These issues were broken down into more specific engineering challenges, including which languages should be used to express collective reliability properties (which is closely related to the discussion of the first group) and how such properties should be verified, how to engineer reliable coordination mechanisms when we have only partial access to the states of agents, how to decompose and/or distribute the monitoring and control of individual agents (and associated failure recovery) necessary for reliable coordination, how to do all of the above in systems where agents learn (closely related to the discussion of the second group), and, finally, how to allocate responsibility to individual agents when behaviour is not reliable.

A more detailed research agenda for engineering reliable multiagent systems is in preparation, which we plan to publish as a "position paper" in a journal special issue arising from the work at the workshop.

2 Table of Contents

3 Overview of Talks

3.1 Exploring properties of reliable multi-agent systems through simulation

Tobias Ahlbrecht (TU Clausthal, DE)

One specific question for this seminar is aimed at methods to identify properties that a particular reliabale multi-agent system should satisfy. We argue, that simulation can be used as a tool for requirements engineering, especially when there is no clear position on where or how to start eliciting these properties. Simulatons are then well suited, since they are rather tangible, allow to focus on key elements, while abstracting the more intractable parts of the environment, and can be used early in the process to explore the problem space. Then, mechanisms for explaining agent behaviour could be used (a) to identify the causes of problematic behaviour in the simulation (i.e. where unknown properties still have to be satisfied) and (b) to check whether acceptable simulation runs can be ascribed to previous changes and not only chance.

3.2 Challenges in Self-Driving Cars

Stefano Albrecht (University of Edinburgh, GB)

Joint work of FiveAI
URL https://five.ai

The coming decades will see the creation of autonomous vehicles (AVs) capable of driving without human intervention. Among the expected benefits of AVs are a significant reduction in traffic incidents and congestion while improving cost-efficiency. The UK aims to be a leader in the AV industry and is investing significantly in this area. The Innovate UK-funded project "StreetWise", led by UK-based company FiveAI, is the largest project to date and aims to demonstrate safe autonomous driving in London by 2020. Supported by a Royal Society Industry Fellowship, I work with FiveAI to develop artificial intelligence technologies for safe autonomous driving. I will present some of the research challenges along the road to AVs: How can an AV predict the behaviour of other vehicles in an uncertain and dynamically changing environment? And how can the AV make robust decisions and drive safely in such environments?

3.3 High level specifications and some associated problems

Natasha Alechina (University of Nottingham, GB)

This talk attempts to summarise some common themes in the preceding talks, and challenges related to them.

In order to engineer reliable multi-agent systems, we need to be able to either formally verify them, or synthesise provably correct systems. However given the systems are very complex and there are very many properties that they should satisfy, exhaustively enumerating and verifying the required properties seems impossible. A possible way forward would be specifying properties on a very high abstract level and coming up with some way of decomposing them (a theme from Michael Winikoff's talk, and also from Michael Fisher's talk, cf everyone agrees the system should be "safe", but what exactly is involved in being "safe"?). Examples of high level properties are: high-level descriptions of actions the agent should perform ("make furniture") or the abstract properties such as "do this safely".

With high level actions, automatic decomposition into concrete steps is possible; in fact this is what HTN (Hierarchical Task Networks) planning does.

With high level properties, this is harder. To do this automatically requires a domain theory which supports deriving definitions of high level properties in terms of low level ones. This may be possible if all low level properties are specified in terms of actions; then it is possible to derive a disjunction of low-level properties/actions as in Michael Winikoff's talk, e.g. "do no harm = (not)action 1 or (not)action2 ... or not(action_n)" where actions with "not" in front of them have "harm" in their effects, and actions without "not" prevent harm somehow. The situation is harder if there is no domain theory, no action theory, or action theory is undecidable (like movement in 3D space). Then the possible way forward may be to produce an approximation/abstraction of the corresponding transition system (discrete actions and discrete time, abstracted state descriptions, state test for "unsafe"); generate states as needed; use run-time monitoring with bounded lookahead (a bit like in Stefano Albrecht's talk).

3.4 Engineered Adaptivity through Clonal Plasticity

Rem Collier (University College Dublin, IE)

This talk explores the use of an adaptation strategy, known as clonal plasticity, to inject run-time adaptation mechanisms into Agent Oriented Programming languages. The strategy is formed through the combination of two processes present in plants: Clonal Reproduction and Phenotypic Plasticity. The former process relates to the fact that daughter plants are genetically identical to parent plants, hence the term clone. Conversely, the latter process refers to the ability of an organism to vary its structure (height of stem, width of leaves, etc.) or behaviour (connect to its clonal sibling or disconnect) according to environmental influences. The resulting strategy is distinuished from other evolutionary mechanisms, such as genetic algorithms, in that the genome of the organism is not modified at all.

In the talk, we explore three ways in which Clonal Plasticity could be integrated with the AgentSpeak(L) language: parameter-based adaptation, statement-based adaptation and rule-based adaptation. The former technique refers to an approach in which adaptation arises through the modification of various well-defined parameters within the program. Statement-based adaptation refers to an approach where adaptation operators are embedded within the core planning language. This allows for the speciifcation of variant plans. Adaptation adaptation arises through the selection of a specific variant of that plan. Finally, rule-based adaptation refers to the use of multiple rules to capture variant plans that could be used to achieve a given goal. Rule order, which is commonly used to select which of a set of applicable plans should be selected is then adapted, enabling rules that were previously unreachable to be selected.

3.5 What can go wrong when running AgentSpeak?

Niklas Fiekas (TU Clausthal, DE)

Classical programming languages come with tools to catch programming mistakes as early as possible, at compile time, and strongly typed versions of popular languages keep appearing (e.g., TypeScript, typing for Python).

When designing agent programming languages, we should consider static analysis early on.

In the talk we look at common mistakes in Jason-style AgentSpeak programs, if we can catch them at compile time, or which design elements of the Jason programming language make this difficult.

3.6 Verification of Autonomous Systems Software

Michael Fisher (University of Liverpool, GB)

Joint work of Louise Dennis, Michael Fisher

An overview of issues in the verification of software involved in autonomous systems.

References

1 L. A. Dennis, M. Fisher, N. K. Lincoln, A. Lisitsa, and S. M. Vefres. Practical Verification of Decision-Making in Agent-Based Autonomous Systems. *Automated Software Engineering* 23(3):305–359, 2016.
2 M. Fisher, L. A. Dennis, and M. Webster, Verifying Autonomous Systems, *CACM*, 56(9):84–93, 2013.

3.7 Reliable Supportive Agents – What do Agents need to know about the user?

Malte S. Kließ (TU Delft, NL)

An agent system intended to provide support for a user's daily routines should be informed about the user's needs in order to be considered reliable – in the sense that the agent delivers support in the right way at the right time.

This implies that the user needs to trust the system to act according to the user's norms and values. Thus the system needs the capability of handling and reasoning about norms and values.

References

1 Ajmeri, N., Guo, H., Murukannaiah, P. K., and Singh, M. P. *Robust norm emergence by revealing and reasoning about context: Socially intelligent agents for enhancing privacy.* In *International Joint Conference on Artificial Intelligence*, 2018.

2 Cranefield, S., Winikoff, M., Dignum, V., and Dignum, F. *No pizza for you: Value-based plan selection in BDI agents.* In *International Joint Conference on Artificial Intelligence*, 2017.

3 Kayal, A., Brinkman, W.-P., Gouman, R., Neerincx, M. A., and van Riemsdijk, M. B. *A value-centric model to ground norms and requirements for epartners of children.*, In *Coordination, Organizations, Institutions, and Norms in Agent Systems*, 2014.

4 Kließ, M. S., and van Riemsdijk, M. B. *Requirements for a temporal logic of daily activities for supportive technology.* In *Proceedings of the First Workshop on Computational Accountability and Responsibility in Multiagent Systems co-located with 20th International Conference on Principles and Practice of Multi-Agent Systems*, CARe-MAS@PRIMA 2017, Nice, France, October 31st, 2017.

5 Tielman, M., Jonker, C., and van Riemsdijk, M. *What should I do? Deriving norms from actions, values and context.* In *International Workshop Modelling and Reasoning in Context (MRC), Held at FAIM*, 2018. Under revision at the AAMAS/IJCAI Workshop on Modeling and Reasoning in Context.

6 van Riemsdijk, M. B., Jonker, C. M., and Lesser, V. *Creating socially adaptive electronic partners.* In *International Conference on Autonomous Agents and Multiagent Systems*, 2015.

3.8 Synthesising Reliable Multi-Agent Programs

Brian Logan (University of Nottingham, GB)

Joint work of Giuseppe De Giacomo, Paolo Felli, Fabio Patrizi, Sebastian Sardina

I present some work in progress on the synthesis of controllers that implement the abstract actions in a high-level multi-agent program in terms of low-level programs performable by individual agents. In contrast to previous work, we consider this problem in a first-order setting, allowing us to synthesise controllers for data-aware multi-agent systems and programs. I sketch a controller synthesis framework based on Situation Calculus action theories and ConGolog programs in which states have a first-order representation, and briefly describe techniques for the synthesis of controllers for the special case of bounded action theories.

3.9 What can we prove about Ethical Reasoning Systems?

Louise Dennis

Joint work of Louise Dennis, Michael Fisher, Paul Bremner, Alan Winfield, Marija Slavkovik, Martin Bentzen, Felix Lindner, Matt Webster

Prompted by a question in the seminar proposal I discussed work on the verification using model-checking of ethical reasoning and specifically the properties that were validated and tentatively assigned these to a taxonomy. Some properties sought to check the implementation of the relevant ethical theory and generally involved checking the property "the least worst outcome according to the theory was selected" on highly abstract models. Models then became more concrete and also more specific, in some cases the property verified was still of the form "least worst..." but in other cases specific desired outcomes were checked (e.g., "in the event of a fire, evacuation will be performed") and served as a kind of 'sanity check' that the encoding of the ethics aligned with user values.

References

1 Bremner, P., Dennis, L. A., Fisher, M. and Winfield, A. (2019) On proactive, transparent and verifiable ethical reasoning for robots. *Proceedings of the IEEE*, 107 (3). pp. 541–561.
2 L. A. Dennis, M. Fisher, and A. F. T. Winfield. Towards Verifiably Ethical Robot Behaviour. In *Proc. AAAI Workshop on AI and Ethics*, 2015.
3 L. Dennis, M. Fisher, M. Slavkovik, and M. Webster. 2016. Formal Verification of Ethical Choices in Autonomous Systems. *Robotics and Autonomous Systems* 77 (2016), 1–14
4 L. A. Dennis, M. M. Bentzen, F. Lindner and M. Fisher. Verifiable Machine Ethics in Changing Contexts. Submitted to *Artificial Intelligence*.

3.10 Threat-oriented engineering of autonomous systems

Viviana Mascardi (University of Genova, IT)

A system is reliable if it can be relied on to meet its requirements consistently. Nevertheless, a system may be considered reliable even in situations where it does not meet its requirements, provided there is a convincing/acceptable explanation of why the requirement was not met. The explanation might be related with the trade-off of achieving some different requirements, including the "always present" requirement of keeping users (and people in general) safe. The explanation could be better understood if the notion of threat to a goal was explicitly associated with the goal. The idea of making "threats-to-goal" explicit in the engineering process in discussed in this presentation, as well as its connections with explainability.

3.11 Programming Reliable Agent-Based Systems in the Software 2.0 Era (?)

Alessandro Ricci (Università di Bologna, IT)

Recent advances and results in AI are paving the way to a "Software 2.0 era", where a software system or a part of it is not hand-coded by programmers but learnt by exploiting e.g. machine learning techniques. In the case of BDI agent-oriented programming, this accounts for introducing learning in the loop of agent development, so that e.g. a part of the plans of an agent are programmed and a part is learnt by the agent itself, through e.g. reinforcement learning techniques. In that perspective, a main question interesting for the Dagstuhl seminar is about reliability, how to engineer reliable agents and MAS that integrate machine learning at that level.

3.12 RoboCup Rescue Simulation and Reliable Multi-Agent Systems

Holger Schlingloff (HU Berlin, DE)

In this talk, we propose the RoboCup Junior Rescue Simulation (formerly called CoSpace) Challenge as a case study for the design of reliable multi-agent systems. In the game, two simulated robots compete in collecting objects and depositing them in a dedicated safety zone. Thus, the setting is related to real scenarios as in cleaning robots or driverless automated transport robots in industrial production. In the competition, which is targeted towards 12-16 year old high school students, the participants have to design rules for the robots which control their moves. Depending on the available information (map, location of objects, own position, position of the opponent) different variants of the game can be defined. Thus, it is a well-suited example for the engineering of reliable MAS: The focus is on logical, not mechanical challenges; it is simple enough to be approached by current technologies; it can be adapted to different software engineering aspects; and it is of industrial relevance. Specifically, we are asking for techniques to verify the control program with respect to the objectives and rules of the game. An even more challenging question is asking for synthesis techniques to generate the control program automatically from the given rules and objectives.

References

1 Robocup Junior Rescue Simulation (former CoSpace). https://junior.robocup.org/rescue-simulation-league-former-cospace-league/
2 CoSpace Robotics. http://cospacerobot.org/

3.13 Multi-reward Learning for Reliable Long-term Autonomy

Kagan Tumer (Oregon State University, US)

Many real world problems are too complex to allow the agent-environment interaction to be captured by a well-formulated, single reward function. Using multiple rewards to learn different aspects of the problem, potentially across different time scales, offers a promising solution. But this approach introduces a new problem: how does an agent determine which reward is relevant in which state and at what time? Put another way, how does an agent know "what matters when?"

This talk addresses how to navigate multiple rewards to enable learning in complex, long-term tasks. In addition, this paradigm naturally extends to multiagent settings and enables the injection of concepts such a reliability into the system.

3.14 Towards Deriving Verification Properties

Michael Winikoff (Victoria University of Wellington, NZ)

Main reference Michael Winikoff: "Towards Deriving Verification Properties", CoRR, Vol. abs/1903.04159, 2019.
URL https://arxiv.org/abs/1903.04159

Formal software verification uses mathematical techniques to establish that software has certain properties. For example, that the behaviour of a software system satisfies certain logically-specified properties. Formal methods have a long history, but a recurring assumption is that the properties to be verified are known, or provided as part of the requirements elicitation process. This working note considers the question: where do the verification properties come from? It proposes a process for systematically identifying verification properties.

3.15 Reliable Alignment of Goals and Norms

Neil Yorke-Smith (TU Delft, NL)

Joint work of Pankaj R. Telang, Munindar P. Singh, Neil Yorke-Smith
Main reference Pankaj R. Telang, Munindar P. Singh, Neil Yorke-Smith: "A Coupled Operational Semantics for Goals and Commitments", J. Artif. Intell. Res., Vol. 65, pp. 31–85, 2019.
URL https://doi.org/10.1613/jair.1.11494

Commitments capture how an agent relates to another agent, whereas goals describe states of the world that an agent is motivated to bring about. Commitments are elements of the social state of a set of agents whereas goals are elements of the private states of individual agents. It makes intuitive sense that goals and commitments are understood as being complementary to each other. More importantly, an agent's goals and commitments ought to be coherent, in the sense that an agent's goals would lead it to adopt or modify relevant commitments and an agent's commitments would lead it to adopt or modify relevant goals. However, despite the intuitive naturalness of the above connections, they have not been adequately studied in a formal framework. This article provides a combined operational semantics for goals and commitments by relating their respective life cycles as a basis for how these concepts (1) cohere for an individual agent and (2) engender cooperation among agents. Our semantics yields important desirable properties of convergence of the configurations of cooperating agents, thereby delineating some theoretically well-founded yet practical modes of cooperation in a multiagent system.

Participants

- Tobias Ahlbrecht
TU Clausthal, DE
- Stefano Albrecht
University of Edinburgh, GB
- Natasha Alechina
University of Nottingham, GB
- Rem Collier
University College Dublin, IE
- Mehdi Dastani
Utrecht University, NL
- Louise A. Dennis
University of Liverpool, GB
- Frank Dignum
Utrecht University, NL
- Virginia Dignum
University of Umeå, SE
- Jürgen Dix
TU Clausthal, DE
- Niklas Fiekas
TU Clausthal, DE
- Michael Fisher
University of Liverpool, GB
- Koen V. Hindriks
University of Amsterdam, NL
- Alexander Birch Jensen
Technical University of Denmark – Lyngby, DK
- Malte S. Kließ
TU Delft, NL
- Yves Lesperance
York University – Toronto, CA
- Brian Logan
University of Nottingham, GB
- Viviana Mascardi
University of Genova, IT
- Ann Nowé
Free University of Brussels, BE
- Alessandro Ricci
Università di Bologna, IT
- Kristin Yvonne Rozier
Iowa State University, US
- Holger Schlingloff
HU Berlin, DE
- Marija Slavkovik
University of Bergen, NO
- Kagan Tumer
Oregon State University, US
- Michael Winikoff
University of Otago, NZ
- Neil Yorke-Smith
TU Delft, NL