Journal of Reliable Intelligent Environments (2018) 4:211–224
https://doi.org/10.1007/s40860-018-0070-5

ORIGINAL ARTICLE

# A survey on verification strategies for intelligent transportation systems

Hedda R. Schmidtke[1]

Received: 16 April 2018 / Accepted: 10 October 2018 / Published online: 19 October 2018

**Abstract**

As intelligent systems are increasingly entering everyday life, in domains such as transportation, resource distribution, health care, or retail, developing suitable verification mechanisms for such systems becomes vital. From a formal point of view, the employed intelligent sensor actuator systems (ISAS) constituting such intelligent systems combine three different technologies: control systems, distributed systems, and learning and reasoning. While each of the parent domains features tested and proven verification methods, simply combining the tasks unfortunately leads to a combinatorial explosion of complexity. This paper presents an overview and classification of currently employed techniques for handling ISAS in terms of: cyber-physical systems, intelligent autonomous robots, or intelligent agents. The article argues that each of the three classical perspectives misses one important characteristic of ISAS and proposes to combine the three for a full solution. The paper argues that in particular two mechanisms are promising: an intelligent environments perspective that verifies local safety and techniques for context-aware monitoring that allow a mobile system to leverage context-awareness to reduce complexity for self-monitoring tasks.

**Keywords** Autonomous vehicles · Intelligent transportation · Verification · Machine learning · Context

## 1 Introduction

The last decade for the first time had hardware widely available that was sufficient for running Artificial Intelligence (AI) algorithms in end-user applications. Moreover, the proliferation of networked computing technologies equipped with sensors throughout all segments of the population, from children to retirees, in the form of smart phones, has led for the first time to large amounts of freely available and representative data repositories sufficient to train machine learning classifiers applicable to a majority of the whole population.

A wealth of novel technologies has resulted from this situation with several common features: systems are equipped with sensors and network access and provide their services to users dependent on what is appropriate in a context. Experimentally but also increasingly with commercial prototypes, smart environment technologies enter safety-critical domains traditionally reserved to systems tested with the closest scrutiny, such as the electricity grid (smart grid), the transportation system (autonomous vehicles and smart city infrastructures), and even the health care system (smart hospitals and ambient assisted living). The majority of earlier successful applications involving machine learning technologies, such as spam filters, location-aware recommender systems, and even factory robots were not safety critical, at least not on a geographic scale. With increasing proliferation of the new technologies, however, consumers and citizens are increasingly willing to trust such intelligent sensor–actuator systems (ISAS), even with their life, as in the case of autonomous vehicles or smart hospitals. ISAS manufacturers and associated stakeholders, such as the automobile industry, cities, or the insurance industry, are struggling with the difficult task to navigate between consumer/citizen convenience and trust, on the one hand, and questions of predictability and consequently responsibility and liability, on the other hand. From the perspective of safety, ISAS pose considerable opportunities, e.g., to make driving safer by detecting and reacting to driver fatigue, or to reduce energy consumption, but challenges are major: ranging from concrete practical questions regarding the distribution of liability between manufacturers and consumers/citizens to fundamental questions regarding the responsibility for AI systems.

✉ Hedda R. Schmidtke
schmidtke@acm.org

[1] University of Oregon, Eugene, OR 97403-1251, USA

A key role in this discussion is played by the question whether complex robotic AI systems operating within the complexity of everyday life such as autonomous vehicles are verifiable at all. This paper surveys the different possible strategies to approach the verification of ISAS to resolve this question. We show that none of the conventional strategies are applicable per se, as each is missing one different crucial component, and that a combination of techniques is needed that can carefully avoid scalability issues. We argue that a promising approach is to combine a perspective of verification of intelligent environments with a novel type of self-monitoring context-aware mobile AI systems.

**Structure of the Article** We present a classification of different approaches to ISAS verification (Sect. 2), in general, and where possible make reference to autonomous transport, in particular. Autonomous vehicles are of particular importance, given their power to cause fatalities. We outline that each type of approach misses a different aspect of the complex problem. In Sect. 3, we discuss the verification of algorithms generated by machine learning and reasoning. We finally show how the high complexity problem of general verification of autonomous vehicles can be broken down spatiotemporally, so as to become a problem solvable in a piecewise manner (Sect. 4). We discuss the result and its ethical ramifications in Sect. 5. We argue that a key to a successful transition into the age of ISAS is to create more human-like intelligent systems while at the same time avoiding anthropomorphizing AI systems. In conclusion (Sect. 6), we argue that verification of ISAS is a solvable problem if it is not seen purely as a computer engineering problem but also as a civil engineering problem that can only be solved with a novel type of trustworthy human-like AI systems.

## 2 Three perspectives on ISAS

From a formal point of view, system verification methods for ISAS need to address a complex combination of systems. ISAS combine three different technologies: control systems, distributed systems, and learning and reasoning (Fig. 1, 2). While each of the parent domains features tested and proven verification methods, the combined task unfortunately leads to a combinatorial explosion of possible situations, in particular, as classical engineering strategies, such as layering and modularization approaches, are not applicable to the wealth of possible situations such a system may face when interacting with the world.

Parts of the research task have been addressed by other recent subdomains. Three areas lie in the intersection of control systems, distributed systems, and learning and reasoning. Cyberphysical systems (CPS) combine control systems and distributed systems or networking: extensions of control systems theory with discrete state-based protocols, and program verification techniques have been proposed to verify the combination of distributed system protocols operating in the real world, e.g., in factories. However, CPS usually do not involve AI technologies, or verification of the AI technologies is not addressed. Intelligent agents combine reasoning and learning with interaction capabilities—in interactions with human users or other agents of the same or different type. Verification techniques focus on consistency of the knowledge base of the agent and game theoretic evaluation of its interaction strategies. However, these systems rarely comprise sensor–actuator facilities or embodiment. Pervasive computing systems combine the categories of intelligent agents with basic cyberphysical systems components, and thus extend both categories. Classical intelligent autonomous robots, as a third candidate category, combine control system components with reasoning or learning, but are seldom designed to interact with other robots. Experimental systems, such as the teams in the RoboCup competition [2], or autonomous vehicles therefore extend the classical area. Verification of such systems was classically not necessary, as they had exploratory rather than product character. Autonomous intelligent robots interacting with human beings, apart from autonomous vehicles, thus have not entered the market place, yet. Evaluation of such robots has been a focus mostly from a human–robot interaction point of view using methods of traditional human–computer interaction, in particular, user studies.

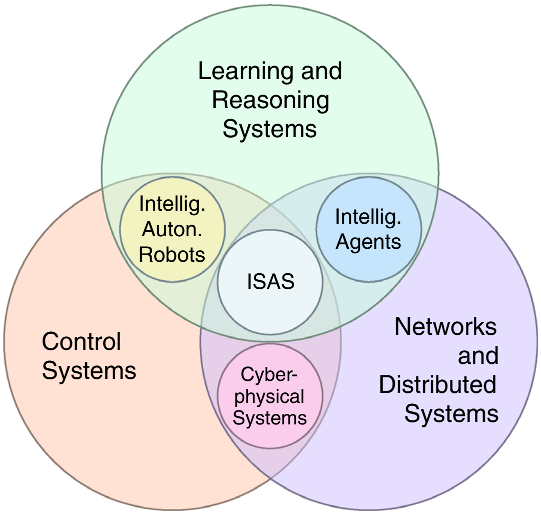

**Fig. 1** Intelligent sensor–actuator systems in comparison to related technologies

We discuss methods from the three perspectives in the following three subsections. While we reference some textbooks regarding the verification methods for the three fundamental technologies, control systems, distributed systems, and reasoning and learning, it is beyond the scope of this sur-

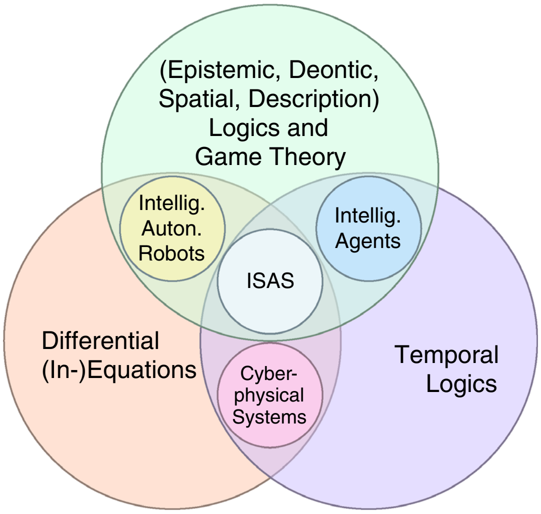
(Epistemic, Deontic, Spatial, Description) Logics and Game Theory
Intellig. Auton. Robots
Intellig. Agents
ISAS
Differential (In-)Equations
Cyber-physical Systems
Temporal Logics

**Fig. 2** Prevalent verification methods for ISAS and adjacent technologies

vey to provide an introduction or more detailed discussion of standard techniques. We highlight the fundamental differences between the three perspectives each approach takes, but focus on the ways proposed to interface between them. This necessarily leaves out interesting advances that have been proposed in each area. Mathematical and computational tools for solving or approximating differential equations, for instance, drive advances in classical control systems verification and allow increasingly complex systems to be modeled in this manner. Distributed systems verification is based on the logical analysis of a discrete process abstraction and provides a wealth of approaches applicable in different phases of the product development. Learning and reasoning have their foundations in millennia of research on logic and the theory of science, and are verified using tools of logics, game theory, and statistics.

The key reason for leaving out a more detailed discussion of the parent areas is that each makes considerable simplifications that are broken with a complex ISAS scenario, such as autonomous vehicle verification. A pure distributed systems approach, for instance, can verify fundamental program and network protocol constraints, but cannot handle verification of the actuator response or sensor properties, or verification of the machine learning behavior of an autonomous vehicle.

We here survey verification perspectives rather than specific approaches, and reference specific tools or systems only as examples, as there is a general problem in integrating the three different perspectives. Most approaches, for instance, neglect or simplify the system intelligence and/or the modeling of the continuous world, which poses, as we will try to motivate, both a problem and an opportunity for a solution.

In the discussion, we will take a birds-eye perspective trying to bring together different approaches rather than going deeper into more specific proposals, as there are categorical shortcomings having their roots in the underlying perspectives. To give concrete examples, employing the systems modeling language SysML, an extension of the Unified Modeling Language (UML) to embedded systems design, for instance, allows embedded system specifications to be analyzed and verified during the design process [41], but does not support verification of system intelligence. Coronato and Pietro [12], to give another example, such as many other proposals in Ambient Intelligence leverages a network process abstraction, such as Ambient Calculus [8], Bigraphs [64], or $\pi$-calculus [33], extended to handle space in the same manner as networks or folder hierarchies, extending the notion of concurrency to capture mobile devices. This simplifies the problem of handling continuous domains as well as the intelligent reasoning capabilities considerably. By focussing on the underlying formalisms—sporadically drawing upon specific proposals as examples, but sacrificing any attempt to do justice to the wealth of approaches—and the perspectives they entail, we can address generic properties in a more targeted way and with a neutral birds-eye perspective.

## 2.1 ISAS as CPS

The type of system that is maybe most similar to a system operating within a physical context is a control system or more specifically a cyber-physical system. In both ISAS and control systems, the difficult task is to evoke actuation in response to input from sensors. Cyber-physical systems, such as modern aircraft, factories, or vehicles, in contrast to classical control systems and similar to ISAS, contain more complex computational components that influence how the system invokes actuation depending on sensory input. Cyber-physical systems (CPS) are systems that combine a software (cyber) with a control (physical) component and often also a network component. They play an important role in today's advanced control systems with applications ranging from pace makers to power grids [42] and are indispensable as fundamental components of ISAS. The classical approach to verification of cyber-physical systems uses reachability analysis to compute over the (possibly infinite) set of all reachable states whether for any given situation an unsafe state is reachable. Modeling both the continuous, dynamic, physical side of the control system part and the discrete, state-based, programmatic side of the computational system part, such models describe systems in a hybrid manner in terms of both finite state machines and partial differential equations.

While control theory traditionally described and verified systems in terms of mathematical descriptions of their physical behavior [7], networks are usually verified by analyzing properties of the state space [36]. More complex networks, however, already suffer from an explosion of state space size [55], and advanced methods are required for reducing the number of states [29]. This problem increases with networked control systems [91], e.g., if networks connect sensing and actuation.

Embedded system software design takes a similar approach [23] focussing on verification of concurrency requirements using reachability analysis of the state space of an automaton or Petri net. On the programming side, computational complexity for this type of lightweight devices has to be minimal so as to be highly predictable. Embedded device software, thus often has constant runtime requirements so as to ensure real-time compliance in network response as well as actuation and sensing tasks [23].

Hybrid approaches were designed to provide better analyses [47] of cyber-physical system performance. A hybrid approach for the verification of intelligent environments taking a CPS perspective was proposed by del Mar Gallardo et al. [60]. A critical conceptual gap exists, however, between the continuous time of control systems and the discrete time assumed with state space models [1], which—when considering larger systems—requires dealing with different temporal and spatial scales [50,71]. The latter problem is still unsolved within the cyber-physical systems domain.

Hybrid approaches of cyber-physical systems design and description [6,47] thus require the combination of the traditional control system verification with a distributed systems verification in a two-layered architecture focussing on the interface between continuous and discrete. The focus is on the integration of the distributed system state as used in verifying a protocol into the control system. The model of the discrete system in the hybrid systems literature is simplified to an automaton [6]. From this perspective, the networked computational components appear in the form of potential network computational delays [47] or as triggers switching control behavior depending on state [6].

Control systems are designed and tested using equations that describe their behavior in terms of relationships between input and output variables [7, cf.]. Accordingly, they can be verified by mathematical analysis with respect to properties such as stability, robustness to disturbance, overshoot, or steady-state errors. Mathematical analysis is only possible if the complexity of the controller allows it: simple classical controllers can be described with linear equations, whereas non-linear adaptive controllers or non-linear fuzzy controllers may require other approaches, including simulation.

The main characteristic of the control systems description is the quantitative nature of descriptions. A classical simple control system, such as a temperature controller in an HVAC or the cruise control in a vehicle, can be a simple analog electronic feedback circuit translating quantitative measurements of temperature or velocity into quantitative actuator response. When looking at the interface to the discrete system, different types of interactions are possible [6] from two-layered systems consisting of a discrete system communicating with a controller over DA/AD transformers to controlled general hybrid dynamical systems which can switch between different controlled behaviors based on state.

A special role within the control systems area falls to fuzzy control [69]. Fuzzy membership functions add a layer of abstraction transforming measurement values, not into discrete values like an AD transformer, but into values in the interval [0, 1]. Fuzzy controllers thus remain in the continuous, quantitative domain of controllers, while performing operations guided by a logical formalism. This allows fuzzy controllers to be evaluated both on the quantitative control level—where, however, the non-linearity of the transformation makes verification a complex task—and on a logical level allowing automated logical proof methods. Fuzzy controlled hybrid systems are thus of particular interest with respect to ISAS, which feature a logic-based learning or reasoning component.

## 2.2 ISAS as intelligent autonomous robots

Robots controlled by simple automata switching between controlled behaviors can be studied using hybrid control systems approaches, as cyber-physical systems. Intelligent autonomous robots, however, such as self-driving vehicles usually contain additionally a learning or reasoning component that controls the state change. If the reasoning component is logic-based, logical consistency as well as entailment of desirable properties can be decided using a logical approach. Moreover, the design of the robot can be derived as a logical consequence of the description of a particular environment [44,45], thus at the same time proving that the robot is fit for its purpose within this environment.

In the case of logic-based intelligent robots, reasoning is qualitative. We will look at this closer below with respect to intelligent agents. Most learning mechanisms, in contrast, operate on quantitative input vectors, either classifying, i.e., producing qualitative output, or predicting, i.e., producing quantitative output, e.g., for actuator control. Both methods pose interesting questions for verification, especially if such systems are to interact with human beings. Taking a black-box perspective, a trained classifier or prediction system is not different from a system developed by an engineer. The control mechanisms it implements can, in theory, be analyzed using the same techniques. The main issue is the high complexity a machine learning mechanism may introduce and the unconventional approach it may take, as machine learning, in contrast to a human engineer, is not required to produce human-readable designs, easy to interpret and verify by human engineers. The issue of high complexity of verification for autonomous robots partially arises from the recently increased power of hardware and thus machine learning and reasoning mechanisms, but it also appears in more classical autonomous robot applications designed by engineers. Where robots are deployed for critical tasks, e.g., in defense or rescue missions, lightweight verification methods have been proposed that evaluate a robot's behavior with respect to an environment and mission specification[59], a methodology we propose to transfer to autonomous vehicles.

The examples cited above, exhibit only a rudimentary level of intelligence and autonomy. Robots with more advanced reasoning abilities and thus verification potential were developed in the area of qualitative reasoning (QR). Given the computational limitations of early robots and the intractability of many AI problems in a complex world, the focus of QR was on providing lightweight yet powerful reasoning capabilities to autonomous robots and other AI systems. The focus of qualitative reasoning is directed on domains that are classically handled using quantitative approaches. It emerged following the Naive Physics Manifestos [31,32] and aimed to facilitate reasoning about space, time, and measurements or estimates of quantities in a similar way to how human beings are able to reason about these domains, i.e., without the complexity of solving difference equations. Qualitative reasoning comprises temporal [22] and spatial [10] reasoning, as well as combinations of these with sensory value domains [19,24]. As a domain of particular importance to autonomous robot applications, Qualitative Spatial Reasoning (QSR), in particular, targets autonomous way-finding and also allows robots to create qualitative representations of space [46].

Verification of systems developed automatically by machine learning techniques has not received much interest from research, yet. These systems were traditionally widely employed only in non-critical domains, such as spam filters, or as systems supporting but not replacing a human professional. Liability and responsibility thus are lying with the human user. The interaction with humans accordingly has received much interest [81], given that, e.g., a smart car operates not within a plant maintained by engineers but in the socio-technical everyday environment of consumers. The problem of system verification from this perspective can then also be framed as a usability issue, i.e., as lying within the area of human–computer interaction (HCI). An increasing number of papers and venues[1] within the areas of intelligent autonomous robots and ISAS, such as pervasive computing systems, thus looks at the issues of fitness for purpose from an HCI point of view using classical HCI methodology, evaluating technologies, e.g., by user studies. The metrics proposed by Steinfeld et al. [86] assess navigation, perception, management, manipulation, and social performance. As an example, they discuss the evaluation of a remote control application following a user study approach, experimentally evaluating the number of errors (collisions), time to complete a navigation task, and subjective ratings by the user regarding effort, learnability, and confidence.

A domain in which robots communicate and collaborate is swarm robotics. Dependability and reliability of swarms has been ensured for swarms using failure mode and effect analysis and reliability modeling [97]. In autonomous vehicles, coordinated swarm behavior exists in the form of platoons or convoys of autonomous vehicles. Verification is possible by combining model checking and agent-based verification techniques [40].

### 2.3 ISAS as agents

Software agents [98] can be viewed as another predecessor of ISAS. Like intelligent autonomous robots they can act autonomously within an environment, this environment, however, is not the physical world but, e.g., a network: software agents lack the hardware to interact with the real world. The focus accordingly is on their social behavior within a network populated by other agents or human users. Depending on the degree of intelligence the agent has, any distributed system can be viewed as a type of software agent, with verification criteria for network protocols such as fairness in access to resources arising as minimal conditions. Agents of higher levels of intelligence feature a knowledge base consisting of one or more ontologies that guide their behavior. Ontologies are logical systems specifying the vocabulary and its meaning for a domain [26]. A knowledge base accordingly can be verified using logical proof of consistency. Moreover, interactive behavior of the agent in a certain environment can be analyzed using methods for analyzing the interaction of strategies with game theory [68], adherence to norms and legal codes with deontic logics [57] and legal ontologies [82], and the development of knowledge states with epistemic logics [17].

To obtain a formal verification method for multi-agent systems (MAS), the logical components of the agent need to be integrated with a formalism for verification of distributed systems [58] such as computation tree logic (CTL) [9] or temporal logic of actions [48]. While agents are sometimes written in purely logic-based formalisms [98] and verification can thus remain largely on the logical level, distributed systems, including some MAS and ISAS, e.g., in pervasive computing, have imperative program parts. Formalisms for handling imperative programs leveraged include, e.g., Floyd–Hoare Logic [18,35]. However, such systems can be evaluated in a two-level process, by proving first that the imperative subsystems provide desired properties and behaviors, which can then be reasoned about on the logical level [30,79].

[1] http://humanrobotinteraction.org

## 3 Machine learning and logical reasoning

The notions of an intelligent environment, intelligent system, or artificial intelligence all refer to the presence of intelligence in the behavior of the system. While the notion of intelligence seems elusive, it is clear that an intelligent system requires reasoning capabilities, whether from machine learning or logic, to accomplish the complex variety of tasks we associate with intelligence. The last decades made considerable advances in both areas. The family tree of logical languages was considerably expanded since the discovery of the expressiveness-tractability tradeoff, and developments in deep learning and reinforcement learning, in particular, led to the current proliferation of AI technologies.

Much of the progress made in AI products over the last decade was made in the machine learning (ML) area. But the gap between learning and reasoning has a deeper correspondence in the gap between perception and logic in general cognitive systems, which is at the heart of such fundamental problems as the symbol grounding problem [28], the question how the symbols of logic and language are grounded in perception. ML is fundamentally perceptual. Complex ISAS, such as autonomous vehicles, operate using classifiers that process sensory input into classes, which in turn can be associated with behaviors, either in the same step or in multiple steps. The resulting intelligence is that of trained behaviors, leaving uncertainty about how the system will behave under unusual situations as the key question. The gap between learning and logic making it hard to apply verification techniques, which are logic-based.

The symbol grounding problem seems easily solved from a perceptual, ML perspective [85,89]: two systems trained with the same data in the same manner will obtain the same classifiers, that is, will agree in their judgement of new input data and the reactions they initiate. From a logical point of view, grounding seems to be much harder: how can we know whether a person actually means what they are saying. The lack of trust in AI systems is on this level. We do not doubt the systems' ability to, e.g., statistically classify reliably, we doubt that they have a sense of what they are classifying or why a certain reaction is trained over another. One may argue that these questions are academic rather than concrete questions for engineering. However, attached to these questions are the crucial notions of trust and responsibility, and the ability to generate explanations, main points in question for ISAS and a focus of increasing attention. In contrast to ML-based systems, logic-based systems can generate explanations for their actions and provide guarantees. In fact, logic underlies all of the above mentioned verification techniques, except those for control systems.

Logical languages have a long tradition. From a historical point of view, the modern, most widely employed first-order logic is young, and dates back to the mid nineteenth century. Before that time, the most widely used logic in Europe and the Middle East was term logic, a logic dating back to Aristotle [for an analysis in modern terms cf.11]. Term logic has limited expressiveness as discovered already by Leibnitz [52]. Propositional logic was formalized by Boole [4], first-order logic with a clear set-theoretical semantics by Frege [20]. The expressive power of formal logics led to the endeavor towards a formalization of the fundamental of mathematics started in [83,96]. The discovery of undecidability of even the seemingly simplest mathematical system that contains only the Peano axioms in Goedel's famous proof [65] as well as conceptual considerations regarding the set-theoretical underpinnings [96] led to multiple endeavors over the twentieth century focussing on decidable, yet expressive logical axiomatizations for specific domains of interest, such as time [70] and space [87]. Computational considerations in the area of artificial intelligence rekindled interest in weaker languages that have a tractable proof mechanism of PTIME or lower complexity [54], which led to the invention and particularly fruitful study of the broad class of Description Logics and their properties at the end of the twentieth century and in the beginning of the twenty-first century [66].

Description logics being designed for the representation of conceptual hierarchies and particularly useful in domains such as object-oriented software engineering, however, do not provide adequate support for representing continuous domains, such as time, space, or temperature, which is a fundamental requirement for reasoning in ISAS. While the objects and classes in ISAS are usually simple and do not require much modeling, in contrast to other domains, ISAS receive sensory input, which is usually continuous numeric, e.g., amount of light, acceleration, noise level, and also output continuous numeric information, such as desired temperature, shade level. This type of information has been modeled in DL as the so-called concrete domains [27] which are not well integrated into the overall theory and lead to a language of high complexity [95].

These results are in contrast to the evolutionary hierarchy of cognitive abilities in nature [25]. In natural cognitive systems, such as in autonomous robots, the interaction with the physical environment has to be handled in real-time with a mechanism of the lowest complexity, while reasoning about classes, as focused by Description Logics, can be provided more time. The Context Logic program [78] is an attempt to follow the cognitive hierarchy, providing a layering of languages, stretching from a real-time compliant representation and reasoning mechanism in the Horn-fragment of propositional logic [76] for reasoning about continuous domains, such as space and time but also measured quantities from sensors such as temperature, to a quantified language with the expressiveness of first-order logic, and sufficient to represent a granularity-dependent characterization of geometry [74] oriented on a mereological base theory [83].

A key result from research in Context Logics (CL) is the recent result [75] suggesting a potential link between logic and analogous representations. The surprising result shows that truth table and DNF for propositional formulae corresponding to formulae of the most fundamental CL language [76] have a quantitative property that is analogous to the content of the formulae, a result that sheds new light on the grounding problem from an unexpected angle. Logical formulae thus could be grounded in reality in a more literal manner than expected pointing towards the potential for a new category of AI system that unites ML-based perceptual intelligence with logical reasoning. This type of AI system could have the potential to generate explanations for its actions and provide guarantees, which would make it both self-explainable and thus more trustworthy. While the current ML-based systems thus are the first feasible real-time reactive ISAS, but due to their black-box character lack predictability, explainability, and verifiability posing a high risk, the new AI systems, such as the CL-based systems, able to closely integrate perception and logic would allow a direct integration of verification and monitoring strategies due to their logic-based component. Given [75], such systems may be considerably closer than previously assumed.

## 4 Verification of ISAS

As discussed, ISAS are different from cyber-physical systems in that they require a reasoning component; they are different from intelligent agents in that they require interaction with quantitative domains; and they are different from autonomous intelligent robots in that they require interaction with other ISAS and human beings within a larger concurrent network domain. Logic plays a dominant role both as the framework for conventional computational verification mechanisms and as the language of logic-based reasoning. In reasoning, the intelligent system determines how it reacts given a specification of desired behaviors and the complex context of situational factors at a time. This aspect gives additional flexibility to the way the system reacts under different circumstances and with different other devices present.

Figure 3 illustrates the architecture of an ISAS in contrast to earlier sensor–actuator systems. Whereas an ISAS is a more generic intelligent system (Fig. 3a) with components for interaction with the environment on several layers and the actual functionality embedded in the reasoning and learning components, the classical controller (Fig. 3b) and CPS (Fig. 3c), in contrast, are developed for a more narrowly defined specific task. Figure 3 uses the often invoked HVAC example to illustrate this: a classical analog control system regulates temperature on the basis of a hardware that directly relates input to output (Fig. 3b); a cyber-physical system is a control system that additionally involves a software component whose states influence the way the system reacts to input and controls different components (Fig. 3c); the context-aware ISAS in contrast creates behavior from a specification of knowledge about desirable behavior (Fig. 3a).

To take the HVAC example: an ISAS building will regulate the heaters, lights, windows, shades, and air conditioning based on the building administrators' specification, but also taking into account specific users' and groups of users' situational needs. In general, good office working conditions, for instance, require sufficient light; however, when the ISAS meeting room detects that a presentation is about to start, shades will be activated to dim the light to a level that is below this general work level. To perform actuation, the ISAS will itself rely on control systems or CPS functionality on its lower levels to successfully perform subtasks. The system, however, detects and reacts to a complex situation in a way that fulfills requirements of the situation enabled to handle error conditions or contradictions on several levels. For example, when multiple components belonging to users with different preferences about an environmental parameter, expressing different specifications, are present the system may need to arbitrate between conflicting specifications, leveraging means of meta-level reasoning and communication so as to resolve conflicts resulting from errors or conflicting information from different users.

In basic context-aware systems, the so-called context models [cf. [3], for a survey] provide basic reasoning functionality. They can be provided in a constraint format [13], as a tree structure [37,39], or in an explicitly logical format [34,79]. For verification purposes, constraints can be encoded in all three cases in terms of logical rules. Depending on the context, i.e., the informational environment, the system changes behavior according to such logical rules, which may be set not only by system designers, but also by local system administrators, and even by end-users. A reachability analysis, under these circumstances, becomes an infeasible task, not only because of the combinatorial explosion of all possible system configurations, but also because of the system under study not being fully specified: the ultimate system behavior in a specific space and situation is thus a function of the interplay of several subsystems that are not modular, but logically dependent on one another. As Milner [63] pointed out, we need to revisit our conceptions of modularity, to understand such systems.

A number of approaches have been brought forward to verify pervasive computing systems. The prevalent strategy is to model pervasive computing systems as extensions of distributed systems. Examples of this approach are ambient calculus [94], ambient logic [8,73], bigraphs [62,64], or $\pi$-calculus [33,51]. However, the continuous physical reality of sensors and actuators, space and time, is usually mapped into discrete states or network devices. Ambient Logic [8] upon which [73] is built, for instance, assumes tree-based location

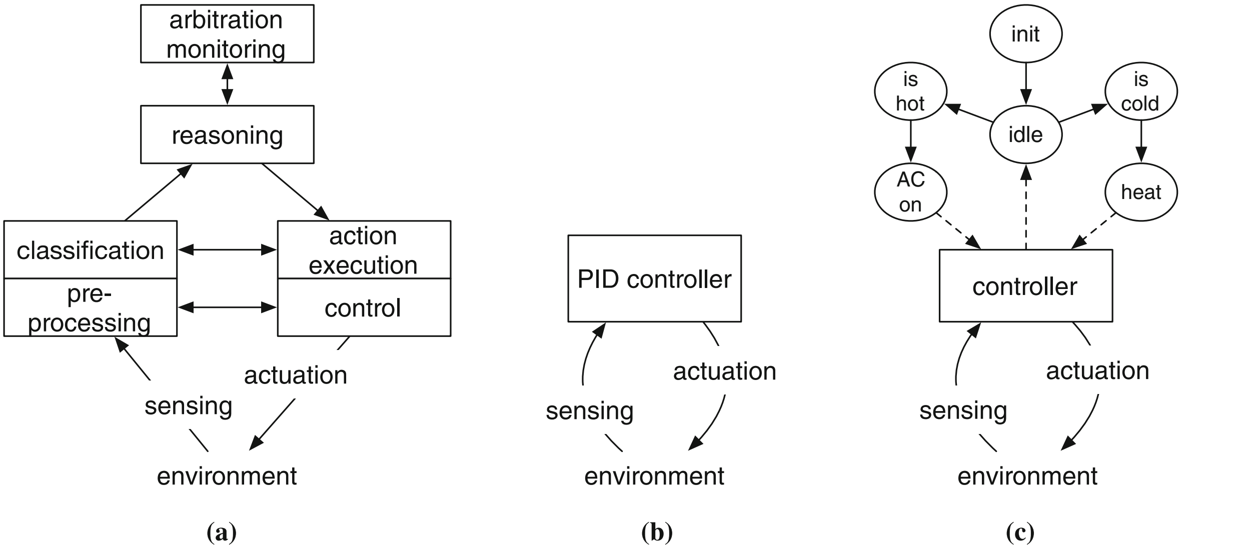

**Fig. 3** Architectural components of an ISAS **a** in comparison to a classical control system **b** and a cyber-physical system **c** for the standard HVAC example

hierarchies as its representation of space. This requires a clear partitioning which is problematic from a point of view that takes sensory uncertainty and the continuity of physical space into account [77].

Other approaches derived from a classical distributed systems perspective focus on system design or monitoring. Kawahara et al. [41], for instance, propose to use the systems modeling language (SysML) for embedded system specifications to be analyzed and verified during the design process. The monitoring using runtime verification [53] of intelligent environments is another way to ensure reliability or at least controlled shutdown mechanisms [12]. Self-monitoring is a key design component of advanced ISAS (Fig. 3a).

Few approaches [5,60] represent the sensor point of view. Boytsov and Zaslavsky [5] propose a geometric approach to characterize sensor value state spaces which are partitioned to yield the states of automata that thus formalize the context-dependent reaction component. Similarly, constraint-based hybrid automata are leveraged by del Mar Gallardo et al. [60]. This perspective is very close to the control theoretic and CPS perspective, and as in CPS, the expressiveness of the contextual reasoning system is limited to the expressiveness of automata. The intelligence of the systems, which with the human ideal plays an important role for the trust we have, e.g., in human drivers, does not play a prominent role in either approach [5,60].

The crucial point is how to build a bridge between equations formalizing the quantitative domain of sensors and actuators and the qualitative domain of logics. In collaborations with students, the Context Logics (CL) family of logical languages was built over the last ten years, to facilitate this crucial task for both reasoning within ISAS and reasoning about ISAS. From the existence of fast systems with good ISAS reasoning abilities, it was clear that there was a class of primary reasoning tasks in context modeling that was tractable. Formalizing how these systems reasoned, we therefore developed CL as a simple logical language [80], similar in style to DL and the simple ancient term logic as well as the familiar equation formalism for quantitative domains. The advantage of such a logic was that we thus could not only make systems more interoperable [38] but also provide reasoning about ISAS giving rise to the CL-based ontology-based verification method [79]. This method can be extended to the intelligent environments domain, as illustrated in Fig. 4.

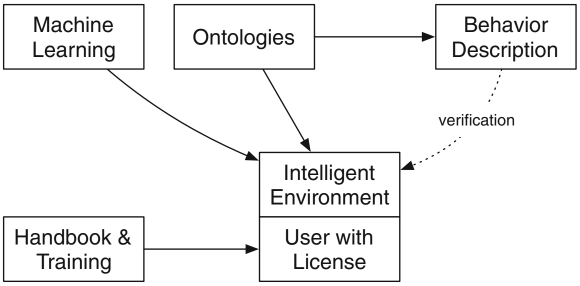

**Fig. 4** Ontology-based verification method for intelligent environments: while administrators' licenses and trainings have to be extended to allow them to properly operate advanced robotics applications that can do much harm, including building automation applications, verification mechanisms have to be employed to ensure that intelligent environments can be guaranteed by manufacturers to operate according to the legal framework before deployment

In contrast to approaches more oriented towards verification of the distributed computing components [8,12,33, 51,62,64,73], the approach [79] focused on allowing greater complexity on the side of the contextual reasoning. A key design goal of the CL family of logics was to have a language that allows seamless integration of several levels of expressiveness that are of high importance for reliable intelligent environments: the general [80] designed for the reasoning level of ISAS contains a fragment that directly corresponds to the real-time reactive light-weight PQSR [76], which was even developed into a qualitative spatial reasoning (QSR) calculus [21]. Adding quantifiers [74], in contrast, adds full first-order logic expressiveness to the decidable [80] allowing the specification of a mereogranular geometry, a formal framework that facilitates, inter alia, reasoning about trajectories of extended objects.

### 4.1 Locally safe environments

A key to making ISAS verification possible is the reduction of complexity. A promising approach well compatible with the framework of Schmidtke and Woo [79] is the adoption of detailed models of an environment as used in the verification of safety critical robotics applications, e.g., with the MissionLab editor of Lyons et al. [59]. While the approach of encoding the complete legal traffic framework and employing reasoning to derive proper behavior in a situation is not a feasible approach and cannot be real-time compliant, it is clear that human intelligence also does not function that way.[2] Navigating in unknown environments is harder than navigating through a well-known environment, for instance. A driver encountering a specific complex road arrangement or dynamic situation for the first time is more likely to make errors than an experienced driver. Applied to the problem of verification, the key is to verify the proper behaviors as mission specifications in a spatially piecewise manner (Fig. 5), i.e., to prove that a vehicle will perform according to specification with respect to the large but tractable space of a specific intersection and not the intractable space of all possible intersections.

The necessary data to geometrically represent intersections or other parts of a road network is often already available from municipalities and other local stakeholders, e.g., in the form of digital surface models and detailed location surveys. Such information is collected and processed through geographical information systems (GIS). The quantitative–qualitative barrier [19,24] has been a long-standing topic in efforts to represent space qualitatively, with applications not only in intelligent autonomous robot research (Sect. 2.2) but also in GIS [15]. A wealth of dedicated approaches exist to represent space qualitatively in light-weight calculi [10,14, 21,72]. Within the Context Logic hierarchy of languages a light-weight spatial reasoning calculus was presented in [76]. Given that such a language necessarily has limited expressiveness but efficient performance [54], it fulfills the criteria for a real-time component that could be installed for controlling and monitoring [53] an autonomous vehicle's operation. In order for generally characterizing admissible situations on an intersection, however, such vehicle-dependent descriptions need to be linked to a complete description of the intersection, which requires a more expressive language. Within the Context Logic hierarchy, [74] is an expressive framework that allows the description of spatial structures in a flexible and detailed manner, and providing qualitative geometric concepts—not only, e.g., topological notions [72]—translation from quantitative geometric data is facilitated. The qualitative nature of the resulting descriptions makes it then possible to fully specify an intersection as a spatiotemporal entity affording certain behaviors.

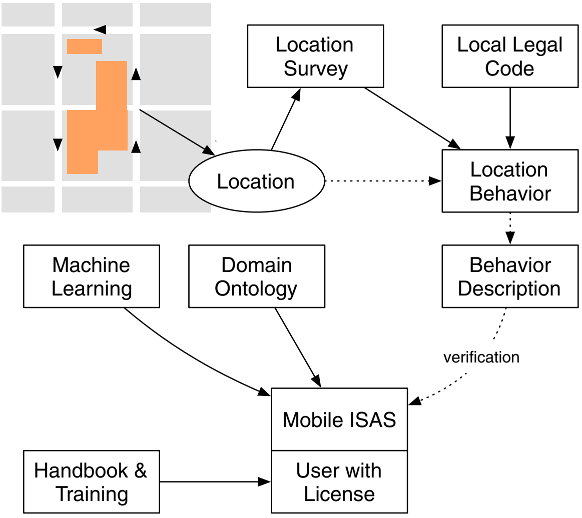

**Fig. 5** Ontology-based verification method for mobile ISAS: while vehicles operate in autonomous mode, manufacturers carry full responsibility and liability for the vehicle, taking legally the role of the driver. Ontological specifications of specific locations can be leveraged by manufacturers to reduce their risk in a systematic way: when vehicles enter an area or situation, in which the manufacturer does not want to take responsibility, a well-defined handover process—or, if the user does not take over, shutdown—is initiated while still within the safe area. Turning autonomous operation off in a well-defined manner, the manufacturer does not incur damages from a user's action or inaction. The device is maximally safe if used as prescribed

Based on this formal description of a location, any basic verification mechanism successfully used for handling concurrency in intelligent environments can be applied, be it

[2] This is one of the reasons why drivers of autonomous cars fail to prevent accidents: it takes considerable time for a human being to understand a complex situation, so as to filter and select among the wealth of available possible actions an appropriate one. While an alert driver handling an ongoing incrementally changing driving context, is at any time within a properly filtered context and able to react within a one second delay, a driver relying on self-driving capabilities of an autonomous car, will require a considerably extended comprehension period for acquiring the specific driving context to be added to reaction time. With respect to the literature on driver's reaction times, this case corresponds to one of reduced visibility [84], known to increase reaction times.

Linear Temporal Logic (LTL) as in [60] or Communicating Sequential Processes (CSP) as in [79].

With a formal verification method possible, a dangerous void of responsibility for autonomous vehicles [56], and ISAS in general, can be addressed: if autonomous behavior can be guaranteed to be safe for specific verifiably safe locations and situations, manufacturers can take full responsibility for autonomous operation of their product within those situations, including full product liability for actions committed by the ISAS. Before leaving such a locally safe area or situation, manufacturers could disable the autonomous behavior in a controlled fashion so as to not be held accountable: the transition between safe and unsafe situations would become a detectable feature. Failure of the driver to take over, would then be similar to other improper system operation, such as running a red light or stalling the engine with the legal code and law enforcement accordingly updated.

Ontological specifications of locations against which autonomous vehicle behavior could be verified could come from a range of sources. Possible sources could be public–private partnerships or industry consortia. In any case, manufacturers could decide whether they want to support autonomous vehicle operation within a specific area and take responsibility given the risk or not and can then locally enable or disable the autonomous driving functionality. With a clearly defined handover phase initiated with sufficient lookahead, drivers would have the necessary time to acquaint themselves with the situation so as to be able to take over, and knowing their vehicle will otherwise come to a halt would be motivated to do so.

## 5 Discussion

The CL-based approach unites all three areas: the handling of quantitative values received from sensors and sent to actuators, the imperative and network component, and the reasoning employed by an intelligent system. We discuss in this section, that this shows that autonomous transportation system verification is possible Sect. 5.1 and argue that it is a necessity of responsible engineering Sect. 5.2.

### 5.1 Safe autonomous transport systems are possible

The last section introduced the notion of locally safe systems. As location together with other parameters important for, e.g., traffic light recognition success, such as time of day, can be determined by GPS, clocks, and other sensors, a manufacturer can know with high reliability how a given situation will be handled by the system, and accordingly, what its probability of success is. If this probability is too low, e.g., because the sun has a certain angle with respect to the traffic lights at a certain intersection as approached from a certain direction at a certain time, the necessity of sensory compensation, human support, or complete handover to a driver can be predicted. With a large portion of a situation predictable, the system runs less like a human-like AI, and more like a space agency mission, where ideally nothing is left to chance.

Today's AI is not the human-like robot taxi from science fiction. In many basic ways, it still is far less sophisticated than animal intelligence, not to mention human intelligence: able to reproduce trained patterns like a trained dog, where the dog is still in many ways superior to the AI, showing that there is a long way to go to reach a human-like intelligence. The current situation of astonishing performance, e.g., in games by AI is largely due to the scalability of ML techniques. ML mechanisms, in contrast to reasoning, for instance, can make full use of large-scale computing power. Human intelligence, in contrast, is so remarkable due to its ability to handle such tasks on the basis of a vastly less powerful and fast hardware substrate. It is the ability of the human mind to beat a vastly more powerful computing platform, e.g., in chess or go that is remarkable, not that a six or nine orders of magnitude faster large computing platform can be constructed that can beat a human being at a single task.

Our attribution of intelligence to this type of system is a result of anthropomorphism, which is known to increase trust [93] and inspire forgiveness [92]. It is, at least with the current state of the art, a misplaced label, whose main benefit unfortunately is in marketing a technology that is currently not safe.

Looking at the wider ramifications for transportation systems, the task of assessing the impact of the new intelligent versions of classical domains, such as transportation, but likewise electricity distribution and health care, requires rethinking assumptions, for example, while some drivers may employ certain driving styles, behavior being randomly distributed is unlikely to systematically impact the functioning of a municipality's transportation system. A manufacturer equipping its autonomous vehicles with a certain behavior, in contrast, may produce a considerably different transportation system. This opens new opportunities. Transport simulation systems, for instance, can feature autonomous vehicle behavior as a controllable parameter.

A disadvantage is that large-scale complex systems, in particular, when they are the result of machine learning techniques, are usually difficult to understand even by their designers. The issues resulting from this have already led to new legislation, such as the EU's General Data Protection Regulation [16], requiring that machine learning systems be enabled to explain their decisions [cf. [90], for a more in-depth discussion]. In smart city contexts, for instance, we want to avoid "buggy, brittle and hackable cities" [43]. Even more so than with previous Pervasive Computing systems [63], we need means to understand and control ISAS.

To do this, verification needs to take a different approach that not only encompasses system models but also models of the intelligence operating them. The system model needs to encompass a suitable description of the involved smart systems themselves. Until now this was posing the question as to how the resulting complexity could possibly be handled. Local safety breaks the problem down into a large set of much smaller problems of making a distinct intersection or road segment safe, potentially even split up further temporally into times of day and week, or seasons. The first autonomous vehicles may need to be "Sunday drivers" operating under considerably restricted conditions only. In this way, the context dependency, which at the first glance seemed to be the problem, may turn out to provide the solution to making the verification problem incremental, decomposable, and thus tractable.

As a second aspect, formal models of systems involved and verification mechanisms can also allow much better traffic control and safety, and considerable interest may exist from private and public stakeholders to contribute to the effort. We outlined how a piecewise process and potentially concerted effort of formal modeling can make verification computationally tractable. With tractable verification possible, ISAS manufacturers are enabled to prevent problematic ISAS behavior that infringes on existing laws, as for any other device's malfunctioning.

### 5.2 The ethical dimension

Before concluding this paper, this section outlines that the stakes may be even higher than the human lives in danger. The Universal Declaration of Human Rights (UDHR) [88] declares in article 12:

> No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honor and reputation. Everyone has the right to the protection of the law against such interference or attacks.

In today's reality, this right is considerably limited. Although it exists in principle, it is not only infringed upon under the premise of national security reasons in a number of ways, but we all quasi-voluntarily relinquish it regularly. So much so, that, in fact, it would be a considerable impediment to everyday life not to relinquish it. Privacy policies allow us to accept or decline using a software, but not to choose alternatives that do not force us to give up on this fundamental right, but the reason for this is not that alternatives do not exist or have fundamental technical drawbacks. Alternatives are not only possible but can be considered technically superior in additional ways. Langheinrich [49] in his seminal article "Privacy by Design" proposed several families of mechanisms that are not only superior in terms of privacy, but also have a lower environmental footprint and are even more natural and easier to understand as well as computationally faster than the solutions we have come to rely on.

While the right to privacy may be lost, the new ISAS technologies operating in the physical world affect an even more fundamental right, the right to live. The UDHR [88] in Article 3 declares:

> Everyone has the right to life, liberty and security of person.

Two drivers of autonomous cars and one pedestrian have lost their lives so far, with responsibility in each case carried by the human being sitting in the driver's seat, the reasoning being that they should have been taken control to prevent the accident. In the case of the first human life claimed by an autonomous vehicle, for instance, the driver was made responsible, since he was using the self-driving capability without providing full supervision [67]. Engineering solutions proposed to address the problem seek to control the driver leading to a situation in which the deceased driver is discussed like the faulty part in the machine, but has to take full responsibility for the machine.

Much in this argument hinges on the assumption that formal analysis of ISAS would be intractable and that machine learning creates non-verifiable technologies. As this paper has argued, this is not the case. The key issue is that today's AI is not yet trustworthy. Similar to the situation before the first AI winter, what we can achieve is producing solutions to selected problems of intelligence that work surprisingly well within bounds not well understood by the public.

Anthropomorphizing a device as still learning, evoking emotions of forgiveness usually reserved for children, while withholding compassion for its victims, e.g., with justifications, such as that they had been jaywalking or taking drugs, is a path we will not want to take. Bringing products to the market which are not technically mature should not become the norm. We have been getting used to increasing frequencies of software product updating. With autonomous robots this poses a risk to human safety. Decreasing reliability in software and increasing speed in obsolescence of hardware due to auto-update functionalities in networked devices are of considerable concern in safety-critical domains, and for example, avoided in the defense sector [61].

## 6 Conclusions

With the first fatalities caused by autonomous cars and discussions partially suggesting a license agreement type of solution, formal verification of ISAS has become a research topic of highest priority. The human user can either be required to be alert and in control, or relieved from control and not responsible, a mixture of both creates a contradictory legal double bind, which cannot lead to a just solution. In the case of privacy, strongly felt to be a human right in 1948, an impossible luxury today, we have witnessed that it is possible to erode human rights, and thus human dignity. The argument that a user can opt out of these new technologies is doubtable. Like paper maps, travel by horse, and public telephones are no longer a realistic option for the everyday employee worried to keep their position, autonomous vehicles and other autonomous robots will soon be a convenience society as a whole, including government agencies and employers will rely on.

Unfortunately, no approach for verifying autonomous car behavior in its three dimensions of CPS, autonomous robot, and multi-agent system is currently available. This leaves the door wide open for new technologies we do not have control over and license-type agreements to be established as the norm. This paper aims to propose an alternative path. We gave a short overview of the different approaches that could be applicable to the task of analyzing and verifying intelligent sensor–actuator systems. We outlined the fundamental landscape of earlier theoretical perspectives, more recent attempts to address combined types of systems, such as cyber-physical systems, agents, and autonomous robots, as well as recent approaches to address the verification of ISAS put forward in the area of pervasive computing systems.

We showed that the ontology-based method based on the Context Logic (CL) family of languages may provide a pathway to a new holistic approach that unites key properties of importance for ISAS: logic-based reasoning about rules, dynamic context-dependent triggering of behavior in complex multi-actor scenarios, handling of sensory information, and a well-defined interface to lower-level and hardware-related computing components and their verification mechanisms. The language was developed specifically to enable decidable qualitative reasoning over continuous domains, such as space, time, and sensor values in ISAS. The CL-framework can support verification of ISAS as it integrates a mechanism for both reasoning within context, with the decidable core language usable by ISAS, and reasoning about contexts with a first-order variant of the same language usable in the development stage. The verification mechanism can thus be connected to an ontology describing admissible behaviors as well as to a verification mechanism for the underlying hardware-related context-triggered code fragments.

The key role, however, is taken by a particularly lightweight fragment of the language, which is fundamentally more perceptual in nature than other logical languages. It is not only suitable for monitoring ISAS in real time as required for run-time verification, but may provide much more: a logical link to the perceptual component of ISAS.

A crucial problem of any verification method for autonomous systems verification, the explosion of state-space complexity for a general specification of behavior in the real world, was addressed by the proposal of a spatially and situationally piecewise verification of behavior as locally safe. While the problem of generic verification of a system for the complete legal framework for any possible road and situation is not feasible, distinct situations, such as to turn right at a given intersection on the basis of the legal code, are. By restricting autonomous vehicle operation to distinct locations, whose safe navigability can be guaranteed, full manufacturer liability for autonomous vehicles can be made a requirement without sacrificing the technology. Manufacturers can reduce their risk by turning off autonomous operation in a controlled and enforced way outside of these distinct areas, thus requiring a licensed driver to take control and responsibility, as also recommended by the National Transportation Safety Board [67].

The path to full self-driving vehicles may be prolonged by several years following these suggestions, and one might argue that each year we delay costs tens of thousands of lives, some of which immature self-driving cars while killing others might save. However, a safer technology will save more lives in the long run, as it will save our standards and values. The Universal Declaration of Human Rights [88] declares our human rights as universal and unalienable. Abandoning these rights means denying them to the people of the future upon whose ability to enjoy them we would thus infringe.

## References

1. Baheti R, Gill H (2011) Cyber-physical systems. Impact Control Technol 12:161–166
2. Behnke S, Sheh R, Sarıel S, Lee DD (2017) RoboCup 2016: Robot World Cup XX. Springer
3. Bettini C, Brdiczka O, Henricksen K, Indulska J, Nicklas D, Ranganathan A, Riboni D (2010) A survey of context modelling and reasoning techniques. Pervas Mobile Comput 6(2):161–180
4. Boole G (1854) An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities. Dover Publications, New York
5. Boytsov A, Zaslavsky A (2013) Formal verification of context and situation models in pervasive computing. Pervas Mobile Comput 9(1):98–117
6. Branicky MS, Borkar VS, Mitter SK (1998) A unified framework for hybrid control: model and optimal control theory. IEEE Trans Autom Control 43(1):31–45
7. Brogan WL (1990) Modern control theory. Pearson,
8. Cardelli L, Gordon AD (2000) Mobile ambients. Theoret Comput Sci 240(1):177–213
9. Clarke E, Emerson E (1982) Design and synthesis of synchronization skeletons using branching time temporal logic. Logics Program 52–71
10. Cohn AG, Hazarika SM (2001) Qualitative spatial representation and reasoning: an overview. Fundamenta Informaticae 46(1–2):1–29
11. Corcoran J (1973) A mathematical model of Aristotle's syllogistic. Archiv für Geschichte der Philosophie 55(2):191–219

12. Coronato A, Pietro GD (2012) Tools for the rapid prototyping of provably correct ambient intelligence applications. IEEE Trans Softw Eng 38(4):975–991
13. Dey AK, Abowd GD (2000) Towards a better understanding of context and context-awareness. In: Workshop on the what, who, where, when, and how of context-awareness. ACM
14. Egenhofer MJ (1994) Spatial SQL: a query and presentation language. IEEE Trans Knowl Data Eng 6(1):86–95
15. Egenhofer MJ, Mark DM (1995) Naive geography. In: Frank AU, Kuhn W (eds) Information Spatial Theory, A Theoretical Basis for GIS. Springer, pp 1–15
16. European Union (2016) General data protection regulation. http://data.europa.eu/eli/reg/2016/679/oj. Accessed 4 Oct 2018
17. Fagin R, Halpern JY, Moses Y, Vardi M (2004) Reasoning about knowledge. MIT press, USA
18. Floyd RW (1967) Assigning meanings to programs. Program Verif 14:65–81
19. Forbus KD (1984) Qualitative process theory. Artif Intell 24(1):85–168
20. Frege G (1879) Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. L. Nebert
21. Freksa C (1991) Qualitative spatial reasoning. In: Cognitive and linguistic aspects of geographic space. Springer, New York, pp 361–372
22. Freksa C (1992) Temporal reasoning based on semi-intervals. Artif Intell 54(1–2):199–227
23. Gajski DD, Vahid F, Narayan S, Gong J (1994) Specification and design of embedded systems, vol 13. Prentice Hall, Englewood Cliffs
24. Galton A (2000) Qualitative spatial change. Oxford University Press, Oxford
25. Gärdenfors P (2005) The detachment of thought. In: Erneling C, Johnson D (eds) The mind as a scientific subject: between brain and culture. Oxford University Press, Oxford, pp 323–341
26. Guarino N (1998) Formal ontology and information systems. In: Guarino N (ed) Formal Ontol Inf Syst. IOS Press, Amsterdam, pp 3–15
27. Haarslev V, Lutz C, Möller R (1999) A description logic with concrete domains and a role-forming predicate operator. J Logic Comput 9(3):351–384
28. Harnad S (1990) The symbol grounding problem. Phys D Nonlinear Phenom 42(1–3):335–346
29. Havelund K, Shankar N (1996) Experiments in theorem proving and model checking for protocol verification. In: International symposium of formal methods Europe. Springer, pp 662–681
30. Hawblitzel C, Howell J, Kapritsos M, Lorch JR, Parno B, Roberts ML, Setty S, Zill B (2015) Ironfleet: proving practical distributed systems correct. In: Proceedings of the 25th symposium on operating systems principles. ACM, pp 1–17
31. Hayes P (1985) The second naive physics manifesto. In: Hobbs J, Moore R (eds) Theories of the commonsense world. Ablex Publishing Corporation, Norwood, pp 1–36
32. Hayes PJ et al (1978) The naive physics manifesto. Tech. rep., Université de Genève, Institut pour les études sémantiques et cognitives
33. Hennessy M (2007) A distributed pi-calculus. Cambridge University Press, Cambridge
34. Henricksen K, Indulska J (2006) Developing context-aware pervasive computing applications: models and approach. Pervas Mobile Comput 2:37–64
35. Hoare CAR, Jifeng H (1998) Unifying theories of programming, vol 14. Prentice Hall, Englewood Cliffs
36. Holzmann GJ (1990) Algorithms for automated protocol verification. AT&T Techn J 69(1):32–44
37. Hupfeld F, Beigl M (2000) Spatially aware local communication in the RAUM system. In: IDMS. Springer, pp 285–296
38. Jang S, Woo W (2003) ubi-UCAM: a unified context-aware application model. In: Blackburn P, Ghidini C, Turner RM, Giunchiglia F (eds) International conference on modeling and using context, pp 178–189
39. Jiang C, Steenkiste P (2002) A hybrid location model with a computable location identifier for ubiquitous computing. In: Borriello G, Holmquist LE (eds) Ubiquitous Comput. Springer, Gothenburg, pp 246–263
40. Kamali M, Dennis LA, McAree O, Fisher M, Veres SM (2017) Formal verification of autonomous vehicle platooning. Sci Comput Program 148:88–106
41. Kawahara R, Dotan D, Sakairi T, Ono K, Nakamura H, Kirshin A, Hirose S, Ishikawa H (2009) Verification of embedded system's specification using collaborative simulation of sysml and simulink models. In: Model-based systems engineering, 2009. MBSE'09. International Conference on, IEEE, pp 21–28
42. Khaitan SK, McCalley JD (2015) Design techniques and applications of cyberphysical systems: a survey. IEEE Syst J 9(2):350–365
43. Kitchin R (2014) The real-time city? big data and smart urbanism. GeoJ 79(1):1–14
44. Kloetzer M, Belta C (2010) Automatic deployment of distributed teams of robots from temporal logic motion specifications. IEEE Trans Robot 26(1):48–61
45. Kress-Gazit H, Fainekos GE, Pappas GJ (2009) Temporal-logic-based reactive mission and motion planning. IEEE Trans Robot 25(6):1370–1381
46. Kuipers B (2000) The spatial semantic hierarchy. Artif Intell 119(1–2):191–233
47. Kumar P, Goswami D, Chakraborty S, Annaswamy A, Lampka K, Thiele L (2012) A hybrid approach to cyber-physical systems verification. In: Proceedings of the 49th annual design automation conference. ACM, pp 688–696
48. Lamport L (1994) The temporal logic of actions. ACM Trans Program Lang Syst (TOPLAS) 16(3):872–923
49. Langheinrich M (2001) Privacy by design—principles of privacy-aware ubiquitous systems. In: Abowd GD, Brumitt B, Shafer S (eds) Ubiquitous computing. Springer, Heidelberg, pp 273–291
50. Lee EA (2008) Cyber physical systems: Design challenges. In: Object oriented real-time distributed computing (ISORC), 2008 11th IEEE international symposium on IEEE, pp 363–369
51. Lekshmy VG, Bhaskar J (2015) Programming smart environments using $\pi$-calculus. Procedia Comput Sci 46:884–891
52. Lenzen W (2004) Calculus Universalis. Studien zur Logik von GW Leibniz, Mentis, Paderborn
53. Leucker M, Schallhart C (2009) A brief account of runtime verification. J Logic Algebraic Program 78:293–303
54. Levesque HJ, Brachman RJ (1987) Expressiveness and tractability in knowledge representation and reasoning. Comput Intel 3(2):78–93
55. Lin FJ, Chu P, Liu MT (1987) Protocol verification using reachability analysis: the state space explosion problem and relief strategies. ACM SIGCOMM Comput Commun Rev 17(5):126–135
56. Liu HY (2017) Irresponsibilities, inequalities and injustice for autonomous vehicles. Ethics Inf Technol 19(3):193–207. https://doi.org/10.1007/s10676-017-9436-2
57. Lomuscio A, Sergot M (2003) Deontic interpreted systems. Studia Logica 75(1):63–92
58. Lomuscio A, Qu H, Raimondi F (2009) Mcmas: A model checker for the verification of multi-agent systems. In: International conference on computer aided verification. Springer, pp 682–688
59. Lyons DM, Arkin RC, Jiang S, Liu TM, Nirmal P (2015) Performance verification for behavior-based robot missions. IEEE Trans Robot 31(3):619–636
60. del Mar Gallardo M, Lavado L, Panizo L, Titolo L (2017) A constraint-based language for modelling intelligent environments. J Reliab Intell Environ 3(1):55–79

61. Merola L (2006) The COTS software obsolescence threat. In: Fifth international conference on commercial-off-the-Shelf (COTS)-based software systems (ICCBSS'05), pp 127–133. https://doi.org/10.1109/ICCBSS.2006.29
62. Milner R (2006a) Pervasive process calculus. Electron Notes Theoret Comput Sci 162:255–259
63. Milner R (2006b) Ubiquitous computing: shall we understand it? Comput J 49(4):383–389
64. Milner R (2008) Bigraphs and their algebra. Electron Notes Theoret Comput Sci 209:5–19
65. Nagel E, Newman JR, Hofstadter DR (2001) Gödel's proof. New York University Press, New York
66. Nardi D, Brachman RJ (2002) An introduction to description logics. In: McGuinness D, Nardi D, Patel-Schneider P (eds) F Baader DC. Description Logic Handbook. Cambridge University Press, Cambridge, pp 5–44
67. National Transportation Safety Board (2017) Collision between a car operating with automated vehicle control systems and a tractor-semitrailer truck near williston, florida may 7, (2016) Highway Accident Report NTSB/HAR-17/02. National Transportation Safety Board, Washington, DC
68. Nisan N, Roughgarden T, Tardos E, Vazirani VV (2007) Algorithmic game theory, vol 1. Cambridge University Press, Cambridge
69. Passino KM, Yurkovich S, Reinfrank M (1998) Fuzzy control. Addison-Wesley, USA
70. Prior A (1968) now. Nous 2:101–119
71. Rajkumar RR, Lee I, Sha L, Stankovic J (2010) Cyber-physical systems: the next computing revolution. In: Proceedings of the 47th design automation conference. ACM, pp 731–736
72. Randell D, Cui Z, Cohn A (1992) A spatial logic based on region and connection. In: Knowledge representation and reasoning. Morgan Kaufmann, pp 165–176
73. Ranganathan A, Campbell RH (2008) Provably correct pervasive computing environments. In: PerCom, pp 160–169
74. Schmidtke HR (2016) Granular mereogeometry. In: Ferrario R, Kuhn W (eds) Formal ontology in information systems. In: Proceedings of the 9th international conference (FOIS 2016), IOS Press, Frontiers in Artificial Intelligence and Applications, vol 283, pp 81–94
75. Schmidtke HR (2018) Logical lateration—a cognitive systems experiment towards a new approach to the grounding problem. Cognit Syst Res. https://doi.org/10.1016/j.cogsys.2018.09.008
76. Schmidtke HR, Beigl M (2011) Distributed spatial reasoning for wireless sensor networks. In: Modeling and using context. Springer, pp 264–277
77. Schmidtke HR, Woo W (2007) A size-based qualitative approach to the representation of spatial granularity. In: Veloso MM (ed) Twentieth international joint conference on artificial intelligence, pp 563–568
78. Schmidtke HR, Woo W (2008) Partial ordering constraints for representations of context in ambient intelligence applications. In: Villadsen J, Christiansen H (eds) Constraints and language processing, pp 61–75
79. Schmidtke HR, Woo W (2009) Towards ontology-based formal verification methods for context aware systems. In: Tokuda H, Beigl M, Brush A, Friday A, Tobe Y (eds) Pervasive 2009. Springer, pp 309–326
80. Schmidtke HR, Hong D, Woo W (2008) Reasoning about models of context: A context-oriented logical language for knowledge-based context-aware applications. Revue d'Intelligence Artificielle 22(5):589–608
81. Sheridan TB (2016) Human-robot interaction: status and challenges. Hum Factors 58(4):525–532
82. Singh MP (1999) An ontology for commitments in multiagent systems. Artif Intell Law 7(1):97–113
83. Srzednicki JJ, Stachniak Z (eds) (2012) Leśniewski's Systems Protothetic, Nijhoff International Philosophy Series, vol 54. Springer, Netherlands
84. Stanisław Jurecki R, Lech Stańczyk T, Jacek Jaśkiewicz M (2017) Driver's reaction time in a simulated, complex road incident. Transport 32(1):44–54
85. Steels L (2008) The symbol grounding problem has been solved. so what's next. Symbols and embodiment: Debates on meaning and cognition pp 223–244
86. Steinfeld A, Fong T, Kaber D, Lewis M, Scholtz J, Schultz A, Goodrich M (2006) Common metrics for human-robot interaction. In: Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction. ACM, pp 33–40
87. Tarski A (1956) Foundations of the geometry of solids. In: Logic, Semantics, Metamathematics. Papers from 1923 to 1938. Clarendon, Oxford, pp 24–29
88. UN General Assembly (1948) Universal declaration of human rights http://www.un.org/en/universal-declaration-human-rights/. Accessed 16 Apr 2018
89. Vogt P (2002) The physical symbol grounding problem. Cognit Syst Res 3(3):429–457
90. Wachter S, Mittelstadt B, Floridi L (2017) Transparent, explainable, and accountable ai for robotics. Sci Robot 2(6)
91. Walsh GC, Ye H, Bushnell LG (2002) Stability analysis of networked control systems. IEEE Trans Control Syst Technol 10(3):438–446
92. Waytz A, Epley N, Cacioppo JT (2010) Social cognition unbound: Insights into anthropomorphism and dehumanization. Curr Direct Psychol Sci 19(1):58–62
93. Waytz A, Heafner J, Epley N (2014) The mind in the machine: anthropomorphism increases trust in an autonomous vehicle. J Exp Soc Psychol 52:113–117
94. Weis T, Becker C, Brändle A (2006) Towards a programming paradigm for pervasive applications based on the ambient calculus. In: Workshop on combining theory and systems building in pervasive computing
95. Wessel M (2001) Obstacles on the way to qualitative spatial reasoning with description logics: some undecidability results. Descrip Logics 49
96. Whitehead AN, Russell B (1912) Principia mathematica. University Press,
97. Winfield AF, Nembrini J (2006) Safety in numbers: fault-tolerance in robot swarms. Int J Modell Identif Control 1(1):30–37
98. Wooldridge M (1997) Agent-based software engineering. IEE Proc Softw 144(1):26–37