# Towards Trusting Autonomous Systems

Michael Winikoff(✉)

Department of Information Science, University of Otago,
Dunedin, New Zealand
michael.winikoff@otago.ac.nz

**Abstract.** Autonomous systems are rapidly transitioning from labs into our lives. A crucial question concerns trust: in what situations will we (appropriately) trust such systems? This paper proposes three necessary prerequisites for trust. The three prerequisites are defined, motivated, and related to each other. We then consider how to realise the prerequisites. This paper aims to articulate a research agenda, and although it provides suggestions for approaches to take and directions for future work, it contains more questions than answers.

## 1 Introduction

The past few years have witnessed the rapid emergence of autonomous systems in our lives. Whether in the form of self-driving cars on the road, Unmanned Aerial Vehicles (UAVs) in the skies, or other, less media-grabbing forms, autonomous systems have recently been transitioning from labs and into our lives at a rapid pace.

A crucial question that needs to be answered before deploying autonomous systems is that of *trust*: to what extent are we comfortable with trusting software to make decisions, and to act on these decisions, without intervening human approval?

This paper explores the question of trust of autonomous systems. Specifically, it seeks to answer the question:

> ***In what situations will humans (appropriately) trust autonomous systems?***

In other words, assume that we are dealing with a specific problem and its context, where the context includes such things as the potential consequences (safety, social, etc.) of the system's behaviour. We then seek to know what prerequisites must hold in order for people to be able to develop an appropriate level of trust in a given autonomous system that solves the specific problem. By "appropriate level of trust" we mean that a system that is worthy of being trusted becomes trusted, but a system that is not worthy of trust becomes untrusted.

We consider the question of trust from the viewpoint of individual people. We choose to adopt this lens, rather than, say, considering the viewpoint of society as a whole, for a number of reasons. Firstly, individual trust is crucial:


A. El Fallah-Seghrouchni et al. (Eds.): EMAS 2017, LNAI 10738, pp. 3–20, 2018.
https://doi.org/10.1007/978-3-319-91899-0_1

the viewpoint and policies of a society are clearly based on the viewpoints of the individuals in the society[1]. Secondly, individuals are more familiar to us, and hence easier to analyse. Finally, and most importantly, we can study individual humans through various experiments (e.g. surveys). This allows us to seek to answer the question of the prerequisites for trust using experimental methods (e.g. social science, marketing, psychology).

Before proceeding to explore the prerequisites for trust, we need to briefly clarify what this paper is *not* about, and indicate the assumptions that we are making. This paper is about trusting autonomous systems (i.e. systems empowered to make decisions and act on them). Although autonomous systems often use Artificial Intelligence (AI) techniques, they are not required to be intelligent in a general sense. Thus this paper is *not* about the issues associated with trusting human-level AI, nor is it about issues relating to hypothetical superintelligence [31]. This paper is also not about the broader social consequences of deploying autonomous systems. For example, the impact of AI and automation on the patterns and nature of employment [4,12,46,47]. These are important issues, and they do affect the extent to which a society will allow autonomous systems to be deployed. However, they are out of scope for this paper, since they require social rather than technological solutions.

We make two assumptions. Firstly, we assume that we are dealing with systems where the use of autonomy is acceptable. There are some systems where human involvement in decision making is essential. For example, an autonomous system that handed down prison sentences instead of a human judge may not be socially acceptable. There is also a strong case for banning the development of autonomous weapons[2]. We do note that cases where autonomy is unacceptable are not fixed, and may vary as trust develops. For instance, if it is shown that software systems are able to make more consistent and less biased decisions than human judges, then it may become acceptable to have autonomous software judges in some situations. Secondly, in this paper we do not consider systems that learn and change over time. Learning systems pose additional challenges, including the potential inadequacy of design-time verification, and dealing with emergent bias [5].

The sorts of systems that are within scope include autonomous UAVs, self-driving cars, robots (e.g. nursebots), and non-embodied decision making software such as personal agents and smart homes.

This paper is a "blue sky" paper in that it doesn't provide research results. Instead, it seeks to pose challenges, and articulate a research agenda. The paper does provide some answers in the form of suggestions for how to proceed to address the challenges, but largely it provides questions, not answers.

[1] Although not all individual viewpoints receive equal prominence, which can lead to government policies being out of step with the desires of the population.

[2] http://futureoflife.org/open-letter-autonomous-weapons/.

### 1.1 Related Work

Whilst there is considerable literature devoted to the fashionable question of trusting human-level or super-intelligent AI, there is considerably less literature devoted to the more mundane, but immediate, issue of trusting autonomous (but less intelligent) systems.

Fisher *et al.* [23] consider trust in driverless cars. Like us, they flag legal issues and the importance of verification. This paper differs from their work in considering legal factors in a broader context of *recourse* (where legal recourse is only one of a range of options), and in considering additional factors relating to formal verification. We also posit that *explanation* is important to trust. On the other hand, they also consider human factors, such as driver attention, which are relevant for cars that have partial autonomy, where the human driver needs to be ready to take back control in certain situations.

Helle *et al.* [28] consider, more narrowly, challenges in testing autonomous systems. They also reach the conclusion that formal verification is required, and, like the earlier work of Fisher *et al.* [19,22,23], propose verifying the decision making process in isolation. However, they also highlight the need to do complete system testing to ensure that the system works in a real environment. Where extensive real-world testing is impractical, they highlight *virtual testing* (with simulations) as an approach that can help. Helle *et al.* also have other recommendations that concern testing, such as using models, testing early and continuously, and automating test generation.

A recent Harvard Business Review article [5] argued that "*Trust of AI systems will be earned over time, just as in any personal relationship. Put simply, we trust things that behave as we expect them to*". The article went on to highlight two key requirements for trust: *bias*, and more generally *algorithmic accountability*, and *ethical systems*. They argue that for AI to be trusted, there need to be mechanisms for dealing with bias (detecting and mitigating). More relevant to this paper, they go on to argue that bias is a specific aspect of the broader issue of algorithmic accountability, and they argue that "*AI systems must be able to explain how and why they arrived at a particular conclusion so that a human can evaluate the system's rationale*". They further propose that this explanation should be in the form of an interactive dialogue. They also argue that AI systems should include explicit representation and rules that embody ethical reasoning (see Sect. 3).

Abbass *et al.* [1] discuss the relationship between trust and autonomy, considering high-level definitions of concepts such as trust. The paper does not provide clear answers to what is required for trust. Similarly, a meta-analysis of literature on factors affecting trust in human-robot interaction [26] found that the most important factors affecting trust related to the *performance* of the robot (e.g. behaviour, predictability, reliability). However, they did not provide a clear picture of which specific factors, and also noted that further work was required, since some factors were not adequately investigated in the literature.

In parallel with the original version of this paper being written, a report on Ethically Aligned Design [45] was being developed. This report is broader in scope than this paper, but provides independent support for the points made here.

Finally, there is also a body of work on computational mechanisms to make recommendations, and to manage reputation and trust between software agents (e.g. [36,38]). However, this work focuses on trust of autonomous systems *by other software*, rather than by humans. This makes it of limited relevance, since it does not consider the complex psychological and social factors that inform human trust. A human does not decide to trust a system using just a simple calculation based on the history and evidenced reliability of the system in question.

The remainder of this paper is structured as follows. Section 2 introduces, defines, and motivates the three prerequisites that we identify. Section 3 introduces a fourth element (representing human values and using them in a reasoning process), that we do not consider essential, but that supports the prerequisites. Sections 4 and 5 discuss how to tackle the prerequisite of being able to explain decisions, and verification & validation, respectively. We conclude in Sect. 6.

## 2 Prerequisites to Trust

We propose that there are three required prerequisites to (appropriately) trusting an autonomous system:

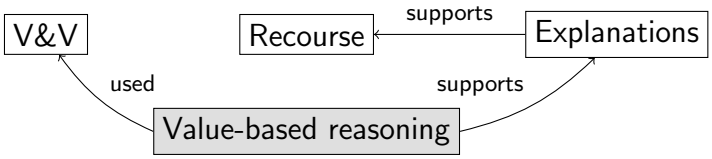
1. a social framework that provides **recourse**, should the autonomous system make a decision that has negative consequences for a person;
2. the system's ability to provide **explanations** of its behaviour, i.e. why it made a particular decision; and
3. **verification & validation of the system**, to provide assurance that the system satisfies key behavioural properties in all situations.

However, we do not claim that these three prerequisites are *sufficient*. We do argue that all three are *necessary*, but it may be that other prerequisites are also necessary. Identifying other prerequisites to trust is therefore an important part of answering the key question posed in Sect. 1.

A key message of this paper is that answering the key question requires a broad programme of research that spans technological sub-questions (e.g. formal verification, explanation) as well as social science sub-questions (e.g. when would humans trust autonomous software, what sort of explanations are helpful), and psychological sub-questions (e.g. how is trust affected by anthropomorphism, and how do characteristics of software affect the extent to which it is ascribed human characteristics).

The remainder of this section briefly outlines the three prerequisites. For each prerequisite we briefly define *what* it is, and motivate the need for that prerequisite ("*why*"). We also draw out the relationships between the three prerequisites (summarised in the diagram below). The subsequent sections consider for each prerequisite *how* that prerequisite might be addressed. Note that for recourse we only discuss "what" and "why" in this section, not "how" in a subsequent

section. This is because the "how" is a social and legal question, and is out of scope for this paper. On the other hand, Sect. 3 discusses value-based reasoning, which is not an essential prerequisite (hence not in this section), but which can support both verification & validation, and explanations.




### 2.1 Recourse: Law and Social Frameworks

We begin with the notion of *recourse*. In a sense, this prerequisite provides a safety net. We know that no person or system is perfect, and that even given a best possible set of practices in developing an autonomous system, it will have a non-zero rate of failure. The notion of recourse is that if an autonomous system does malfunction, that there is some way to be compensated for the negative consequences. We therefore argue that recourse is a necessary prerequisite to trust because it supports trusting a system that is less than 100% perfect, and in practice no system is 100% perfect.

Although the term "recourse" may suggest a mechanism where an affected individual uses the legal system to obtain compensation from another "person" (for autonomous systems, likely the corporation that developed the system), there are other possible social mechanisms that could be used, such as following an insurance model. For example, a form of "autonomous cars insurance" could cover people (pedestrians, cyclists, passengers, and other drivers) in the event that an autonomous vehicle malfunctioned in a way that caused harm. This insurance would ideally cover all people, and there are various models for universal insurance that could be used. For instance, New Zealand has a national comprehensive insurance scheme that automatically provides all residents and visitors with insurance for personal injury (www.acc.co.nz).

Being able to establish a justification for compensation, be it via legal proceedings or as some sort of insurance claim, would require that autonomous systems record enough information to permit audits to be undertaken, and the cause of harm identified. The ability of an autonomous system to explain why it made a decision can therefore support the process of seeking recourse by providing (part of) the evidence for harm.

While the existence of a recourse mechanism is identified as a prerequisite for trust of autonomous systems, this area is not a focus of this paper, and we do not discuss it further. More broadly, but also out of scope for this paper, are issues relating to governance, regulation, and certification.

### 2.2 Explanations

> "*... for users of care or domestic robots a why-did-you-do-that button which, when pressed, causes the robot to explain the action it just took*" [45, p. 20]

A second prerequisite that we argue is essential to (appropriate) trust is the ability of an autonomous system to explain why it made a decision. Specifically, given a particular decision that has been made, the system is able to be queried, and provide to a human user an explanation for why it made that decision. The explanation needs to be in a form that is comprehensible and accessible, and may be interactive (i.e. take the form of a dialogue, rather than a single query followed by a complex answer).

There is a range of work, conducted in the setting of expert systems, rather than autonomous systems, that considers what is required for experts to trust systems. This work highlights explanation as an important factor in trust. For example, Teach and Shortliffe [44] considered attitudes of physicians (medical practitioners) towards decision support systems, including exploring the functionality and features that such systems would require in order to be acceptable to physicians. They noted (all emphasis is in the original) that

> "An ability of a system to explain its advice was thought to be its most important attribute. Second in importance was the ability of a system to understand and update its own knowledge base. ... Physicians did not think that a system has to display either perfect diagnostic accuracy or perfect treatment planning to be acceptable" (p. 550)

They go on to recommend (p. 556) that researchers should:

> "Concentrate some of the research effort on *enhancing the interactive capabilities* of the expert system. The more natural these capabilities, the more likely that the system will be used. At least four features appear to be highly desirable:
> (a) *Explanation.* The system should be able to justify its advice in terms that are understandable and persuasive. ...
> (b) *Common sense.* The system should "seem reasonable" as it progresses through a problem-solving session. Some researchers argue that the operation of the program should therefore parallel the physician's reasoning processes as much as possible. ...
> (c) *Knowledge representation.* The knowledge in the system should be easy to bring up to date, ...
> (d) *Useability* [sic] ..."

they also recommend (p. 557) that researchers

> "Recognize that *100% accuracy is neither achievable nor expected.* Physicians will accept a system that functions at the same level as a human expert so long as the interactive capabilities noted above are a component of the consultative process."

In other words, the system always being right was seen by physicians as being less important, whereas the system being able to be understood was more important.

Stormont [43] considers trust of autonomous systems in hazardous environments (e.g. disaster zone rescue). He notes that while reliability is important, "*a more important reason for lacking confidence may be the unpredictability of autonomous systems*" [43, p. 29]. In other words, autonomous software can sometimes do unexpected things. This can be a good thing: in some cases a software system may be able to find a good solution that is not obvious to a human. We argue that this supports the need for explanations: if a system is able to behave in a way that doesn't obviously make sense to a human, but is nonetheless correct, then in order for the system to be appropriately trusted, it needs to be able to explain why it made its decisions. These explanations allow humans to understand and learn to trust a system that performs well. A difference between Stormont and Teach & Shortliffe is that the latter argue for the system to mirror human decision-making in order to be comprehensible (point (b) quoted above), whereas Stormont sees the benefit of allowing software to find solutions that may not be obvious to humans.

As noted earlier, providing explanations can support the process of building a case for compensation. The provision of explanations can benefit from using value-based reasoning (see Sect. 3).

### 2.3 Verification and Validation (V&V)

> "*It is possible to develop systems having high levels of autonomy, but it is the lack of suitable V&V methods that prevents all but relatively low levels of autonomy from being certified for use*" [15, p. ix].

Before deploying any software system, we need to have confidence that the system will function correctly. The strength of the confidence required depends on the consequences of the system malfunctioning. For non-safety-critical software, this confidence is obtained by software testing. However, autonomous systems can exhibit complex behaviour that makes it infeasible to obtain confidence in a system via testing [51,53]. This therefore necessitates the use of formal methods as part of the design process.

While there may be situations where humans are willing to trust their lives to systems that have not been adequately verified, we argue that this is a case of excessive, and inappropriate, trust. If a system can potentially make a decision that, knowingly, results in harm to a human, then we should have strong assurance that this either does not occur, or occurs only under particular conditions that are well understood, and considered acceptable. The need for confidence in a system's correct functioning, and, for autonomous systems, the need to use formal methods, has been well-recognised in the literature (e.g. [15,17,23,28]).

## 3 Value-Based Reasoning

We have argued that recourse, explanations, and V&V are prerequisites that are essential (but not necessarily sufficient) to having appropriate human trust in

autonomous systems. In addition we now propose a fourth element: value-based reasoning. We do not consider value-based reasoning to be an essential prerequisite, but explain below why it may be desirable, and how it supports two of the prerequisites. As noted earlier, a recent HBR article [5] argued that ethics can, and should, be codified and used in reasoning. Similarly, van Riemsdijk *et al.* [40] had earlier argued that socially situated autonomous systems (e.g. personal assistants and smart homes) should represent and use norms to reason about situations where norms may conflict.

By *value-based reasoning* we mean that the autonomous system includes a representation for human values (e.g. not harming humans), and that it is able to conduct reasoning using these human values in order to make decisions, where relevant. One (widely discussed) example is the use of ethical reasoning in autonomous vehicles [6]. However, using human values in the reasoning process can be beneficial not just in life-and-death situations. Consider a system that takes care of an aged person, perhaps with dementia or Alzheimer's disease. There are situations where competing options may be resolved by considering human values, such as autonomy vs. safety, or privacy vs. health. Perhaps the elderly person wants to go for a walk (which is both healthy, and is aligned with their desire for autonomy), but for safety reasons they should not be permitted to leave the house alone. In this example, the system needs to decide whether to allow the person it is caring for to leave the house, and, if so, what other actions may need to be taken. The key point is that in different situations, different decisions make sense. For instance, if a person is at a high risk of becoming lost, then despite their desire for autonomy, and the health benefits of walking, they should either be prevented from leaving, or arrangements should be made for them to be accompanied.

Value-based reasoning can be used to support two of the prerequisites. Firstly, we conjecture that the existence of a computational model of relevant human values could be used as a basis for providing higher level, more human-oriented, explanations of decisions. Secondly, in some situations, having an explicit model of values (or, perhaps more specifically, ethics) would be required to be able to verify certain aspects of an autonomous system's behaviour, for instance that the system's reasoning and decisions take certain ethical considerations into account. For example, a recent paper by Dennis *et al.* [20] proposes to use formal methods to show that an autonomous system behaves *ethically*, i.e. that it only selects a plan that violates an ethical principle when the other options are worse. For instance, a UAV may select a plan that involves colliding with airport hardware (violating a principle of not damaging property) only in a situation where the other plans involve worse violations (e.g. collision with people or manned aircraft).

In some situations doing value-based reasoning will not be feasible. For instance, in a real autonomous vehicle, the combination of unreliable and noisy sensor data, unreliable actuators, the inherent unpredictability of consequences (partly due to other parties acting concurrently), and the lack of time to reason, means that in all likelihood, an autonomous vehicle will not be able to make

decisions using utilitarian ethical reasoning. On the other hand, there may be applications (e.g. military) where software being able to conduct ethical reasoning would be considered to be very important [2].

Key research questions to consider in order to achieve value-based reasoning are:

- What values should be represented, and at what level of abstraction?
- How should reasoning about values be done, and in particular, how does this interact with the existing decision making process?
- How can values be utilised in providing explanations? And are such explanations more accessible to people than explanations that do not incorporate values?
- Given an agent with value-based reasoning, what sort of verification can be done that makes use of the existence of values?

Cranefield *et al.* [14] present a computational instantiation of value-based reasoning that provides initial answers to some of these questions. Specifically, they present an extension of a BDI language that takes simply-represented values into account when selecting between available plans to achieve a given goal.

## 4 Explanations

As noted earlier, an important element in trust is being able to understand why a system made certain decisions, leading to its behaviour. Therefore, there is a need to develop mechanisms for an autonomous system to explain why it chose and enacted a particular course of action.

Since explanations can be complex (e.g. "I performed action $a_1$ because I was trying to achieve the sub-goal $g_2$ and I believed that $b_3$ held ..."), in order to be comprehensible, they need to be provided in a form that facilitates navigation of the explanation. This navigation can be in the form of a user interface that allows the explanation to be explored, or by having the explanations take the form of a dialog with the system (e.g. [13]).

Although there has been earlier work on explaining expert system recommendations, which may be useful as a source of ideas, the problem here is different in that we are explaining a *course of action* (taken over time, in an environment), not a (static) recommendation. Consequently, we are not dealing with deductive reasoning rules (as in expert systems), but with *practical* reasoning (although more likely to focus on means-end-reasoning than on deliberation, i.e. the focus is more likely to be on *achieving* rather than *selecting* goals).

Mechanisms for providing explanations obviously depend on the internal reasoning mechanism used and the representation of practical reasoning knowledge. For instance, Broekens *et al.* [11] assume a representation in terms of a hierarchy of goals, also including beliefs and actions. If it turns out that explanations in terms of goals and beliefs are natural for humans to understand (which we might expect to be the case, since we naturally use "folk psychology" to reason about the behaviour of other humans), then that may imply that we want to

have the autonomous system represent its knowledge in the form of plans to achieve its goals. However, it may also be possible to explain decisions made by a non-goal-based reasoning process, by using a separate representation in terms of goals. Although this would mean that the agent reasoning can use any mechanism and representation, it introduces the potential for the actual reasoning and the goal representation used for explaining to differ. Finally, it is also possible to provide explanations based solely on the observed behaviour of the system (i.e. without having an accessible or useful internal representation of the system's decision making process), but this approach has drawbacks due to the limited information available [25].

There has been some work on mechanisms for autonomous systems to provide explanations (e.g. [11,27,54]), but more work is needed. In particular, it is important for future work to take into account insights from the social sciences [35]. Although there may well be differences between how humans explain behaviour and how we want autonomous systems to explain their behaviours, it makes sense to at least be aware of the extensive body of work on how humans explain behaviour, e.g. [34].

Harbers [27] assumes that there is a goal tree that captures the agent's reasoning. The goal tree relates each goal to its sub-goals, and is indicated as being an "or" decomposition, "all" decomposition, "seq" (sequence) decomposition, or "if" decomposition. Each goal to sub-goal relationship is mediated by an optional belief that allows the sub-goal to be adopted (e.g. the sub-goal "prepare the fire extinction" is mediated by the belief "at incident location" [27, Fig. 4.4]). The leaves of the tree are actions. The goal-tree is the basis for the implementation of the agent (using the 2APL agent programming language). A number of different explanation rules are considered. For instance, explaining an action in terms of its parent goal, or in terms of its grandparent goal, or in terms of beliefs that allowed the action to be performed, or in terms of the *next* action to be done (e.g. "I did action $a_1$ so I could then subsequently do action $a_2$"). Harbers reports on an experiment (with human subjects) using a simple fire fighting scenario, where the tree of goals contains 26 goals, and where the agent executes a sequence of 16 actions. The experiment aims to find out which explanation rules are preferred. She finds that in general there is not a consistent preference: for some actions a particular rule (e.g. the parent goal) is the commonly preferred explanation, whereas for other actions, the next action is the commonly preferred explanation. Harbers proposed that an action ought to be explained by the combination of its parent goal and the belief that allowed the action to be performed (which was not an explanation rule used in her experiment), but also defined two exceptional situations for which different explanations should be used. Broekens *et al.* [11] report on a similar experiment, and also find that there is not a single explanation rule that is the best for all situations.

One characteristic of the rules used by Harbers and by Broekens *et al.* is that they are (intentionally) incomplete: given an action, each rule selects only part of the full explanation. For instance, a rule that explains an action in terms of its parent goal ignores the beliefs that led to that goal being selected. By contrast,

Hindriks [29] defines (informal) rules that yield a more complete explanation. More recently, Winikoff [54] builds on Hindriks' work by systematically deriving formally-defined rules that are then implemented. Winikoff also explicitly defines (but does not prove) a completeness result: that, given their derivation, the rules capture *all* the explanatory factors. However, this work aims to support programmers debugging a system, rather than human end-users trying to understand a system's behaviour (presumably without a detailed understanding of the system's internals!). Additionally, the completeness of the rules comes at a cost: the explanations are larger, and therefore harder to comprehend.

Finally, as mentioned in the previous section, it may be desirable to include human value-based reasoning into the decision process, which then poses the question of how to exploit this in the provision of explanations.

We therefore have the following research questions:

- How can an autonomous system provide explanations of its decisions and actions?
- What forms of explanation are most helpful and understandable? Is it helpful to structure explanations in terms of folk psychology constructs such as goals, plans and beliefs?
- How can explanations be effectively navigated by human users? In what circumstances is it beneficial to provide an explanation in the form of a dialogue?
- What reasoning processes and internal representations facilitate the provision of explanations? Does there need to be some representation of the system's goals?
- What is the tradeoff between using the same representation for both decision making and explanation, as opposed to using a different representation for explanation?
- How well can explanations be provided without a representation for the system's decision making knowledge and process (i.e. based solely on observing the system's behaviour)?
- How can explanations be provided that exploit the presence of representations of human values in the reasoning process? Are such explanations more accessible to people than explanations that do not incorporate human values?

Note that we are assuming a setting where a system deliberates and acts autonomously, and may be required to provide after-the-fact explanations (to help a human understand why it acted in certain ways, or to provide evidence for compensation, in the event of harm). However, another setting to be considered is where autonomous software works closely with humans, as part of a mixed team. In this sort of setting it is important not just to be able to explain after the fact, but also to provide updates during execution so that team members (both human and software) have sufficient awareness of what other team members are doing, or are intending to do. Doing this effectively is a challenge, since a balance needs to be struck between sharing too little (leading to inadequate awareness, and potential coordination issues) or too much (leading to overloading human team members with too much information). There has been some work that has explored this issue (e.g. [33,42]). However, this is not related to trusting

autonomous systems in a general setting, but to the effectiveness of working with software in mixed human-agent teams.

## 5 Verification and Validation

We have already noted that we need to have a way of obtaining assurance that an autonomous software system will behave appropriately, and that obtaining this assurance will require formal methods. We now consider the challenges involved in doing so, highlight some approaches, and pose research questions.

Work on techniques for verifying autonomous systems goes back at least 15 years (e.g. [56]). However, current state-of-the-art techniques are still only able to verify small systems [7,17,19,21,22,37,56]. Given the work that has been done, and the foundations provided by earlier work on verification of (non-autonomous) software, continuing to improve verification techniques is important future work, and eventually the techniques will be able to deal with realistically-sized systems. A number of ideas have been proposed that reduce the complexity of verification.

Firstly, Fisher *et al.* [19,22,23] have proposed to reduce the complexity of verifying autonomous systems by focussing on verifying the system's decision making in isolation. The correct functioning of sensors and effectors is assessed separately, which requires end-to-end testing, possibly involving simulation [28]. Verifying decision making not only improves efficiency, but also allows verification to consider whether a bad decision is made in error (e.g. due to missing information), or intentionally, which is an important distinction [3,32].

Secondly, Bordini *et al.* [8] have proposed using slicing to reduce the complexity of verification. The basic idea is that given a particular property to be verified, instead of verifying the property against the agent program, one first generates a specialised version of the program that has been "sliced" to remove anything that does not affect the truth of the property being verified. The property is then verified against the "sliced" program. There is scope for further work, including considering other forms of program transformation prior to verification. For instance, there is a body of work on partial evaluation[3] [30] that may be applicable.

Thirdly, there are various approaches that reduce the complexity of verifying a large system by verifying parts of the system separately, and then combining the verifications. One well-known approach uses assume-guarantee rules (e.g. [24]). It would be useful to consider adapting this approach for use with autonomous systems. In the case that the system's decision making is represented in terms of a hierarchy of goals, it may be that sub-goals provide a natural point of modularity, i.e. that one can verify sub-goals in isolation, and then combine the results.

[3] Partial evaluation is the process of taking a program and some of its inputs and producing a specialised program that is able to accept the remaining inputs and compute the same results as the original program, but more efficiently.

In addition to these research strands, which aim to make verification practical for real agent programs, there is another issue to consider: ***where does the formal specification come from?*** Verification takes a property and checks whether this property holds, but in order to be confident that a system (autonomous or not) will behave appropriately, we need to be confident that the collection of properties being verified adequately capture the requirements for "appropriate behaviour" [41].

In some cases there may be existing laws or guidelines that adequately specify what is "appropriate behaviour" for a given context, for instance, the Rules of the Air[4] describe how a pilot must behave in certain situations[5], and can be used as a source for properties to be verified [50]. However, sometimes such guidelines do not exist, or they may be incomplete. For example, important constraints may not be explicitly stated, if they are "obvious" to humans, such as that a pilot should not accelerate in a way that exceeds human tolerances.

We therefore propose the development of a process for systematically deriving the properties to be verified from the system's design and a collection of high-level generic properties (e.g. "cause no harm", "always ensure others are aware of your intentions" - important for predictability). We assume that the autonomous software is developed using a well-defined methodology [55] which uses design models (e.g. goal model, interaction protocols) as "stepping stones" in the development process that results in software. The properties to be verified ("Formal Specification" in Fig. 1) are derived by taking (1) a collection of generic high-level properties which apply to any system, expressed in an appropriate notation, and applying (2) a well-defined process for deriving a fault model [48] from the high-level properties and the system's design models. We then need a well-defined process (3) for deriving the required formal specification properties from the fault model.

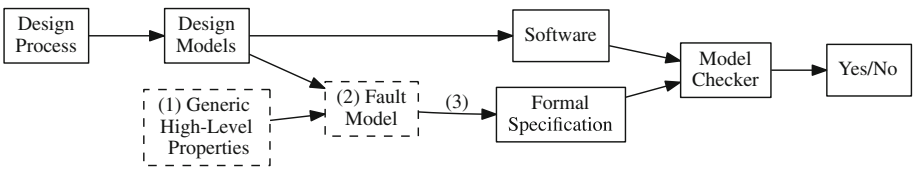



**Fig. 1.** Proposed process for systematically deriving properties to be verified

For instance, given a high-level property of "not harming people", one might examine the system's design (along with information on its environment, and domain knowledge regarding the consequences of various actions) to derive a fault model that captures the specific ways in which the system's decisions might

[4] https://www.easa.europa.eu/document-library/regulations/commission-implementing-regulation-eu-no-9232012.

[5] For example, that when two planes are approaching head on and there is a danger of collision, that the pilots should both turn to their right.

lead to harming people. As an example, consider a robot assistant ("Care-O-bot") [49] that resides in a home along with an elderly person being cared for. We would consider how harm to the person being cared for can occur in relation to the system's requirements. Since the system is responsible for managing medication, we might identify that administering medication incorrectly, or failing to remind the person to take their medication, are possible ways in which harm can be caused. Similarly, the system failing to promptly seek help in the event of an accident, adverse medical event or other emergency (e.g. fire, earthquake) would be another way in which the person being cared for could be harmed. This analysis process *contextualises* the threats to the high-level properties in the circumstances of the system, and results in a fault model, which captures specific ways in which the system at hand might violate the high-level properties. We then need to have a way of deriving from the fault model specific properties to be verified, in an appropriate formal notation. The collection of high-level properties (1), process for deriving a fault model for a given system (2), and method for deriving formal properties from the fault model (3) all need to be developed, along with appropriate notations.

Finally, as noted in the previous section, the internal reasoning process and associated representation matters. What sort of reasoning mechanisms and knowledge representations should be used to facilitate verification? Fisher *et al.* [22] have argued, in the context of verifying autonomous systems, that the systems should be developed in terms of beliefs, goals, plans, and actions, i.e. using a BDI (Belief Desire Intention) [39] agent-oriented programming language such as Gwendolyn[6] [18].

We therefore have the following research questions:

- How can agent program slicing be improved? What other forms of program transformation (e.g. partial evaluation) could be used to reduce the complexity of verification?
- Can the decision making process for a given autonomous system be verified in a modular way, perhaps using assume-guarantee reasoning (e.g. [24])? If so, can goals and sub-goals be used as a natural point to divide into independent components for verification?
- How can the properties to be verified be systematically derived?
- Should autonomous agents be programmed using a notation that supports representations for goals, beliefs, plans, and actions? If so, are existing BDI agent programming languages adequate, or do they need to be extended, restricted, or otherwise modified?

## 6 Discussion

In this paper we have considered the issue of *trust*, specifically posing the question: "*In what situations will humans (appropriately) trust autonomous systems?*"

[6] Other prominent BDI agent-oriented programming languages include Jason [9], Jadex [10], JACK [52], and 2APL [16].

We argued that there are three prerequisites that are essential in order for appropriate trust in autonomous systems to be realised: having assurance that the system's behaviour is appropriate (obtained through verification & validation), having the system be able to explain and justify its decisions in a way that is understandable, and the existence of social frameworks that provide for compensation in the event that an autonomous system's decisions do lead to harm ("recourse"). We also discussed using computational representations of human values as part of the decision making process in autonomous software, and how this can support the other prerequisites.

However, while we have argued that these three prerequisites are necessary, we are not in a position to claim that they are sufficient. Therefore, an overarching piece of research is to investigate experimentally the extent to which humans are willing to trust various autonomous systems given the prerequisites, and, especially, where people are not willing to trust a system, to identify what additional prerequisite might be required in order to enable (appropriate) trust.

We have discussed paths towards achieving the two technical prerequisites, and posed specific research questions, thereby defining a research agenda. There is much work to be done, and I hope that this paper will help to spur further discussion on what is needed to have appropriate trust in autonomous systems, and encourage researchers to work on the problems and questions articulated.

**Acknowledgements.** I would like to thank the anonymous reviewers for their comments, and Michael Fisher for discussions and pointers to literature.

## References

1. Abbass, H.A., Petraki, E., Merrick, K., Harvey, J., Barlow, M.: Trusted autonomy and cognitive cyber symbiosis: open challenges. Cogn. Comput. **8**(3), 385–408 (2016). https://doi.org/10.1007/s12559-015-9365-5
2. Arkin, R.C., Ulam, P., Wagner, A.R.: Moral decision making in autonomous systems: enforcement, moral emotions, dignity, trust, and deception. Proc. IEEE **100**(3), 571–589 (2012). https://doi.org/10.1109/JPROC.2011.2173265
3. Atkinson, D.J., Clark, M.H.: Autonomous agents and human interpersonal trust: can we engineer a human-machine social interface for trust? In: Trust and Autonomous Systems: Papers from the 2013 AAAI Spring Symposium, pp. 2–7 (2013)
4. Autor, D.H.: Why are there still so many jobs? The history and future of workplace automation. J. Econ. Perspect. **29**(3), 3–30 (2015)
5. Banavar, G.: What It Will Take for Us to Trust AI. Harvard Business Review, November 2016. https://hbr.org/2016/11/what-it-will-take-for-us-to-trust-ai
6. Bonnefon, J.F., Shariff, A., Rahwan, I.: The social dilemma of autonomous vehicles. Science **352**(6293), 1573–1576 (2016). https://doi.org/10.1126/science.aaf2654
7. Bordini, R.H., Fisher, M., Pardavila, C., Wooldridge, M.: Model checking AgentSpeak. In: Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 409–416. ACM Press (2003)
8. Bordini, R.H., Fisher, M., Wooldridge, M., Visser, W.: Property-based slicing for agent verification. J. Log. Comput. **19**(6), 1385–1425 (2009). https://doi.org/10.1093/logcom/exp029

9. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming Multi-agent Systems in AgentSpeak Using Jason. Wiley (2007). ISBN 0470029005
10. Braubach, L., Pokahr, A., Lamersdorf, W.: Jadex: a BDI-agent system combining middleware and reasoning. In: Unland, R., Calisti, M., Klusch, M. (eds.) Software Agent-Based Applications, Platforms and Development Kits, pp. 143–168. Birkhäuser, Basel (2005). https://doi.org/10.1007/3-7643-7348-2_7
11. Broekens, J., Harbers, M., Hindriks, K.V., van den Bosch, K., Jonker, C.M., Meyer, J.C.: Do you get it? User-evaluated explainable BDI agents. In: Dix, J., Witteveen, C. (eds.) MATES 2010. LNCS, vol. 6251, pp. 28–39. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16178-0_5
12. Brynjolfsson, E., McAfee, A.: The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies. W.W. Norton & Company, New York (2014)
13. Caminada, M.W.A., Kutlák, R., Oren, N., Vasconcelos, W.W.: Scrutable plan enactment via argumentation and natural language generation (demonstration). In: Bazzan, A.L.C., Huhns, M.N., Lomuscio, A., Scerri, P. (eds.) International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), pp. 1625–1626. IFAAMAS (2014). http://dl.acm.org/citation.cfm?id=2616095
14. Cranefield, S., Winikoff, M., Dignum, V., Dignum, F.: No pizza for you: value-based plan selection in BDI agents. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, pp. 178–184 (2017). https://doi.org/10.24963/ijcai.2017/26
15. Dahm, W.J.: Technology Horizons: A Vision for Air Force Science & Technology During 2010–2030. Technical report, AF/ST-TR-10-01-PR, US Air Force (2010)
16. Dastani, M.: 2APL: a practical agent programming language. Auton. Agents Multi Agent Syst. **16**(3), 214–248 (2008)
17. Dastani, M., Hindriks, K.V., Meyer, J.J.C. (eds.): Specification and Verification of Multi-agent Systems. Springer, Heidelberg (2010). https://doi.org/10.1007/978-1-4419-6984-2
18. Dennis, L.A., Farwer, B.: Gwendolen: a BDI language for verifiable agents. In: Löwe, B. (ed.) AISB 2008 Workshop on Logic and the Simulation of Interaction and Reasoning (2008)
19. Dennis, L.A., Fisher, M., Lincoln, N.K., Lisitsa, A., Veres, S.M.: Practical verification of decision-making in agent-based autonomous systems. Autom. Softw. Eng. **23**(3), 305–359 (2016). https://doi.org/10.1007/s10515-014-0168-9
20. Dennis, L.A., Fisher, M., Slavkovik, M., Webster, M.: Formal verification of ethical choices in autonomous systems. Robot. Auton. Syst. **77**, 1–14 (2016). https://doi.org/10.1016/j.robot.2015.11.012
21. Dennis, L.A., Fisher, M., Webster, M.P., Bordini, R.H.: Model checking agent programming languages. Autom. Softw. Eng. J. **19**(1), 3–63 (2012). https://doi.org/10.1007/s10515-011-0088-x
22. Fisher, M., Dennis, L., Webster, M.: Verifying autonomous systems. Commun. ACM **56**(9), 84–93 (2013)
23. Fisher, M., Reed, N., Savirimuthu, J.: Misplaced trust? In: Engineering and Technology Reference. The Institution of Engineering and Technology (2015). https://doi.org/10.1049/etr.2014.0054
24. Gheorghiu Bobaru, M., Păsăreanu, C.S., Giannakopoulou, D.: Automated assume-guarantee reasoning by abstraction refinement. In: Gupta, A., Malik, S. (eds.) CAV 2008. LNCS, vol. 5123, pp. 135–148. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70545-1_14

25. Gomboc, D., Solomon, S., Core, M., Lane, H.C., van Lent, M.: Design recommendations to support automated explanation and tutoring. In: Conference on Behavior Representation in Modeling and Simulation (BRIMS) (2005). http://ict.usc.edu/pubs/Design%20Recommendations%20to%20Support%20Automated%20Explanation%20and%20Tutoring.pdf
26. Hancock, P.A., Billings, D.R., Schaefer, K.E., Chen, J.Y.C., de Visser, E.J., Parasuraman, R.: A meta-analysis of factors affecting trust in human-robot interaction. Hum. Factors **53**(5), 517–527 (2011). https://doi.org/10.1177/0018720811417254
27. Harbers, M.: Explaining Agent Behavior in Virtual Training. SIKS dissertation series no. 2011–35, SIKS (Dutch Research School for Information and Knowledge Systems) (2011)
28. Helle, P., Schamai, W., Strobel, C.: Testing of autonomous systems - challenges and current state-of-the-art. In: 26th Annual INCOSE International Symposium (2016)
29. Hindriks, K.V.: Debugging is explaining. In: Rahwan, I., Wobcke, W., Sen, S., Sugawara, T. (eds.) PRIMA 2012. LNCS (LNAI), vol. 7455, pp. 31–45. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32729-2_3
30. Jones, N.D.: An introduction to partial evaluation. ACM Comput. Surv. **28**(3), 480–503 (1996). https://doi.org/10.1145/243439.243447
31. Kaplan, J.: Artificial intelligence: think again. Commun. ACM **60**(1), 36–38 (2017). https://doi.org/10.1145/2950039
32. Lee, J.D., See, K.A.: Trust in automation: designing for appropriate reliance. Human Factors **46**(1), 50–80 (2004)
33. Li, S., Sun, W., Miller, T.: Communication in human-agent teams for tasks with joint action. In: Dignum, V., Noriega, P., Sensoy, M., Sichman, J.S.S. (eds.) COIN 2015. LNCS (LNAI), vol. 9628, pp. 224–241. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42691-4_13
34. Malle, B.F.: How the Mind Explains Behavior. MIT Press, Cambridge (2004). ISBN 9780262134453
35. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. CoRR abs/1706.07269 (2017)
36. Pinyol, I., Sabater-Mir, J.: Computational trust and reputation models for open multi-agent systems: a review. Artif. Intell. Rev. **40**(1), 1–25 (2013). https://doi.org/10.1007/s10462-011-9277-z
37. Raimondi, F., Lomuscio, A.: Automatic verification of multi-agent systems by model checking via ordered binary decision diagrams. J. Appl. Log. **5**(2), 235–251 (2007)
38. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in multi-agent systems. Knowl. Eng. Rev. **19**(1), 1–25 (2004). https://doi.org/10.1017/S0269888904000116
39. Rao, A.S., Georgeff, M.P.: BDI agents: from theory to practice. In: Lesser, V.R., Gasser, L. (eds.) Conference on Multiagent Systems, pp. 312–319. The MIT Press, San Francisco (1995)
40. van Riemsdijk, M.B., Jonker, C.M., Lesser, V.R.: Creating socially adaptive electronic partners: interaction, reasoning and ethical challenges. In: Weiss, G., Yolum, P., Bordini, R.H., Elkind, E. (eds.) Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 1201–1206. ACM (2015). http://dl.acm.org/citation.cfm?id=2773303
41. Rozier, K.Y.: Specification: the biggest bottleneck in formal methods and autonomy. In: Blazy, S., Chechik, M. (eds.) VSTTE 2016. LNCS, vol. 9971, pp. 8–26. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48869-1_2

42. Singh, R., Sonenberg, L., Miller, T.: Communication and sharedmental models for teams performing interdependent tasks. In: Osman, N., Sierra, C. (eds.) AAMAS 2016 Workshops, Best Papers. LNCS/LNAI, vol. 10002, pp. 163–179. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-46882-2_10
43. Stormont, D.P.: Analyzing human trust of autonomous systems in hazardous environments. In: Metzler, T. (ed.) AAAI Workshop on Human Implications of Human-Robot Interaction, pp. 27–32. The AAAI Press, Technical report WS-08-05 (2008). http://www.aaai.org/Library/Workshops/ws08-05.php
44. Teach, R.L., Shortliffe, E.H.: An analysis of physician attitudes regarding computer-based clinical consultation systems. Comput. Biomed. Res. **14**, 542–558 (1981)
45. The IEEE Global Initiative for Ethical Considerations in Artificial Intelligence and Autonomous Systems: Ethically Aligned Design: A Vision For Prioritizing Wellbeing With Artificial Intelligence And Autonomous Systems, Version 1. IEEE (2016). http://standards.ieee.org/develop/indconn/ec/autonomous_systems.html
46. The White House: Artificial Intelligence, Automation, and the Economy, December 2016. https://www.whitehouse.gov/sites/whitehouse.gov/files/documents/Artificial-Intelligence-Automation-Economy.PDF
47. The White House: Preparing for the Future of Artificial Intelligence, October 2016. https://www.whitehouse.gov/sites/default/files/whitehouse_files/microsites/ostp/NSTC/preparing_for_the_future_of_ai.pdf
48. Vesely, W.E., Goldberg, F.F., Roberts, N.H., Haasl, D.F.: Fault tree handbook. Technical report, NUREG-0492, US Nuclear Regulatory Commission, January 1981
49. Webster, M., Dixon, C., Fisher, M., Salem, M., Saunders, J., Koay, K.L., Dautenhahn, K., Saez-Pons, J.: Towards reliable autonomous robotic assistants through formal verification: a case study. IEEE Trans. Hum. Mach. Syst. **46**(2), 186–196 (2016)
50. Webster, M., Cameron, N., Fisher, M., Jump, M.: Generating certification evidence for autonomous unmanned aircraft using model checking and simulation. J. Aerosp. Inf. Syst. **11**(5), 258–279 (2014). https://doi.org/10.2514/1.I010096
51. Winikoff, M., Cranefield, S.: On the testability of BDI agent systems. J. Artif. Intell. Res. (JAIR) **51**, 71–131 (2014). https://doi.org/10.1613/jair.4458
52. Winikoff, M.: JACK$^{TM}$ intelligent agents: an industrial strength platform. In: Bordini, R.H., Dastani, M., Dix, J., Fallah-Seghrouchni, A.E. (eds.) Multi-Agent Programming: Languages, Platforms and Applications, vol. 15, pp. 175–193. Springer, Boston (2005). https://doi.org/10.1007/0-387-26350-0_7
53. Winikoff, M.: How testable are BDI agents? An analysis of branch coverage. In: Osman, N., Sierra, C. (eds.) AAMAS 2016 Workshops, Best Papers. LNCS/LNAI, vol. 10002, pp. 90–106. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46882-2_6
54. Winikoff, M.: Debugging agent programs with "Why?" questions. In: Das, S., Durfee, E., Larson, K., Winikoff, M. (eds.) Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2017)
55. Winikoff, M., Padgham, L.: Agent oriented software engineering. In: Weiß, G. (ed.) Multiagent Systems, Chap. 15, 2 edn., pp. 695–757. MIT Press (2013)
56. Wooldridge, M., Fisher, M., Huget, M.P., Parsons, S.: Model checking multi-agent systems with MABLE. In: Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), pp. 952–959. ACM Press (2002)