

“Dave...I can assure you ...that it’s going to be all right ...” A Definition, Case for, and Survey of Algorithmic Assurances in Human-Autonomy Trust Relationships

BRETT W. ISRAELSEN and NISAR R. AHMED, University of Colorado Boulder, Cooperative Human-Robot Intelligence Laboratory (COHRINT), and Research and Engineering Center for Unmanned Vehicles (RECUV)

People who design, use, and are affected by autonomous artificially intelligent agents want to be able to *trust* such agents—that is, to know that these agents will perform correctly, to understand the reasoning behind their actions, and to know how to use them appropriately. Many techniques have been devised to assess and influence human trust in artificially intelligent agents. However, these approaches are typically ad hoc and have not been formally related to each other or to formal trust models. This article presents a survey of *algorithmic assurances*, i.e., programmed components of agent operation that are expressly designed to calibrate user trust in artificially intelligent agents. Algorithmic assurances are first formally defined and classified from the perspective of formally modeled human-artificially intelligent agent trust relationships. Building on these definitions, a synthesis of research across communities such as machine learning, human-computer interaction, robotics, e-commerce, and others reveals that assurance algorithms naturally fall along a spectrum in terms of their impact on an agent’s core functionality, with seven notable classes ranging from integral assurances (which impact an agent’s core functionality) to supplemental assurances (which have no direct effect on agent performance). Common approaches within each of these classes are identified and discussed; benefits and drawbacks of different approaches are also investigated.

CCS Concepts: • General and reference → Surveys and overviews; • Human-centered computing; User models; User studies; Interactive systems and tools; Interaction design theory, concepts and paradigms; Systems and tools for interaction design; Collaborative and social computing theory, concepts and paradigms; Interaction paradigms; Interaction techniques; Visualization application domains; • Computing methodologies; Artificial intelligence; Machine learning; • Applied computing; Engineering; Psychology; Online shopping; Law; Computing in government;

Additional Key Words and Phrases: Human-computer trust, interpretable machine learning, explainable artificial intelligence, transparency, accountability, fairness, algorithmic assurances

HAL 9000, 2001 *A Space Odyssey*, full quote: “Just what do you think you’re doing, Dave? Dave, I really think I’m entitled to an answer to that question. I know everything hasn’t been quite right with me, but I can assure you now, very confidently, that it’s going to be all right again.”

The authors acknowledge the helpful feedback of Michael Mozer and Eric Frew, as well as that of the reviewers. This work was funded by a research gift from Northrop-Grumman Aerospace Systems and by the Center for Unmanned Aircraft Systems (C-UAS), a National Science Foundation Industry/University Cooperative Research Center (I/UCRC) under NSF Award No. CNS-1650468 along with significant contributions from C-UAS industry members.

Authors’ addresses: B. W. Israelsen and N. R. Ahmed, 1111 Engineering Drive, 429 UCB, ECAE 175, Boulder, CO 80309, USA.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Association for Computing Machinery.

0360-0300/2019/01-ART113 \$15.00

<https://doi.org/10.1145/3267338>

ACM Reference format:

Brett W. Israelsen and Nisar R. Ahmed. 2019. “Dave...I can assure you ...that it’s going to be all right ...” A Definition, Case for, and Survey of Algorithmic Assurances in Human-Autonomy Trust Relationships. *ACM Comput. Surv.* 51, 6, Article 113 (January 2019), 37 pages.

<https://doi.org/10.1145/3267338>

1 INTRODUCTION

Trust plays a key role in interpersonal relationships. For example, a supervisor asks a subordinate to accomplish a task based on several factors that indicate the subordinate can be trusted to do so. Likewise, when using something like an autonomous vehicle, users must trust it appropriately in order to use it properly. With the rapid advancement of artificially intelligent technology and autonomous systems to do tasks that were previously assumed to be too complicated for machines, there is now much discussion in public [33, 122, 139], business [10, 72, 129], and academic settings [18, 38, 79] on how humans can trust said technology—although, the connection to trust is not always made explicit from a technical standpoint. Those who discuss *how* to trust a specific technology are really referring to the need to identify *indicators of the appropriate level of trust*. In other words, it is desirable to *design* capabilities and methods into intelligent technology that help designers, users, and other stakeholders achieve appropriate levels of trust in that technology. These capabilities and methods are collectively referred to as *assurances*. The field of formal Validation and Verification (V&V) also uses the term assurances to refer to structured evidence that indicates whether or not a system is functioning according to *a priori* design specifications [17]. These assurances will be referred to here as “hard assurances.” Hard assurances are often not relatable to users of systems or used to adjust levels of trust with a user in real time but are used for certification and meeting certain qualifications such as safety. This is in contrast to soft assurances, which are meant to affect user trust and *trust-related behaviors*. In this article, “assurances” will refer to soft assurances only.

This survey investigates what assurances an *Artificially Intelligent Agent* (AIA) can provide to a human user in order to affect their trust. The colloquial definitions of appropriate use, assurance, AIA, and trust should suffice for now to give the reader a general idea of the motivation; more formal definitions will be presented in Section 2. Many researchers from different disciplines will potentially be interested in this work, which includes fields like machine learning, artificial intelligence, robotics, and unmanned systems. More broadly, it includes any disciplines that deal in some way with the interface between humans and technology; particularly those who are interested in working with, trusting, interpreting, understanding, and/or regulating AIAs. As such, this article cuts across multiple disciplines and ties together concepts from several important research topics, such as trustworthy and explainable learning and AI, ethical and transparent autonomy, and safety-/user-aware intelligent systems.

Figure 1 is a simple diagram of the trust cycle that exists between a human user and an AIA: user trust is affected by assurances from the AIA, which in turn affects the user’s interaction with the AIA (e.g., to trust AIA with responsibilities or not). To fully understand and appreciate the importance of assurances, one must have a more formal understanding of the components of the trust cycle. This article provides an overview of the trust cycle elements and then turns more focused attention to assurances, surveying related research to date. From this survey, properties and classifications of assurances are defined, and considerations for further research are presented. Some of the novel contributions of this article include: creating a detailed description and definition of assurances in general human-AIA relationships (based in research from several diverse

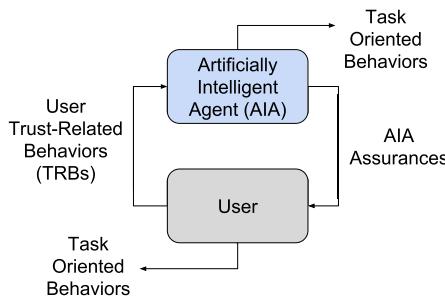


Fig. 1. Simple one-way trust relationship between a human user and an AIA: based on a user's level of trust, they take certain actions (e.g., give AIA commands), which can lead to AIA actions and/or to assurances that affect user trust. Some Task Oriented Behaviors can fall within the trust cycle, but the addition of a separate arrow is meant to encompass any actions that are not strictly trust related.

research communities); making a detailed breakdown of the different components of assurances; and identifying design considerations for implementing assurances at different levels of integration within the AIA. To this end, Section 2 provides definitions for each of the terms. Section 3 presents a more detailed version of Figure 1. We classify existing work into seven categories of integration within the AIA; detailed definitions, discussion, and examples of assurances from these categories help readers to understand how to apply assurances in their specific applications. Finally, recommendations for future work are discussed in Section 4, and conclusions are presented in Section 5.

2 MOTIVATION AND BACKGROUND

The notion of algorithms being used to create assurances for users is *not* new. However, the creation of these assurances (for example, in various engineering disciplines, software, science, economics, and others) has historically been done in an ad hoc manner. The need for designed assurances has grown considerably in recent years, as the advanced capabilities of intelligent systems have become more difficult to comprehend and predict [34, 50, 86, 142]. Advanced intelligent systems share capabilities with less-advanced counterparts, but generally possess much more delegated responsibility, autonomous functionality, are employed in more uncertain environments, and are operated by a wider demographic of users with different levels of understanding and technical skills. These kinds of technologies are going to be more prolific in number and influence than any other previous technology known to date (consider the number of people already using digital assistants and content recommendation, as well as the impact that autonomous vehicles are likely to have throughout the world). In this atmosphere, the practice of designing assurances with little formal understanding is no longer viable; in short: *the existing informal approach to assurance design is no longer sufficient due to the new challenges that advanced intelligent systems introduce*.

When researchers discuss concepts like "comprehensible systems," "interpretable learning," "transparent systems," and "explainable AI," they are really interested in making deliberately designed mechanisms to help designers and users appropriately "trust" autonomous and artificially intelligent systems as they perform their tasks. For example, many systems are designed to learn from extremely large amounts of data and are expected to regularly perform on never before seen data—yet, it is rarely obvious if such data conforms to assumptions made at design time. Other systems are designed to perform tasks that are too "dirty, dull, and dangerous" for humans; the separation of users from these tasks often makes it difficult for them to understand whether these

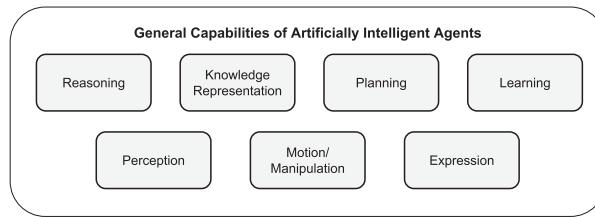


Fig. 2. Set of possible AIA capabilities.

systems are performing as desired. The authors, for instance, are interested in the design of unmanned robotic vehicle systems that operate in concert with remote human operators in uncertain dynamic environments. Since operators will generally not be computer scientists or roboticists, it is desirable for such systems to behave/communicate in ways that help operators properly use their abilities in scenarios featuring unexpected or incomplete information, time-critical decisions, and risky outcomes [61, 126]. This application is explained in more detail later in relation to Figure 1. These issues also have relevance and analogues in other applications of autonomous artificial intelligence, robotics, machine learning, and decision-making/support systems [7, 44, 105, 123], e.g., for scientific data analysis [37], public policy and medicine [64, 139], and cognitive assistance [52].

Some fields have formally and explicitly considered trust between humans and specific forms of intelligent technology, e.g., e-commerce, automation, and human-robot interaction. However, these research efforts have focused largely on developing formal cognitive and psychological models of trust, rather than system behaviors or algorithms that designers can exploit as assurances. Other fields that have explored assurance design only provide an informal connection to trust and applications to other disciplines, so it is unknown how effective their developed assurances might be in practice, or what principles ought to be considered for other kinds of autonomous and artificially intelligent systems. This article surveys assurance metrics and methods across relevant application domains, with the goal of identifying common principles, approaches, and questions related to trust-based interaction. To begin with, definitions for the trust cycle elements in Figure 1 are given to formally ground the concept of assurances. An example application is then provided as a means to compare/contrast technical ideas and implementations of algorithmic assurances throughout the survey in Section 3.

2.1 Trust Cycle Definitions

Artificially Intelligent Agents. Herein the term AIA will be used in order to encompass a broad range of technologies that can be considered autonomous. An AIA is defined here as an agent that acts on an internally or externally generated goal, and possesses, to some extent, at least one of the capabilities shown in Figure 2 [90, 103, 116]. While the term AIA can describe anything from a simple assembly line robot (which only possesses a single capability from Figure 2) to the fabled HAL 9000 (who presumably possesses all of the AIA capabilities), this definition underscores the idea that many assurances that exist for one set of (perhaps less capable) AIAs can be adapted and generalized for use in other AIAs. In other words, this definition sets a scope for the bodies of research that are likely to have investigated assurances and assurance principles, which can be extended to any “intelligent” computing system. The range of AIA capabilities also helps establish what kinds of assurances might be needed in future systems. For example, assurances for an AIA that only carry out planning tasks will probably differ in design or implementation from assurances for an AIA that only carry out perception tasks.

It should be noted that an AIA is assumed to operate with a degree of autonomy that is *delegated* by a user. That is, an AIA is self-directed and self-sufficient in its task to the extent that the user's "intent frame" (desired goals, plans, constraints, stipulations, and/or value statements) can be met by the AIA, regardless of how it actually accomplishes this. Following Miller [98], this view of autonomy as a delegation relationship refines the need for "transparent AIAs" by avoiding a contradiction of purpose that stems from an otherwise naive interpretation. From a naive standpoint, one could argue that if AIAs are developed primarily to alleviate the burden of complex reasoning and other undesirable workloads by removing users from the task at hand entirely, then this purpose is undercut by exposure and explanation of sophisticated AIA inner workings to the user. However, if AIAs are subordinates that are delegated tasks by users (who must still act as supervisors), the meaning of "transparency" shifts away from concern over how exactly an AIA accomplishes a task, toward concern over whether or not an AIA can execute the task as per the user's intent frame. This delegation-based view naturally sets up the question of user trust in AIAs.

User Trust. Trust is widely recognized as a critical part of intelligent multi-agent system dynamics—from those involving only simple one-on-one interactions [83], to more complex ones describing markets and governments [41]. Because of interest spanning many disciplines, it is difficult (if not impossible) to write a succinct definition of trust that would completely satisfy all interested parties. However, following Ref. [96], for the purposes of this work, trust is defined here as a psychological state in which an agent willingly and securely becomes vulnerable, or depends on, a trustee (e.g., another person, institution, or an AIA), having taken into consideration the characteristics (e.g., benevolence, integrity, competence) of the trustee.

This raises two important questions. Firstly, since trust is generally understood to exist between people, is it possible for a human to enter into a trusting relationship with an AIA? It has been confirmed several times that humans actually do develop trust in autonomous machines [8, 32, 39, 67, 92, 102, 113, 119, 141]. Lacher et al. [78] also notes that people trust AIAs for transportation systems at different levels (i.e., an engineer trusts differently than an operator or passenger). Secondly, in designing assurances that affect trust-based user behaviors, is it possible to know what drives those behaviors and thus have some working model of user trust that can be mapped to AIAs? McKnight et al. [95] (and later, Ref. [94]) performed what is arguably the first multi-disciplinary survey and unification of trust literature, which also condensed it into a single typology consisting of four major related components. Adapted to AIAs, these are: *Disposition to Trust*: the extent to which one displays a tendency to be willing to depend on AIAs in general across a broad spectrum of situations and persons; *Institution-Based Trust*: the extent to which one believes that regulations are in place that are conducive to situational success in an endeavor; *Trusting Beliefs*: the extent to which one believes that the AIA has one or more characteristics beneficial to oneself; *Trusting Intentions*: the extent to which one is willing to depend on, or intends to depend on, the AIA even though one cannot control its every action. Dispositional Trust is generally considered by psychologists and deals with long-term psychological traits that develop in a person from childhood (e.g., is someone pre-disposed to trusting technology?). Institutional Trust is generally studied by sociologists and represents the level to which a person trusts social/commercial structures. Finally, Interpersonal Trust (encompassing both "trusting beliefs" and "trusting intentions") deals directly with one-on-one relationships and tends to fluctuate most quickly. Each of these trust components has sub-components defined in Figure 3, which were identified by compiling many research studies across several disciplines. These components are the principal drivers of user trust-related behaviors and are the general notional targets of AIA assurances.

Trust-Related Behaviors. Trust ultimately leads to some kind of meaningful behavior or action that reflects the level of an individual's trust [84]. These actions are called "trust-related behaviors"

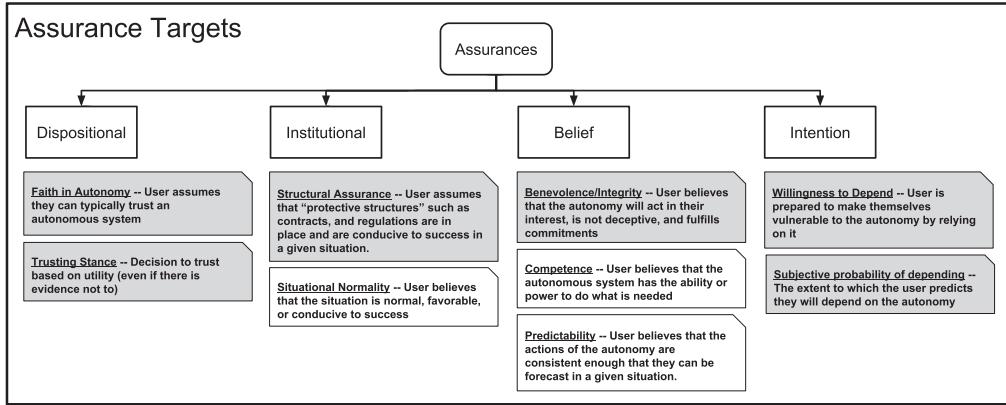


Fig. 3. Notional assurance targets based on the component definitions of the main trust categories. While any of these could be considered targets for assurances, the focus here is only on "Situational Normality," "Competence," and "Predictability."

(TRBs) [94]. In the case of a human-AIA relationship per Figure 1, some example TRBs could include the kinds of tasks the human user assigns to the AIA, accepting and following through on a plan produced by the AIA, directing that a new plan be made, or switching off autonomous capabilities altogether to teleoperate and perform tasks manually through a physical mechanism that the AIA otherwise controls.

Trust is not a univariate quantity that can be objectively measured. Rather, it is a multidimensional phenomenon whose "relative magnitudes and directions" must be observed through changes in TRBs, or qualitative self-reports gathered via surveys [102]. It thus comes as no surprise that TRBs are the more objective method of observation due to the fact that people are not always consistent in their ratings and may sincerely feel different levels of trust while performing similar TRBs [36]. Parasuraman and Riley [106] were interested in understanding the use of automation by humans and defined terms to describe that use. Here, it is proposed that, by extension, those terms also apply to the behaviors of humans toward more advanced AIAs. Within this scope the definitions are as follows: *Misuse*: over-reliance on an AIA (which could manifest itself in a user's unrealistically optimistic expectations of performance); *Disuse*: under-utilization of an AIA (e.g., a user turning off the AIA prematurely or failing to use all of its capabilities); *Abuse*: inappropriate application of an AIA (where *application* in this case means the choice to deploy an AIA in a certain context).

Following Figure 1, AIA assurances should ideally be designed to steer the user away from misuse, disuse, or abuse of the AIA, i.e., toward otherwise appropriate TRBs, by properly "calibrating" assurances to suitably influence user trust. This point, to some extent, has been alluded to in [61, 81, 85, 101]. Other researchers who propose "calibration" (or related concepts) suggest calibrating *trust* as opposed to TRBs. Dzindolet et al. [36] found that providing system performance feedback tended to increase users' *self-reported trust*, even though resulting TRBs did not reflect self-reported trust levels. This highlights the danger of calibrating "trust," as opposed to calibrating the TRBs. Whereas TRB calibration focuses on concrete and measurable behaviors, trust calibration involves influencing something that is directly immeasurable and subject to individual biases when indirect measurements are attempted.

Assurances. An assurance is an AIA property or behavior that can either increase or decrease user trust. The term "assurance" is perhaps earliest used in the context of human-AIA relationships

by Sheridan and Hennessy [120]. McKnight and Chervany [94] allude to this kind of feedback in e-commerce relationships as “Web Vendor Interventions.” Corritore et al. [27] refer to assurances as “trust cues” that can influence how online users trust e-commerce vendors. Lee and See [81] discuss “display characteristics,” which are methods by which an autonomous system can communicate information to an operator. More recently, Lillard et al. [85] provided a formal definition of assurances for autonomous systems that is similar to the one used here.

Assurances can be classified in several different ways. One way to classify an assurance is by its *Information Source*: Assurances must be informed by some kind of information, whether that means real-time observation of TRBs in order to have feedback, or well-accepted concepts of cognitive science as guiding principles of design. Another approach is to identify the *Source/Target* pair: In a human-AIA trust relationship, assurances link the AIA to the user. The user has multi-dimensional trust in the AIA (see Figure 3), and each AIA capability has multiple dimensions of “trustworthiness.” In designing an assurance, it is useful to explicitly identify the source capability and the target trust dimension (i.e., a certain assurance may have been designed as a “planning-competence” assurance). An assurance can be considered *Component* or *Composite*: A component assurance stems from one AIA capability to one trust dimension. A composite assurance originates from multiple AIA capabilities to one trust dimension. Another consideration is whether the assurance is *Tutoring* or *Telling*: An assurance that is dynamic to the different characteristics, and experience of users is a “tutoring” assurance. It is designed to help a user learn, over time, to trust appropriately. Conversely, all other assurances are “telling,” in that they are static in regards to separate users. *Mode of Expression*: Assurances can also be classified by their mode of expression. This includes the method and medium by which the assurance is expressed. There are many open questions regarding each of these categories; they are discussed further in Section 4 regarding future work.

Level of Integration. Herein, the “level of integration” of assurances are surveyed. This is useful because it addresses a natural consideration in the design process of AIAs; it also encapsulates well the key approaches that are in use. In this context “integration” refers to the level of effect the assurance has on the core functions of the AIA. As an example: an assurance that, if missing, greatly effects the AIA functionality is considered integral to the AIA. Conversely, a missing assurance that has no effect on the AIA functionality is not integral; we also call this “supplemental.” Between these two extremes, there is a natural continuum of integration on which we can classify the different algorithmic approaches to designing assurances; we do so in Section 3.

Summary. Each of the elements of Figure 1 has been defined in this Section (2.1). Figure 4 illustrates how these concepts fit together. In this document, algorithmic assurances are surveyed through the lens of their “Level of Integration”; more detailed discussion of the other elements are found in Section 4.

2.2 Recurring Example Application

To illustrate the assurances surveyed in the next section, a recurring example application based on the “VIP escort” problem [60] is provided, motivated by the authors’ work in unmanned systems. As depicted in Figure 5, an unmanned ground vehicle (UGV) leads a small convoy through a road network monitored by unattended ground sensors (UGS). The road network also contains a hostile pursuer that the UGV is trying to evade while exiting the network as quickly as possible. The pursuer’s location is unknown but can be estimated using intermittent data from the UGS, which only senses portions of the network and can produce false alarms. The UGV’s decision space involves selecting a sequence of actions (i.e., go straight, turn left, turn right, go back, stay in place). The UGS data, UGV motion, and pursuer behavior are all stochastic, and the problems of

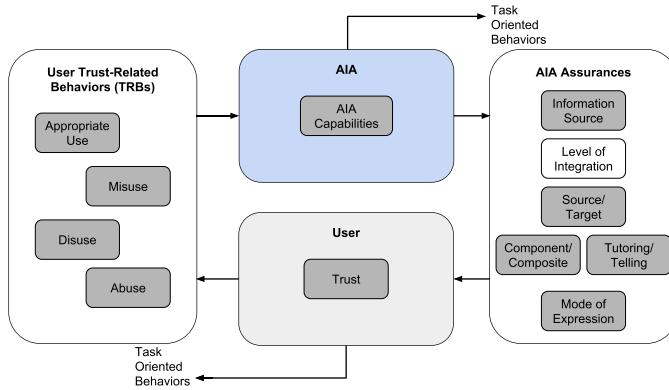


Fig. 4. Figure depicting the details of the human-AIA trust cycle.

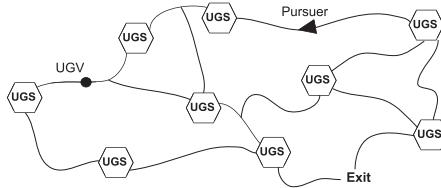


Fig. 5. Application example of an unmanned ground vehicle (UGV) in a road network, trying to evade a pursuer, using information from unmanned ground sensors (UGSs), as well as information from a human supervisor.

decision-making and sensing are strongly coupled: some trajectories through the network allow the UGV to localize the pursuer before heading to the exit (but incur a high time penalty); other trajectories afford rapid exit with high pursuer location uncertainty (increasing the risk of getting caught by the pursuer, which can take multiple paths). A human supervisor monitors the UGV during operation. The supervisor does not have detailed knowledge of the UGV—but can interrogate its actions, modify its decision-making stance (“aggressive” vs. “conservative”), and provide extra information about the pursuer (which is sporadically observed and follows an unknown course).

One way to construct an autonomous UGV path planner is to discretize time and spatial variables to build a partially observable Markov decision process (POMDP) model [73] of the navigation task. The ideal POMDP solution is an optimal UGV action selection policy that will, *on average*, maximize some utility function whose optimum value coincides with desirable UGV behaviors (i.e., avoiding the pursuer and exiting quickly). POMDP policies can be calculated by any number of sophisticated approximations that operate on probability distributions for the unknown pursuer state, which, in turn, can be found via Bayesian sensor fusion [4]. This defines at least two AIA capabilities per Figure 2: knowledge representation and planning.¹ The trust-cycle terms here can then be defined as follows relative to the supervisor (user)

- *AIA*: the combined POMDP planning and data fusion agent, which must make decisions under uncertainty;
- *Trust*: the supervisor’s willingness to rely on the UGV’s planning and data fusion algorithms when the safety of the VIP being escorted is at stake;

¹Consideration of lower-level UGV state estimation and control also leads to perception and motor control/execution.

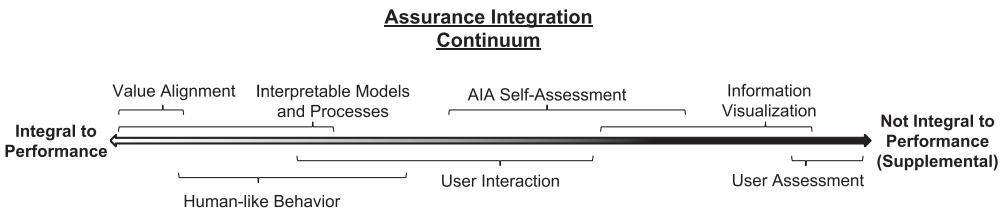


Fig. 6. The continuum of the level of integration of algorithmic assurances: to the left are those assurances that are integral to the key functions of the AIA; on the right are assurances that are not integral to performance, i.e., “supplemental” assurances.

- *TRBs*: supervisor’s behaviors that indicate trust (or lack thereof) in the UGV’s planner; these include approving/rejecting the planner’s actions, or real-time adjustments of the data fusion output based on what the supervisor receives from other intelligence sources;
- *Assurances*: properties and behaviors of the planning agent that effect the supervisor’s trust, e.g., communication of the escape success probability, reports that unexpected UGS data have been registered, or explanations of actions taken.

3 SURVEY OF ALGORITHMIC ASSURANCES

Whereas other researchers have noted the *existence* of assurances, we now directly consider the question: what exactly *are* assurances, and how can they be *practically designed* into AIAs? This section surveys the related literature to understand what algorithmic approaches can be used to design AIA assurances.

As discussed in Section 2.1, there are many different ways of classifying assurances. In evaluating different practical approaches to designing assurances, we have found that it is easiest to consider the “level of integration” of the assurance in the AIA. The level of integration of an assurance refers to the extent to which the core functionality of the AIA is dependent on the existence of that assurance. Assurances naturally lie on a continuum between being totally integral to the core function of the AIA, and not being integral at all but being generated by artifacts of the underlying task or AIA functionality. Figure 6 illustrates this continuum. For simplicity, we will sometimes refer generally to assurances as “integral” or “supplemental” based on whether they lie on the left or right side of the figure, respectively.² In the literature, we have identified seven main categories for designed assurances that span this continuum.

Practically, understanding the level of integration of different assurances is useful because doing so can indicate at which point different assurances need to enter the design process. For example, an assurance that is integral to the AIA must necessarily be considered from early on in the design process, whereas one that is supplemental can feasibly be added much later. Also, assurances at different levels of integration have similarities in their affects; because of this, designers may make different decisions regarding assurance design based on their specifications and goals.

While assurances cannot *guarantee* appropriate TRBs from a user, integral assurances are generally built with the aim of intrinsically guaranteeing—as nearly as possible—certain effects on user trust and TRBs. In contrast, supplemental assurances are typically weaker, and *encourage* appropriate TRBs; they rely much more on the uncertain relationships with human users. Problems can arise, for example, when a designer expects supplemental assurances to have the same effects

²Note that, while Figure 6 shows that the assurance classes occupy large spaces on the continuum, this is not referring to individual “component” assurances. An individual component assurance cannot be both integral and supplemental at the same time; it is located at a point on the continuum. This is not to say that an AIA cannot, simultaneously, have many assurances distributed over the assurance integration continuum, but that these assurances must be considered as separate.

on TRBs as those of integral assurances. This should generally not be expected. The remainder of this section is dedicated to discussing each of the seven categories in more detail.

Survey Methodology. While, theoretically, a two-way trust model could be considered (i.e., in which the AIA also has trust in the user), attention is restricted here to a one-way trust relationship that considers only how user trust (and TRBs) evolves in response to assurances from the AIA.

It should be noted that it is practically impossible to perform a fully comprehensive survey of all AIA assurances, due to the broad spectrum of possible assurances, and AIAs in general. As an example, one could rightly argue that control engineers treat metrics like gain and phase margins as assurances for automatic feedback control systems, in much the same way that machine learning practitioners treat training and test accuracy as assurances for learning algorithms—and, hence, concepts related to robustness, stability, and the like, for feedback control systems ought also be included in this survey. Similar arguments exist for assurances developed in fields like econometrics, software testing, aeronautical engineering and many others. While assurances can, in theory, be applied in both the most simple “automatic” systems (like a thermostat), this survey will focus on assurances in more advanced AIAs that make decisions under uncertainty. However, the admittedly narrow scope of this survey does not impede the development of fundamental insights and principles in designing assurances.

Initially, in order to find applicable research, papers that formally addressed trust, and tried to create models of it, were investigated. This was done with the aim of trying to understand how trust might be influenced. Secondly, literature regarding trust between humans and some form of machine entity was reviewed; this led to research in fields like e-commerce, automation, and human-robot interaction. Third, research on “interpretable,” “comprehensible,” “transparent,” “explainable,” and other similar types of learning and modeling methods were examined. Finally, with that literature as a background, research disciplines investigating computational methods that can be useful as assurances, but in which trust itself is not the main focus, were considered. This information was then used to construct an informed definition and classification of assurances based on methods that are currently in use or being investigated.

We now proceed to discuss each of the categories from Figure 6, starting from the most integral to the AIAs core functionality and proceeding to the least integral.

3.1 Value Alignment

AIAs operate autonomously in delegated tasks, with the expectation that they behave according to users’ intent frames. Optimization-based algorithms are arguably among the most common and direct approaches for accomplishing this. The general idea is to define a *utility function* that normatively governs the AIA’s abilities so that desirable behaviors are elicited through maximization of the utility, i.e., such that the AIA behaves rationally in accordance with the user’s intent frame. A utility function describes the “long-term desirability” of taking certain actions in certain conditions, i.e., beyond immediate benefits or penalties, and should coherently reflect user preferences about the state of the world and AIA behaviors [116]. Such mapping of user intent frames to utility functions has two positive benefits. Firstly, it ensures that AIA behaviors can themselves be used as assurances: users will tend to trust AIA’s more if they are “well-behaved” and acting in accordance with their desired intent than if they are not. Secondly, an AIA can generate assurances via auxiliary behaviors that help ensure its utility function is aligned with the user’s intent frame. Since it is practically quite challenging to encode user preferences and intent frames into utility functions, the process of *value alignment*³ leads to many different algorithmic strategies for generating assurances.

³Value alignment is more commonly known as “AI Alignment” in AI research [12, 148].

Consider a generic decision-making problem where an AIA that must make choice $a \in \mathcal{A}$ given some task state $s \in \mathcal{S}$, with scalar utility function $U_A(a, s)$. If a user’s true utility is represented by scalar function $U_H(a, s)$, then, in the ideal situation, the AIA seeks the optimal decision $a^* \in \mathcal{A}$ such that, for any $s \in \mathcal{S}$,

$$a^* = \arg \max_{\mathcal{A}} U_A(a, s) = \arg \max_{\mathcal{A}} U_H(a, s).$$

Hence, value alignment tries to minimize the difference between the utilities of the AIA and the user. When the utility of the robot $U_A(a, s)$ and the human $U_H(a, s)$ are approximately equivalent (within some tolerance), then the values of the AIA are *aligned* with those of the human. An AIA with aligned values will be considered by users to be more predictable (and thus more competent) because the AIA will be more likely to act in desirable ways. Bostrom [14] provides a well-known example of an AIA whose value is *not* aligned: an autonomous robot is designed, and deployed, with the intent that it make paper clips. To maximize $U_A(s, a)$, the robot then decides to take over the world in order to maximize its resources and ability to make more paper clips. To reasonable human users, this was clearly *not* the intended behavior; the utilities that the robot used for making decisions did not match those that the human must have had. Therefore, the robot’s resulting behavior was intrinsically an assurance that reduced trust. On the other hand, if the robot was to try to learn from its mistakes and improve (i.e., make $U_A(s, a)$ closer to $U_h(s, a)$) that could be perceived as an assurance that increases trust—the robot can be “forgiven” for making honest mistakes in trying to optimize an ill-posed/under-specified utility function, as long as it is able to recognize and remedy this.

3.1.1 Common Approaches: There are two algorithmic strategies for value alignment—(i) indirect: approximate $U_H(a, s)$ explicitly via $U_A(a, s)$, and then use this approximation to find a^* ; (ii) direct: identify $a^* = \arg \max_{\mathcal{A}} U_H(a, s)$ directly via the use of optimal state-action value functions $Q^*(s, a)$ (which give the utility to be gained if the AIA were to proceed optimally starting from s , regardless of its past states or actions). These strategies closely resemble techniques used for reinforcement learning problems and their variants (especially inverse reinforcement learning); not surprisingly, most value alignment techniques are rooted in this domain. Value alignment research tends to focus on several different issues [7, 47]; some of the more directly applicable topics and associated methods that point to useful assurance strategies are described below. The solutions to these problems are assurances because they afford opportunities for users to better understand the actual intentions and goals of the AIA, as well as understand how the AIA actually interprets intent frames.

Reward Hacking and Human-Guided Learning. The reward hacking problem deals with avoiding and removing unintended consequences in AIA behaviors that arise from imperfections in the specification of $U_A(s, a)$ (as in the paper clip-making robot example above). The most popular solution strategies use some form of offline supervisory human guidance or training data feedback in the utility function learning process. This approach recognizes the intrinsic difficulty of mapping user preferences to a single scalar utility $U_H(s, a)$ for complex tasks, and leverages sophisticated machine learning and reasoning strategies to identify relevant preferences within $U_A(s, a)$ or $Q^*(s, a)$, depending on the kinds of tasks considered. For instance, one-shot/non-sequential decision-making tasks like image recognition or object perception do not necessarily have dynamical state considerations, but may require potential expansion of the action space for sensible labeling of new object categories.

In the context of sequential decision-making problems, Hadfield-Menell et al. [56], Hadfield-Menell et al. [55], and Huang et al. [59] consider variations of the “inverse reward design” problem using inverse reinforcement learning techniques. In these works, discounted cumulative rewards

are used to model utility functions $U_A(s, a)$ and $U_H(s, a)$, where the actual reward factors contributing to $U_H(s, a)$ are unknown but can be inferred from user-generated contextual information at design time. Specifically, Ref. [56] notes that reward factors provided by users in limited training contexts serve as “noisy evidence of intent”. Hence, to avoid situations where an AIA trainee demonstrates desirable behaviors in specific training scenarios but later demonstrates undesirable behaviors in novel scenarios, the AIA must be able to reason over the uncertainty in the user’s intent in order to fully capture the context in which it was trained. In a different task setting, Freitas [40] compared two approaches to discovering “interesting” knowledge from large datasets, based on the idea that human users require assistance from complex systems in order to find useful patterns and other interesting insights. He mentions “user-driven” methods that involve a user suggesting interesting templates or providing general impressions in the form of IF-THEN rules. A subsequent comparison to different “data-driven” methods suggests that the latter are not very effective in practice. Having said that, user-driven approaches may not fare any better when compared over many users, as each user will likely have different preferences. Other scaled-up user-driven approaches, e.g., based on crowd-sourcing Chang et al. [20], can also achieve better accuracy for labeling tasks while also exploring new or ambiguous classes that can be ignored with traditional approaches (especially if training datasets are biased or very limited). Chang et al. [20] also consider a similar, scaled up, user-driven approach called “Revolt” that crowd-sources the labeling of images. It is able to attain high accuracy labeling, while also exploring new or ambiguous classes that might be ignored with traditional approaches.

Some other methods for designing, learning, and eliciting appropriate utility functions are also discussed in Refs [29], [44], and [55]. Despite the differences in AIA application contexts, these methods all provide the user with better context for what *should* be known by the system, and for how well it can interpolate/extrapolate. These processes allow users to refine their own intent in complex settings, e.g., to reveal or resolve subtle low-level inconsistencies in desired task requirements that would otherwise lead a rational AIA to undesirable behaviors.

Safe Learning and Correct-by-Construction Synthesis.. In many applications, $U_H(s, a)$ must be safely approximated when certain combinations of (s, a) lead to irreversibly bad consequences. Hence, as AIAs try to learn what a user’s utility is, they must do so in a safe manner. For instance, humans do not learn about the dangers of heights from falling off of skyscrapers. Instead we have to do so cautiously over time and extrapolate from a much less drastic experience (i.e., tripping on a curb). Safe reinforcement learning (safe RL) methods offer formal strategies and assurances for AIAs to learn in similar ways. Safe RL has been defined as the process of avoiding “unintended and harmful behavior that [emerges] from machine learning systems” [7]. Two ways to approach safe RL are: (i) modification of the optimality criterion with a safety factor, and (ii) modification of the exploration process through the incorporation of external knowledge [44].

For example, Lipton et al. [87] designed an “intrinsic fear” RL approach that uses a deep Q-network and a “supervised danger model.” The danger model stores the likelihood of entering a catastrophe state within a “short number of steps.” This model can be learned by detecting catastrophes through experience and can be improved over time. Curran et al. [28], in a more specific application, asks how a robot can learn when a task is too risky and then avoid those situations or ask for help. Similarly, Kahn et al. [65] use Bayesian Deep Neural Nets (using bootstrapping and dropout) to learn about the probability (with uncertainty) of an autonomous vehicle colliding in an environment given its current state, observations, and sequence of controls. Using this model they formulate a “velocity-dependent collision cost” that is used for model-based reinforcement learning. With this approach, the vehicle naturally proceeds slowly when there is an elevated risk of collision. This “safety-aware” behavior provides an assurance signal to the user.

Aside from purely learning-based approaches, we can also consider V&V methods. Not all practitioners are aware that V&V techniques can generate soft assurances for users. This is because V&V typically refers to the use of formal methods to guarantee the behavior of a system within some set of specifications, which are handed down by a certification authority as requirements to system designers to generate "hard assurances" (formal proofs of the functionality of the system). Although these "hard assurances" are not primarily designed for user consumption, they could, in principle, be exposed to and interpreted for users in certain contexts. A prime example is given by Raman et al. [110], who developed a formal way for non-expert users to provide structured natural language task specifications to a robot, such that a "correct-by-construction" controller will be built if the specification is valid. Otherwise, the robot will provide an explanation about which specification(s) are unrealizable/inconsistent and will cause failure. In the context of a practical self-driving car application, Ghosh et al. [45] presents a framework called Trusted Machine Learning (TML) for learning models from dynamically generated data that fit pre-determined "trustworthiness" constraints. These approaches are promising in that they not only present a way to communicate when and why specified tasks cannot be performed or certain actions cannot be taken, but also provide positive assurances in the form of guaranteed, formally verified, AIA processes for performing desired tasks (plans, models, etc.). While this directly addresses the competence and predictability components of AIA trust, the "raw" expression of these assurances does not formally account for effects on user trust or TRBs in formulating explanations.

Robustness to Context Shifts. How can an AIA determine when the basis and provenance of its approximation to $U_H(s, a)$ or $Q^*(s, a)$ is no longer valid for a particular task? This problem has attracted much recent attention in the learning literature under the guise of "nonstationary" learning. Nonstationarity refers to the complex challenge of training a model based on data from one distribution D while taking into account that the test distribution D' will likely shift through time [109]. For instance, in the context of classification problems, Sugiyama et al. [123] propose using importance sampling Monte Carlo to formally detect events related to "covariate shift" (training and test input data follow different distributions) and "class-balance change" (where the class-prior probabilities are different in training and test phases, but where there is no covariate shift). Similarly, Charikar et al. [21] address learning from "untrusted" data, which could be subject to adversarial attack or unknown nonstationarity.

These methods can be more generally adapted and developed beyond learning tasks, in order to evaluate the sensitivity of as-designed AIA's capabilities to possible changes in task context not captured/considered at design time (an example of coping with "unknown unknowns"). If the sensitivities imply a significant deviation in $U_A(s, a)$ or a^* from expected values (i.e., from user intent frame as initially understood), or indicate the presence of new (s, a) pairs that are not accounted for by $U(s, a)$ (e.g., test data that is very far from the training set), then the AIA can inform the user accordingly and thus possibly opt out of performing tasks that are now potentially "out of scope." This provides direct low-level behavioral assurances about changes in predictability, competence, and situation normality, though these may not be immediately understood by non-expert users.

3.1.2 Grounding Example. In the case of the "VIP Escort" problem (described in Section 2.2), value alignment might be used as an assurance in the following way, starting with the assumptions that:

- The UGV has just begun an attempt to escape the road-network.
- The UGV uses safe RL to learn its escape policy.
- The operator is able to observe the UGV during its entire escape attempt.

The operator has used several different UGVs for similar tasks. This newer model uses “safe RL” to learn its policy. When observing the UGV’s attempt at escape, the operator notices a difference in how the UGV operates. Whereas the older UGV models would sometimes do risky things, this UGV seems to navigate dangerous situations much better.

Discussion of Example: In this case, safe RL enabled the UGV to treat situations that an operator might classify as “dangerous” with more care. With this integral capability, the UGV assures the operator that it is more competent.

3.2 Interpretable Models and Processes

Another way to provide assurances about AIA conformance to user intent frames is to expose the models and algorithmic processes governing its actions directly to the user. If these models and processes also happen to be easy for users to interpret, then the user can (ideally) acquire a well-formed and highly predictive “theory of mind” for the AIA’s behavior, with little or no effort. Doshi-Velez and Kim [34] give an argument for why interpretability is critical in AIA systems since interpretability “is used to confirm other important desiderata of [machine learning] systems.” Yet, perhaps unsurprisingly, “interpretability” and the attendant desiderata still elude formal universally accepted definitions. They also use the words “interpretable” and “explainable” interchangeably. In contrast, we treat them as distinct descriptors. We discuss models that are inherently interpretable here, and models that can be understood by explanation in Section 3.5.1. The difference is that interpretability (in our view) implies that the actual process/model used by an AIA is self-explanatory, whereas explainable models can be made interpretable by post hoc operations but do not necessarily explain the actual model/process used by an AIA. Being able to interpret the actual model/process used by an AIA helps human users to more appropriately understand their behaviors, and thus exhibit appropriate TRBs in turn. This approach to assurance also captures broader AIA processes and models that rely on rules, heuristics, and the like, rather than just those that rely on optimization of some particular utility.

3.2.1 Common Approaches. Two main approaches to designing interpretable AIA models and processes are considered here. The first is to *assess an existing set of candidate models/processes* in order to evaluate their interpretability in the context of a particular task, and then select the best candidate. This is typically done with certain classes of models or solution processes, e.g., whether to use decision trees vs. decision tables for a given planning task. The second is to *synthesize interpretable models/processes* by leveraging human designer input during the model/solution-building process. The first approach requires pre-defined measures of interpretability, and thus some mechanism for capturing ability to gain insights into competence, predictability, and situation normality. This also presupposes that the candidates are inherently interpretable along these lines to begin with, which may rule out methods that perform well on certain tasks. The second approach allows designers to apply domain knowledge to determine metrics for interpretability, although this can lead to solutions that do not perform as well as those that are less interpretable.

What is the assurance mechanism that potentially leads to proper TRBs in either approach? Essentially, allowing the user to access and examine an interpretable model/process also allows them to simultaneously assess competency, predictability, and situational normality components of trust. If the models/processes are perfectly interpretable, then a user could understand exactly how the AIA would perform its task (i.e., down to a mechanical/programmatic level). This gives the user a “mental model” of what the AIA would consider to be situational normality and how AIA would respond in different situations (predictability and competence). The caveat here is that incorrect TRBs may arise if the user misinterprets or only understands part of the model/process. This is a significant risk in highly complex or specialized problems, where users may not actually

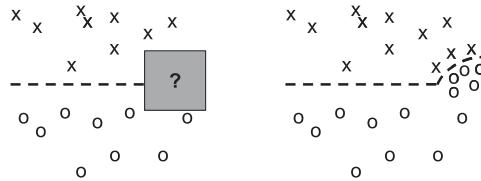


Fig. 7. Example of simple global interpretable learning model on the left, and on the right, a more complex locally interpretable learning model that can be used when more precise understanding of a specific decision made by the learner is required.

have sufficient training or expertise. This also poses concerns for how/when users can access interpretable AIA components. Unlike value alignment (where a user accesses assurances only through behavior of AIA itself), the user has more freedom in deciding when and how to “peek under the hood.” This relates to assurances based on information visualization discussed later, except that here, the information being given to the user are the actual AIA algorithms themselves, as opposed to byproducts or after effects of those algorithms.

Assessing Interpretability. Van Belle and Lisboa [135] suggested three ways to ascertain the level of interpretability and potential utility of learned models (compare to categories proposed by Lipton [86]): (1) Map them to domain knowledge; (2) Ensure safe operation across the full operational range of model inputs; and (3) Assess whether important non-linear effects are accurately accounted for. This work identifies certain strengths and weaknesses of different techniques, but ultimately concludes that no method is clearly best in all situations. Along similar lines, Huysmans et al. [62] compared decision trees, decision tables, propositional if-then rules, and oblique rule sets to understand which set of methods is “most interpretable.” It was experimentally determined that decision trees and tables tend to be easier to interpret, but it is noted that each method could perform better than others in different applications. For example, decision trees and tables are typically better suited for answering a symbolic question (which requires a local understanding of a model) like: *how does the model classify observation X?* This is in contrast to a spatial question (which requires a global understanding of the model) like: *is it correct that applicants with a high income are more likely to be accepted than applicants with a low income?* Having quantified the interpretability of a model given different classes of problems and different requirements of users, the appropriate model can then be selected during design to fit the needs of a specific application.

Interpretable Model Synthesis. Ruping [115] asks how classification results and the accuracy-interpretability tradeoff can be made more transparent to those who design and use classifiers. He explores one approach by combining simpler global models with more complex local models that are built around learning results (Otte [105] and Ribeiro et al. [111] implement similar ideas as well). Figure 7 illustrates this idea. The explanation of Figure 7 could be something like: “The classification boundary is generally a horizontal line. However, for a small region on the right-hand side, the boundary is shaped roughly as an inverse quadratic starting from the horizontal line.”

Considerable effort has also gone into endowing “grey box” and “black box” models with interpretable features. For instance, Abdollahi and Nasraoui [1] investigate making collaborative filtering models more interpretable by using a conditional restricted Boltzmann machine (RBM). Ridgeway et al. [112] use “weight of evidence” (WoE) as a boosting method that is more amenable to interpretation, and show that the performance WoE is on par with AdaBoost. Choi et al. [23] construct a recursive attention neural network to remove recurrence on the hidden state vector, and instead add recurrence on the visits of patients to doctors, as well as on different diagnoses

during those visits. In this way,² the model is able to predict possible diagnoses in time, and a visualization can be that that indicates the critical visits and diagnoses that lead to that prediction.

Learning of human-understandable representations for data and feature selection also provides another avenue for developing assurances [11, 53]. For instance, Mikolov et al. [97] studied how to represent words and phrases in a vector space for natural language text learning; this enables simple vector operations for understanding word sense similarity and relative relationships learned from text corpora. For example, the vector addition operation *airlines+German* yields similar entries that include *Lufthansa*. Such representations encode knowledge that can be easily checked and understood by humans, and thus implicitly facilitate interaction and calibration of trust (see Ref. [57] for another example). The problem of discovering human understandable features and representations in more general settings still remains an open question. Currently, the main question for representation learning is how to find the “best representations” for a particular application—not necessarily the representations and features that are “most humanly understandable.” This is not surprising, since human-understandable representations and features are not necessarily optimal for the criteria that AIAs are typically designed against.

Contrary to the belief that interpretable models are necessarily worse performing than their less interpretable counterparts, several researchers have shown that this is not always the case (at least in the context of machine learning). However, the real tradeoff is the amount of work that goes in to crafting the interpretable model from the start; these methods are often custom designed for certain tasks and are not easily transferable to other problems. Because of this, AIA designers must strike a balance between *interpretable models*, *explainable models*, and *black-box models*.

Park et al. [107] point out that real interpretability in complex tasks still requires expert knowledge to make sense of complicated features; in essence: *people are needed at both ends of interpretable models*. For instance, Jovanovic et al. [64] use “Tree-Lasso” (TL) logistic regression with domain knowledge (i.e., medical diagnostic codes) to group similar conditions, and then use TL regression again on that information to develop a sparser model. Zycinski et al. [152] also use domain knowledge to structure a data matrix before feature selection and classification. See also Zhang et al. [150] and Khoa et al. [71] for other related examples. This kind of approach is also illustrated by those who use “theory guided data science” (TGDS [37, 68]). As one example Morrison et al. [100] address the situation where an imperfect analytical model is available for chemical reaction kinetics: the theoretical reaction equations are well known, but a “stochastic operator” is added on top of this to account for uncertainties and modeling errors. In adopting this approach, the model becomes interpretable (to experts).

3.2.2 Grounding Example: In the case of the “VIP Escort” problem (described in Section 2.2), interpretable models might be used as an assurance in the following way, starting with the assumptions that:

- The UGV has just begun an attempt to escape the road-network.
- The UGV is using a decision-tree for selecting different movements.
- The operator is able to view the decision-tree model the UGV is using.

While the operator is monitoring the progress of the UGV in its attempt to escape the road-network, they are able to consult the decision-tree model. In this case, the operator chose to consult the table when they saw the UGV make an unexpected turn at a given intersection. The operator identified the conditions that led to the decision and found that the UGV was not well-equipped to execute the decision the operator thought was best.

Discussion of Example: In this example, the use of a decision-tree as a model enabled the operator to investigate unexpected behavior. During inspection, they identified certain conditions that led to

a decision, and they found that the UGV was not *competent* to perform what the operator thought was a better decision. Because of this, the operator better understood the decision the UGV made and will have a more appropriate level of trust in future interaction.

3.3 Human-Like Behavior

Since humans are accustomed to forming and evaluating trusting relationships with each other, imitation of human–human communication and interactive behaviors provides yet another avenue for developing AIA assurance strategies. Support for this idea is given by Tripp et al. [134], who compared human trust in other humans against human trust in intelligent interactive technology. They found that as the technology becomes more “human-like,” self-reported levels of trust in technology become more similar to levels of trust in other humans.

Also, de Visser et al. [30] specifically discusses different methods by which AIAs can be more human-like in order to “repair trust” with users (here, trust repair is roughly analogous to assurances, but focuses on re-building trust after it is lost). Among several other possibilities, they suggest that an AIA might repair trust by anthropomorphizing (responding using a human communication channel), or by explaining its actions in the same way a person would. Such human-like behavior opens the door for AIAs to exhibit “non-rationally motivated” behaviors (i.e., suboptimal, as opposed to irrational actions), if these conform to social norms or other psychological cues that provide useful assurances about predictability (e.g., a robot arm that executes legible motions), competency (e.g., a robot which slowly backs away from unfamiliar or potentially dangerous objects), or situation normality (e.g., a robot car that apparently rubbernecks near an unfamiliar scene on the road).

3.3.1 Common Approaches. Generally, we do not have algorithms that describe how humans interact with each other (yet), and must settle for heuristics or best attempts to create human-like behavior via algorithms. From a high level, researchers have addressed these: nonverbal communication and mannerisms.

Nonverbal Communication. Nonverbal communication can take many different forms. One popular approach is to use motion or gestures. Szafir et al. [127] investigated how to enable “Assisted Free Flyer” robots (quad-copters that are made to interact with humans in close spaces) to communicate by using gestures. In doing so, they use “motion primitives” (a basic vocabulary of movements) that were inspired by basic “character animation” principles [136]. In their evaluations of these primitives with human participants in the presence of free flyers, they found that human users significantly found the free flyers to be more natural, and felt safer around them. Later, Szafir et al. [128] also experimentally showed the effectiveness of using illuminated “turn signals” and pairs of human-like “eyes” that shifted with free flyer heading (much as human eyes do when people walk in a crowd) to help users more easily interpret the vehicle’s intended movements and actions. These works provide strong support for “commonsense communication” assurances aimed at predictability in physical user-AIA interactions (even if indicators like “moving eyes” do not actually see anything). Likewise, Dragan and Srinivasa [35] investigate “legible motion planning,” i.e., planned robotic physical movements and gestures that, by themselves, convey intended actions and goals. For example, a table-setting robot may grasp a plate on both sides from the top using two end effectors if it intends to shift the position of the plate along the table surface, whereas it may grasp the same plate with only one hand from the side if it intends to pull away and remove the plate from the table. Legible motion is used by humans working in close proximity, and so can also be useful and important for situations in which a physically embodied AIA and person are collaboratively working in close proximity to each other. Similarly, in

more recent work, Kwon et al. [76] investigates calculating trajectories that convey “incapability,” which is *what* the AIA is trying to do, and *why* it is unable to do so. See also Ref. [3] for related work.

Mannerisms. Humans are naturally inclined to leverage social interaction cues and adherence to/violation of social norms as evidence for assessing the trustworthiness of other humans in everyday interactions. AIAs can leverage these inclinations to provide simultaneous assurances of their competence, predictability, and situation normality. Consider, for instance, a recent “mini-Turing Test” example from the popular media: at Google/IO 2018, Google Duplex [46] was introduced through a demo where it placed a phone call to make a reservation. An oft-remarked feature of this demo is the great difficulty (if not near impossibility) of detecting whether or not the Duplex voice is human—down to the words spoken, tone of voice, and speech mannerisms (which included “um . . .,” pauses, and shortened sentences). The human on the other end of the call was none the wiser, and trusted that they were, in fact, speaking to a regular human customer—when, in fact, they were speaking in a completely natural manner to the product of a recurrent neural network (RNN) trained on anonymized phone conversation data.

More formally, Salem et al. [119] investigated the effects of autonomous task errors, task types, and “system personality” on cooperation and trust for humans who observed a domestic robot performing house tasks, such that the robot implicitly showed competence by its mannerisms and successes/failures during tasks. In this case, the mannerisms and competency of the robot were completely under control and hard-coded into the system. Regardless, when participants were asked to cooperate with the robot on certain other tasks, the strange/unexpected operation of the robot was found to influence the self-reported trust levels of the participants.

Wu et al. [145] investigated how a person’s decisions in a coin entrustment game are affected by their belief in whether they are competing against an AIA or another human player (which, unbeknownst to participants, was in fact an AI with some programmed human-like idiosyncrasies, e.g., variable wait times between turns). Trust in this context was measured directly by the number of coins a participant was willing to lose by putting them at risk to the other player. The experiment found that the participants trusted the AI opponent more than they trusted the “human” opponent; the authors suggest that this may be due to the perception that the AI opponent did not have feelings and operated in a more predictable and consistent “machine-like” way. Given that, in this situation, the “human” player was represented by an AI as well, this experiment illustrates that “machine-like” behavioral consistency can lead to implicit positive effects and the trust of the participant in certain contexts.

3.3.2 Grounding Example. In the case of the “VIP Escort” problem (described in Section 2.2), human-like behavior might be used as an assurance in the following way, starting with the assumptions that:

- The UGV is about to begin an attempt at escaping the road-network.
- The operator can observe all the actions of the UGV via video feeds at intersections.
- The UGV has been designed with the ability to use gestures in order to indicate its ‘incapability’ as in Ref. [76].

As the UGV begins the escort problem, the human supervisor is monitoring progress. When the UGV reaches a certain intersection of the road network, the supervisor expects the UGV to take a path *A*, but it does not. However, before choosing to take path *B*, the UGV made a movement that, to the operator, indicated that it considered attempting to traverse *A*. Due to the attempt, the supervisor was able to surmise that the UGV wanted to take that path but couldn’t due to some limitation.

Discussion of Example: In this case, the UGV is able to maintain appropriate trust of the supervisor because the supervisor was able to interpret the “gesture” that UGV was using. This highlights the assuring effects that human-like communication/behaviors can have on users.

3.4 User Interaction

Despite the oft-repeated sentiment that advanced AIAs will “soon” be able to operate with little or no human involvement, those who have more practical experience with AIAs are much more skeptical of this claim, and point out that it is highly unrealistic to expect AIAs to ever function “perfectly out of the box” with true total autonomy [15]. A popular and promising avenue for surmounting the inevitable shortcomings of AIAs, and thus engendering trust in users, has therefore been to put the users “in-the-loop” (or “on-the-loop”) as collaborative partners who can augment (or supervise) AIA capabilities. In formulating algorithms for AIA capabilities that leverage user inputs, the user becomes analogous to a supervisor working alongside those they supervise; in doing so, they are able to provide useful feedback in real time, lend their expertise, and better appreciate the decisions and outcomes of the team’s work. Such collaborative problem solving not only gives users a chance to directly assess AIA competence and predictability through experience (assurances), but also provides a way for users to continuously engage AIAs in accordance with their actual capabilities (appropriate TRBs). Note that user interaction techniques are *not* the same as user assessment techniques discussed later, since user assessment techniques do not involve fundamentally changing AIA algorithms or capabilities to exploit user interaction.

3.4.1 Common Approaches: Users can be exploited to provide or augment any of the AIA capabilities in Figure 2 on many different levels. At one extreme, a user might fully replace or augment a subset of core AIA capabilities, e.g., to act as a high-level “sensor” and planner for an autonomous robot in a navigation task [69]. On the other extreme, the human might have a very weak involvement in the core perception functionality of an AIA, e.g., to validate the labeling of image data. Since the literature in this area is quite vast and ongoing research quite active, we focus here, for the sake of brevity, only on a few typical methods from the human-robot interaction literature where the AIA (an autonomous robot) engages the user as an additional “sensor”/perception agent or “controller”/planning agent. The references cited in these works also point to a host of other related and relevant techniques, which, in turn, can (and have) been adapted to other AIA capabilities such as learning, reasoning, knowledge representation, etc.

Sweet and Ahmed [125] investigate the use of humans as “soft” sensors for target localization tasks, whereby semantic natural language observations (“Target is by the bridge,” “Nothing in the street”) can be directly combined with conventional “hard” robot sensor data (from cameras, lidar, sonar, etc.) in order to improve and augment the robot’s Bayesian state estimation algorithms. They apply their approach in a scenario called “Cops and Robots” where a single “cop” robot tries to locate mobile “robber” robots in a semantically rich indoor environment. In this case, the human acts as a “deputy” that remotely interacts with the system. The human can see security camera footage of the building in which the cop is searching and can offer natural language feedback to the cop robot when appropriate. If the human offers information, it can be fused into the cop robot’s estimation model, but in the meantime, the cop robot operates autonomously to plan its motion without human assistance. Along similar lines, Kaupp and Makarenko [69] empirically identify the appropriate level of autonomy for a robotic navigation system while taking into account the amount of sensory interaction required by a human supervisor. In this case, the robot has sensors of its own but can also ask for user input when the value of information (VOI) is high enough (i.e., is it worth asking a human sensor for information given that there is a cost?); they define the

threshold VOI by performing human trials before deployment of the system in order to optimize the involvement of the human user.

Tellex et al. [130] consider planning algorithms that are augmented by human natural language commands for an autonomous assembly robot that can detect when it has failures (conditions that don't match expectations based on internal models). When this occurs, the robot requests help from the human user to resolve the problem. In this way the human and robot are dependent on each other to accomplish a task. Since the user knows that, if needed, the robot will ask for help, they can more appropriately trust that unknown problems won't occur without them being informed. Freedy et al. [39] studied performance measures for mixed-initiative human-autonomous robot teams (where users and robots share planning and decision authority), and examined the extent to which such teams can only be successful if "humans know how to appropriately trust and hence appropriately rely on the automation." They explore this idea by using a tactical reconnaissance scenario where human participants supervised a UGV platoon with three levels of autonomous targeting/firing capability (low, medium, high); these levels were dependent on the experimental conditions. The operator needed to monitor the UGV in case it couldn't perform as desired; in such cases, the operator could intervene to resolve the problem. Operators were trained to recognize signs of task failure and to only intervene if they thought the mission completion time would suffer.

3.4.2 Grounding Example: In the case of the "VIP Escort" problem (described in Section 2.2), user interaction might be used as an assurance in the following way, starting with the assumptions that:

- The UGV has just begun an attempt to escape the road-network.
- An interface system exists by which the operator can receive and provide information to the UGV.

The UGV is capable of operating autonomously, but also can benefit by asking for assistance or information when necessary, e.g., using a natural language interface for augmented planning and sensor fusion. In this way, the functionality of the UGV can be greatly improved via interaction with the user. As the user interfaces with the UGV and is able to provide feedback and information about the best known location of the pursuer based on information unavailable to the UGV, they have more trust in the competence, predictability, and situational normality of the UGV.

Discussion of Example: In this scenario, the user is more immersed in the functioning of the UGV. Not only are they able to respond to queries from the UGV, but they can also provide direct observations as well. Subsequently, the user feels more immersed in the functioning of the UGV and is more cognizant of appropriate TRBs.

3.5 AIA Self-Assessment

The techniques of previous sections generally tend to provide integral assurances (i.e., designed as part of core functionality of AIA capabilities) that are artifacts of interactive algorithms designed to compensate for shortcomings in AIA capabilities. This section focuses on 'introspective' assurances that inform users of competency limits and boundaries of AIA capabilities without requiring user interaction, and that can generally be separated from core AIA functionality (i.e., without requiring modification of core, underlying, AIA design). These self-assessments can provide users with insights regarding either or both of the following related issues: (i) what information and tasks are actually within the AIA's reach, and (ii) what is required by the AIA to actually do its assigned task? In contrast to user interaction techniques: the analogy here is of a subordinate telling a supervisor what she is/is not capable of, or telling the supervisor what she would need in order

to carry out the specific task at hand to achieve a specific outcome, or what the possible outcomes actually would be for that specific task.

3.5.1 Common Approaches. The literature in this section can be split into two high-level categories. The first set deals with how an AIA can algorithmically account for its uncertainties in its models of its task, environment, operating context, and capabilities. These kinds of assurances help inform the predictability and situation normality aspects of trust. The second set of methods attempts to algorithmically reduce complex “uninterpretable” models or processes that underlie AIA capabilities into more interpretable ones by providing explanations. Here, the AIA makes an active attempt at processing data and making information available to the user to inform the competency aspect of trust.

Accounting for Uncertainty. An AIA that can predict its performance on different tasks can provide assurances about competence, predictability, and the situational normality of a given task. Several researchers have worked to improve this ability in visual classification [25, 51, 66, 151]. For example, to ensure that visual classifiers don’t fail silently in novel scenarios, Zhang et al. [151] learned models of errors on training images to predict errors on test images. Kaipa et al. [66] consider a 3D visual classification of assembly line parts for robotic pick and place tasks, and develop statistical goodness-of-fit tests to estimate the likelihood that robots can use their sensors to find parts matching desired ones. These approaches allow the AIA to assess capability and present appropriate assurances to users, though without any formal notions of trust.

Mitchell et al. [99], discuss, in the context of a “never ending learning problem” (i.e., where the AIA perpetually learns over time), how an agent can quantify uncertainty on unlabeled data given three requirements: (1) three or more approximations of a function are available, (2) the assumption that these functions are more accurate than chance, and (3) these functions have independent errors. The rates at which these functions agree on classification of unlabeled examples can be used to solve for their exact accuracies. Doing this allows the system to actively reduce uncertainty by seeking relevant data. In the context of image classification, Paul and Newman [108] introduced “perplexity” as a metric that represents uncertainty in predicting a single class and is used to select the “most perplexing” images for further learning. There have also been several attempts to use Gaussian processes (GPs) to actively learn and assign probabilistic classifications [13, 31, 48, 49, 91, 131–133]. As with perplexity-based classifiers, the key insight is that if a classifier possesses a measure of uncertainty, then that uncertainty can be used for efficient instance searching, comparison, and learning, as well as reporting a measure of confidence to users. The key property of GPs to this end is their ability to produce output confidence/uncertainty estimates that grow more uncertain away from the training data. This information can be readily assessed and conveyed to users, even in high-dimensional problems. This property has also found much use in other AIA active learning problems, e.g., Bayesian optimization [16, 63, 121].

Neural network (NN) models are commonly considered black-box models, and methods to represent uncertainty have not historically been available. However, there have been several recent advances to make this possible to some extent [42, 43]. Bayesian neural networks (BNNs) are a method by which we can draw insight about the uncertainty of a neural network’s predictions; this is possible by placing prior distributions over the weights in an NN. Kendall and Gal [70], in the context of computer vision, also use deep BNNs to help visualize epistemic (input) and aleatoric (model) uncertainty for each pixel of an image. Similarly, Kahn et al. [65] use deep BNNs to learn about the probability (with uncertainty) of an autonomous vehicle colliding in an environment given its current state, observations, and sequence of controls. Using this model, they formulate a “velocity-dependent collision cost” that is used for model-based reinforcement learning. In order to help predict uncertainty in real-time robotic applications that learn from demonstrations,

Choi et al. [24] use mixture density networks (MDNs)—neural networks that learn parameters of a Gaussian mixture distribution—to model complex distributions from human demonstrations.

Models and logic are not trustworthy by themselves; they may be flawed to begin with, or become invalid when assumptions or specifications are violated. Thus, there is great interest in providing assurances that the models and assumptions underlying different AIA processes are, in fact, sound. Laskey [80]—with the intention of communicating model validity to users of “probability-based decision aids”—notes that it is infeasible to perform a decision-theoretic calculation to determine if model revision is necessary. She presents a class of theoretically justified model revision indicators, based on the idea of constructing a computationally simple alternate model and then initiating model revision if the likelihood ratio of the alternate model becomes too large (see also Habbema [54] and Zagorecki et al. [149]). Ghosh et al. [45] present “model repair” and “data repair” strategies that can be used when the current model does not match the observed data, at which point the model and data can be repaired, and control actions can be replanned in order to conform with the formal method specifications. One challenge is how the “trustable” constraints should be identified, as this places a strong burden on the certifying authorities and system designer to foresee all possible failures.

Reduce Complexity. Representations within an AIA are often complex. Sometimes using inherently less complex, “interpretable models” (as discussed in 3.2), is the most straightforward way to address this challenge. However, in some cases, it is desirable to maintain complex, less interpretable representations (e.g., for performance reasons) and then reduce the inherent complexities (possibly post-hoc) to aid human users.

One typical approach is to generate explanations, but how should explanations be provided? There are also considerations regarding whether explanations should occur by two-way interaction between system and user, by natural language interaction, or by probabilities. Some of the answers to these questions lie more in the realm of cognitive science. Still, natural language and other communication modalities could be used [58]. Specifically, Olah et al. [104] investigate how predictions of NNs can be explained through visualizing how different parts of the network respond to certain images. They propose combining several different approaches to get a holistic view of the NN behavior. Specifically, they use feature visualization (what a neuron is looking for), and attribution (how it affects the output).

There are several classes of explanations. Abdollahi and Nasraoui [2] propose three in the context of collaborative filtering: “neighbor style” (explanation based on examples from similar situations), “influence style” (present the most influential items that led to a certain model output), and “keyword style” (identify common features between user keywords and content). Otte [105] and Ribeiro et al. [111] implement analogous ideas in the realm of safe ML and interpreting classifiers, respectively. Huang et al. [59] use “algorithmic teaching” (see Ref. [9]) as inspiration for helping human users learn a robot’s true objectives. Algorithmic teaching involves having a model of a student’s learning algorithm and then presenting training examples to allow the student to learn a target model. In this case the “student” is the human user, and the “teacher” is the robot that is trying to teach the human its own objective function by presenting a set of (optimal) training examples. Here, we would consider the robot’s training examples as assurances.

Another consideration is whether an explanation is meant to be descriptive or aimed at ensuring comprehension, as well as whether explanations need to be on a macro or micro scale relative for parts of the Bayesian network (similar to globally/locally interpretable learned models [115]). Lacave and Díez [77] address the AIA reduction of complexity from the perspective of explaining probabilistic inference in Bayesian networks—specifically, *how* and *why* a Bayesian network reaches a conclusion given some imputed evidence. They present three properties of explanation:

(1) content (what to explain), (2) communication (how to explain), and (3) adaptation (how to adapt based on who the user is). Several key points for designing assurances arise from considering the differences between explaining evidence (i.e., data), the model (i.e., the Bayesian network itself), or the reasoning (i.e., the inference process).

Aitken [5] propose a metric called "machine self-confidence" for providing users with better insight into autonomous decision-making under uncertainty. This insight comes from breaking down the complex influences of uncertainty on the decision-making process and presenting them to the user in a simple way. Self-confidence is defined as the machine's own perception of its ability to carry out tasks in light of uncertainties in its knowledge of the world, its own self/states, and its reasoning process and execution abilities. In this sense, self-confidence is an AIA's metacognitive assessment of its own behavior and "competency boundaries." A computational measure for POMDP-based autonomous planning is defined from five component assurances (which are fairly general and applicable to most other kinds of planners): (1) Model Validity, (2) Expected Outcome Assessment, (3) Solver Quality, (4) Interpretation of User Commands, and (5) Past Performance. The key idea behind this set of measures is to assess where and when approximations required for planning under uncertainty are expected to break down. Model validity attempts to quantify the validity of a model within the current situation. The expected outcome assessment uses the distribution over rewards to indicate how beneficial or detrimental the outcome is likely to be. Solver quality quantifies how a specific POMDP solver is likely to perform in the given problem setting (i.e., how close to optimal the solution policy and approximate solution policy can get). The interpretation of commands component is meant to quantify how well the objective has been interpreted (i.e., how sure is the AIA that it correctly interpreted mission specifications into relevant tasks and suitable goals). Finally, past performance is meant to add in empirical experience from past missions in order to make up for theoretical oversights and account for learning-based processes.

Aitken [5] proposes that self-confidence values could, for instance, be reported as a single value between -1 (complete lack of confidence in achieving mission objectives) and 1 (complete confidence in achieving mission objectives); a self-confidence value of 0 reflects total uncertainty. Each of the component assurances could be useful on its own, but the composite "sum" of the factors is meant to distill the information from the five different areas, so that a (possibly novice) user can quickly and easily evaluate the ability of the AIA to perform in a given situation. Currently, only two of the five metrics (Expected Outcome Assessment and Solver Quality) have been developed quantitatively, but there is continuing work on the other metrics and they plan to perform human experiments to evaluate the usefulness of the self-confidence metrics for AIAs. Other approaches for computing and communicating AIA self-confidence have also been proposed for more specific applications [61, 66, 75, 149].

3.5.2 Grounding Example. In the case of the "VIP Escort" problem (described in Section 2.2), self-assessment might be used as an assurance in the following way, starting with the assumptions that:

- The UGV is about to begin an attempt to escape the road-network.
- The UGV is using the "solver quality" metric mentioned by Aitken et al. [6].
- The operator has access to an interface where they can view the self-confidence metric calculated by the UGV

Before the UGV begins its attempt, it is able to assess its "solver quality" given the specific, previously unseen road-network based on similarities between the current network and ones that it has encountered before (i.e., problem features that are important to determining the quality of

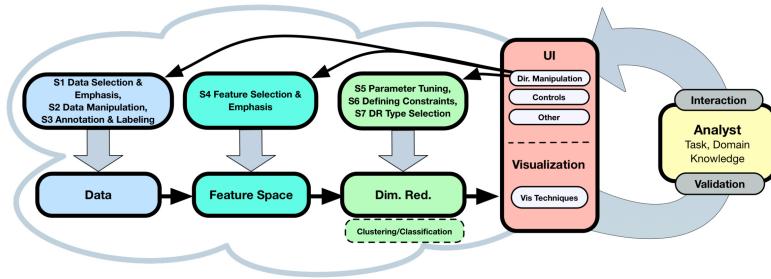


Fig. 8. Human-in-the-loop process model [118] ©2017 IEEE. Reprinted, with permission, from Sacha, Dominik, et al. “Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis.” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, Jan. 2017, pp. 241–50.

the approximate solution produced by the policy). The UGV reports that it has high confidence in its solver quality, and the operator is assured that they can trust the solver in this situation.

Discussion of Example: In this case, the UGV is able to assure the operator of the quality of the solver in the specific road-network. Generally, the UGV reduced what could be a very complex analysis into a simple format for the operator to interpret. This is in contrast to the operator viewing policies, models, algorithms, and complex probability distributions.

3.6 Information Visualization

We define “information visualization” as the act of displaying artifacts generated by AIA models or processes for the task at hand in such a way as to communicate to one of the trust dimensions of a human user. Specifically, we consider the “competence” and “predictability” of the AIA, as well as the “situational normality” of the task at hand. This can overlap but is not necessarily the same as generating self-assessments, which are introspective and *process-based* assurances (i.e., which are descriptive and reflective of the AIA’s capabilities); rather, information visualization tends to more broadly include or revolve around *outcome-based* assurances (i.e., which focus on results or expected results of applying the AIA’s capabilities).

3.6.1 Common Approaches. Liu et al. [88] review several of the current methods that exist for visualizing machine learning models. They identified three main purposes for which visualizations are useful in this context: (1) understanding (why models behave the way they do on certain problems), (2) diagnosis (failures or unexpected behavior on certain tasks), and (3) refinement (ability to improve performance on tasks). We consider two common methods that assist in these processes: dimensionality reduction and visualization of uncertainty.

Dimensionality Reduction. Dimensionality reduction (DR) is one of the key methods used in creating visualizations. Sacha et al. [118] identify seven different methods by which users interact with DR techniques. They use this to make the human-in-the-loop process model for interactive DR that is shown in Figure 8. This interactive nature of their model helps users to better understand the information that they are viewing.

Venna [138] discusses DR for ML and reviews many linear and non-linear projection methods. Vellido et al. [137] also discusses the importance of DR for making ML models interpretable. As one example, Chipman and Gu [22] applied this idea by constraining principle component analysis (PCA) in an attempt to make the resulting linear combinations of variables more interpretable (more homogeneous or more sparse).

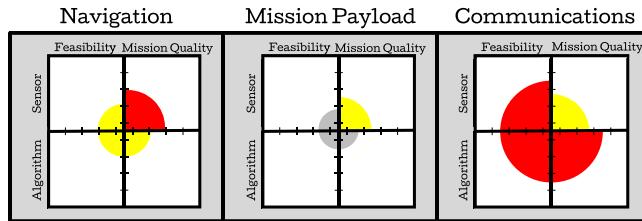


Fig. 9. Proposed “trust annunciator panel” [61] Hutchins, Andrew R., et al., Proceedings of the Human Factors and Ergonomics Society Annual Meeting, “Representing Autonomous Systems’ Self-Confidence through Competency Boundaries” Vol. 59, Issue 1, pp. 279–83. copyright ©2015 by SAGE Journals, Reprinted by Permission of SAGE Publications, Inc.

At times, a simple visualization is the most efficient way to communicate the results of decision-making or planning. For example: Chadalavada et al. [19] enable a robot to project its path onto the ground so users can see.

Treatment of Uncertainty. In the previous section, we have already visited the importance of an AIA being able to quantify its uncertainty. Visualization researchers are concerned with how to *convey* that uncertainty to human users (and quantify uncertainty inherent in making visualizations). Sacha et al. [117] discuss how the propagation of uncertainty through visual analytics systems can affect the trust of human users (see also Ref. [26]). One excellent example of this is the work by Wu et al. [146], who create a tool to visualize the flow and propagation of uncertainty in the visualization process. In this way, users can understand where uncertainty enters the data visualization process.

The relationship between systemic uncertainties and their effects on system performance can be very complex. Hutchins et al. [61] address this by using expert knowledge and a “trust annunciator panel” (TAP) that has several “uncertainty level indicators” in order to display how uncertainties in sensors will effect the output quality and the mission impact; the same goes for the planning algorithm (see Figure 9).

3.6.2 Grounding Example. In the case of the “VIP Escort” problem (described in Section 2.2), information visualization might be used as an assurance in the following way, starting with the assumptions that:

- The UGV has just begun an attempt to escape the road-network.
- The user has access to an interface like that proposed in Ref. [61].

During the attempt, the user is able to see how the sensor uncertainty might possibly effect the outcome of the mission. In this case, the user is assured that the sensors will have little negative impact on the outcome of the mission given the current weather conditions.

Discussion of Example: Here we see how a visualization is able to assist the user in correlating the effects between sensor uncertainty and mission outcome. This is not a simple relationship for operators (especially untrained) to learn on their own; even if they were able to learn, the time required to do so can be very detrimental.

3.7 User Assessment

In this section, we address assurances that are based solely on user assessment; in other words, the AIA expends little or no computational effort to “digest data” to turn into assurance information for the user, and instead relies on the user’s own cognitive abilities to draw assurances from their observations. Such assurances are clearly not integral to the function of the AIA, as they might,

for example, be realized by incorporating simple displays, print statements, or other “raw data” indicators into a basic user interface. This category is in contrast to Section 3.6, where the AIA is designed to process data to assist the user in understanding how to trust the AIA appropriately. This approach is enticing for many system designers given how easy it is to implement at any stage of AIA design (even as an after-thought). However, the effectiveness of this approach rests on several, strong assumptions:

- The user can form a “good enough” mental model of the AIA on their own to inform appropriate TRBs;
- Different users have “similar enough” capabilities and experiences to draw appropriate inferences on their own;
- There are no other compelling sources of information that will confound the assurance;
- Common cognitive biases won’t interfere with the long-term operation/supervision of the system (e.g., recency, framing, or anchoring effects that skew user’s perception of non-linear changes in performance variables like power/fuel consumption).

The weight of each of these assumptions relies heavily on the task to be performed and the characteristics of the typical users. For example, in situations with highly trained personnel (i.e., military or manufacturing facility), all users will have a similar level of capability; thus, “user assessment” is a viable and effective solution. In other scenarios with more diverse users and operating environments, these assumptions begin to break down (i.e., mass market consumer products).

3.7.1 Common Approaches. As suggested in the section’s name, users can form assurances by any method of perception. The most commonly investigated approaches are: simple, visual, “display of raw data,” and “by inspection” performance-based assessment.

Display of Raw Data. Assurances associated with displaying AIA performance variables sound banal (e.g., flow rate for an automated pump [102]), but they actually make use of a nuanced point: the displayed performance value actually serves to inform the user’s own mental model of the trustworthiness of an AIA capability. That is, the user’s trust in the AIA’s capability does not change only in response to the instantaneous “goodness/badness” of the AIA’s performance, but accounts for the past history of the AIA’s performance as well as any observed discrepancies between the AIA’s expected behavior and its actual behavior. The user’s trust dimensions (“competence,” “predictability,” etc.) are then affected by their perception of trustworthiness according to the combined model and data delivered by the display. This approach (also noted and discussed by Refs [61], [120], and [144]) is effective but relies heavily on the implicit assumption that the user will create a “good enough statistical model” of the AIA’s behavior from data presented by the AIA. With this in mind, one might train a user to recognize signs of failure/success in different interactions with an AIA as assurances [32, 39, 119]. The main drawback of this idea is that it still relies on users’ ability to construct “good enough” mental models of AIA behavior and characteristics from noisy observations to avoid misinterpreting AIA behaviors. It can also require intensive and costly special effort for non-expert or non-specialist users. A more ideal approach in such cases would be to design explicit assurances that help users construct correct/consistent mental process models of AIA behavior and thus reduce the risk of misinterpretation.

Performance-based. Users can also be assured by directly assessing the performance of an AIA on their own without any additional aiding or prompting. Put simply: *making stuff that (obviously) doesn’t break improves trust*. Riley [113] investigated how reliability and workload affected the participant’s likelihood of trusting in automation. Two simulated environments were created to this end. First was for participants to use/not use an automated aid (with variable reliability) to

classify characters while also performing a distraction task. Interestingly, they found that pilots (those with extensive experience working with automated systems) had a bias to use more automation but reacted similarly to students in the face of dynamic reliability changes.

In a similar vein, Desai et al. [32] investigated the effects of robot reliability on the trust of human operators. In this case, a human participant needed to work with an autonomous robot to search for victims in a building while avoiding obstacles. The operator had the ability to switch the robot from manual (teleoperated) mode, to semi-autonomous, or autonomous mode depending on how they thought they could trust the system to perform. During this experiment, the reliability of the robot was changed in order to observe the effects on the operator's reliance to the robot. Trust was measured by the amount of time the robot spent in different levels of autonomy (i.e., manual vs. autonomous), and it was found that trust changed based on the levels of reliability of the robot. Yu et al. [147] also had similar findings in their study of operators utilizing an "automatic quality monitor."

3.7.2 Grounding Example. In the case of the "VIP Escort" problem (described in Section 2.2), user assessment might be used as an assurance in the following way, starting with the assumptions that:

- The UGV has just begun an attempt to escape the road-network.
- The user can observe the location of the UGV on the road network.
- The user has access to the speedometer of the UGV.
- The user has been trained and understands how the UGV functions.

As the user monitors the UGVs progress, they notice that, on a particular stretch of road, the speedometer reading seems very high, and the UGV stops moving. They recall from training that in situations where the speedometer shows a high speed and the UGV isn't moving, it is likely that the UGV is spinning out or high-centered. They are able to diagnose the failure and dispatch the appropriate assistance.

Discussion of Example: In this case, the user was able to diagnose a problem based on the UGV not moving and the speedometer being high. They were able to do so because they were familiar with the system and were trained to be able to recognize this kind of situation. In future interaction, the user *might* associate the failure to certain characteristics of the road or other properties of the task...or just feel like the UGV isn't very competent.

4 FUTURE WORK

The formal design of algorithmic assurances is still an emerging field. Consequently, there are many opportunities for further research along different lines. This section outlines some possible promising directions for future work.

4.1 Properties of Assurances

Figure 4 gives some hints about how designers might be able to fully characterize the properties of AIA assurances. In this survey, we investigate in some detail "Level of Integration." However, all of the other grayed-out boxes in Figure 4 have open questions that should still be investigated.

Source-Target Classification. It would be especially convenient for designers of AIAs to be able to refer to assurances by way of their source AIA capability (see Figure 2) and their target user trust dimension (see Figure 3). For example, human-like gestures could be considered a "motion-predictability" assurance. For instance, the reader may be able, in retrospect, to identify work from Ref. [35] as a "motion-predictability" assurance, while the work by Wang et al. [141] could be

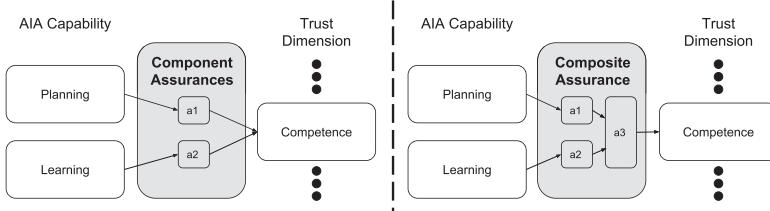


Fig. 10. Component vs. composite assurances: the combination of multiple component assurances into a single assurance is a composite assurance. On the left are two component assurances a_1 and a_2 ; on the right, only a_3 affects the trust dimension.

considered to describe “perception-competence” and “planning-predictability” assurances (among others). Meanwhile, Aitken et al. [6] considered a large set of assurances that span several source capabilities and target trust dimensions. However, it is not always easy to clearly separate AIA capabilities or trust dimensions due to the inherent cross-over.

Still, such classifications are useful because different classes of algorithms will likely present themselves as useful in applications for which assurances must target “predictability” dimensions of trust, for example, as opposed to “situational normality.” Given the inherent difficulty of precisely modeling and measuring trust, it is not immediately clear how such mappings can be precisely delineated. Future research might begin by looking for missing correspondences or correlations in the literature for notional capability source-trust target pairs. For example, have satisfactory assurances been developed for the “learning-situational-normality” source-target pair? Or, to what extent (if any) can assurance x for “perception-competence” also be applied to “learning-competence”? Finally, are there certain classes of algorithms that are suited for communicating to the “predictability” dimension of trust, and can they be adapted from one AIA capability to another?

Component and Composite. A component assurance is an assurance that describes a single AIA capability (or one aspect of an AIA model/process for a capability), and targets one or more trust dimension targets. Component assurances are the most well-researched in the existing literature. A component assurance might include displaying the confidence of a classification prediction, or visualizing a model. A composite assurance is the combination of more than one component assurance into a single assurance. A notable example is the machine self-confidence work by Aitken [5], which notionally combines five component assurances into a single composite assurance interpretable to non-experts.

Figure 10 illustrates the concepts of component and composite assurances. Some open questions here are: how can component assurances generally be combined to create a composite one? Also, to what extent can component and composite assurances be used in concert to provide assurances for users of differing expertise? And, are the effects of a composite assurance equal to the sum of its components?

Explicit and Implicit Assurances. This work has only considered *designed algorithmic assurances*. However, users will always form some kind of trust relationship to an AIA, even if deliberately designed assurances are not available. In the absence of designed assurances, user trust is informed by *implicit* undesigned assurances. These can be thought of as artifacts or side-effects of other design decisions not meant to directly influence user trust.

Why is it important to consider implicit assurances? There is always a danger that users attend to the “wrong” assurances, i.e., AIA features that are not meant to be interpreted as assurances but are nevertheless easily perceived as such (possibly more so than intended explicit assurances). For

example, a designer may create a planning-predictability assurance for an autonomous wheeled mobile robot, which could be rendered ineffective by an implicit assurance given by the appearance of that robot, e.g., the user may trust it less if the robot has old tires or has a large tool attached to its front end which makes it "look unsafe."

It remains an open question as to how designers can identify and mitigate the impact of implicit assurances, especially so that they do not confound the intended effects of explicit assurances. User studies will undoubtedly be helpful in obtaining feedback about which AIA characteristics most affect user trust, e.g., if explicit assurances are being perceived, and if there are implicit assurances whose effects overwhelm those of explicitly designed assurances. With such feedback, designers would have a realistic idea about whether their explicitly designed assurances are having the desired effect on user TRBs. However, the design and analysis of this issue remains open for further study, and is likely to have many application-specific dependencies (though, in the spirit of this article, cross-domain comparisons would also likely prove valuable).

Tutoring vs Telling. Assurances investigated to date are largely designed for one-way "telling" of information, i.e., that they do not consider and adapt to the experience or other traits of different users. The ability to adapt to different users and tutor them to appropriately trust AIAs will become more critical as time passes due to the diversity of user bases for advanced AIAs and time that users will spend interacting with them on complex tasks. A tutoring assurance might, for example, be a planned dynamic sequence of assurances that would change in time to adapt to the user's needs via two way user-AIA communication. This might include modification of assurances to help a user avoid boredom or fatigue in long-duration applications requiring user supervision, or to use the system differently in varying circumstances. It is not surprising that, to our knowledge, no research has been done with respect to tutoring a user in a trust relationship. This is a complex problem that requires understanding how different users learn and identifying potential strategies for eliciting appropriate TRBs from them. However, many interesting avenues for pursuing these ideas may come from the work on educational tutoring systems [143] and algorithmic teaching [9].

4.2 Trust and Distrust

The treatment of assurances in this survey is based, in part, on a model of interpersonal trust. For completeness, it will be important to further investigate *distrust*, as reviewed and discussed by Lewicki et al. [82], and formalized in McKnight and Chervany [93]. Low trust is not the same as distrust, and low distrust is not the same as trust. McKnight and Chervany [93] suggest that "the emotional intensity of distrust distinguishes it from trust," and they explain that distrust comes from emotions like wariness, caution, and fear—whereas trust stems from emotions like hope, safety, and confidence. Trust and distrust are orthogonal elements that define a person's TRB toward a trustee. Since distrust was not considered here, it is not clear to what extent the human-AIA trust model remains effective in the presence of user wariness, caution, or fear. Questions for future work include: to what extent can behaviors driven by distrust be isolated from those originating from trust? How can those behaviors be detected to begin with? And in what circumstances is the extra effort necessary?

4.3 Human Limitations

Dealing with human users requires consideration of their cognitive limitations. For instance, cognitive biases known as "framing effects" (reacting to the same choice in different ways depending on how it is presented) will be important to consider for designing usable AIAs that must make decisions under uncertainty [39, 113]. The existence of framing effects are not surprising to those familiar with cognitive science, but they will likely be unanticipated phenomena to many AIA

system designers. Other related cognitive biases and limitations such as “recency effects” (being biased in making choices based on recent experience), “focusing effects” (being biased in choice selection based on a single aspect of a correlated event), or “normalcy biases” (failure to consider situations that have never occurred before) are also important to consider.

Besides cognitive biases, humans are also limited in their ability to understand certain kinds of information. Communities that investigate how probabilistic and statistical explanations can be presented to humans will have many insights that are relevant for AIA designers and assurance design [74, 89, 114, 124, 140]. But it is not immediately clear what methods are most appropriate for application in assurance design, or how they might be applied. For instance, can the AIA detect when cognitive limitations are effecting TRBs? What other user limitations need to be characterized?

4.4 Expression and Perception of Assurances

Although specific algorithms can be used to build the contents of assurances, it is also critical to consider the actual communication of assurances. The expression (and subsequent perception) of an assurance involves considering mediums, methods, and efficacy. The medium of an assurance includes the form in which it is expressed, e.g., visually, audibly, or otherwise. The method of expression includes, for example, using a plot or a natural language phrase (which could be text-based or speech-based, depending on the medium). Finally, the factors influencing the efficacy of the assurance must also be considered (e.g., consider using an audible assurance in a noisy environment). Humans generally utilize different methods/mediums when communicating assurances to each other to maintain efficacy when potential “losses in transfer” might occur. However, arguably the greatest challenge in using different mediums and methods is not in their implementation, but in designing the ability to recognize and decide when they should be applied. Some interesting questions are: In what circumstances are different methods most useful? And the same for mediums? How can different methods/mediums be selected in order to maximize assurance efficacy while also taking into account that using all possible combinations will *not* help the user? How, and to what extent, can AIAs assess the efficacy of an assurance before, during, or after operation?

4.5 Observing Effects of Assurances

Since assurances are meant to influence TRBs, it is important to quantify these effects so that: (1) the AIA system designer can understand how effective the assurances actually are; and (2) the AIA can evaluate the efficacy of its assurances and adapt them as needed. To our knowledge, there has not been any work that enables an AIA to observe user responses to assurances and then adapt behaviors appropriately (at least not in the trust cycle setting).

There are two known approaches to measuring the effects of assurances: gathering self-reported changes [67, 92, 102, 119, 144] and measuring changes in TRBs [8, 32, 39, 119, 145]. Measuring changes in TRBs is the more objective approach, generally speaking, but the choice between one method and the other depends on the application. Still, more investigation is needed to identify the *principles* behind measuring the effects of assurances. Some interesting, yet unanswered questions include: are there some TRB measurement strategies that fare better than others for particular kinds of applications or assurances? In what ways, if any, do these methods need to be adapted to suit different kinds of users? Is it possible to show that there are, in fact, causal relations from specific assurances to specific TRBs?

5 CONCLUSIONS

The issues of user trust in AIAs and appropriate deployment/use of AIAs have become very prominent. Assurances are the method by which AIAs can influence humans to trust and (more

importantly) *use* them appropriately. We have presented here a definition, case for, and survey of algorithmic assurances in the context of human-AIA trust relationships. A formal treatment of this topic is necessary because the ecosystem of AIAs is evolving more rapidly than ever before; consequently, previous informal approaches to designing algorithmic assurances are insufficient.

This survey was performed, to some extent, from a standpoint of designing intelligent unmanned vehicle systems that must work in concert with a human supervisor. However, the theoretical framework and categorization of assurances is meant to be generally applicable to a broad range of AIAs. A major motivation for this survey was the observation that there are many researchers in different but related domains such as human factors, robotics, machine learning, artificial intelligence, and others who are (unknowingly) working along different parts of the same human-AIA assurance spectrum. It is important for members of each community to recognize this, so that research efforts can be methodically organized to answer related open questions in this important area. Assurances have historically been ignored from a practical standpoint and are the least understood component of human-AIA trust relationships. There have been many researchers who have recognized the concepts behind assurances, but no detailed definitions have been given until now.

There are three main contributions from this work: (1) we have drawn from multiple bodies of research in order to fill in the missing details for the human-AIA trust cycle (Figure 1) and to formally define assurances within this cycle; (2) we present a classification of assurances in Section 2.1; (3) we identify an “assurance integration continuum” shown in Figure 6. On that continuum, seven different classes of algorithms were identified. Practitioners can use these classes to select and design assurances for AIAs. Given the material provided herein, those who design assurances should have the tools required to approach design and future research from a solid theoretical foundation.

A final important and sobering takeaway is that there is not a single “silver bullet” algorithmic assurance that will perform the best in all situations. Given enough time, it is quite possible that highly specialized assurances could be designed for many situations. Even so, we warn that, for the research and design of assurances to be sustainable in the current environment of fast-paced development of new technology, it is important to consider approaches that are as principally grounded as possible, in order to be more easily used with yet-to-be-invented methods for implementing various AIA capabilities. We have identified many future opportunities for research on AIA assurance design and their influence on human trust, and hope researchers will begin looking outside of their own disciplines to discover, design, and formally test new tools and ideas for assurance design and implementation. The framework presented here should unify research efforts by providing a common taxonomy in relation to human-AIA trust relationships. We believe it will help researchers see the field from a larger perspective, classify the type of research they are performing, and consider the greater implications of their work. The field of algorithmic assurances has an abundance of avenues for new and challenging research, and we encourage researchers to pursue them.

REFERENCES

- [1] Behnoush Abdollahi and Olfa Nasraoui. 2016. Explainable restricted Boltzmann machines for collaborative filtering. (June 2016). arxiv:stat.ML/1606.07129
- [2] Behnoush Abdollahi and Olfa Nasraoui. 2018. Transparency in fair machine learning: The case of explainable recommender systems. In *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*, Jianlong Zhou and Fang Chen (Eds.). Springer International Publishing, Cham, 21–35.
- [3] Henny Admoni, Thomas Weng, Bradley Hayes, and Brian Scassellati. 2016. Robot nonverbal behavior improves task performance in difficult collaborations. In *Proceedings of the 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI'16)*. IEEE Press, Piscataway, NJ, 51–58.

- [4] Nisar Ahmed, David Casbeer, Yongcan Cao, and Derek Kingston. 2017. Multitarget localization on road networks with hidden Markov Rao–Blackwellized particle filters. *J. Aerosp. Inf. Syst.* 14, 11 (Sept. 2017), 573–596.
- [5] Matthew Aitken. 2016. *Assured Human-Autonomy Interaction through Machine Self-Confidence*. Master’s thesis. University of Colorado at Boulder.
- [6] Matthew Aitken, Nisar Ahmed, Dale Lawrence, Brian Argrow, and Eric Frew. 2016. Assurances and machine self-confidence for enhanced trust in autonomous systems. In *Proceedings of the RSS 2016 Workshop on Social Trust in Autonomous Systems*.
- [7] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. (June 2016). arxiv:cs.AI/1606.06565.
- [8] Wilma A. Bainbridge, Justin W. Hart, Elizabeth S. Kim, and Brian Scassellati. 2011. The benefits of interactions with physically present robots over video-displayed agents. *Adv. Robot.* 3, 1 (Jan. 2011), 41–52.
- [9] Frank J. Balbach and Thomas Zeugmann. 2009. Recent developments in algorithmic teaching. In *Language and Automata Theory and Applications*. Springer, Berlin, 1–18.
- [10] Guru Banavar. 2016. What It Will Take for Us to Trust AI. Retrieved February 16, 2017 from <https://hbr.org/2016/11/what-it-will-take-for-us-to-trust-ai>.
- [11] Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 8 (Aug. 2013), 1798–1828.
- [12] Rob Bensinger. 2014. Stuart Russell: AI value alignment problem must be an “intrinsic part” of the field’s mainstream agenda. Retrieved June 20, 2018 from <https://www.lesswrong.com/posts/S95qCHBxtASmYyGSs/stuart-russell-ai-value-alignment-problem-must-be-an>.
- [13] L. P. Berczi, I. Posner, and T. D. Barfoot. 2015. Learning to assess terrain from human demonstration using an introspective Gaussian-process classifier. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*. 3178–3185.
- [14] Nick Bostrom. 2014. *Superintelligence: Paths, Dangers, Strategies*. OUP Oxford.
- [15] Jeffrey M. Bradshaw, Robert R. Hoffman, David D. Woods, and Matthew Johnson. 2013. The seven deadly myths of “Autonomous Systems.” *IEEE Intell. Syst.* 28, 3 (2013), 54–61.
- [16] Eric Brochu, Vlad M. Cora, and Nando de Freitas. 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. (Dec. 2010). arxiv:cs.LG/1012.2599.
- [17] R. Calinescu, D. Weyns, S. Gerasimou, M. U. Iftikhar, I. Habli, and T. Kelly. 2017. Engineering trustworthy self-adaptive software with dynamic assurance cases. *IEEE Trans. Software Eng. PP*, 99 (2017), 1–1.
- [18] Davide Castelvecchi. 2016. Can we open the black box of AI? *Nature* 538, 7623 (Oct. 2016), 20–23.
- [19] R. T. Chadalavada, H. Andreasson, R. Krug, and A. J. Lilienthal. 2015. That’s on my mind! Robot to human intention communication through on-board projection on shared floor space. In *Proceedings of the 2015 European Conference on Mobile Robots (ECMR’15)*. 1–6.
- [20] Joseph Chee Chang, Saleema Amershi, and Ece Kamar. 2017. Revolt: Collaborative crowdsourcing for labeling machine learning datasets. In *Proceedings of the 2017 Conference on Human Factors in Computing Systems (CHI’17)*. ACM, New York, NY, 2334–2346.
- [21] Moses Charikar, Jacob Steinhardt, and Gregory Valiant. 2017. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC’17)*. ACM, New York, NY, 47–60.
- [22] Hugh A. Chipman and Hong Gu. 2005. Interpretable dimension reduction. *J. Appl. Stat.* 32, 9 (Nov. 2005), 969–987.
- [23] Edward Choi, Mohammad Taha Bahadori, Joshua A. Kulas, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2016. RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. (Aug. 2016). arxiv:cs.LG/1608.05745.
- [24] Sungjoon Choi, Kyungjae Lee, Sungbin Lim, and Songhwai Oh. 2017. Uncertainty-aware learning from demonstration using mixture density networks with sampling-free variance modeling. (Sept. 2017). arxiv:cs.CV/1709.02249.
- [25] W. Churchill, Chi Hay Tong, C. Guräu, I. Posner, and P. Newman. 2015. Know your limits: Embedding localiser performance models in teach and repeat maps. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*. 4238–4244.
- [26] C. D. Correa, Y. H. Chan, and K. L. Ma. 2009. A framework for uncertainty-aware visual analytics. In *Proceedings of the 2009 IEEE Symposium on Visual Analytics Science and Technology*. 51–58.
- [27] Cynthia L. Corritore, Beverly Kracher, and Susan Wiedenbeck. 2003. On-line trust: Concepts, evolving themes, a model. *Int. J. Hum. Comput. Stud.* 58, 6 (2003), 737–758.
- [28] William Curran, Cameron Bowie, and William D. Smart. 2016. POMDPs for risk-aware autonomy. In *Proceedings of the 2016 AAAI Fall Symposium Series*.
- [29] Sébastien Da Veiga and Amandine Marrel. 2012. Gaussian process modeling with inequality constraints. In *Annales de la Faculté des Sciences de Toulouse*, Vol. 21. 529–555.

[30] Ewart J. de Visser, Richard Pak, and Tyler H. Shaw. 2018. From “automation” to “autonomy”: The importance of trust repair in human-machine interaction. *Ergonomics* (March 2018), 1–33.

[31] J. Dequaire, C. H. Tong, W. Churchill, and I. Posner. 2016. Off the beaten track: Predicting localisation performance in visual teach and repeat. In *Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA)*. 795–800.

[32] M. Desai, M. Medvedev, M. Vázquez, S. McSheehy, S. Gadea-Omelchenko, C. Bruggeman, A. Steinfeld, and H. Yanco. 2012. Effects of changing reliability on trust of robot systems. In *Proceedings of the 2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 73–80.

[33] David DeSteno. 2014. Can You Trust Technology? Retrieved June 14, 2017 from http://www.huffingtonpost.com/david-desteno/can-you-trust-technology_b_4683614.html.

[34] Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. (Feb. 2017). arxiv:stat.ML/1702.08608.

[35] Anca Dragan and Siddhartha Srinivasa. 2013. Generating legible motion. In *Proceedings of Robotics: Science and Systems*.

[36] Mary T. Dzindolet, Scott A. Peterson, Regina A. Pomranky, Linda G. Pierce, and Hall P. Beck. 2003. The role of trust in automation reliance. *Int. J. Hum. Comput. Stud.* 58, 6 (2003), 697–718.

[37] James H. Faghmous and Vipin Kumar. 2014. A big data guide to understanding climate change: The case for theory-guided data science. *Big Data* 2, 3 (Sept. 2014), 155–163.

[38] Katherine Ellen Foley. 2017. A pioneering computer scientist wants algorithms to be regulated like cars, banks, and drugs. Retrieved June 5, 2017 from <https://qz.com/998131/>.

[39] A. Freedy, E. DeVisser, G. Weltman, and N. Coeyman. 2007. Measurement of trust in human-robot collaboration. In *Proceedings of the 2007 International Symposium on Collaborative Technologies and Systems*. 106–114.

[40] Alex A. Freitas. 2006. Are we really discovering interesting knowledge from data. *Expert Update (the BCS-SGAI magazine)* 9, 1 (2006), 41–47.

[41] Francis Fukuyama. 1995. *Trust: The Social Virtues and the Creation of Prosperity*. Free Press Paperbacks.

[42] Yarin Gal. 2016. *Uncertainty in Deep Learning*. University of Cambridge.

[43] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the International Conference on Machine Learning*. 1050–1059.

[44] J. Garcia and F. Fernández. 2015. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.* 16, 1 (2015), 1437–1480.

[45] Shalini Ghosh, Patrick Lincoln, Ashish Tiwari, and Xiaojin Zhu. 2016. Trusted machine learning for probabilistic models. *International Conference on Machine Learning (ICML)*. <https://pdfs.semanticscholar.org/bca5/34070de2057728860d60a12fd92944798ecb.pdf>.

[46] Google. 2018. Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone. Retrieved June 15, 2018 from <http://ai.googleblog.com/2018/05/duplex-ai-system-for-natural-conversation.html>.

[47] G. Gordon Worley, III. 2018. Formally Stating the AI Alignment Problem—Map and Territory. Retrieved June 20, 2018 from <https://mapandterritory.org/formally-stating-the-ai-alignment-problem-fe7a6e3e5991>.

[48] H. Grimmett, R. Paul, R. Triebel, and I. Posner. 2013. Knowing when we don’t know: Introspective classification for mission-critical decision making. In *Proceedings of the 2013 IEEE International Conference on Robotics and Automation*. 4531–4538.

[49] Hugo Grimmett, Rudolph Triebel, Rohan Paul, and Ingmar Posner. 2016. Introspective classification for robot perception. *Int. J. Rob. Res.* 35, 7 (2016), 743–762.

[50] David Gunning. 2017. Explainable artificial intelligence (XAI). *Defense Advanced Research Projects Agency (DARPA), nd Web* (2017).

[51] Corina Gurău, Chi Hay Tong, and Ingmar Posner. 2016. Fit for purpose? Predicting perception performance based on past experience. In *Proceedings of the 2016 International Symposium on Experimental Robotics*. Springer, Cham, 454–464.

[52] Dan Gutfreund, Yoav Katz, and Noam Slonim. 2016. Automatic arguments construction—From search engine to research engine. In *Proceedings of the 2016 AAAI Fall Symposium Series*.

[53] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *J. Mach. Learn. Res.* 3, (Mar.2003), 1157–1182.

[54] J. D. F. Habbema. 1976. Models for diagnosis and detection of combinations of diseases. *Decision Making and Medical Care: Can Information Science Help?: Proceedings of the IFIP Working Conference on Decision Making and Medical Care* (1976), 399–411.

[55] Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. 2016. The off-switch game. (Nov. 2016). arxiv:cs.AI/1611.08219.

[56] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J. Russell, and Anca Dragan. 2017. Inverse reward design. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 6765–6774.

[57] Anne-Claire Haury, Pierre Gestraud, and Jean-Philippe Vert. 2011. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PLoS One* 6, 12 (Dec. 2011), e28210.

[58] Bradley Hayes and Julie A. Shah. 2017. Improving robot controller transparency through autonomous policy explanation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction (HRI'17)*. ACM, New York, NY, 303–312.

[59] Sandy H. Huang, David Held, Pieter Abbeel, and Anca D. Dragan. 2017. Enabling robots to communicate their objectives. (Feb. 2017). arxiv:cs.RO/1702.03465.

[60] Laura Humphrey. 2012. Model checking UAV mission plans. In *Proceedings of the AIAA Modeling and Simulation Technologies Conference*. American Institute of Aeronautics and Astronautics.

[61] Andrew R. Hutchins, M. L. Cummings, Mark Draper, and Thomas Hughes. 2015. Representing autonomous systems' self-confidence through competency boundaries, Vol. 59. SAGE Publications, 279–283.

[62] Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. 2011. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decis. Support Syst.* 51, 1 (April 2011), 141–154.

[63] Brett W. Israelsen, Nisar Ahmed, Kenneth Center, Roderick Green, and Winston Bennett Jr. 2017. Towards adaptive training of agent-based sparring partners for fighter pilots. In *InfoTech@Aerospace Conference*. Grapevine, TX, 1–15.

[64] Milos Jovanovic, Sandro Radovanovic, Milan Vukicevic, Sven Van Poucke, and Boris Delibasic. 2016. Building interpretable predictive models for pediatric hospital readmission using Tree-Lasso logistic regression. *Artif. Intell. Med.* 72 (2016), 12–21.

[65] Gregory Kahn, Adam Villaflor, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. 2017. Uncertainty-aware reinforcement learning for collision avoidance. (Feb. 2017). arxiv:cs.LG/1702.01182.

[66] Krishnanand N. Kaipa, Akshaya S. Kankanhalli-Nagendra, and Satyandra K. Gupta. 2015. Toward estimating task execution confidence for robotic bin-picking applications. In *Proceedings of the 2015 AAAI Fall Symposium Series*.

[67] Poornima Kaniarasu, Aaron Steinfeld, Munjal Desai, and Holly Yanco. 2013. Robot confidence and trust alignment. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI'13)*. IEEE Press, Piscataway, NJ, 155–156.

[68] Anuj Karpatne, Gowtham Atluri, James Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. 2016. Theory-guided data science: A new paradigm for scientific discovery. (Dec. 2016). arxiv:cs.LG/1612.08544.

[69] T. Kaupp and A. Makarenko. 2008. Measuring human-robot team effectiveness to determine an appropriate autonomy level. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, 2146–2151.

[70] Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). 5574–5584.

[71] Nguyen Lu Dang Khoa, Mehrisadat Makki Alamdari, Thierry Rakotoarivelo, Ali Anaissi, and Yang Wang. 2018. Structural health monitoring using machine learning techniques and domain knowledge based features. In *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*, Jianlong Zhou and Fang Chen (Eds.). Springer International Publishing, Cham, 409–435.

[72] Bijan Khosravi. 2016. Will You Trust AI to Be Your New Doctor? Retrieved February 16, 2017 from <http://www.forbes.com/sites/bijankhosravi/2016/03/24/will-you-trust-ai-to-be-your-new-doctor-a-five-year-outcome/>.

[73] Mykel J. Kochenderfer. 2015. *Decision Making Under Uncertainty: Theory and Application*. MIT Press.

[74] Kristine M. Kuhn. 1997. Communicating uncertainty: Framing effects on responses to vague probabilities. *Organ. Behav. Hum. Decis. Process.* 71, 1 (July 1997), 55–83.

[75] Ugur Kuter and Chris Miller. 2015. Computational mechanisms to support reporting of self confidence of automated/autonomous systems. In *Proceedings of the 2015 AAAI Fall Symposium Series*.

[76] Minae Kwon, Sandy H. Huang, and Anca D. Dragan. 2018. Expressing robot incapability. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction (HRI'18)*. ACM, New York, NY, 87–95.

[77] Carmen Lacave and Francisco J. Díez. 2002. A review of explanation methods for Bayesian networks. *Knowl. Eng. Rev.* 17, 2 (June 2002), 107–127.

[78] A. Lacher, R. Grabowski, and S. Cook. 2014. Autonomy, trust, and transportation. In *Proceedings of the 2014 AAAI Spring Symposium Series*.

[79] Morteza Lahijanian and Marta Kwiatkowska. 2016. Social trust: A major challenge for the future of autonomous systems. In *Proceedings of the AAAI Fall Symposium on Cross-Disciplinary Challenges for Autonomous Systems, AAAI Fall Symposium. AAAI*.

[80] Kathryn Blackmond Laskey. 1991. Conflict and surprise: Heuristics for model revision. In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence (UAI'91)*. Morgan Kaufmann, San Francisco, CA, 197–204.

[81] J. D. Lee and K. A. See. 2004. Trust in automation: Designing for appropriate reliance. *Hum. Factors: J. Hum. Factors Ergon. Soc.* 46, 1 (2004), 50–80.

[82] Roy J. Lewicki, Daniel J. McAllister, and Robert J. Bies. 1998. Trust and distrust: New relationships and realities. *Acad. Manage. Rev.* 23, 3 (1998), 438–458.

[83] Roy J. Lewicki, Edward C. Tomlinson, and Nicole Gillespie. 2006. Models of interpersonal trust development: Theoretical approaches, empirical evidence, and future directions. *J. Manage.* 32, 6 (Dec. 2006), 991–1022.

[84] J. David Lewis and Andrew Weigert. 1985. Trust as a social reality. *Soc. Forces* 63, 4 (June 1985), 967–985.

[85] Austin Lillard, Eric W. Frew, Brian Argrow, Dale Lawrence, and Nisar Ahmed. 2015. Assurances for enhanced trust in autonomous systems. In *Proceedings of the 2015 AAAI Fall Symposium*.

[86] Zachary C. Lipton. 2016. The mythos of model interpretability. (June 2016). arxiv:cs.LG/1606.03490.

[87] Zachary C. Lipton, Jianfeng Gao, Lihong Li, Jianshu Chen, and Li Deng. 2016. Combating reinforcement learning's Sisyphean curse with intrinsic fear. (Nov. 2016). arxiv:cs.LG/1611.01211.

[88] Shixia Liu, Xiting Wang, Mengchen Liu, and Jun Zhu. 2017. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics* 1, 1 (March 2017), 48–56.

[89] Meghann Lomas, Robert Chevalier, Ernest Vincent Cross, II, Robert Christopher Garrett, John Hoare, and Michael Kopack. 2012. Explaining robot actions. In *Proceedings of the 7th Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI'12)*. ACM, New York, NY, 187–188.

[90] George F. Luger. 2008. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (6th ed.). Addison-Wesley.

[91] D. J. C. MacKay. 1992. Information-based objective functions for active data selection. *Neural Comput.* 4, 4 (July 1992), 590–604.

[92] D. Harrison McKnight, Michelle Carter, Jason Bennett Thatcher, and Paul F. Clay. 2011. Trust in a specific technology: An investigation of its components and measures. *ACM Trans. Manage. Inf. Syst.* 2, 2 (July 2011), 12:1–12:25.

[93] D. Harrison McKnight and Norman Chervany. 2001. While trust is cool and collected, distrust is fiery and frenzied: A model of distrust concepts. *AMCIS 2001 Proceedings* (2001), 171.

[94] D. H. McKnight and N. L. Chervany. 2001. What trust means in E-commerce customer relationships: An interdisciplinary conceptual typology. *Int. J. Electron. Commerce* 6, 2 (2001), 35–59.

[95] D. Harrison McKnight, Larry L. Cummings, and Norman L. Chervany. 1998. Initial trust formation in new organizational relationships. *Acad. Manage. Rev.* 23, 3 (July 1998), 473–490.

[96] D. H. McKnight, Charles J. Kacmar, and Vivek Choudhury. 2004. Dispositional trust and distrust distinctions in predicting high- and low-risk Internet expert advice site perceptions. *e-Service J.* 3, 2 (2004), 35–55.

[97] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger (Eds.), 3111–3119.

[98] Christopher A. Miller. 2014. Delegation and transparency: Coordinating interactions so information exchange is no surprise. In *Virtual, Augmented and Mixed Reality. Designing and Developing Virtual and Augmented Environments*. Springer International Publishing, 191–202.

[99] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. 2018. Never-ending learning. *Commun. ACM* 61, 5 (April 2018), 103–115.

[100] Rebecca E. Morrison, Todd A. Oliver, and Robert D. Moser. 2018. Representing model inadequacy: A stochastic operator approach. *SIAM/ASA J. Uncertainty Quantification* 6, 2 (2018), 457–496.

[101] Bonnie M. Muir. 1994. Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics* 37, 11 (1994), 1905–1922.

[102] B. M. Muir and N. Moray. 1996. Trust in automation. Part II. Experimental studies of trust and human intervention in a process control simulation. *Ergonomics* 39, 3 (March 1996), 429–460.

[103] Nils J. Nilsson. 2009. *The Quest for Artificial Intelligence*. Cambridge University Press.

[104] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. 2018. The building blocks of interpretability. *Distill* 3, 3 (March 2018), e10.

[105] Clemens Otte. 2013. Safe and interpretable machine learning: A methodological review. In *Computational Intelligence in Intelligent Data Analysis*, Christian Moewes and Andreas Nürnberger (Eds.). Springer Berlin, 111–122.

[106] Raja Parasuraman and Victor Riley. 1997. Humans and automation: Use, misuse, disuse, abuse. *Hum. Factors: J. Hum. Factors Ergon. Soc.* 39, 2 (1997), 230–253.

- [107] Yubin Park, Joyce Ho, and Joydeep Ghosh. 2016. ACDC: α -carving decision chain for risk stratification. (June 2016). arxiv:stat.ML/1606.05325
- [108] R. Paul and P. Newman. 2011. Self help: Seeking out perplexing images for ever improving navigation. In *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*. 445–451.
- [109] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. 2009. *Dataset Shift in Machine Learning*. MIT Press.
- [110] Vasumathi Raman, Constantine Lignos, Cameron Finucane, Kenton C. Lee, Mitch Marcus, and Hadas Kress-Gazit. 2013. *Sorry Dave, I'm Afraid I Can't Do That: Explaining Unachievable Robot Tasks Using Natural Language*. Technical Report. University of Pennsylvania, Philadelphia.
- [111] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1135–1144.
- [112] G. Ridgeway, D. Madigan, T. Richardson, and J. O’Kane. 1998. Interpretable boosted naïve Bayes classification. In *KDD*.
- [113] Victor Riley. 1996. *Operator Reliance on Automation: Theory and Data*. Lawrence Erlbaum Associates, Inc.
- [114] William B. Rouse. 1986. Design and evaluation of computer-based decision support systems. In *Microcomputer Decision Support Systems*.
- [115] Stephan Ruping. 2006. *Learning Interpretable Models*. Ph.D. Dissertation. University of Darmstadt.
- [116] Stuart Jonathan Russell and Peter Norvig. 2010. *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall.
- [117] Dominik Sacha, Hansi Seuratne, Bum Chul Kwon, Geoffrey Ellis, and Daniel A. Keim. 2016. The role of uncertainty, awareness, and trust in visual analytics. *IEEE Trans. Vis. Comput. Graph.* 22, 1 (Jan. 2016), 240–249.
- [118] Dominik Sacha, Leishi Zhang, Michael Sedlmair, John A. Lee, Jaakkko Peltonen, Daniel Weiskopf, Stephen C. North, and Daniel A. Keim. 2017. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. Vis. Comput. Graph.* 23, 1 (Jan. 2017), 241–250.
- [119] Maha Salem, Gabriella Lakatos, Farshid Amirabdollahian, and Kerstin Dautenhahn. 2015. Would you trust a (faulty) robot?: Effects of error, task type and personality on human-robot cooperation and trust. In *Proceedings of the 10th Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI’15)*. ACM, New York, NY, 141–148.
- [120] Thomas B. Sheridan and Robert T. Hennessy. 1984. *Research and Modeling of Supervisory Control Behavior. Report of a Workshop*. Technical Report. DTIC Document.
- [121] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical Bayesian optimization of machine learning algorithms. In *Adv. Neural Inf. Process. Syst.* 25. 2951–2959.
- [122] IEEE Spectrum. 2016. Can We Trust Robots? Retrieved June 14, 2017 from <http://spectrum.ieee.org/robotics/artificial-intelligence/can-we-trust-robots>.
- [123] Masashi Sugiyama, Makoto Yamada, and Marthinus Christoffel du Plessis. 2013. Learning under nonstationarity: Covariate shift and class-balance change. *Wiley Interdiscip. Rev. Comput. Stat.* 5, 6 (2013), 465–477.
- [124] William R. Swartout. 1983. XPLAIN: A system for creating and explaining expert consulting programs. *Artif. Intell.* 21, 3 (Sept. 1983), 285–325.
- [125] N. Sweet and N. Ahmed. 2016. Structured synthesis and compression of semantic human sensor models for Bayesian estimation. In *Proceedings of the 2016 American Control Conference (ACC)*. 5479–5485.
- [126] Nicholas Sweet, Nisar R. Ahmed, Ugur Kuter, and Christopher Miller. 2016. Towards self-confidence in autonomous systems. In *AIAA Infotech @ Aerospace*. 1651.
- [127] Daniel Szafir, Bilge Mutlu, and Terrence Fong. 2014. Communication of intent in assistive free flyers. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction (HRI’14)*. ACM, New York, NY, 358–365.
- [128] Daniel Szafir, Bilge Mutlu, and Terry Fong. 2015. Communicating directionality in flying robots. In *Proceedings of the 10th Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI’15)*. ACM, New York, NY, 19–26.
- [129] Colin Tankard. 2016. What the GDPR means for businesses. *Network Security* 2016, 6 (June 2016), 5–8.
- [130] Stefanie Tellex, Ross Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. 2014. *Asking for Help Using Inverse Semantics*. Robotics: Science and Systems Foundation.
- [131] Rudolph Triebel, Hugo Grimmel, Rohan Paul, and Ingmar Posner. 2013. Introspective active learning for scalable semantic mapping. In *Workshop. Robotics Science and Systems (RSS)*. 809–816.
- [132] Rudolph Triebel, Hugo Grimmel, Rohan Paul, and Ingmar Posner. 2016. Driven learning for driving: How introspection improves semantic mapping. In *Robotics Research*, Masayuki Inaba and Peter Corke (Eds.). Springer International Publishing, 449–465.
- [133] Rudolph Triebel, Hugo Grimmel, and Ingmar Posner. 2013. Confidence boosting: Improving the introspectiveness of a boosted classifier for efficient learning. In *Workshop on Autonomous Learing*.
- [134] J. Tripp, D. H. McKnight, and N. K. Lankton. 2011. Degrees of humanness in technology: What type of trust matters? In *AMCIS*.

- [135] Vanya Van Belle and Paulo Lisboa. 2013. Research directions in interpretable machine learning models. In *ESANN*.
- [136] A. J. N. van Breemen. 2004. Bringing robots to life: Applying principles of animation to robots. In *Proceedings of Shaping Human-Robot Interaction Workshop (CHI)*.
- [137] A. Vellido, J. D. Martín-Guerrero, and P. J. G. Lisboa. 2012. Making machine learning models interpretable. *ESANN*, Citeseer. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.431.5382&rep=rep1&type=pdf>.
- [138] Jarkko Venna. 2007. *Dimensionality Reduction for Visual Exploration of Similarity Structures*. Helsinki University of Technology.
- [139] Julian Wagner and Alexander Benecke. 2016. National legislation within the framework of the GDPR. *Eur. Data Prot. L. Rev.* 2 (2016), 353.
- [140] Richard J. Wallace and Eugene C. Freuder. 2001. Explanations for whom. In *CP01 Workshop on User-Interaction in Constraint Satisfaction*.
- [141] Ning Wang, David V. Pynadath, and Susan G. Hill. 2016. Trust calibration within a human-robot team: Comparing automatically generated explanations. In *Proceedings of the 11th ACM/IEEE International Conference on Human Robot Interaction (HRI'16)*. IEEE Press, Piscataway, NJ, 109–116.
- [142] Adrian Weller. 2017. Challenges for transparency. (July 2017). arxiv:cs.CY/1708.01870.
- [143] Etienne Wenger. 2014. *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann.
- [144] Christopher D. Wickens, Rena Conejo, and Keith Gempeler. 1999. Unreliable automated attention cueing for air-ground targeting and traffic maneuvering. *Proc. Hum. Fact. Ergon. Soc. Annu. Meet.* 43, 1 (Sept. 1999), 21–25.
- [145] Jane Wu, Erin Paeng, Kari Linder, Piercarlo Valdesolo, and James C. Boerkoel. 2016. Trust and cooperation in human-robot decision making. In *Proceedings of the 2016 AAAI Fall Symposium Series*.
- [146] Yingcai Wu, Guo-Xun Yuan, and Kwan-Liu Ma. 2012. Visualizing flow of uncertainty through analytical processes. *IEEE Trans. Vis. Comput. Graph.* 18, 12 (Dec. 2012), 2526–2535.
- [147] Kun Yu, Shlomo Berkovsky, Dan Conway, Ronnie Taib, Jianlong Zhou, and Fang Chen. 2018. Do I trust a machine? Differences in user trust based on system performance. In *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*, Jianlong Zhou and Fang Chen (Eds.). Springer International Publishing, Cham, 245–264.
- [148] Eliezer Yudkowsky. 2001. *Creating Friendly AI 1.0: The Analysis and Design of Benevolent Goal Architectures*. The Singularity Institute, San Francisco, CA. (2001).
- [149] Adam Zagorecki, Marcin Kozniewski, and Marek Druzdzel. 2015. An approximation of surprise index as a measure of confidence. In *Proceedings of the 2015 AAAI Fall Symposium Series*.
- [150] Bang Zhang, Ting Guo, Lelin Zhang, Peng Lin, Yang Wang, Jianlong Zhou, and Fang Chen. 2018. Water pipe failure prediction: A machine learning approach enhanced by domain knowledge. In *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*, Jianlong Zhou and Fang Chen (Eds.). Springer International Publishing, Cham, 363–383.
- [151] Peng Zhang, Jiuling Wang, Ali Farhadi, Martial Hebert, and Devi Parikh. 2014. Predicting failures of vision systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3566–3573.
- [152] Grzegorz Zycinski, Margherita Squillario, Annalisa Barla, Tiziana Sanavia, Alessandro Verri, and Barbara Di Camillo. 2012. Discriminant functional gene groups identification with machine learning and prior knowledge. In *ESANN*.

Received November 2017; revised June 2018; accepted August 2018