Safety Beyond Verification: The Need for Continual, User-Driven Assessment of AI Systems

Siddharth Srivastava[1] and **Georgios Fainekos**[2]
[1] Arizona State University
[2] Toyota Motor North America, Research & Development
siddharths@asu.edu, georgios.fainekos@toyota.com

How should we assess the safety and functionality of taskable AI systems that are designed to continually learn and solve user-desired tasks in user-specific environments? From household robotics to digital assistants that can make potentially dangerous changes to their operational environments, this question is central to realizing the promise of AI.

We investigate why answering this question requires more than an extrapolation of existing paradigms for verification and validation, and identify concrete desiderata and promising directions for research on formal assessment of AI systems. Throughout this document, we use the term “AI system” to refer to taskable AI systems that to try to achieve the task that their user has in mind. Such systems are commonly formulated as agents that carry out some form of sequential decision making, a.k.a. planning. Such systems often utilize machine learning to improve their computational performance, although our discussion also applies to AI systems that do not utilize learning.

Conventional Approaches

The vast majority of today’s engineered systems operate in an ecosystem where limited functionality yields safety. Designers play a key role in evaluating safety and defining operational envelopes for systems with narrow scope of functionality. E.g., conventional automobile systems run through various empirical tests and formal verification pipelines. In addition, they are supported by an eco-system of product support, safety and maintenance organizations, all of which make system expertise readily available to non-expert users. Taskable AI systems invalidate both of these conventional avenues for ensuring safe operation.

Verification and Validation Verification and validation (V&V) of systems has a rich history of research and development. These paradigms evaluate whether a given component or system satisfies designer-formulated functional properties such as safe lead distance in adaptive cruise-control (Loos, Platzer, and Nistor 2011; Hasuo et al. 2023). The designers (broadly construed as the team or the organization responsible for creating the product) take the responsibility for designing safety properties, and iterating over system designs to create specifications of expected behavior (possible executions) and safety constraints, and designs that match these specifications.

However, *taskable AI systems are designed to address situations where the designer need not know the objectives that their users may have in mind* – prior knowledge of expected behaviors is even less likely. A system doesn’t need to change after deployment to invalidate the assumption of prior knowledge of expected behaviors. Indeed, taskable AI systems are typically designed to adapt to the environment and compute new behaviors for achieving user-desired tasks even when they are not actively learning and/or changing the algorithms or heuristics used to plan.

As a result, the conventional notions of verification and validation have limited applicability and utility for AI systems. They can still be used to assert and verify physical safety properties that are expected to be maintained across all possible tasks and environments. E.g., robot designers can develop physical safety and operability envelopes for their robot and for specific environments, e.g., maximal accelerations and velocities. While such properties are necessary, they are clearly not sufficient for ensuring safety.

For instance, safety assessment for a hospital robot goes beyond physical movement. It is essential to determine whether it could deliver critical medication to the wrong room, and whether it could be relied upon to assure delivery of life-saving medication in an emergency situation. Knowledge of possible objectives, possible executions or user-specific safety constraints is untenable as an assumption in ensuring the safety of such systems.

Product Support and Maintenance for Safety The current paradigm for safe usability of complex systems relies on an eco-system of product support driven by a diverse body of technicians with low-barriers to entry. If a driver experiences unexpected vibrations while braking, a stop at the local garage can help diagnose and repair possible safety issues. This may be feasible due to the finite number of components and specific functionality and variability among similar products being deployed.

AI systems, on the other hand, are expected to adapt to their environments. With systems changing to meet indiosyncrasies of user-specific tasks and environments, it becomes all but impossible to utilize the economies of scale in product support: A request for determining whether a current system is safe for the task that the user has in mind would result in an infeasible engineering effort V&V of that

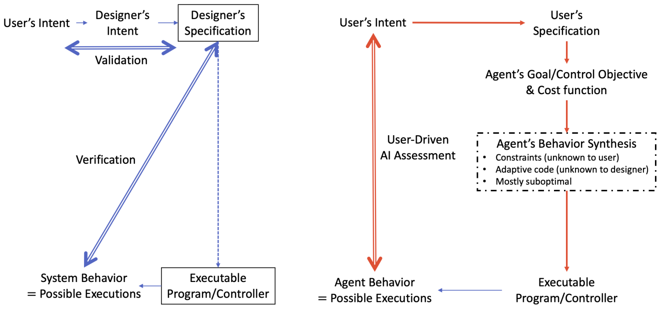
User's Intent
Designer's Intent
Designer's Specification
Validation
Verification
System Behavior = Possible Executions
Executable Program/Controller
User's Intent
User's Specification
Agent's Goal/Control Objective & Cost function
User-Driven AI Assessment
Agent's Behavior Synthesis
Constraints (unknown to user)
Adaptive code (unknown to designer)
Mostly suboptimal
Agent Behavior = Possible Executions
Executable Program/Controller

Figure 1: Conventional system verification (L) and user-driven assessment of AI systems (R). Solid boxes indicate components available at design stage. Dashed boxes indicate components available for systems that don't use learning after deployment.

particular system – an effort whose results would not easily transfer to other instances of the same system.

Continual User-Driven AI Assessment

We argue that the assessment of AI systems needs to address fundamentally different questions that go beyond those addressed in existing paradigms for system evaluation and safety assurance. Fig. 1 illustrates these differences. In the conventional paradigm (shown on the left), the designer plays a central role in transferring users' intent to specifications and ensuring, through formal and empirical methods that the system design meets these specifications (Tuncali et al. 2020; Hashemi et al. 2023; Yaghoubi and Fainekos 2019a). In some forms of this paradigm the designer uses automated synthesis from specifications to go directly from the functional specifications to correct-by-construction system designs (Hashemi et al.; Yaghoubi and Fainekos 2019b).

In contrast, assessment of an AI system includes several new components. These differences are essential to empowering users and placing them closer to the central role in utilizing their AI agents in tasks that they desire. Unsuprisingly, this also diminishes the designer's control on the overall behavior of the AI system thereby necessitating a new, user-driven AI assessment paradigm.

Dynamic Synthesis and Incorporation of Safety Properties Since users' tasks and environments are not known a priori, one of the major open questions involves effectively generating, with feedback from the user, safety properties relevant to the user's intent. This is a critical departure from the conventional paradigm, where experts carefully scope operating environments and corresponding safety properties for a limited range of functionality. In addition, once acquired, these safety properties need to be incorporated in planning and reasoning algorithms used for behavior synthesis, and they need to be updated during execution while incorporating interventions and feedback from the user.

Overall Capability Assessment While the central question for conventional systems can be stated as "Will a given implementation achieve (the designers') functional specifications under assumptions on the environment?", the central question for AI assessment is significantly more user-centric: "Will it be safe for a user to use their AI system for the task and environment that they have in mind?" This question necessitates that the user understands the scope of safe operation of their current AI system. Addressing this problem requires approaches for dynamically identifying what an AI system can and can't do and the impact of these capabilities on user-desired notions of safety as well as safety considerations stemming from regulatory guidelines. Early work in this direction shows promise in identifying AI system capabilities by interrogating the system through a minimal, query-response interface (Verma, Marpally, and Srivastava 2021, 2022; Verma, Karia, and Srivastava 2023).

Accuracy of Capturing User Intent Typically, users express their intent inaccurately through an instruction or a command to the AI system, which needs to be translated into a goal or an objective function and associated cost functions for the agent's behavior. Absence of robust methods for addressing this aspect leads to problems such as reward miss-specification and wireheading (Russell, Dewey, and Tegmark 2015; Amodei et al. 2016).

Reconciling Behavior Synthesis with User Intent While conventional V&V paradigms assume that designers have access to the code that controls a system's sensors and actuators, in AI systems, the code available at design stage (e.g., the DQN algorithm (Mnih et al. 2015)) controls the agent's computation, which generates, post deployment task-specific executable sensing and control actions. In the case of AI systems, the executable controller is therefore specific to the user's intent and the current operating environment, and un-knowable during system design.

Almost all practical implementations of planning and reasoning algorithms produce suboptimal behavior. Furthermore, users are often unaware of constraints on the AI system's abilities (e.g. a robot's kinematic constraints). Consequently, as evidenced by research on explainable planning and learning, the computed behavior often belies users' expectations for what the system should be doing. User-driven assessment of AI systems needs to ensure that the algorithms used for behavior synthesis yield executions that comply with the safety properties acquired as discussed above, in the context of user-specific tasks in user-specific environments.

Differential Assessment Currently deployed AI systems already feature dynamic updates (e.g., (Jones 2021)). This can leave users unable to determine whether the updated system can still perform the tasks *they* had in mind, in *their* environments. A full re-assessment of the AI system from scratch would be wasteful with every change in the task, the environment or the system itself. Early work in this direction indicates that *differential assessment* paradigms can be more efficient (Nayyar, Verma, and Srivastava 2022), although much remains to be done in making these methods practical and more robust for the real world.

Requirements Monitoring Even though the verification of safety requirements at design time may not be possible, it may be possible to monitor safety requirements that are identified during design stage, at runtime (Yamaguchi, Hoxha, and Nickovic 2023). New opportunities arise on how such safety requirements can be extracted from user intent.

References

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

Hashemi, N.; Hoxha, B.; Yamaguchi, T.; Prokhorov, D.; Fainekos, G.; and Deshmukh, J. 2023. A Neurosymbolic Approach to the Verification of Temporal Logic Properties of Learning-Enabled Control Systems. In *ACM/IEEE 14th International Conference on Cyber-Physical Systems (ICCPS)*, 98–109.

Hashemi, N.; Qin, X.; Deshmukh, J. V.; Fainekos, G.; Hoxha, B.; Prokhorov, D.; and Yamaguchi, T. ???? Risk-Awareness in Learning Neural Controllers for Temporal Logic Objectives. In *American Control Conference (ACC)*, 4096–4103.

Hasuo, I.; Eberhart, C.; Haydon, J.; Dubut, J.; Bohrer, R.; Kobayashi, T.; Pruekprasert, S.; Zhang, X.-Y.; Pallas, E. A.; Yamada, A.; Suenaga, K.; Ishikawa, F.; Kamijo, K.; Shinya, Y.; and Suetomi, T. 2023. Goal-Aware RSS for Complex Scenarios via Program Logic. 8(4): 3040–3072.

Jones, C. 2021. Tesla self-driving software update begins rollout though company says to use with caution. *USA Today*, (July 12, 2021).

Loos, S. M.; Platzer, A.; and Nistor, L. 2011. Adaptive Cruise Control: Hybrid, Distributed, and Now Formally Verified. In *Formal Methods*, volume 6664 of *LNCS*, 42–56. Springer.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-Level Control through Deep Reinforcement Learning. *Nature*, 518(7540): 529–533.

Nayyar, R. K.; Verma, P.; and Srivastava, S. 2022. Differential Assessment of Black-Box AI Agents. In *Proc. AAAI*.

Russell, S.; Dewey, D.; and Tegmark, M. 2015. Research priorities for robust and beneficial artificial intelligence. *AI magazine*, 36(4): 105–114.

Tuncali, C. E.; Fainekos, G.; Prokhorov, D.; Ito, H.; and Kapinski, J. 2020. Requirements-driven Test Generation for Autonomous Vehicles with Machine Learning Components. *IEEE Transactions on Intelligent Vehicles*, 5: 265–280.

Verma, P.; Karia, R.; and Srivastava, S. 2023. Autonomous Assessment of Sequential Decision-Making Systems in Stochastic Setting. In *Proc. NeurIPS*.

Verma, P.; Marpally, S. R.; and Srivastava, S. 2021. Asking the Right Questions: Learning Interpretable Action Models through Query Answering. *Proc. AAAI*.

Verma, P.; Marpally, S. R.; and Srivastava, S. 2022. Discovering User-Interpretable Capabilities of Black-Box Planning Agents. In *Proc. KR*.

Yaghoubi, S.; and Fainekos, G. 2019a. Gray-box Adversarial Testing for Control Systems with Machine Learning Components. In *ACM International Conference on Hybrid Systems: Computation and Control (HSCC)*.

Yaghoubi, S.; and Fainekos, G. 2019b. Worst-case Satisfaction of STL Specifications Using Feedforward Neural Network Controllers: A Lagrange Multipliers Approach. *ACM Transactions on Embedded Computing Systems*, 18(5S).

Yamaguchi, T.; Hoxha, B.; and Nickovic, D. 2023. RTAMT – Runtime Robustness Monitors with Application to CPS and Robotics.