# AI Assurance Needs a Systems Engineering Approach

Robin Bloomfield
*City St George's, University of London*
London, UK
r.e.bloomfield@citystgeorges.ac.uk

John Rushby
*Computer Science Laboratory, SRI International*
Menlo Park CA, USA
Rushby@csl.sri.com

***Abstract*—As systems become increasingly composed of AI/ML elements, the focus of assurance tends to shift from "safe use of AI" to "safe AI." In this position paper, we argue this is opposite to what is required and urge a systems engineering approach.**

***Index Terms*—AI alignment, AI safety, runtime verification**

## I. Introduction

There are general systems engineering principles for dependable systems and their safety assurance that apply to AI systems just as they do to traditional software and systems [1]. These principles are described in Section 1 of our report on AI Dependability and Assurance [2], but a central point is that testing is not enough: we need to know how the system works, how it fails, and why it has the critical properties required of it. An "assurance case" allows us to understand and document the evidence and arguments supporting this belief; we advocate a recent formulation that we call Assurance 2.0 [3].

How the general dependability and assurance principles are applied to AI depends on the purpose and architecture of the system concerned. We propose a strong determinant at present is the ratio of conventional software and systems engineering to AI, and there is a continuum along this dimension from "AI just does perception" to "AI does everything." The basic tension is then between "Safe use of AI" vs. "Safe AI" [4]. As more of the system is AI (and ML), so there is a tendency to depend more on the second of these; that is, to attempt to make the AI "trustworthy." We think this approach is intrinsically difficult and typically provides little credible assurance.

## II. Extent of AI Usage

We outline how concerns and assurance methods typically evolve as more of the system is based on AI; these paragraphs roughly correspond to the sections of our report [2].

At one end of the continuum we have autonomous systems such as self-driving cars with plenty of traditional engineering, although the AI may provide large added value and much safety risk. The general principles apply here very similarly to the way they apply to traditional safety-critical systems (traditional software can be assured because we know exactly how it works), with the exception that the AI/ML components have to be regarded as untrusted black boxes (because we do *not* know exactly how they work) and externally checked or guarded at runtime. The guards may use highly assured traditional software, or possibly AI/ML software that is diverse from the mainline software (e.g., it is difficult to provide perception without AI, so the guard itself may use AI).

Next, we have systems that are engineered for some specific purpose that is enabled by AI (e.g., logistics); typically the AI elements are developed or customized for that purpose. The fact that there is a specific purpose means that we can identify the potential harms and hazards of the system, and can quantify these and thereby derive the level of dependability and assurance required of the system (as opposed to basing assurance on generic properties of the AI). Knowing these levels helps us design the system architecture and mitigations for its hazardous elements. Again, these elements will include the AI and ML components, which must be externally checked or guarded. The architecture should generally provide defense in depth, so that single failures cannot cause accidents. In addition to those due to component faults we must also consider system failures due to overall complexity and coupling.

Then we have systems where most of the functionality is provided by AI and ML, and the rest of the system is little more than a wrapper around these. Increasingly, the AI is provided by generic software such as an LLM rather than specifically constructed. This means that general fallibilities of the LLM must be considered as well as the specific harms and hazards due to the system's purpose. Furthermore, because the LLM is the heart of the system, it is often expected that mitigations should be programmed into the LLM rather than provided externally. In this regard, it is important to realize that everything an LLM does is a "hallucination," otherwise it would simply behave as a search engine over its training data. The LLM is taught to suppress harmful and useless hallucinations by reinforcement learning in secondary training but this is bound to be imperfect because there are huge numbers of individually rare defects. Fine tuning or prompt engineering for mitigation will likewise be imperfect. The consequence is that such systems cannot perform dependably, nor be assured, without some form of external checking—but since the function of the system is enabled by AI, it is likely that checking involves more AI (although some tasks that require AI-generated solutions can be checked by traditional software, and some other tasks can be probed—e.g., for bias—by running multiple instances with slightly different inputs).

Finally, at least for the present, we come to systems that are more extreme instances of those just considered. These are systems that are basically a generic LLM initialized with specific prompts and possibly an agentic interface (meaning they can do things in the world, such as press buttons on a computer screen, or string actions together as when

Partially funded by City St George's and SRI International.

making travel reservations). They are used for purposes such as software programming, customer relationship management, research, correspondence, and so on: basically any task that is currently performed by a human sitting at a computer. Here, it is not easy to specify correctness, nor to check for it ("I know it when I see it"), yet failures can have significant consequences. Furthermore, harms and hazards may not be due to AI failures but to the system's capabilities. For example, LLM-based research in the hands of bad actors may disclose so-called CBRN threats (e.g., how to make a bio-weapon), or may generate social network campaigns to sow discord or influence elections, or may aid in the search for computer vulnerabilities and assist in generating attacks.

Beyond the present is the possibility of AGI and superintelligence. These would supercharge the hazards described above and introduce "existential" harms, such as societal takeover. These harms are the focus of several regulatory groups but we are skeptical of their analysis. First, existential threats, even at low probability, are of such significance that massive assurance is required. Yet much regulatory attention is on trivial tests and protections built into the LLMs themselves. In reality, the mitigations and responses to these threats would be outside the system: e.g., coordinated destructive resistance. Second, we believe the threats are remote: AGI requires real intelligence and current technology is a long way from delivering that. Finally, it is not specifically intelligence that empowers humanity, but cooperation, society, and culture. We think the urgent concern is harms perpetrated by AFGI systems (pronounced AFF-GEE, FG = Fairly Good or Fairly General) only a little more powerful than those available today, acting together with humans. We suggest that research is urgently needed to mitigate these hazards by mechanisms within and outside the system, and through societal adjustment and adaptation that develops resilience against them.

## III. Conclusions and Call to Actions

Engineers for traditional systems, especially those considered critical, are generally aware of the principles for dependable systems, but may not yet have considered their application to systems that are mostly AI; furthermore, AI engineers are often unaware of these principles and instead speak of making their systems "trustworthy" and "aligning" them. We argue that "aligning" opaque AI/ML systems is an intractable problem and the challenge of AI security and safety should instead be reframed as a more tractable, albeit difficult, systems engineering problem. Thus, our messages for policy makers and shapers (governments, regulators, philanthropic funders and specialized AI safety/security institutes) are:

- *Focus on the application and system, not just the AI model*: regulation and policy should be targeted at the deployed system and its application context. It should encourage the development of systems-level assurance cases to extend and enhance existing "Model Cards" [5].
- *Prioritize the development of tools, technologies and evidence* to support risk owners in understanding the extent of hazards, risks and benefits of AI-enabled systems.
- *Mandate architectural resilience*: incentivize or require architectural principles such as external monitoring, diversity, partitioning, and fail-safe mechanisms, particularly in high-risk applications.
- *Prioritize near-term, high-impact threats*: policy attention and resources should focus on mitigating the clear and present dangers of failures, misuse, and societal degradation by capable, general-purpose AI ("AFGI"), rather than being disproportionately directed toward speculative, long-term existential risks (which we see as a distraction amounting to regulatory capture). A focus on AFGI provides a more concrete, actionable and relevant agenda.

This dependability view provides a crucial complementary perspective to the AI-model-centric view of AI safety. It argues that focusing on improving AI benchmark performance or reducing "hallucinations" through internal tuning, while welcome and necessary, are insufficient approaches for building AI systems that are truly dependable.

The key message for developers, engineers, and researchers is the need to shift the mindset and practice from model-centric evaluation to systems-level assurance. This implies embracing, and extending classical systems engineering, specifically:

- *Architecture*: Prioritize architectures that feature defense-in-depth, diversity, partitioning, robust monitoring, and external runtime verification of AI components. Provide APIs so that risk owners and application developers can access assurance artifacts and integrate external checks.
- *Analysis*: Develop hazard analysis techniques to address the plethora of potential harms, including acute vs chronic risks. Recognize that AI systems may become deeply embedded in their social context so the system boundary is wider than anticipated, inviting failures due to Bak's "self-organizing criticality" and "highly optimized tolerance," and leading to Perrow's "normal accidents."
- *Multidisciplinarity*: Foster integration between AI and ML development and traditional dependability and safety engineering practices.
- *Verification Technologies*: Invest in research and tools for the external checking of complex AI outputs, including AI-based checkers, that are diverse and demonstrably more reliable than the systems they monitor.

Overall we should stop trying to align AI models in pursuit of Safe AI, and start engineering systems that Use AI Safely.

## References


[1] INCOSE, *Artificial Intelligence Systems*, https://www.incose.org/communities/working-groups-initiatives/artificial-intelligence-systems.

[2] R. Bloomfield and J. Rushby, "Assurance of AI systems from a dependability perspective," Computer Science Laboratory, SRI International, Menlo Park CA, available as arXiv:2407.13948.

[3] R. Bloomfield, J. Rushby *et al.*, *Assurance 2.0 home page*, http://www.csl.sri.com/users/rushby/assurance2.0.

[4] R. Winther and R. Fredriksen, "Safe AI vs safe use of AI," in *Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference*, 2025, https://rpsonline.com.sg/proceedings/esrel-sra-e2025/pdf/ESREL-SRA-E2025-P7886.pdf.

[5] M. Mitchell *et al.*, "Model cards for model reporting," arXiv:1810.03993, Oct. 2018.