Javen W. Zamojcin
CSE870 Homework 2
2025/01/28

The “silver bullet” metaphor refers to a “magic” solution to complex, perennial problems—in terms of efficiency, effectiveness, and reliability. In particular, Brooks is referring to the software project problem and the outcries for a single solution for reducing software development costs [1, para. 2]. However, Brooks argues that there is no single development, and likely never will be, that can address the demanded improvements of software development in productivity, reliability, or simplicity [1, para. 3]. To explain why, Brooks takes a philosophical approach and first circumscribes the software project problem into its fundamental essence and accidental components. Then arguing that the essence of the software problem is never addressed with any one advancement in technological or managerial technique—that is, the conceptual design work [1, p. 17] or fashioning of complex conceptual constructs [1, p. 1]. Rather, the accidents of the problem, or the design and requirement implementation techniques, are often focused upon in technology innovations. An example Brooks gives to contrast these fundamentals is the introduction of high-level languages; their abstractions free a program from many accidental complexities which were never inherent to begin with and improve some productivity, yet, all the high-level languages can do is furnish the programming constructs and fail to address the underlying problem essence [1, p. 12].

Regarding the integration of AI with safety-critical systems, there exist several essential difficulties inherent to the problem and compliance with current safety standards. By the “black box” nature of AI system solutions, our ability to explain and analyze their decision-making is limited [2, para. 3]. For neural networks in particular, once the system is trained, new data can be fed to it and a useful interpretation is produced, but the decision-making process that occurs between the input and output layers is not intuitive, making extracting their rationale challenging [3, para. 7]. If a system, especially a safety-critical one, makes an incorrect decision, it is imperative to be able to understand what went wrong and how to rectify the error. An instance of this essential problem is evident in autonomous vehicle research, where learning systems can help with perception and decision making, but explicit rules and human knowledge also have to be incorporated to ensure safety [3, para. 8]. Partially due to this limitation, AI systems have not been recommended for use in safety-critical systems [2, para. 3].

An accidental, or non-essential, problem with AI-based safety-critical systems is a lack of consolidated industry best practices and the availability of safety standards with consideration for AI technology [2, para. 5]. In part, this is due to AI still being an immature and rapidly expanding technology, to which best practices in the industry will naturally be discovered, developed, and integrated into use over time; similar to how general software engineering practices and standards have matured with time and unfortunately, tragic experiences—e.g., Therac-25. Additionally, there is still a lack of consensus regarding the taxonomy of AI, or how we define, map, and classify the many different types of AI rapidly developing in academia and industry; this has resulted in high variability of terms and concepts from research contributions and safety standards, slowing the standardization of safety in practice [2, para. 15]. Without a matured taxonomy, the adaptation of traditional safety-critical FuSa (functional safety) standards to AI will be hindered.

Today we see requests for the integration of AI with safety-critical systems analogous to the software engineering “hopes for the silver bullet” discussed by Brooks [1]. E. Jenn et al. states, “ML seems to be one of the only technically and economically viable solution to automate some complex tasks usually realized by humans, such as driving vehicles” [4, para. 1], and J. Perez-Cerrolaza et al. states, “Driven by AD and UAV engineering challenges and the associated economic investment, there is a significant research and engineering effort to define novel technical solutions for developing AI-based autonomous systems” [2, para. 2]. In my opinion, the most valuable lesson by Brooks is that no amount of abstractions can replace good system design. And in the case of safety-critical systems, hasty over-reliance on abstracting controls over to AI without a safety-based design in mind can have fatal consequences.

Bibliography

[1] F. P. Brooks, “No silver bullet essence and accidents of software engineering,” Computer, vol. 20, no. 4, pp. 10–19, Apr. 1987. doi:10.1109/mc.1987.1663532

[2] J. Perez-Cerrolaza et al., “Artificial Intelligence for safety-critical systems in industrial and Transportation Domains: A survey,” ACM Computing Surveys, https://dl.acm.org/doi/10.1145/3626314 (accessed Jan. 29, 2025).

[3] E. Ackerman, “How Drive.ai is mastering autonomous driving with Deep Learning,” IEEE Spectrum, https://spectrum.ieee.org/how-driveai-is-mastering-autonomous-driving-with-deep-learning (accessed Jan. 29, 2025).

[4] E. Jenn et al., “Identifying Challenges to the Certification of Machine Learning for Safety Critical Systems,” ERTS 2020, https://www.erts2020.org/uploads/program/ERTS2020_paper_17.pdf (accessed Jan. 29, 2025).