2024 ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS)

Poster Abstract: Assuring LLM-Enabled Cyber-Physical Systems

Weizhe Xu[†], Mengyu Liu[†], Steven Drager[‡], Matthew Anderson[‡], Fanxin Kong[†]
[†]*Department of Computer Science and Engineering, University of Notre Dame, South Bend, IN*
[‡]*Air Force Research Laboratory, Rome, NY*
wxu3@nd.edu, mliu9@nd.edu, steven.drager@us.af.mil, matthew.anderson.37@us.af.mil, fkong@nd.edu

***Abstract*—Cyber-Physical Systems (CPS) are integrations of computation, networking, and physical processes. The autonomy and self-adaptation capabilities of CPS mark a significant evolution from traditional control systems. Machine learning significantly enhances the functionality and efficiency of Cyber-Physical Systems (CPS). Large Language Models (LLM), like GPT-4, can augment CPS's functionality to a new level by providing advanced intelligence support. This fact makes the applications above potentially unsafe and thus untrustworthy if deployed to the real world. We propose a comprehensive and general assurance framework for LLM-enabled CPS. The framework consists of three modules: (i) the context grounding module assures the task context has been accurately grounded (ii) the temporal Logic requirements specification module forms the temporal requirements into logic specifications for prompting and further verification (iii) the formal verification module verifies the output of the LLM and provides feedback as a guideline for LLM. The three modules execute iteratively until the output of LLM is verified. Experiment results demonstrate that our assurance framework can assure the LLM-enabled CPS.**

I. INTRODUCTION

Cyber-Physical Systems (CPS) represent a fusion of computing, networking, and physical operations. Distinguished by their autonomous and adaptive features, CPS shows a remarkable advancement beyond traditional control systems. Large Language Models (LLMs), such as GPT-4, significantly enhance the capabilities of CPS by providing sophisticated intelligence support. Researchers have explored using LLMs for task planning synthesis in controllable robots or agents. In some applications, LLMs serve as assistants in various domains, including data processing and contextual grounding [1]. Alternatively, LLMs can perform as decision-makers in planning and navigating application [2]. In other words, LLMs are responsible for determining the agents' motions to accomplish specific tasks.

However, the deterministic requirements in CPS contradict the probabilistic nature of LLMs. This contradiction subjects the above applications to risks of vulnerability and unreliability, potentially leading to catastrophes. Therefore, when deploying LLMs in CPS, it necessitates experts to manually inspect LLM outputs, which is expensive and time-consuming. Hence, an automated framework for verifying LLM outputs is essential to deploying LLM to real-world systems. Although some testing and assurance methods for learning-enabled CPS have been proposed [3], [4], the vast number of parameters in LLMs and their variable use cases make these methods unsuitable. Meanwhile, traditional verification and safe assurance methods for CPS [5], [6] are also inadequate due to the learning characteristics of LLMs. Users seek to both verify and guide LLMs to ensure correct and safe outcomes. The goal of this paper is to provide a comprehensive and general assurance framework for LLM-enabled CPS.

Assurance of LLM-enabled CPS presents significant challenges. The first is **context grounding**. LLMs cannot inherently understand the physical world and CPS contexts. Therefore context grounding is necessary. Our approach should get the actionable, context-aware formula for CPS tasks instead of abstract, language-based outputs from LLMs. The second challenge is **safety and temporal requirements**. CPS tasks demand precise execution in terms of temporal order and timing to ensure safety. The planning must satisfy both the safety and temporal requirements. The third is **Formal Verification**. After LLM generates a plan, verifying it meets all requirements is crucial for safety. In addition, verifying LLM-generated plans' safety should go beyond mere true/false assessment. LLM requires detailed feedback for improvement.

In summary, our research presents several key contributions: (1) We have developed a framework that ensures the reliability of LLM-enabled CPS. (2) This framework guarantees that plans generated by LLMs meet both safety and temporal requirements. (3) We conduct an evaluation of our proposed framework, showcasing the enhanced effectiveness of our proposed solution.

II. FRAMEWORK

Overview of the framework. Fig 1 illustrates our iterative assurance framework, comprising three primary components highlighted in blue: context grounding, logic specification, and formal verification.

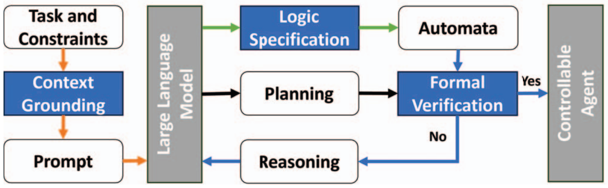

Fig. 1. Iterative Assurance Framework

In the context grounding component, depicted by orange paths, task descriptions are converted into natural language

---

This material was assigned a clearance of CLEARED on 14 Mar 2024, SAF/PA case number 2024-0208, AFRL case number 2024-1061.

979-8-3503-6927-4/24/$31.00 ©2024 IEEE
DOI 10.1109/ICCPS61052.2024.00039

prompts by prompt engineering [7]. The LLM then generates formal expressions to ground the context. If the design of the prompt is not adequate, leading to the LLM being unable to fully ground in the context, it may result in some basic errors. Typically, we try some simple problems to ensure that the LLM has understood the question. For the logic specification component, indicated by green paths, safety and temporal requirements are similarly transformed into natural language prompts. The LLM then gives output in the form of logic specifications corresponding to these requirements, such as LTL formulas. We convert these formulas to automata using formal tools after expert review. The problem's formulas are identical in iterations, so the manual effort required is minimal. Once context grounding and logic specification formulation are complete, the LLM can produce a preliminary plan for the controllable agent, though its safety remains unverified. The formal verification component, shown in blue paths, takes two inputs: the planning candidate and the automaton derived from the temporal requirements. If the plan passes formal verification, it is sent to the controllable agent. If not verified, the process offers detailed feedback for the LLM to adjust its plan until it passes or hits the iteration limit.

III. RESULTS

Our evaluation aims to test the hypothesis that our framework can ensure the reliability of LLM-enabled CPS. The framework ensures LLM finally gives us the correct output or reaches the iterative limit after iteratively executing the three modules. The LLM used in the experiment is GPT-4. In our framework, we use Z3 python API to verify safety constraints expressed as first-order logic (FOL) formulas and use the verification tool SPOT to check the temporal constraints expressed as linear temporal logic (LTL) formulas.

1st Prompt

You are a planner for Uber drivers. There are several cities on the map and some paths between these cities, for example, A-B means there is one path between city A and city B. All the path ... you also have some constraints on the visiting orders.
Here is an example problem and the correct result.
...
Now please give me the result of Problem 2 in the same format. You do not need to give an explanation.
Problem 2:
Cities: A, B, C, D, E, F, G
Paths: A-B, B-C, A-E, E-D, B-F, F-G
Constraints: You should have been to C and D before you go to G.
Init: City A, Object: City G

Illustrative Example. In this example, we consider a navigation problem that requests LLM to find a plan for a driver while not violating the safety and temporal constraints. In the beginning, we introduce the problem to LLM and provide an example problem with the correct result. Then ask LLM to solve a new problem in the same format, the details are shown in the 1st prompt above. Meanwhile, we ask LLM to give us the FOL and LTL formulas for the safety and temporal constraints. The LTL formula for this problem is shown as: $\mathrm{G}(g \rightarrow (c \wedge d))$ where g, c, and d indicate the proposition that the driver has already visited cities G, C, or D, respectively.

Then we get the 1st output from LLM as shown below. We discovered the first three steps broke safety constraints, as verified by Z3, including details like invalid steps. Based on that, we construct the 2nd prompt to re-query the LLM, detailing the earliest invalid steps. We get the 2nd output and find that LLM has already corrected its previous mistake. However, we can find a violation of the temporal constraint by using SPOT. With this information, we re-query LLM using the 3rd prompt. Finally, after three rounds of iteration, we obtain a correct plan that passes all verifications, as shown in the 3rd output. This experiment demonstrates that our method can obtain the correct plan from LLM while assurance safety and temporal constraints.

1st Output

A → B
B → C
B → F
F → G

2nd Prompt

A → B
B → C
B → F
is invalid.

2nd Output

A → B
B → C
C → B
B → F
F → G

3rd Prompt

A → B
B → C
C → B
B → F
F → G
is invalid.

3rd Output

A → B
B → C
C → B
B → A
A → E
E → D
D → E
E → A
A → B
B → F
F → G

IV. POSTER DESCRIPTION

In our proposed poster, we do an overall introduction of the project in the first part, including the purpose, focus of our work, etc. In the second part, we introduce our proposed framework, and in the third part, we give an illustrative example to demonstrate the effectiveness of the framework.

Acknowledgement

This work was supported in part by NSF CNS-2333980.

REFERENCES

[1] S. Huang et al., "Instruct2act: Mapping multi-modality instructions to robotic actions with large language model," *arXiv preprint arXiv:2305.11176*, 2023.
[2] K. Lin et al., "Text2motion: From natural language instructions to feasible plans," *arXiv preprint arXiv:2303.12153*, 2023.
[3] X. Zheng et al., "Testing learning-enabled cyber-physical systems with large-language models: A formal approach," *arXiv preprint arXiv:2311.07377*, 2023.
[4] G. Katz et al., "Reluplex: An efficient smt solver for verifying deep neural networks," in *CAV 2017*. Springer, 2017.
[5] M. Liu et al., "Fail-safe: Securing cyber-physical systems against hidden sensor attacks," in *RTSS*. IEEE, 2022.
[6] L. Zhang et al., "Ai-enabled real-time sensor attack detection for cyber-physical systems," in *AI Embedded Assurance for Cyber Systems*. Springer, 2023.
[7] P. Liu et al., "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.