Doc. No. P4023R0
Date: 2026-02-23
Audience: WG21
Authors: (Directions Group)
Jeff Garland
Paul E. McKenney
Roger Orr
Bjarne Stroustrup
David Vandevoorde
Michael Wong
Reply to: fraggamuffin@gmail.com

Title: Strategic Direction for AI in C++: Governance, and Ecosystem

Target Audience: ISO C++ Directions Group (DG) / WG21

Purpose: To update the C++ Directions Paper (P2000) with a unified strategy for AI, focusing on governance and ecosystem encouragement.

1. Executive Summary

The Vision: The Directions Group (DG) recommends a cohesive strategy for the AI era. We address the interaction between C++ standardization and Artificial Intelligence in two specific areas: governing its use within the committee and encouraging ecosystem improvements for AI generation.

The Scope: This paper covers three distinct thrusts:

- **Thrust I:**Governance & Guidelines. Aligning WG21 with ISO/IEC JTC1 directives on AI usage..
- **Thrust II: Ecosystem Encouragement.** Calling on the community to improve C++ training data ("ImageNet for C++") and tooling-friendly semantics.

---

2. Thrust I: Governance & Guidelines

This thrust focuses on maintaining the integrity of the standardization process in the age of generative AI. We align with and reinforce the guidance provided by ISO/IEC JTC1.

Committee Governance (Ethics of AI in Standardization)

- **The Principle:** The author is the "intelligence of record." While AI tools can assist in productivity, the ultimate responsibility for accuracy, logic, and normative quality rests entirely with the human author.
- **Permitted Use:** AI is a valid tool for research, summarizing unfamiliar domains, checking consistency, or updating text to align with code changes.
- **Prohibited Use:**
  - **Normative Wording:** It is inappropriate to use AI to generate normative wording or core design proposals without rigorous human verification. AI-generated "slop"—voluminous but low-quality submissions—wastes valuable committee time and should not be tolerated.
  - **Bots in Meetings:** Automated AI agents are strictly forbidden from attending or recording ISO meetings due to privacy and intellectual property risks.
  - **Copyright & Origin:** Authors must be mindful of copyright risks. AI generation can inadvertently reproduce copyrighted material. Authors must ensure they own or have rights to every text submitted.
- **Reference:** We refer members to *ISO/IEC JTC 1-SC 22 N5991: Guidance on the use of AI for ISO Committees*.
  - https://sd.iso.org/documents/nxfile/default/8500be37-54e5-417c-b6d6-2d698d28541f/file:content/ISO-IEC%20JTC%201-SC%2022_N5991_Guidance%20on%20the%20use%20of%20AI%20for%20ISO%20Committees%20March%202025-FA.pdf?changeToken=12-0

---

3. Thrust II: Improving the Ecosystem (The "ImageNet" Challenge)

The DG sees or recognizes a critical "Garbage In, Garbage Out" problem facing C++ developers using AI. Current models are trained on legacy C++ (C++98/03), vendor-specific dialects, and unsafe patterns found online. This leads to AI tools generating code that violates modern safety profiles. ImageNet in AI is this https://en.wikipedia.org/wiki/ImageNet

The "ImageNet for C++" Community Challenge WG21 cannot solve this alone, but we strongly encourage the ecosystem (e.g., Boost, Beman Project, Open Source foundations, academic research) to address this gap.

- **The Need:** Just as ImageNet provided the labeled dataset that powered computer vision, the C++ ecosystem needs a curated, human validated collection of modern examples of high-quality dataset of **Modern, Idiomatic C++** (C++20/23/26).
- **The Goal:** A public corpus tagged by domain (e.g., `ai/`, `embedded/`, `finance/`) that favors modern idioms; for example (not exhaustive):
  - **Spans over Pointers:** Reinforcing `std::span` and `std::mdspan` to eliminate buffer overflow hallucinations.
  - **Sender/Receiver over Callbacks:** Promoting `std::execution` patterns over legacy concurrency.

    - **Algorithms over Loops:** Biasing generation toward `<algorithm>` and `<numeric>`.
    - Null ptr checks
    - Range for over traditional loops
- Specific prompts engineered for C++
- Quality of comments in code matter

Tooling and Semantics (The Agent Imperative) We note that AI agents struggle with "Hidden Intent" in C++ APIs (e.g., `foo(x)` needs you to look for the definition or consult a compiler). While we do not propose changing the language syntax to address this, we encourage the development of **Guidelines and Tooling** that favor explicit semantics.

- Tools and linters that surface intent at the call site (e.g., inlay hints for `out` parameters) are critical for the "Agentic" future.
- Make compilers answer questions about argument types and modern usage possibly connecting to MCP
- RAG or Retrieval-Augmented-Generation, basically adding modern C++ knowledge post training. There is also supervised fine-tuning techniques to help improve LLMs after or during pre-training.
- Library designers are encouraged to create APIs that are human consumable because it also helps agents to consume. Make intent evident, reducing the context required for an AI agent to use them correctly.

5. Conclusion

C++ remains the dominant language of AI infrastructure. By enforcing high standards for authorship (Thrust I) and encouraging the ecosystem to improve training data (Thrust II), the Directions Group aims to ensure the language remains robust and relevant in the AI era.