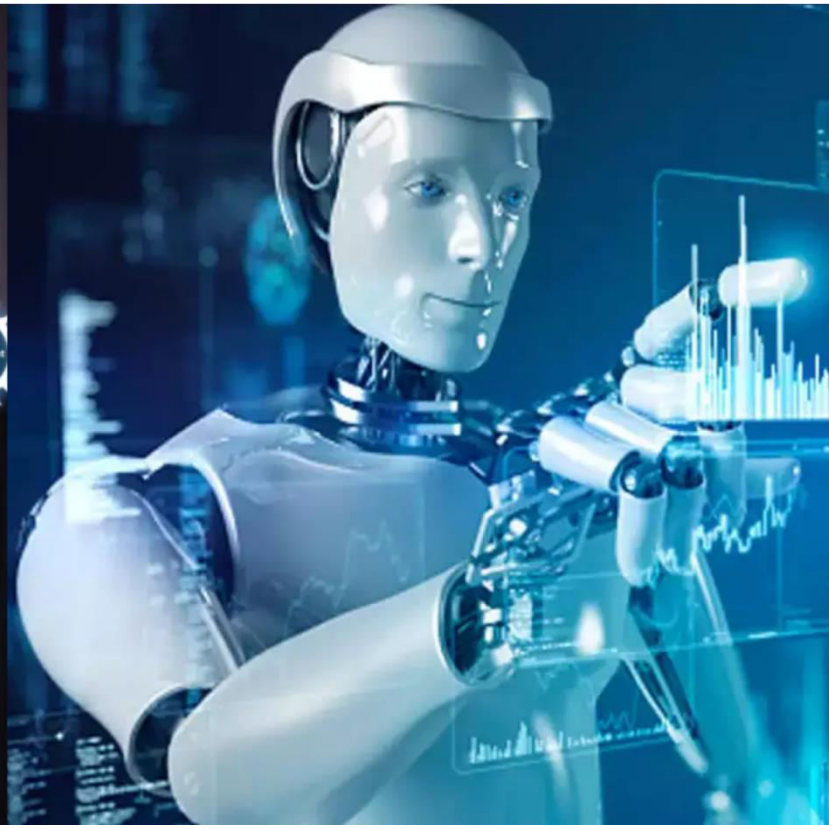
IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

ISSN(O): 2319-8753
ISSN(P): 2347-6710

International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.699

Volume 14, Issue 3, March 2025

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403012|

Unlocking Context for Intelligent Agents: The Model Context Protocol as a Standardized Integration Framework

Ashish Kattamuri

Senior Software Engineer, Proofpoint, Colorado, USA

ABSTRACT: The Model Context Protocol (MCP) is an open standard introduced by Anthropic to establish seamless integration between Large Language Model (LLM) applications and external data sources and tools1. Addressing the challenge of AI models operating in isolation, MCP provides a universal connector. that simplifies integrations, enhances interoperability, and promotes scalability. By offering a standardized way for AI models to access relevant context and perform actions on external systems, MCP aims to resolve the fragmentation and complexity associated with bespoke integrations. This paper explores the architecture, core components, benefits, and potential future implications of MCP, highlighting its role in enabling more context-aware and agentic AI systems.

KEYWORDS: Model Context Protocol (MCP), Large Language Models (LLMs), AI Integration, Standardization, Context Management, AI Agents, Tool Use.

I. INTRODUCTION

The rapid advancements in Large Language Models have showcased remarkable capabilities in reasoning and quality. However, a significant limitation remains their isolation from real-world data, confined by information silos and legacy systems. Previously, each new data source or tool required custom implementation to connect with AI assistants, leading to a complex and unscalable landscape.

To overcome these challenges, Anthropic developed the Model Context Protocol (MCP), an open standard for connecting AI assistants to the systems where data resides. MCP acts as a "USB port" for AI applications, providing a universal interface3 that allows any AI assistant to interact with any data source or service without requiring custom code for each integration. This initiative seeks to simplify the integration process, enabling AI models to generate better and more relevant responses by directly leveraging necessary information and tools. The development of MCP represents a shift towards a more standardized and collaborative approach to AI integration, aiming to unlock richer, context-aware AI behavior across various applications

II. LITERATURE SURVEY

The introduction of MCP in late November 2024 by Anthropic has been widely acknowledged as a significant step towards standardizing AI integrations. InfoQ reported on the release of the Model Context Protocol specification, documentation, and SDKs for Python and TypeScript available on GitHub. This announcement highlighted MCP's goal of facilitating seamless integration between LLM applications and external data sources and tools.

The motivation behind MCP stems from the industry's need for a unified solution to the "M×N problem", where M models and N tools each require custom integration. Anthropic observed that even sophisticated models were limited by their inability to easily access fresh information or act on external systems. MCP aims to transform this into a simpler N+M setup, where tools and models conform to the protocol once, enabling any model to work with any compliant tool. This has led analysts to draw parallels between MCP and ODBC for databases.

Early adopters like Block and Apollo integrated MCP into their systems. Development tools companies such as Zed, Replit, Codeium, and Sourcegraph are also working with MCP to enhance their platforms, enabling AI agents to better

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403012|

retrieve relevant information and understand the context around coding tasks. Block's CTO, Dhanji R. Prasanna, praised MCP's open approach, calling it a "bridge that connects AI to real-world applications".

Runloop AI described MCP as a method for systematically organizing the data, instructions, and domain knowledge an LLM needs to generate consistent outputs14. Instead of mapping a model's response to a predefined function signature, MCP focuses on creating layers of context, allowing developers to prioritize different aspects of the conversation and incorporate updates without rewriting core application logic.

Stackademic presented MCP as a universal connector, similar to USB-C, that simplifies integrations, enhances interoperability, and promotes scalability. They outlined the core components of MCP, including the Host, Client, and Server, and discussed the process of implementing MCP servers and clients using available SDKs.

Weights & Biases highlighted MCP as a new open standard that unifies AI models with external tools and data for smarter, context-rich applications. They echoed the analogy of MCP being a "USB port" for AI, providing a universal interface for AI assistants to plug into any data source or service without custom code. They further elaborated on the origins, technical breakdown, applications, and comparisons with existing context management methods.

III. METHODOLOGY

MCP employs a client-server architecture to establish connections between AI models and external resources. The key components of this architecture are:

- **MCP Host:** The AI-powered application or agent environment that the end-user interacts with. It can connect to multiple MCP servers simultaneously. Examples include Claude Desktop, IDE plugins, or custom LLM-based applications.

- **MCP Client**: An intermediary managed by the host to handle communication with individual MCP servers, providing sandboxing for security. The host spawns a client for each server it needs to use, maintaining a one-to-one link.

- **MCP Server**: A program that implements the MCP standard and offers a specific set of capabilities, typically through a collection of tools, access to data resources, and predefined prompts related to a domain. MCP servers can interface with various data sources like databases, cloud services, or any other data source.

The MCP protocol standardizes how AI applications interact with external systems through three primary interfaces:

- **Prompts**: Predefined templates for common user interactions with a specific server. Users can invoke these prompts (akin to slash commands) for standardized interactions like document Q&A or summarization.

- **Tools**: Model-controlled functionalities exposed by the server that the AI application can invoke to perform specific actions or retrieve information. Examples include listing issues from a GitHub repository or querying a database.

- **Resources**: Data exposed by the server that the AI application can access and incorporate into the context for the LLM. Applications can decide when to put a resource into context, potentially based on predefined rules or LLM calls. Servers can also send resource notifications to clients when updates occur.

MCP offers several benefits over traditional API integrations:

- **Standardization**: MCP provides a common language for AI models and external tools to communicate, promoting interoperability and reducing the need for custom integrations.

- **Enhanced Context Awareness**: By enabling access to real-time data and specialized tools, MCP allows AI models to ground their responses in accurate and relevant information.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403012|

- **Two-Way Communication**: MCP supports bidirectional communication, enabling AI models to not only receive information but also to trigger actions in external systems.

- **Simplified Development**: MCP simplifies the development of AI-powered applications by providing a standardized framework for integrating AI models with external systems.

- **Structured Context**: Unlike simple prompt concatenation or RAG, MCP organizes context into labeled, modular blocks, improving maintainability and clarity.

- **Security and Control**: MCP's design emphasizes controlled access, with the host instantiating clients and approving servers. Explicit permissions are required for each server and its tools. Future developments aim to extend this control to distributed environments.

IV. EXPERIMENTAL RESULTS

The "MCP-Solver" prototype, while based on initial experiments rather than extensive quantitative analysis, yielded significant qualitative findings regarding the integration of LLMs with constraint programming systems via the Model Context Protocol (MCP).

Key Observations from Preliminary Experiments:
- The MCP Solver demonstrated its capability to handle a range of constraint programming problems articulated in natural language, encompassing satisfaction, optimization, and parameter exploration tasks.
- The casting example highlighted the LLM's proficiency in translating intricate logical statements into effective boolean constraints.
- The TSP (Traveling Salesperson Problem) experiment illustrated the system's dual functionality in both optimization modeling and adaptive model modification in response to newly introduced constraints, such as a blocked route.
- The N-Queens problem demonstrated the system's ability to explore varying parameters while preserving the fundamental model structure.
- The current implementation exhibited limitations in solving time for larger problem instances due to a set threshold.
- Autonomous updates to the knowledge base by the LLM were infrequent, indicating a potential area for enhancement in the system's prompting mechanisms.
- A notable strength of the system was observed in iterative model refinement and the ability to recover from errors through natural language dialogue.
- Instances of LLM misinterpretation of solver outputs and translation errors occurred, although these were generally identified and rectified by the implemented verification mechanisms.

V. CONCLUSION

The Model Context Protocol (MCP) represents a significant advancement in the field of AI integration. By providing an open and standardized framework for connecting LLMs with external data and tools, MCP addresses the critical challenge of contextual awareness in AI systems. Its client-server architecture and core components of prompts, tools, and resources offer a structured and flexible approach to augmenting the capabilities of AI models.

The early adoption and positive reception from industry players and the developer community underscore the timeliness and relevance of MCP in solving practical integration challenges. Its potential to foster a standardized ecosystem, enhance agentic AI behavior, and shape the focus of AI research towards the utilization of external knowledge highlight its transformative potential.

While MCP is still in its early stages and faces challenges related to scalability, security in distributed environments, and community governance, its foundational contribution towards standardizing context in AI is undeniable. As the protocol evolves and gains wider adoption, it has the potential to become a critical infrastructure layer that makes AI integration more accessible and manageable, ushering in a new era of highly integrated and context-savvy AI assistants across numerous applications.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 8.699

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 14, Issue 3, March 2025

|DOI: 10.15680/IJIRSET.2025.1403012|

REFERENCES

1. Anthropic. (2024). Model Context Protocol. Retrieved from https://www.anthropic.com/news/model-context-protocol
2. Stackademic. (2024). Everything you need to know about the Model Context Protocol (MCP) from Anthropic. Retrieved from https://blog.stackademic.com/everything-you-need-to-know-about-the-model-context-protocol-mcp-from-anthropic-84acdb3c1a2f
3. AI Engineer. (2024). Building Agents with Model Context Protocol - Full Workshop with Mahesh Murag of Anthropic. YouTube. Retrieved from https://www.youtube.com/watch?v=kQmXtrmQ5Zg
4. RunLoop AI. (2024). Model Context Protocol (MCP): Understanding the Game-Changer. Retrieved from https://www.runloop.ai/blog/model-context-protocol-mcp-understanding-the-game-changer
5. GitHub. (2024). Model Context Protocol. Retrieved from https://github.com/modelcontextprotocol
6. Weights & Biases. (2024). The Model Context Protocol (MCP) by Anthropic: Origins, functionality, and impact. Retrieved from https://wandb.ai/onlineinference/mcp/reports/The-Model-Context-Protocol-MCP-by-Anthropic-Origins-functionality-and-impact--VmlldzoxMTY5NDI4MQ
7. InfoQ. (2024). Anthropic announces the Model Context Protocol. Retrieved fromhttps://www.infoq.com/news/2024/12/anthropic-model-context-protocol/
8. Sakal, N. (2024). MCP vs. API: Model Context Protocol Explained. Retrieved from https://norahsakal.com/blog/mcp-vs-api-model-context-protocol-explained/
9. Szeider, S. (2025). MCP-Solver: Integrating Language Models with Constraint Programming Systems. Retrieved from https://arxiv.org/pdf/2501.00539

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details