MOSAICO: Management, Orchestration and Supervision of AI-agent COmmunities for reliable AI in software engineering

Massimo Tisi[1,*], Jordi Cabot[2], Davide Di Ruscio[3] and Antonio Garcia-Dominguez[4]

[1]*IMT Atlantique, LS2N (UMR CNRS 6004), 4 rue Alfred Kastler, F-44307 Nantes, France*
[2]*Luxembourg Institute of Science and Technology, 5 Av. des Hauts-Fourneaux, L-4362 Esch-sur-Alzette, Luxembourg*
[3]*DISIM – University of L'Aquila, Via Vetoio, Loc. Coppito, L'Aquila, Italy*
[4]*Department of Computer Science, University of York, York, YO10 5DD, United Kingdom*

Abstract

The reliable application of LLM-based agents to software engineering requires a tremendous increase in their accuracy and minimisation of their bias. While LLMs continue increasing in size and performance, it seems that phenomena like hallucinations of a single agent are substantially inevitable, since they are linked to the fundamental inference mechanism in generative models. On the other hand, evidence starts accumulating about the possibility of achieving the required performance by collaboration and debate among groups of agents. As it happens among humans, the quality of work can increase with specialisation of workers on tasks, organised collaboration, and discussion among workers with different backgrounds. Differently from humans, the instantiation of multiple required AI agents, and the collaboration and discussion among them, are very fast and cheap, making this approach particularly convenient.

The MOSAICO EU project proposes the theoretical and technical framework to implement this approach and to scale it to very large groups of collaborating agents, i.e. AI-agent communities. The proposed solutions rely on an integrated platform, handling communication, orchestration, governance, quality assessment, benchmarking and reuse of AI agents. MOSAICO is integrated with existing software development environments, to present the results to software engineers, and allow expert users to intervene in the AI decisions. The performance and reliability of MOSAICO technologies and tools to achieve given software engineering tasks are assessed within four different use-case scenarios coming from immersive technologies, bank/financing, aerospace and Internet of Things sectors. The long-term adoption of MOSAICO results and technologies will be ensured by open-sourcing the code and fostering an open collaboration to enhance user engagement in the MOSAICO community.

Keywords

Generative AI, Large Language Model, AI-Assisted Software Engineering, Responsible AI

1. Introduction

Generative AI based on Large Language Models (LLMs) is increasingly being applied to software engineering (SE) tasks, with very promising results. As assistants in software development, such models are very fast (after training) and relatively cheap, but they can also be highly unreliable and possibly biased. The lack of reliability hampers the general applicability of generative AI to (possibly safety-critical) SE tasks. Current research is mainly focused on improving the reliability of AI assistance by traditional software verification and validation methods. On the other hand, with the landscape of AI quickly evolving, we witness a rapid increase in the variety of high-quality accessible models (e.g., more than 40 independent LLMs are available at present). We anticipate that the cost of accessing such models

RPEatCAiSE25: Research Projects Exhibition at the International Conference on Advanced Information Systems Engineering, June 16-20, 2025, Vienna, Austria

[*] Corresponding author.

massimo.tisi@imt-atlantique.fr (M. Tisi); jordi.cabot@list.lu (J. Cabot); davide.diruscio@univaq.it (D. Di Ruscio); a.garcia-dominguez@york.ac.uk (A. Garcia-Dominguez)
https://massimotisi.github.io/ (M. Tisi); https://jordicabot.com/ (J. Cabot); https://www.disim.univaq.it/DavideDiRuscio (D. Di Ruscio); https://www-users.york.ac.uk/~agd516/ (A. Garcia-Dominguez)
0000-0001-7891-9138 (M. Tisi); 0000-0003-2418-2489 (J. Cabot); 0000-0002-5077-6793 (D. Di Ruscio); 0000-0002-4744-9150 (A. Garcia-Dominguez)

will further decrease in time, as open and self-hosted solutions improve and become widespread. At the same time, the emergence of Small Language Models is reducing the energy footprint of generative AI in several tasks.

MOSAICO vision. Our vision for the future of SE is a reliable application of generative AI to SE tasks, enabled by the coordinated and supervised collaboration of – a possibly large number of – different LLM-based AI agents, that we call a Community of AIs (or AI-agent community). This cooperation will need precise communication among AI agents in all phases of the SE process. Software modelling languages are precise and uniform descriptions of software in all its life-cycle, and have been historically designed as communication tools among software engineers. They are the most natural candidates for exchanging artefacts in the communication among AI agents for SE, and between agents and human engineers.

MOSAICO overall concept. The project aims to produce a holistic methodology and a set of solutions for the engineering and operation of communities of AIs across the SE life-cycle. The solutions will be composed into an integrated MOSAICO platform, handling communication, orchestration, governance, quality assessment, benchmarking, and reuse of AI agents. MOSAICO will be integrated with existing software development environments, to present the results to software engineers, and allow expert users to intervene in the AI decisions.

MOSAICO platform components and capabilities. At the end of the journey, the consortium expects to release a modular platform, working as an "on-demand SE platform" composed of a set of solutions that can be used by themselves or in combination, depending on the SE task to be created. The modules embarked in the MOSAICO platform are: 1) **A protocol for streamlining the communication of SE tools with AI agents, and among AI agents participating in SE activities**, 2) **A management architecture for AI agents for SE**, including inventory, discovery, provisioning, monitoring, and tracking, 3) **A high-performance orchestration system for AI agents**, based on the definition of objectives of an individual agent and its role in bigger objectives, 4) **A trustable supervision and governance layer** parameterizable by different agreement algorithms, coming from the literature on consensus in multi-agent systems, or crowdsourcing.

2. MOSAICO overview

The reliable application of LLM-based agents to SE requires a tremendous increase in their accuracy and minimisation of their bias. While LLMs continue increasing in size and performance, it seems that phenomena like hallucinations of a single agent are substantially inevitable, since they are linked to the fundamental inference mechanism in generative models [1]. On the other hand, evidence starts accumulating about the possibility of achieving the required performance by collaboration among LLM agents [2], and debate among groups of agents [3, 4]. Even simple voting and sampling increase the accuracy of LLM agents in some scenarios [5]. As it happens among humans, the quality of work increases with specialisation of workers on tasks, organised collaboration, and discussion among workers with different backgrounds. Differently from humans, the instantiation of multiple required AI agents, and the collaboration and discussion among them, are very fast and cheap, making this approach particularly convenient. The MOSAICO project proposes the theoretical and technical framework to implement this approach and to scale it to very large groups of collaborating agents, i.e. AI-agent communities.

Project information. Acronym: MOSAICO. **Full name:** Management, Orchestration and Supervision of AI-agent COmmunities for reliable AI in software engineering. **Duration:** from January 2025 to December 2027. **Participants:** Institut Mines Telecom, Luxembourg Institute of Science and Technology, University of York, Università degli Studi dell'Aquila, Netcompany-Intrasoft, Immersion, National Bank of Greece, Collins Aerospace, Unparallel Innovation, Qodo, Eclipse Foundation, F6S Network. **Funding Program:** Horizon Europe - HORIZON-CL4-2024-DIGITAL-EMERGING-01-22 - Fundamentals of Software Engineering (RIA). **URL:** https://mosaico-project.eu/.

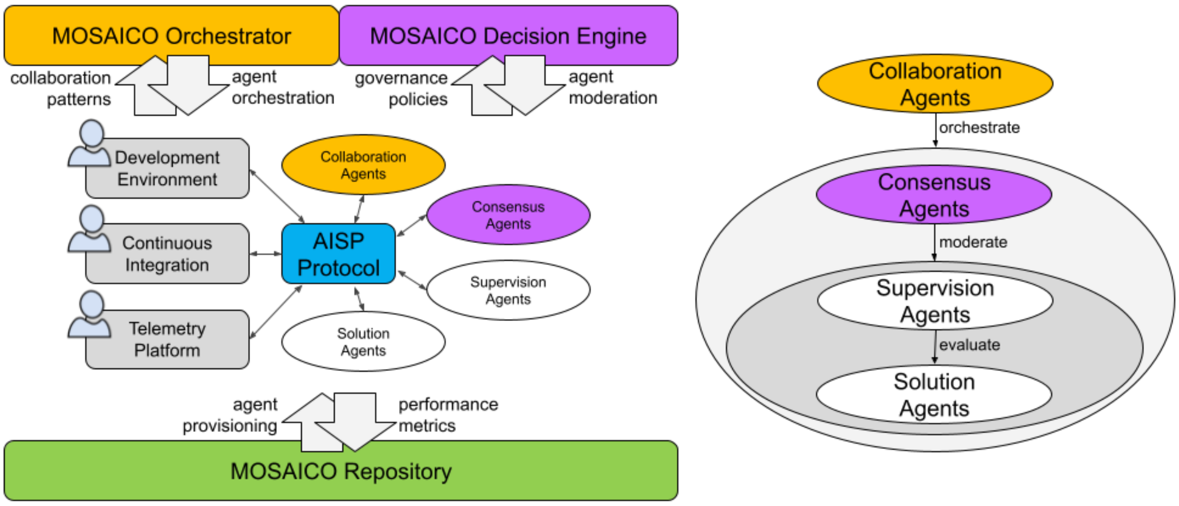

Figure 1: Overview of the MOSAICO platform (left) and categories of MOSAICO agents (right)

The diagram in Figure 1 (left-hand side) gives a high-level view of the components of the MOSAICO platform. In order to be effective, the concept of AI-agent communities needs to be pervasive in the development environment. First of all, it impacts the communication mechanism among agents and between agents and all standard software engineering tooling (including IDEs, CI platforms, and telemetry platforms). MOSAICO proposes the AISP Protocol (Solution 1), for standardised, precise and fine-grained communication initiated by agents or tools. Communicated artefacts (e.g. requirement, design models, code) will depend on a taxonomy of SE tasks (and related inputs/outputs), that will accommodate each agent. MOSAICO proposes the MOSAICO Repository (Solution 2), based on such a taxonomy. The framework will be able to search and provision agents from the repository. During the agents' activity, the repository will store metrics about the performance of the agents in the community performing the given task, according to a set of provided KPIs. These metrics will be used for choosing suitable agents for subsequent iterations of the task. Once the needed agents are instantiated, the MOSAICO Orchestrator (Solution 3) efficiently coordinates their execution, based on given collaboration patterns. The collaboration pattern for a given (sub)task may be provided by the user, or automatically computed by specific MOSAICO Collaboration Agents, fine-tuned for computing collaboration patterns. A key part of the collaboration will be dedicated to supervision and governance. MOSAICO proposes the MOSAICO Decision Engine (Solution 4) that, given a governance policy, moderates the discussion among agents to reach a consensus. The engine supports user-provided rule-based policies, but also intelligent consensus strategies computed by specific MOSAICO Supervision Agents, fine-tuned on the literature on consensus dynamics.

The right-hand side of Figure 1 summarises the categories of AI agents proposed by MOSAICO. Starting from the bottom, *Solution Agents* exploit generative AI to compute proposed solutions to given SE tasks. For instance, a set of solution agents may generate different technical models starting from the application requirements given by the user. *Supervision Agents* evaluate the work of solution agents. For instance, a set of supervision agents may evaluate the generated technical models, e.g., for coverage of the requirements and conciseness. *Consensus Agents* moderate a discussion involving supervision agents, solution agents, and humans if needed, with the support of the MOSAICO Decision Engine, in order to reach a consensus. For instance, a consensus agent may be charged with identifying the most concise technical model that covers all the requirements, by coordinating generations by solution agents and evaluations by supervision agents. Finally, *Collaboration Agents* deal with the decomposition in subtasks, assignment of subtasks to other kinds of agents (solution, supervision, or consensus), and orchestration of their work, by communicating with the MOSAICO Orchestrator. For instance, a collaboration agent identifies the generation of an optimal technical model as a sub-task of the global development task, assigns it to the consensus agent, and connects its input/outputs to other subtasks.

Note that a collaboration agent is also a particular kind of solution agent, since it generates collaboration patterns. Hence, collaboration agents can be evaluated by supervision agents, and the hierarchy in the right-hand side of Figure 1 recurs.

3. Project objectives

The project aims at achieving 6 key specific objectives, detailed in the following.

3.1. AI-agent server protocol

The challenge is to design and implement a protocol with which AI agents can interoperate with each other and with software development tools (such as Integrated Development Environments or Continuous Integration platforms) in a disciplined and uniform way. We draw inspiration from Language Server Protocol (LSP [10]), which standardised operations related to static program analysis and code completion and navigation, and allowed language servers offering such capabilities to be reused across different tools. The envisioned AI Agent Server Protocol (AISP) enables AI agents to declare their capabilities in terms of activities they can perform, languages and file-types they support, receive input (context) from the tool and feedback from the user and other agents, and return their output (e.g., completion suggestions, new artefacts) to the development environment.

For the architecture of the server for the protocol, the approach outlined in Fig. 2 will be followed. An open-source LLM abstraction library such as LiteLLM[1] will be used in order to support running locally a variety of LLM models, in combination with the open-source LangChain framework for context-aware reasoning applications. This will allow us to continue reusing state-of-the-art LLM models throughout the duration of MOSAICO. For the reference implementation of the client for the protocol, a VS Code extension which communicates with the above server will be developed and tested on both VS Code (as an example of a desktop IDE) and Eclipse Theia (as an example of a web-based IDE, which is compatible with VS Code extensions). The reference server will publish anonymised usage and task performance metrics to an event bus, which will feed a telemetry platform for agent observability (e.g., by indexing into Elasticsearch and providing Kibana visualisation dashboards, or by feeding into the tracing capabilities of LLM engineering platforms such as LangFuse[2]).

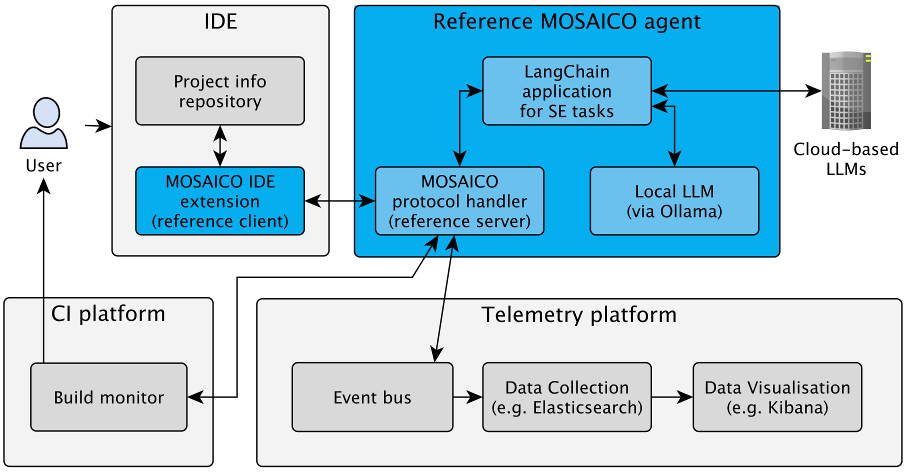
IDE
Project info repository
MOSAICO IDE extension (reference client)
Reference MOSAICO agent
LangChain application for SE tasks
MOSAICO protocol handler (reference server)
Local LLM (via Ollama)
Cloud-based LLMs
User
CI platform
Build monitor
Telemetry platform
Event bus
Data Collection (e.g. Elasticsearch)
Data Visualisation (e.g. Kibana)

Figure 2: Overview of the AI-agent server protocol

[1] https://www.litellm.ai/
[2] https://langfuse.com/

3.2. Repository of AI agents for SE

The second objective aims to create a repository of AI agents tailored for SE tasks. The repository's design will facilitate the effective management of AI agents based on both functional characteristics and quality attributes (KPIs), such as accuracy, failure rate, and latency. The repository will offer standardised metadata for the AI agents it houses, ensuring that detailed and consistent information about their capabilities, limitations, training data, fine-tuning options, and recommended use cases is readily available. This categorization is crucial, particularly when AI agents provide similar functionalities but exhibit varying quality characteristics. To promote repository acceptance, we will develop languages and tools that allow the specification of custom-quality models for AI agents. These models will serve as the basis for assessing the quality of AI agents in alignment with defined quality characteristics. The quality assessment process will be automated, and the results will be utilised to annotate AI agents stored in the repositories. These annotations will play a key role in searching for relevant AI agents aligning with the specific SE tasks to be supported. The components constituting the repository of AI agents are illustrated in Figure 3 (left).

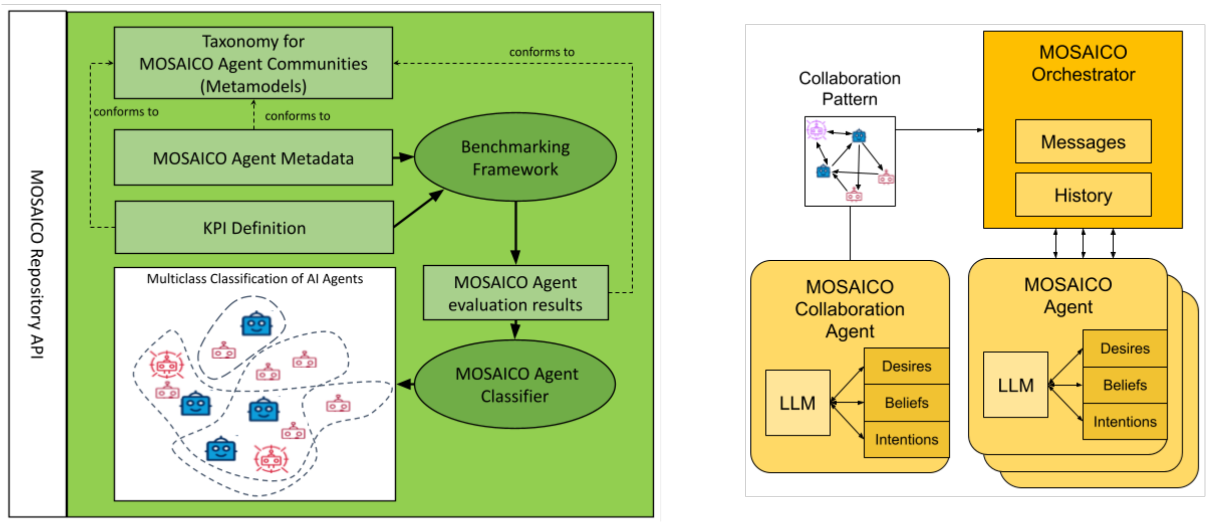

Figure 3: Overview of the repository of AI agents for SE (left) and the coordination and collaboration layer (right)

3.3. Coordination and collaboration of AI-agent communities

Many SE tasks can be achieved by different agents. These agents could just compete among them, or collaborate. For collaboration, we adapt the well-known BDI (Belief-Desire-Intention) framework to LLM-based agents. The BDI model is a way of representing the mental states of an agent, such as beliefs, desires, and intentions. These mental states, represented as a set of variables and then being manipulated by the agent, will support AI agent decisions about what to do. To close the orchestration loop, we will need to define a coordination language that expresses how these agents should work together. We will propose a standardised language to express collaboration patterns for SE tasks. The language will be tailored from existing modelling languages for processes (e.g., BPMN), that are already known by existing LLMs. Such LLMs will be used to automatically extract a dataset of such collaboration patterns from standard operating procedures, technical documents and research papers. The dataset will connect models describing the pattern to textual description of the task performed through the collaboration. Finally, a Collaboration Agent will be trained on this dataset. It will be able to compute suitable collaboration patterns for a given SE task, and to propose alternative patterns in case of low performance. Figure 3 (right) shows the conceptual architecture of the orchestration.

3.4. AI-agent community governance and supervision

Supervision of the agent results is a must, both in competing and collaborating scenarios. Simple supervision just involves checking if the agents are well-behaved (i.e., agents do not just destroy the work of other agents, e.g., an agent that decides the most efficient way to integrate the PR of another agent must not simply delete it). More complex supervision involves evaluating whether the result of an agent (or a community of agents) is good (for any type of definition of "good"). Complex tasks (e.g., assessing whether a piece of generated code is free from vulnerabilities) often require more than one Supervision Agent to participate in the "discussion". Given the individual assessment of each Supervision Agent, we need to reach a conclusion. Such a conclusion can require several iterations and be ultimately based on a voting and agreement policy defined by the project owner.

MOSAICO will provide a governance language to define governance policies, including types of agreement (consensus, majority voting, human-driven, ...) and constraints required to validate the agreement (e.g., minimum number of votes, max deadlines,...). The language will also allow users to express qualitative aspects such as the uncertainty agents can have about their own answers and the uncertainty other agents (or the humans involved) may have about the quality of the involved agents and how this is going to affect the governance.

The architecture is illustrated in the Figure 4. Our first-level agents, in charge of responding to the requests of the agent orchestration system, propose their solutions for the task at hand. These solutions are evaluated by our second-level agents, the Supervision Agents, who give their opinion (e.g., a prioritisation of the quality of the solutions together with their own level of confidence in the evaluation). For simple governance rules (e.g., a simple majority vote) the decision engine will collect those opinions and determine the best solution. For more complex rules (e.g., when the project owner asks for a consensus), a dedicated consensus agent will intervene and will aim to start a discussion among the Supervision Agents to try to reach such consensus or at least to reach enough consensus/majority as requested by the governance policy. This discussion could also be aimed at increasing the confidence in the Supervision Agent's opinion to go over a threshold also stated in the policy. Once such a consensus is reached (potentially with the help of human evaluators if they are also allowed to participate according to the governance policy) the final solution will be selected and communicated back to the Collaboration Agent to advance to the next task. It will be always possible to analyse the trace of the discussion, guaranteeing the explainability of the community decision.

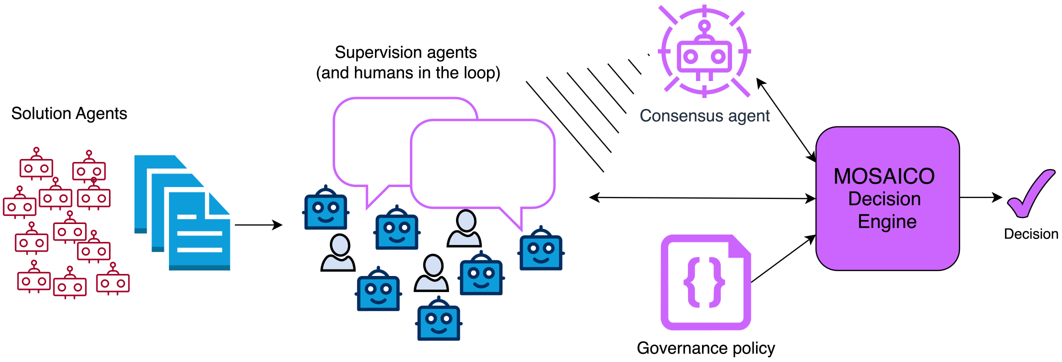

Figure 4: Overview of the AI-agent community governance and supervision

3.5. Validation of MOSAICO

The merit of the MOSAICO SE tools and techniques must be proven in real-life use cases, involving the development of realistic software products and services beyond simple examples and validation scenarios. MOSAICO will be deployed in 4 pragmatic use cases in different sectors, involving a variety of software development actors (e.g., software integrators, end-users of software products, Independent Software Vendors (ISVs) (including SMEs)) and processes (e.g., traditional SE, agile/DevOps process) in different SE scenarios (e.g., develop from scratch, evolve existing software). This validation will showcase the power of the MOSAICO concept and will help the consortium to identify the scenarios where the merits of MOSAICO are maximised. Moreover, MOSAICO must address an evaluation challenge, which lies in the identification of the SE aspects that are essentially improved through MOSAICO such as automation, speed, software quality and developers' satisfaction. To this end, the project will design and use a multi-facet evaluation methodology that will comprise evaluation methods and benchmarks for all of the above-listed evaluation aspects. Furthermore, the project's evaluation methodology will solicit and analyse stakeholders' feedback.

3.6. Long-term adoption of MOSAICO

In the process of building a community to sustain the results of a project, user engagement is essential. Indeed, user engagement through actions like hackathons and webinars involves creating interactive and participatory experiences. MOSAICO ensures long-term adoption of its results by open-sourcing the code and fostering an open collaboration, such as open-source initiatives, to enhance user engagement in the MOSAICO community

4. State of the art

In 2024, different concepts of LLM agents emerged in industry, especially for specialised AI assistants. AutoGPT, an open-source implementation by OpenAI, autonomously pursues predefined goals. LangChain Agents, part of the LangChain framework, employs LLMs to make decisions and choose sequences of actions within applications. The Transformers Agent, developed by HuggingFace, serves as an experimental natural-language API built upon the transformers repository. Academic research on LLM-based multi-agents started in 2023 and is rapidly increasing. See [6] for an overview. CAMEL, a communicative agent framework, showcases the use of role playing for chat agents to effectively communicate [8]. Multi-Agent Debate, explored in [3] and [4], proves to be a compelling approach for encouraging divergent thinking and enhancing the factuality and reasoning of LLMs. AutoGen, an open-source framework [7], enables developers to construct LLM applications through multiple conversing agents. MetaGPT [2], specialises in automatic software development, uses a multi-agent conversation framework for efficient LLM application. In March 2024 Microsoft released a preliminary article on a similar collaborative multi-agent framework [9]. Each one of these articles show that a multi-agent system can outperform a single-agent system in specific scenarios. For instance, [5] shows that a simple voting and sampling strategy can already significantly increase accuracy.

More recently, industry has recognized the need to improve the interoperability of LLM agent solutions, and has started to propose specifications for various communication protocols. Anthropic has proposed the Model Context Protocol[3] to standardize how agents can obtain additional information from other systems. LangChain has open-sourced their Agent Protocol, although at the time of writing it does not support agent-to-agent communication[4]. The most recent development is Google's Agent2Agent protocol[5], which shares many of the goals we set for the AISP. Part of the work of WP1 will be to evaluate these industry-led specifications and consider any adaptations they may require to meet our vision of reliable application of generative AI to SE tasks.

As opposed to the common emphasis on small teams of AI agents, our goal is to transcend these limitations and achieve scalability by extending our framework to encompass entire communities and crowds of AI entities. Current solutions in the field predominantly concentrate on constructing LLM applications. Our approach distinguishes itself by addressing a wide taxonomy of sub-tasks within SE, in collaboration with human counterparts. Finally, unlike most existing solutions that limit themselves to static agent conversation patterns, our framework is designed to embrace and support dynamic patterns defined by AI agents, enabling adaptive and responsive interactions.

[3] https://www.anthropic.com/news/model-context-protocol

[4] https://github.com/langchain-ai/agent-protocol/issues/5

[5] https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/

5. Conclusion, progress and relevance to CAiSE

We described the MOSAICO European project, which aims to produce a framework for the Management, Orchestration, and Supervision of AI-agent COmmunities. The main objective of MOSAICO is to address the complexities of software applications based on multitudes of collaborating LLM-based agents, ensuring higher reliability and mitigating biases through collective intelligence and continuous human-agent interaction.

In the first 3 months (of the 3-year span of the project), efforts focused on a comparison of state-of-the-art multi-agent systems and protocol design approaches. We built early prototypes to experiment ideas about the four technical solutions of the project. We worked at precisely defining the four use cases and we prepared a detailed dissemination strategy.

Relevance to CAiSE 2025. The MOSAICO project fits in the topic "Novel Approaches to Information Systems Engineering - Artificial Intelligence including generative AI and Machine Learning" of CAiSE 2025. Indeed, MOSAICO will contribute significantly to supporting the development of advanced information systems, by promoting the employment of agent interactions that must be aligned with the needs of users and organizations. Today information system projects are becoming more and more complex due to the need of integrating AI components (among others). MOSAICO's goal of taming this complexity by embedding agents that can assist in the development of such systems in a reliable way will be of key importance in future information system engineering.

Acknowledgments

This paper is supported by the European Union under the Grant Agreement No 101189664. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HADEA). Neither the European Union nor the granting authority can be held responsible for them. The paper authors are grateful to all the MOSAICO project participants for their contribution.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Z. Xu, S. Jain, and M. Kankanhalli, "Hallucination is Inevitable: An Innate Limitation of Large Language Models." arXiv, Jan. 22, 2024. Accessed: Mar. 17, 2024. [Online]. Available: http://arxiv.org/abs/2401.11817

[2] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, et al. Metagpt: Meta programming for multi-agent collaborative framework. arXiv preprint arXiv:2308.00352, 2023.

[3] ian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. Encouraging divergent thinking in large language models through multi-agent debate. arXiv preprint, 2023.

[4] Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. CoRR, abs/2305.14325, 2023. doi: 10.48550/arXiv.2305.14325. URL https://doi.org/10.48550/arXiv.2305.14325.

[5] J. Li, Q. Zhang, Y. Yu, Q. Fu, and D. Ye, "More Agents Is All You Need." arXiv, Feb. 03, 2024. Accessed: Feb. 29, 2024. [Online]. Available: http://arxiv.org/abs/2402.05120

[6] T. Guo et al., “Large Language Model based Multi-Agents: A Survey of Progress and Challenges.” arXiv, Jan. 21, 2024. Accessed: Mar. 17, 2024. [Online]. Available: http://arxiv.org/abs/2402.01680

[7] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework, 2023. URL https://doi.org/10.48550/arXiv.2308.08155

[8] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: communicative agents for ”mind” exploration of large scale language model society. CoRR, abs/2303.17760, 2023. doi: 10.48550/arXiv.2303.17760. URL https://doi.org/10.48550/arXiv.2303.17760

[9] M. Tufano, A. Agarwal, J. Jang, R. Z. Moghaddam, and N. Sundaresan, “AutoDev: Automated AI-Driven Development.” arXiv, Mar. 13, 2024. Accessed: Mar. 19, 2024. [Online]

[10] Microsoft Corporation, “Language Server Protocol Specification — 3.17”. October 2022. URL https://microsoft.github.io/language-server-protocol/specifications/lsp/3.17/specification/. Accessed: April 12, 2025. [Online]