

Response to Reviews

Dear Editor,

We are grateful to the reviewers for their substantial effort spent for evaluating our work. We would like to submit the revision of our research article “A Hybrid Role-Based Reference Architecture for LLM-Enhanced Multi-Agent Systems”. We appreciate the reviewers’ comments and have addressed all of them.

Below, we give a list of how we addressed each specific comment. In our responses, we use “initial article” to refer to the article that has previously been reviewed, and we use “revised article” to refer to the article that we submit upon this revision. We have [highlighted the parts of the paper](#) that we revised to help make our modifications easier to identify.

Note on scope. The revisions reported below are those we were able to incorporate within the time available for the *pre-proceedings* version. A subset of the reviewers’ comments—particularly those that require substantive structural changes to the architecture (e.g., explicit memory and communication components in Figure 2, deeper discussion of roles as run-time entities), additional empirical evaluation, and further refinements to the figures and supplementary material—are still being worked on. We are committed to addressing these remaining items in the *post-proceedings* version of the paper, and the responses with empty boxes below indicate the items that are currently in progress.

Sincerely,

Tansu Asici, Önder Gürcan and Geylani Kardaş

Reviewer 1

Comment 1:

The related work section omits several relevant papers on the integration of BDI agents and LLMs, including:

- Dynamic plan generation with LLMs: automatic execution of abstract BDI-agent goals (<http://doi.org/10.1080/17445760.2025.2541956>)
- Integrating Belief-Desire-Intention agents with large language models for reliable human–robot interaction and explainable Artificial Intelligence (<https://doi.org/10.1016/j.engappai.2024.109771>)
- Let Me Talk to You! Natural Language Interaction Between Humans and BDI Agents via ChatBDI (<https://doi.org/10.3233/FAIA251242>)
- BDI Agents in Natural Language Environments (<https://www.ifaamas.org/Proceedings/aamas2024/pdfs/p880.pdf>)

Response: We thank the reviewer for pointing out these highly relevant works. We have added these four BDI+LLM integration papers to the Related Work section and discussed them in relation to our approach.

Comment 2:

The authors should also discuss traditional normative technologies used in BDI agents, such as Moise, and explain why these approaches were deemed unsuitable for their solution.

Response: We have added a discussion in the Related Work section about traditional BDI normative technologies (in particular, Moise) and explained how our role-based approach differs from and complements these existing solutions.

Comment 3:

It is also unclear what, exactly, was adapted in Figure 1 from the previously published paper cited as reference [5]. More generally, the paper needs a clearer explanation of how it differs from that earlier work. In particular, the authors should explain what their current approach enables that was not possible before.

Response: We have clarified in the caption and the surrounding text of Figure 1 exactly which elements are adapted from reference [5] and which are new contributions of this work. In addition, we have extended the text to explicitly explain how our current approach differs from our previous work and what it enables that was not possible before.

Comment 4:

Figure 2 presents a deliberation mechanism with six different decision-making mechanisms, but the paper does not clearly explain how these mechanisms relate to the proposed role structure.

Response:

Comment 5:

Finally, the paper does not provide experimental results or empirical evidence to demonstrate the advantages of the proposed architecture over existing frameworks. Aside from code snippets in a case study, there is no real validation. For a regular paper, some form of evaluation is necessary.

Response:

Reviewer 2

Comment 1:

The authors do a great job in explaining the technical details of the proposed framework but in my opinion the paper lacks comparison with related works in the area of LLM-based agent engineering. The related work section focuses on other hybrid approaches but to be fair it is not entirely clear to me how the work positions itself within the context of “pure” LLM frameworks such as LangGraph. In my (limited) experience, frameworks for LLM agents allow to define workflows similarly to how the authors do involving LLM and non-LLM related action steps so I would be able to appreciate more the paper if the authors could show what would be key differences in their approach compared to the existing frameworks.

Response: We have extended the Related Work section with an explicit comparison to “pure” LLM agent frameworks such as LangGraph. We highlight the key differences between our role-based approach and workflow-oriented LLM frameworks, in particular: (i) roles as first-class design-time and run-time abstractions tied to responsibilities and safety constraints, rather than nodes in a workflow graph, (ii) the separation between decision-making and execution mechanisms with hybrid action types, and (iii) framework-agnostic annotation-based specification that can be mapped onto different underlying agent frameworks.

Comment 2:

I understand the contribution of using the role abstraction and a structured formalization, but a weakness which is not discussed is the heavy reliance on strings that point to method names in the annotations. This can be quite critical when refactoring the codebase and defies the argument that the framework improves maintainability.

Response: We thank the reviewer for raising this valid concern. We have added a discussion of this limitation in the Discussion/Limitations section, acknowledging that string-based method references in annotations can be fragile under refactoring. We also outline possible mitigation strategies, such as compile-time validation via an annotation processor, IDE tooling support, and method-reference alternatives, as future directions for strengthening the maintainability claim.

Comment 3:

The abstract is quite long, I would suggest to reduce it and make it more focused.

Response: We have tightened the abstract by merging the previously fragmented descriptions of the meta-model and its Java realisation into a single, more direct statement, and by removing redundant phrasings throughout. The revised version preserves the problem–contribution–demonstration structure while reducing the overall length, with each paragraph now focused on a single clear message.

Comment 4:

Consider exporting the schemas in pdf for better quality.

Response: We have re-exported all three schematic figures (the reference architecture, the agent internal architecture, and the role meta-model) from the source `.drawio` file as vector PDFs and updated the `\includegraphics` references in the manuscript accordingly. The figures are now rendered as vector graphics, which eliminates pixelation when zooming and yields noticeably better print quality.

Comment 5:

The UML diagram is not very clear, consider using a less busy scheme or partitioning it.

Response:

Reviewer 3

Comment 1:

The agent architecture in Figure 2 seems to miss an explicit representation of memory (in a broad sense); rules and state would typically be managed by such a component.

Response:

Comment 2:

The agent architecture in Figure 2 seems to miss an explicit representation of how communication is handled; is receiving a message treated as perceiving the environment, or is it handled separately? Both options exist in AOSE, though explicit message handling is more prevalent (e.g., Jason, architectural patterns from Danny Weyns [1], work around BSPL [2]).

[1] Danny Weyns. Architecture-based design of multi-agent systems. Springer Science & Business Media, 2010.

[2] Samuel H. Christie, Munindar P. Singh, and Amit K. Chopra. 2023. Kiko: Programming Agents to Enact Interaction Models. In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1154–1163.

Response:

Comment 3:

It is a bit unclear what roles represent in this architecture. The paper presents them as run-time entities. Are roles computational entities — actual architectural components that process data? If so, what is the motivation for this design choice? As presented, most elements associated with a role (state, rules, and action specifications) appear to reduce to factual and procedural knowledge. It seems the proposed reference architecture may implicitly encode implementation choices that overconstrain alternative realizations of roles.

Response:

Comment 4:

One point appears unclear in Section 5.1 (page 9): the annotations are described as framework-agnostic, yet `@AgentSpec` is applied to a class that extends “the framework’s Agent base class”. Are the annotations intended to be standalone, or is there an intended coupling with class inheritance? This apparent tension

between framework-agnostic annotations and framework-dependent inheritance should be clarified.

Response: We have clarified this tension in Section 5.1. The annotations themselves are framework-agnostic metadata that can be processed independently, while the inheritance from the framework’s `Agent` base class in our implementation is an integration choice for the specific framework used in the case study, not a requirement of the annotation-based specification. We have rewritten the relevant paragraph to make this distinction explicit.

Comment 5:

Page 12, “here the hotel provider adopts a PROFESSIONAL tone with CONCISE verbosity, ensuring formal yet brief responses during negotiations”: *ensuring* is a strong claim, as this ultimately depends on the behavior of the underlying model and its adherence to instructions.

Response: We have softened the wording as suggested. The phrase has been changed to “guiding the model toward formal yet brief responses” to reflect that the outcome depends on the underlying model’s adherence to instructions.

Comment 6:

The presentation of the implementation references methods that are not shown in the listings (e.g., `canFulfillRequest()` or methods invoking LLMs to generate negotiation messages); including these in an appendix could improve clarity without affecting space constraints.

Response:

Comment 7:

It would be interesting to expand the discussion on responsibilities and how they are operationalized; including example prompts in an appendix would also be helpful.

Response:

Comment 8:

Typos / editing issues:

- “actions are typed by implementation approach” ⇒ needs reviewing.
- Figure 1: the text is too small; the font size could be easily increased for legibility.
- “It might use Local Code actions for deterministic or performance matter steps” ⇒ this phrase is unclear; for steps where deterministic behavior and performance matter?
- “how Contract Net Protocol (CNP) (a classic AOSE interaction pattern) combines with hybrid action types to create a robust, LLM-enhanced MAS” ⇒ can be combined? integrated?

Response: We thank the reviewer for these careful suggestions. We have applied the following edits in the revised article:

- “actions are typed by implementation approach” has been rephrased to “each action declares its implementation type”.
- “deterministic or performance matter steps” has been rephrased to “steps where deterministic behavior and performance matter”.
- “how Contract Net Protocol (CNP) ... combines with” has been rephrased to “can be integrated with”.
- Regarding the font size of Figure 1: we have updated the underlying source diagram to enlarge the labels, re-exported all schematic figures (the reference architecture, the agent internal architecture, and the role meta-model) from the source `.drawio` file as vector PDFs, and updated the manuscript to include the PDF versions. The figures are now rendered as vector graphics with larger, fully legible text and no pixelation when zoomed.

References