Neapolis University Pafos

Course Code: IS509

Task 3: Academic Introduction to a Proposal

Name: Aleksandr Petrunin

Student ID: 1251114137

November 30, 2025

Contents

1 Introduction

1.1 Background and context

Goal: Establish research territory (broad academic landscape + demo knowledge of the field)
Strategy: Relevant + credible sources + historical context with current issues + balance depth and breadth.

1.1.1 Broad context: Multi-agent systems with LLMs

The advancement of Large Language Models (LLMs) has ushered in a new era of autonomous AI agents capable of performing complex tasks with minimal human intervention [1]. Multi-agent systems (MAS) tend to outperform single-agent systems due to the larger pool of shared resources, coordination and specialization [2]. Businesses and individuals can leverage LLMs to build autonomous multi-agent systems with frameworks, such as AutoGPT, CrewAI, and Langchain to automate workflows and optimize decision-making [3] [4] [5].

1.1.2 Narrow focus: Agent-to-agent communication concepts and standards

However, current implementations of multi-agent systems predominantly rely on natural language processing (NLP) for inter-agent communication. Agent-to-agent communication through natural language introduces several challenges, including ambiguity, misinterpretation, and inefficiencies in task execution [6]. A pool of protocols and orchestration mechanisms for structured communication between agents exists [7].

Older agent communication standards. KQML (Knowledge Query and Manipulation Language) by DARPA is an early standard that provides a framework for knowledge sharing and querying among agents [8]. FIPA Agent Communication Language (ACL) tried to cover KQML limitations and defines a set of communicative acts and message structures for agent communication [9].
These languages found early adoption in academic, research, and experimental environments. They layed the foundation of modern software ecosystems, enabling everything from automated trading to smart home devices to work together [10].
However, FIPA-ACL and KQML were designed for symbolic AI systems with explicit knowledge representations. They are not so suitable for modern LLM-based agents that operate on distributed representations and natural language. Their fixed ontology requirements and predefined performative sets make them incompatible with the emergent behaviors of current multi-agent systems. On top of that, these languages lack of modern web protocol integration like HTTP, JSON, REST APIs. It might have prevented widespread adoption beyond research contexts.

Current solutions. OpenAI Agents SDK is an orchestration mechanism, growing adoption in OpenAI ecosystem, not an inter-agent communication protocol [11]. Model context protocol (MCP) by Anthropic provides a standardized way for AI models to connect with and use external tools and data sources - it is a client-to-agent protocol [12].

Emerging standard. With ACP's (protocol by IBM) recent merger into Google's A2A (Agent to Agent) protocol under the Linux Foundation, A2A is rapidly becoming the dominant open standard for agent-to-agent communication [13]. In contrast to OpenAI Agents SDK, which provides only local IPC semantics, and the Model Context Protocol (MCP), which focuses on client-to-agent tool invocation, A2A is explicitly architected for distributed multi-agent systems. It defines message envelopes, routing, agent metadata, capability discovery, and extensible JSON-based interaction primitives. A2A and MCP are considered as complementary standards [14].

1.1.3 Specific gap: Semantic limitations of A2A protocol

Figure 1 illustrates the communication landscape in multi-agent systems. Traditional formal protocols (FIPA-ACL, KQML) offered high semantic precision and verifiability. But they suffered from inflexibility and poor industry adoption. Modern LLM-based agents communicate through natural language prompts, planning steps, and role assignments. It provides adaptability, but also introduces ambiguity and non-determinism. Current standards like A2A provide syntactic infrastructure (message envelopes, routing, discovery) but lack semantic guarantees, leaving a gap. Our research targets this gap by proposing a typed semantic layer. This layer should bridge formal verification with LLM emergence behavior.

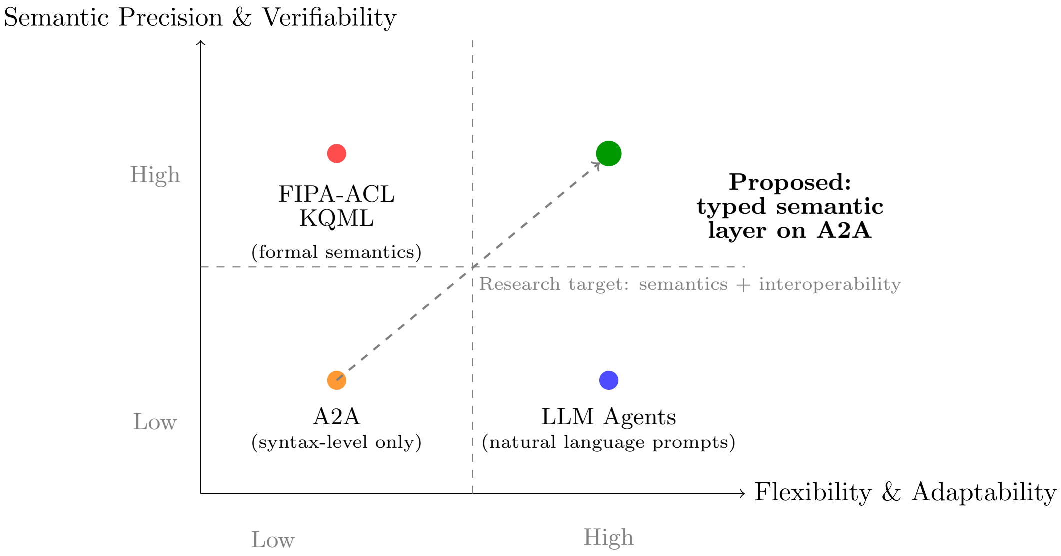
Semantic Precision & Verifiability
High
Low
FIPA-ACL
KQML
(formal semantics)
Proposed:
typed semantic
layer on A2A
Research target: semantics + interoperability
A2A
(syntax-level only)
LLM Agents
(natural language prompts)
Flexibility & Adaptability
Low
High

Figure 1: Positioning of existing agent communication paradigms relative to semantic verifiability and adaptability.

A2A protocol provides a much-needed open standard for agent interoperability. However, A2A standardizes only the syntax of communication: message envelopes, routing, and discovery. Unlike FIPA-ACL and KQML, which defined performatives and semantic frameworks, A2A does not prescribe message semantics, ontologies, shared meaning representations, or reasoning-level guarantees. This semantic gap motivates research into typed, non-NLP communication layers that can enable more reliable, verifiable, and deterministic multi-agent behavior. We want to implement guarantees that agents understand each other beyond message structure.

1.1.4 Solution: Typed semantic extension for A2A protocol

This research will develop a typed semantic extension for the A2A protocol that introduces formal performatives, interaction contracts, and verification constraints for multi-agent communication. We will leverage A2A's built-in extension mechanism (A2A Specification Section 4.6) to define strictly-typed message semantics inside the existing 'Message' metadata and extensions fields. The semantic layer will be implemented as an extension. It will be declared in Agent Cards and enable opt-in enforcement. At the same time backward compatibility will be preserved. We will prototype LLM-based MAS with this semantic layer and evaluate improvements in task reliability, negotiation consistency, and communication determinism. All three evaluation types will be compared to a baseline of natural-language-only A2A implementations.

1.2 Statement of the problem

Goal: Identify knowledge gaps(what remains unknown and unexplored → justify research).
LR → Gap analysis → problem statement/research aims.

The A2A protocol, now the dominant open standard for agent-to-agent communication under the Linux Foundation, provides a comprehensive syntactic infrastructure but deliberately leaves semantic interpretation unspecified. The specification (Section 4.6) explicitly supports extensions for additional functionality, yet provides no guidance on semantic guarantees [14].

Syntactic completeness without semantic guarantees. A2A defines three architectural layers: canonical data model (Task, Message, Artifact), abstract operations (Send Message, Get Task, etc.), and protocol bindings (JSON-RPC, gRPC, HTTP+JSON). However, the protocol intentionally adopts ”opaque execution” as a guiding principle—agents collaborate based on declared capabilities without sharing internal reasoning or imposing semantic constraints on message interpretation [14].

Concrete manifestations of the semantic gap. The absence of semantic layer results in:

1. **No performatives:** Unlike FIPA-ACL (request, inform, query-ref), A2A provides no standard communicative acts. It forces agents to infer intent from natural language;
2. **No interaction contracts:** Message exchange lacks pre/post-conditions or verification constraints. It results in weak formal validation of agent behavior;
3. **No shared ontologies:** The protocol provides no mechanism for establishing common meaning representations;
4. **Task state ambiguity:** A2A state transitions carry no semantic guarantees about what agents have agreed upon or accomplished.

Research questions. This gap raises critical questions: How can agents achieve deterministic communication within A2A's extension mechanism? What formal semantics are necessary for verifiable multi-agent coordination? Can a typed semantic layer maintain A2A's flexibility while enabling reliable task delegation and negotiation?

1.3 Justification of significance

Goal: Present my contribution to the field, show how my research addresses these gaps and advance knowledge. **Strategy**: Magnitude + Urgency + Impact.

This research addresses a gap in the emerging standard for agent-to-agent communication. A2A's explicit extension mechanism could provide a semi-standardized pathway for semantic enhancement, making this work potentially applicable to real-world deployments.

1.3.1 Magnitude: Industry momentum and standardization urgency

a) **Standard consolidation:** A2A emerged from the collaboration of Google's and IBM's competing protocols under Linux Foundation governanc. It is a signal of industry consensus on agent interoperability [13];

b) **Enterprise readiness:** A2A's design principles explicitly target enterprise deployment with built-in authentication, authorization, asynchronous processing, and security considerations (Sections 7, 13);

c) **Ecosystem growth:** Multi-agent frameworks (LangChain, AutoGPT, CrewAI) are adopting A2A as the interoperability layer;

d) **Extension mechanism adoption window:** As A2A moves from draft (v1.0) to production.

1.3.2 Urgency: Real-world deployment challenges

Current state: A2A agents communicate via natural language in Message.parts fields, relying on LLM interpretation without semantic constraints. This creates:

a) **Task coordination failures:** Agents misinterpret delegation intent in multi-turn interactions (contextId-linked tasks), leading to incorrect task state transitions;

b) **Negotiation non-determinism:** Without formal performatives, agents cannot reliably establish commitments, preconditions, or mutual beliefs;

c) **Prompt drift in long-running tasks:** A2A's async-first design supports tasks spanning hours or days, but semantic drift compounds over extended interactions;

d) **Verification impossibility:** Enterprise deployments require audit trails and compliance verification, but natural language messages in Task.history cannot be formally validated.

Consequences of inaction: Production multi-agent systems will continue to exhibit unreliable behavior, limiting deployment to non-critical applications and preventing adoption in regulated industries (healthcare, finance, autonomous systems).

1.3.3 Impact: Research and industry stakeholders

For researchers:

a) Bridges formal agent communication theory (FIPA-ACL, KQML) with modern LLM-based agents;

b) Provides empirical framework for evaluating semantic vs. syntactic protocol layers;

c) Establishes methodology for protocol extension design within industry standards.

For industry practitioners:

a) Immediate applicability: Semantic extension compatible with existing A2A implementations via Message.metadata and AgentCard.capabilities.extensions;

b) Backward compatible: Agents without semantic support continue functioning, enabling gradual adoption;

c) Enterprise value: Enables formal verification, audit trails, and compliance in regulated deployments.

For standardization bodies:

a) Demonstrates practical use of A2A's extension mechanism for protocol evolution;

b) Provides evidence-based guidance for potential inclusion in future A2A core specification;

c) Establishes patterns for domain-specific semantic extensions in multi-agent ecosystems.

While this research primarily targets horizontal agent-to-agent communication, the semantic layer architecture may inform future work on client-to-agent protocols (MCP) and human-agent interaction, where precise intent interpretation is equally critical.

1.4 Scope and limitations

Research scope. This research develops a typed semantic extension for A2A protocol, implementing it as a Message.metadata schema with formal performatives and interaction contracts. The extension will be declared in AgentCard.capabilities.extensions (Section 8) and validated at message send/receive boundaries. Evaluation focuses on multi-turn task coordination, delegation reliability, and negotiation determinism using LLM-based agents in controlled environments.

Protocol boundaries. The semantic layer operates within A2A's extension mechanism (Section 4.6), targeting Message and Task objects only. Client-to-agent communication (MCP protocol) and human-agent interaction are excluded. The work focuses on horizontal agent-to-agent communication using A2A's three standard protocol bindings (JSON-RPC, gRPC, HTTP+JSON).

Implementation constraints. Experiments use simulated multi-agent environments rather than distributed production deployments. LLM computational requirements limit experimental scale to 3-5 concurrent agents. Security implementation (Section 13), cross-organization authentication, and push notification mechanisms (Section 3.5.3) are acknowledged but not evaluated.

Evaluation scope. Semantic layer tested on representative task types: delegation, negotiation, and multi-step coordination. Full coverage of A2A's task lifecycle states (Section 4.1.3) and all possible agent skill combinations is beyond scope. Baseline comparison uses natural-language-only A2A communication without semantic constraints.

Rationale. Focusing on the extension mechanism ensures compatibility with A2A's design philosophy and immediate applicability to existing implementations. Simulated environments provide reproducibility and controlled variables essential for validating semantic guarantees. The subset approach enables completion within research timeframe while demonstrating generalizability to broader agent interaction patterns.

1.5 Methodology overview

1.5.1 LLM selection strategy

Selection criteria. LLMs evaluated based on: (1) instruction-following capability for structured output, (2) context window size for multi-turn interactions, (3) API availability and cost, (4) reasoning capability for semantic constraint interpretation.

Candidate models. Primary evaluation uses GPT-4o (OpenAI) for strong reasoning and Claude 3.5 Sonnet (Anthropic) for long-context handling. Comparative baseline includes GPT-3.5-turbo for cost-effectiveness analysis. Model selection (will be) validated through preliminary tests on semantic constraint parsing accuracy. *Probably small language models will be a better target due to inherited obvious reasoning limitations.*

1.5.2 Experimental design

Table 1: Experimental Design Matrix

Dimension	**Variable**	**Levels**	**Metrics**
Communication Mode	Semantic layer	With / Without (semantic vs. NLP)	Task success rate Negotiation consistency
Task Complexity	Complexity	Simple, Multi-step, Negotiation-heavy	Completion time Message count
Agent Configuration	Model	GPT-4o, Claude 3.5, Mixed combinations	Interpretation accuracy Error rate
Interaction Depth	Turns	2, 5, 10+	Semantic drift Context coherence

1.5.3 Evaluation workflow

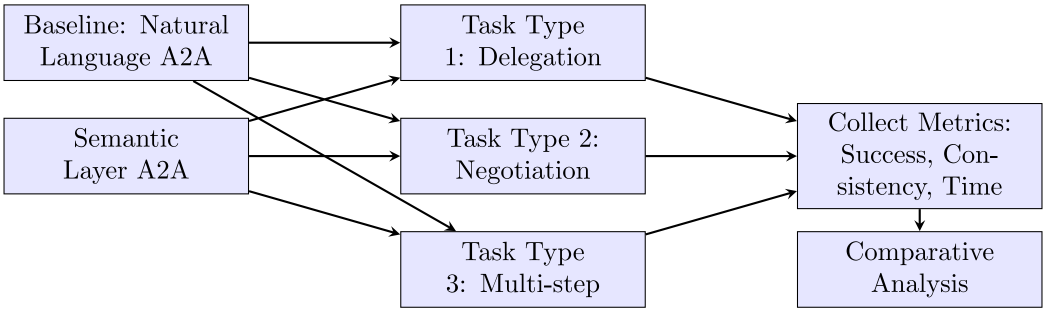

Figure 2: Experimental workflow comparing semantic vs. natural language communication modes

Controlled variables. All experiments use identical A2A infrastructure (JSON-RPC binding), consistent task prompts, and fixed agent role definitions. LLM temperature set to 0.1 for reproducibility. Each experimental condition repeated minimum 10 times to establish statistical significance. *Is 10 times enough? More models to include in a swarm?*

Success criteria. Semantic layer considered successful if: (1) task completion rate improves >15% vs. baseline, (2) negotiation consistency >90%, (3) semantic drift reduced in 10+ turn interactions, (4) no degradation in completion time >20%.

References

[1] Y. Yang, Q. Peng, J. Wang, Y. Wen, and W. Zhang, “Llm-based multi-agent systems: Techniques and business perspectives,” *arXiv preprint arXiv:2411.14033*, 2024.

[2] R. Ye et al., “X-mas: Towards building multi-agent systems with heterogeneous llms,” *arXiv preprint arXiv:2505.16997*, 2025.

[3] A. Sinha, *The rise of multi-agent systems: Collaboration at scale*, Medium blog post, [Online; accessed 26-Nov-2025], Nov. 2025. [Online]. Available: https://medium.com/@akankshasinha247/the-rise-of-multi-agent-systems-collaboration-at-scale-c71e8e60c56d.

[4] A. Sinha, *Building multi-agent architectures – orchestrating intelligent agent systems*, Medium blog post, [Online; accessed 26-Nov-2025], Jun. 2025. [Online]. Available: https://medium.com/@akankshasinha247/building-multi-agent-architectures-orchestrating-intelligent-agent-systems-46700e50250b.

[5] J. Chen, *The promise of multi-agent ai*, Forbes article, [Online; accessed 26-Nov-2025], May 2024. [Online]. Available: https://www.forbes.com/sites/joannechen/2024/05/24/the-promise-of-multi-agent-ai/.

[6] P. Zhou, Y. Feng, H. Julaiti, and Z. Yang, “Why do ai agents communicate in human language?” *arXiv preprint arXiv:2506.02739*, 2025.

[7] Y. Yang et al., *A survey of ai agent protocols*, arXiv preprint arXiv:2504.16736v2, [Online; accessed 26-Nov-2025], Apr. 2025. [Online]. Available: https://arxiv.org/abs/2504.16736v2.

[8] T. Finin, R. Fritzson, D. McKay, and R. McEntire, “Kqml as an agent communication language,” in *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM)*, [Online; accessed 26-Nov-2025], 1994, pp. 456–463. [Online]. Available: https://ebiquity.umbc.edu/_file_directory_/papers/318.pdf.

[9] S. Poslad and P. Charlton, “Standardizing agent interoperability: The fipa approach,” in Jun. 2001, vol. 2086, pp. 98–117, ISBN: 978-3-540-42312-6. DOI: 10.1007/3-540-47745-4_5.

[10] A. D. Ridder, *Agent communication languages and protocols comparison*, SmythOS article, [Online; accessed 26-Nov-2025], 2024. [Online]. Available: https://smythos.com/developers/ai-agent-development/agent-communication-languages-and-protocols-comparison/.

[11] OpenAI, *Openai-agents-python: A lightweight, powerful framework for multi-agent workflows*, GitHub repository, [Online; accessed 26-Nov-2025], 2025. [Online]. Available: https://github.com/openai/openai-agents-python.

[12] Anthropic, *Model context protocol (mcp)*, Online documentation and specification, [Online; accessed 26-Nov-2025], Nov. 2024. [Online]. Available: https://www.anthropic.com/news/model-context-protocol.

[13] The Linux Foundation, *Linux foundation launches the agent2agent protocol project to enable secure intelligent communication between ai agents*, Press release, [Online; accessed 26-Nov-2025], Jul. 2025. [Online]. Available: https://www.linuxfoundation.org/press/linux-foundation-launches-the-agent2agent-protocol-project-to-enable-secure-intelligent-communication-between-ai-agents.

[14] Google LLC / A2A Project, *Agent2agent (a2a) protocol – latest specification*, Official website, [Online; accessed 26-Nov-2025], 2025. [Online]. Available: https://a2a-protocol.org/latest/.

[15] M. Aparicio et al., “Fipa – intelligent agents from theory to practice,” *Telecommunications Journal / Journal of Communications*, 1999, FIPA founding-overview paper; online at FIPA archives. [Online]. Available: https://www.chiariglione.org/standards/fipa/telecom99.htm.

[16] *Multi-agent system market by agent system type (single-agent systems and multi-agent systems), by agent type, by application – global industry outlook, key companies (ibm, microsoft, google deepmind, and others), trends and forecast 2025-2034*, Dimension Market Research report, PDF/online, [Online; accessed 26-Nov-2025], May 2025. [Online]. Available: https://dimensionmarketresearch.com/report/multi-agent-system-market/.