Natural-Language Interfaces for Human–Agent Coordination in Distributed Optimisation

1 Introduction

Effective human–agent coordination in structured problem-solving domains — such as distributed scheduling, resource allocation, or multi-agent negotiation — requires not only accurate reasoning but also accessible communication. Humans are often asked to interact with AI systems grounded in formal logic, symbolic planning, or optimization constraints, yet most natural communication is expressed in flexible, high-level language. Large language models (LLMs) offer a bridge, translating between human-like expressions and machine-operable formats.

However, how this translation is mediated — whether through formal grammars, free-form interfaces, or hybrid models — significantly impacts human experience and system performance. We investigate this question in the context of a constrained cooperative optimization task: a clustered graph colouring game played by mixed teams of humans and agents.

We present a general protocol for agent interaction that permits flexible variation across different interface architectures. This enables rigorous comparison of alternative agent language designs and their consequences for optimization outcomes and user experience. We explore both objective metrics (e.g., convergence speed, solution quality) and subjective metrics (e.g., trust, workload, perceived fluency).

2 Related Work

2.1 Symbolic and Rule-Based Agent Communication

Early approaches to DCOP problems include message passing systems, for example in Max-Sum [8]. Argumentation-based coordination has also been employed, where a predefined grammar is provided to agents, allowing them to exchange plans and agree on joint actions and configurations including simple justification [9]. This naturally extends to human-agent teaming; these structured arguments are based on dialogue models originally developed to remove ambiguity, persuade, and reach consensus, so they can be adapted as interaction protocols for mixed human–agent settings [6]

2.2 LLMs for Argumentation and Structured Reasoning

Being text-based and dialogical, argumentation schemes may be enhanced by the use of LLMs. Similarly, the ability for LLMs to reason and to coordinate with humans about some non-textual domain is limited. Both problems may benefit from a hybrid system. The simplest of these approaches is to use an LLM to extract formal arguments from free-form text input [2, 5], then allowing those arguments to be run through a reliable solver. Similar approaches have been employed for building structures in formal logic [12]. Plan Domain Definition Languages (PDDLs) are often also used to the same end [10, 7, 13].

2.3 Human–Agent Teaming and Usability

Beyond performance metrics, human–agent teaming research emphasises the importance of communication, coordination, and strategic reasoning in mixed teams. Recent high-profile systems demonstrate that language models can be integrated with planning and optimisation procedures to negotiate and coordinate with humans in complex strategic games. For example, the CICERO agent combines large language models with strategic reasoning to negotiate and

cooperate at a human-level in the multiplayer strategy game Diplomacy, using dialogue to form alliances and plan joint actions under uncertainty [3]. Follow-on work such as DipLLM further explores fine-tuned LLM agents that learn equilibrium strategies for multi-party negotiation tasks by decomposing joint decision spaces into sequential action assignments [11].

More generally, frameworks that use LLMs as translators from natural language into formal planning or optimisation specifications have shown promise for grounding user intent in backend solvers. CaStL is one such approach, translating natural language constraints into PDDL and executable solver scripts that facilitate long-horizon task and motion planning with improved handling of complex constraint structures [4]. Similarly, the TIC framework uses LLMs to generate intermediate logical representations and downstream planner invocation, reducing error rates in formal task specification [1].

Existing work largely focuses either on autonomous negotiation agents that converse with humans in competitive games, or on language-to-planner pipelines where the human provides one-off specifications to a solver. Less is understood about peer-to-peer mixed teams in which humans and agents repeatedly coordinate local decisions within a shared optimisation task, as in distributed constraint problems. It is therefore unclear which interaction paradigm users prefer—strict rule-based argumentation, free-form LLM dialogue, or hybrids that translate language into formal acts—and how these choices affect convergence, trust, and workload over sustained collaboration. This study addresses that gap by directly comparing alternative language interfaces for human–agent coordination within a common DCOP backend.

3 Interaction Protocol and System Architecture

We define a general turn-based communication protocol in which agents and humans jointly solve an optimization problem over a distributed constraint network (DCOP). Each player controls a local subgraph (cluster), and must iteratively adjust local values to minimize cross-cluster constraint violations.

Messages are exchanged asynchronously and can carry suggestions, commitments, counter-proposals, or explanations. The protocol supports three core components:

- A message grammar or interpretation layer (rule-based or LLM-mediated)
- A reasoning backend
- A response generator

Reasoning Backend Because each cluster contains only five nodes with three possible colours, the joint search space is small enough to permit exhaustive evaluation. The system therefore uses an **optimal exhaustive solver** to evaluate proposals and determine the minimal-penalty configuration consistent with current commitments. This design choice intentionally removes algorithmic approximation error so that differences between conditions reflect the *communication interface* rather than solver quality.

RB (Simple Argumentation) Participants communicate using a lightweight, established argumentation-style grammar rather than raw utilities. The baseline supports a small set of speech acts:

- `Propose(node=value)`
- `Challenge(node=value)`
- `Justify(node, reason)`
- `Commit(node=value)`

The grammar is intentionally minimal and human-interpretable, reflecting classical dialogue-based coordination approaches, and does not expose internal solver details.

LLM-F (Freeform LLM) Messages are written in unrestricted language and interpreted/generated by an LLM with no symbolic planning or checking.

LLM-API (Hybrid Planner API) LLMs interpret messages and map them to internal API calls (e.g., "get penalty for C equals red", "apply forced C equals blue"). Decisions are made by the structured backend and translated back into language.

LLM-RB (LLM-Augmented Rule-Based) The user types natural language which is translated **in real time** into the RB grammar by an LLM, ultimately producing a formal argumentation statement accompanied by a human-readable justification, similar to systems such as Diplomacy that combine structured syntax with free text. The agent follows the same pattern, emitting both a structured act (e.g., `Propose(C=red)`) and additional explanatory language to support mutual understanding. Reasoning remains symbolic, but the interaction layer is conversational.

4 Experimental Design

We evaluate these modes in a controlled user study using a distributed graph coloring task as a proxy for coordination under constraint. Each participant (human or agent) controls a node cluster, with inter-cluster edges defining constraint penalties for incompatible assignments.

4.1 Conditions

Each participant engages in one of the following interaction modes: RB, LLM-F, LLM-API, or LLM-RB. In each case, the backend remains a consistent optimization problem (e.g., Max-Sum or greedy search), allowing us to isolate the effects of communication modality.

4.2 Participants

The primary condition of interest involves one human collaborating with two agents in a mixed team, reflecting the typical human–agent partnership scenario targeted by this work. To provide extreme baselines, two additional configurations are included:

- **Human-only (3 humans)** – an upper bound on interpretability and natural coordination
- **Agent-only (3 agents)** – an upper bound on optimisation speed without human factors

We recruit 30 participants and assign them to the 1-human/2-agent mixed team setup, with a smaller sample being used for the human-only baseline. Each team plays up to 10 rounds or until convergence, whichever occurs sooner, under different interface modes (within-subjects design). Depending on time required, each user will ideally perform each of the 4 scenarios with a questionnaire between each, counterbalanced by Latin square.

4.3 Measures

Objective

- **Convergence Rounds**: Number of turns to reach a stable, consistent colouring
- **Solution Score**: Total penalty from constraint violations
- **Message Metrics**: Length, frequency, diversity

Subjective

- **Trust**: Post-task trust questionnaire
- **Workload**: NASA-TLX
- **Fluency/Comfort**: Likert-scale fluency and expressiveness ratings

5 Expected Results and Discussion

We expect:

- **RB** will yield strong optimization outcomes but low usability, particularly for naïve users unfamiliar with argumentation formats.
- **LLM-F** will have high expressiveness but frequent coordination failures or inconsistencies due to LLM hallucinations or misalignment.
- **LLM-API** will produce balanced performance: likely it will still suffer from reliability issues.
- **LLM-RB** will offer a compelling trade-off: human-accessible interaction with formal robustness and explainability via structured reasoning.

We hypothesize that LLM-RB will be the most usable format for sustained collaboration.

6 Conclusion

This study proposes and evaluates a comparative framework for language-mediated human–agent coordination, where natural language is translated into structured action via LLMs and symbolic reasoning. We show that neither full-formalism nor unconstrained language suffices alone; instead, hybrids like LLM-RB offer interpretable, efficient, and human-aligned interfaces for cooperative optimization.

Future work includes extending the protocol to richer planning domains, investigating automatic formalization techniques for dynamic tasks, and exploring grounding in physical environments.

7 Remaining Work and Timeline

The project plan assumes that analysis scripts are prepared prior to the pilot and that data analysis and paper writing proceed in parallel with participant running.

7.1 Task List

- Ethics revision (1 day)
- Finalise API mode – mostly working but requires some finishing (1 day)
- Implement RB mode – mostly working but requires some finishing (2 days)
- Implement LLM_F mode – fairly straightforward; may CoT/few-shot (1–2 days)
- Implement LLM_RB properly – real-time translation to structured acts etc (3 days)
- Polishing the whole thing (2 days)
- Data cleaning and preprocessing scripts (2 days)
- Pilot testing with 2–3 colleagues – fix problems (2 days)
- Questionnaires (1 day)
- Data collection – 30 participants × 60–75 minutes (10 days calendar)
- Statistical analysis and visualisation (3 days, parallel with collection)
- Write Results and Discussion sections (3 days, parallel with collection)
- Final revisions and submission preparation (2 days)

7.2 Indicative Calendar (Working Days)

Dates	Activity
19 Jan	Ethics revision
20 Jan	Finalise API mode
21–22 Jan	Implement RB mode
23–26 Jan	Implement LLM_F mode
27–29 Jan	Implement LLM_RB properly
30 Jan–2 Feb	Polishing
3–4 Feb	Data cleaning and preprocessing scripts
5–6 Feb	Pilot testing
7–9 Feb	Questionnaires
10–23 Feb	Data collection (calendar block)
12–16 Feb	Statistical analysis (parallel)
17–19 Feb	Write Results and Discussion (parallel)
24–25 Feb	Final revisions

References

[1] Sudhir Agarwal and Anu Sreepathy. Tic: Translate-infer-compile for accurate "text to plan" using llms and logical representations. In *International Conference on Neural-Symbolic Learning and Reasoning*, pages 222–244. Springer, 2024.

[2] Lucas Anastasiou and Anna De Liddo. A hybrid human-ai approach for argument map creation from transcripts. In *Proceedings of the First Workshop on Language-driven Deliberation Technology (DELITE)@ LREC-COLING 2024*, pages 45–51, 2024.

[3] Meta Fundamental AI Research Diplomacy Team (FAIR)†, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022.

[4] Weihang Guo, Zachary Kingston, and Lydia E Kavraki. Castl: Constraints as specifications through llm translation for long-horizon task and motion planning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11957–11964. IEEE, 2025.

[5] Hao Li, Viktor Schlegel, Yizheng Sun, Riza Batista-Navarro, and Goran Nenadic. Large language models in argument mining: A survey. *arXiv preprint arXiv:2506.16383*, 2025.

[6] Sanjay Modgil and Henry Prakken. The aspic+ framework for structured argumentation: a tutorial. *Argument & Computation*, 5(1):31–62, 2014.

[7] James Oswald, Kavitha Srinivas, Harsha Kokel, Junkyu Lee, Michael Katz, and Shirin Sohrabi. Large language models as planning domain generators. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pages 423–431, 2024.

[8] Alex Rogers, Alessandro Farinelli, Ruben Stranders, and Nicholas R Jennings. Bounded approximate decentralised coordination via the max-sum algorithm. *Artificial Intelligence*, 175(2):730–759, 2011.

[9] Yuqing Tang and Simon Parsons. Argumentation-based dialogues for deliberation. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 552–559, 2005.

[10] Yaqi Xie, Chen Yu, Tongyao Zhu, Jinbin Bai, Ze Gong, and Harold Soh. Translating natural language to planning goals with large-language models. *arXiv preprint arXiv:2302.05128*, 2023.

[11] Kaixuan Xu, Jiajun Chai, Sicheng Li, Yuqian Fu, Yuanheng Zhu, and Dongbin Zhao. Dipllm: Fine-tuning llm for strategic decision-making in diplomacy. *arXiv preprint arXiv:2506.09655*, 2025.

[12] Yu'an Yang, Siheng Xiong, Ali Payani, Ehsan Shareghi, and Faramarz Fekri. Harnessing the power of large language models for natural language to first-order logic translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6942–6959, 2024.

[13] Max Zuo, Francisco Piedrahita Velez, Xiaochen Li, Michael Littman, and Stephen Bach. Planetarium: A rigorous benchmark for translating text to structured planning languages. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11223–11240, 2025.