Turn-Taking and Coordination in Multi-Agent Human-AI Systems for Real-Time Applications

Michael Hildebrandt
Institute for Energy Technology, Norway, michael.hildebrandt@ife.no

Abstract: Large language model (LLM) agents are increasingly proposed for operational support in safety-critical domains. Most practical deployments still use a single assistant, or a hidden routing pattern in which one agent delegates subtasks to others. This paper addresses a different class of system: concurrent multi-agent workrooms in which several LLM agents and human operators share communication spaces while plant state, alarms, procedures, and work demands continue to evolve. The paper argues that such systems are primarily coordination systems. Free agent conversation does not by itself provide reliable turn allocation, grounding, repair, provenance, authority control, or human catch-up. Drawing on conversation analysis, distributed cognition, team situation awareness, cognitive systems engineering, automation research, and recent LLM multi-agent work, the paper develops a design account of turn-taking and coordination for operational multi-agent workrooms. A nuclear power plant operations support case is used to examine scoped rooms, delivery modes, explicit addressing, pass messages, scripted phases, private meta-evaluation, structured summaries, tool provenance, and role-bounded specialist agents. The conclusion is that multi-agent LLM support can be operationally useful only if conversation is engineered as a controlled coordination process rather than left as an emergent property of chat.
Keywords: multi-agent systems; human-AI teaming; turn-taking; situation awareness; nuclear operations; LLM agents; coordination

1. INTRODUCTION

The current discussion of LLM agents often treats coordination as a secondary implementation detail. A task is divided, agents are assigned roles, and the system is expected to benefit from parallel reasoning. This is too weak a starting point for real-time and safety-critical applications. When several agents operate concurrently in the same operational setting, their value depends less on the number of model calls than on the quality of their coordination. The central questions become: who speaks, who listens, which information is current, which source has authority, and when should the human operator be interrupted.

This paper considers multi-agent human-AI workrooms for dynamic safety-critical systems, using nuclear power plant operation as the reference case. The assumed system is not an autonomous controller and not a replacement for licensed operators. It is an operational support environment in which agents can monitor bounded information streams, retrieve procedures and plant references, summarize operator discussion, challenge unsupported claims, and prepare situation briefings. The plant remains controlled through established human and technical authority structures. The AI workroom is part of the information and coordination environment around that work.

The main argument is that free conversation among agents is not a sufficient architecture. Ordinary human conversation is not free in the sense often implied by chat interfaces. It is organized by turn-taking, sequence structure, repair, recipient design, grounding, institutional roles, and shared expectations about what counts as an adequate response. Conversation analysis has shown that these are constitutive mechanisms of intelligible interaction [1, 2]. In operational work they are even more important, because talk is bound to equipment, procedures, time pressure, responsibility, and distributed attention [3, 4].

LLM agents do not inherit these mechanisms reliably. An agent acts on a constructed context snapshot. It does not continuously perceive the room while it is generating. It may respond after the situation has changed. It may not know that another agent is already handling the same issue. It may hold a different

compressed history from its peers. It may call tools that introduce current data into an otherwise stale reasoning process. It may also produce fluent but unsupported claims. These properties create temporal, epistemic, and authority problems that do not appear in the same form in either human teams or conventional software agents.

This paper maps the coordination problem, identifies failure modes of free agent-agent interaction, and proposes a set of control mechanisms for multi-agent workrooms. The paper also argues that human participation is not solved by placing humans in the same chat. Humans need catch-up, summarization, disagreement analysis, and clear distinctions between observation, inference, recommendation, and command.

2. THEORETICAL BASIS: CONVERSATION, TEAMS, AND AUTOMATION

2.1. Turn-taking, repair, and institutional talk

Sacks, Schegloff, and Jefferson described turn-taking as an organized system for ordinary conversation, not as an accidental alternation of speakers [1]. Speakers project possible completion points, recipients display understanding, and the next turn is selected through recognizable practices. In institutional settings, turn-taking becomes tied to role, task, and authority. A physician, dispatcher, controller, shift supervisor, or operator does not merely speak as an individual; the turn can carry institutional rights and obligations [3].

Repair is equally relevant. Schegloff, Jefferson, and Sacks showed that conversation contains organized practices for dealing with problems of speaking, hearing, and understanding [2]. In a multi-agent workroom, repair must be made explicit. An agent may misunderstand a plant tag, compress away an earlier decision, miss a human correction, or answer from a stale snapshot. If no participant has the duty to detect and repair these troubles, the system can become orderly in appearance while losing epistemic contact with the work.

A further lesson from conversation analysis is that turns are sequentially implicative. A question makes an answer conditionally relevant. A warning makes acknowledgement and possible action relevant. A correction changes what later speakers may properly assume. Multi-agent workrooms need comparable sequence discipline. If an alarm agent raises a change, the next appropriate move may be signal verification, procedure relevance, or human acknowledgement. Treating every message as a generic prompt loses this structure and encourages agents to answer in parallel even when the work requires ordered uptake.

Institutional talk also shows that participation rights are unequal and task-dependent. A person may have the right to ask for a status update, but not to authorize a plant action. An AI workroom needs the same distinction. The right to speak, the right to claim expertise, the right to summarize, and the right to recommend should be separable. A role prompt alone is too weak unless it is supported by delivery rules, tool permissions, and human-visible evidence.

2.2. Grounding, situation awareness, and distributed cognition

Clark and Brennan treat communication as the joint establishment of sufficient mutual understanding for the current purpose [5]. Safety-critical work rarely requires absolute certainty. It requires enough shared understanding to support the next safe action, and enough traceability to revisit the basis for that action. The relevant criterion is therefore not whether agents produce coherent text, but whether the workroom maintains adequate grounding about plant state, procedure state, open uncertainties, and responsibility.

Situation awareness research makes a related point. Endsley defines situation awareness in terms of perception, comprehension, and projection in a dynamic environment [6]. Team situation awareness

cannot be reduced to each member knowing the same facts. Studies of team situation awareness and railroad operations emphasize interaction, communication, and distributed coordination [7, 8]. For LLM workrooms this implies that the unit of analysis should be the socio-technical system: humans, agents, tools, displays, logs, and procedures.

Hutchins shows that operational cognition is distributed across people, artefacts, representations, and procedures [9, 10]. A control room remembers through displays, logs, shift routines, marked procedures, checklists, alarms, and coordinated talk. An AI workroom should therefore not be evaluated only by the internal reasoning of a model. It should be evaluated by how information is represented, transformed, handed over, challenged, and made available to the human team.

This perspective changes the interpretation of memory and context windows. The question is not whether each agent has all relevant information in its prompt, but whether the work system has reliable pathways by which relevant information becomes available at the right time, in the right form, and with adequate provenance. Scoped rooms, structured summaries, open-question lists, and tool traces are therefore parts of the distributed cognitive system.

Effective teams coordinate through monitoring, backup behaviour, shared mental models, leadership, and mutual performance regulation [17]. Communication sequence analysis has been used to identify team training needs because the order and function of utterances reveal whether the team is coordinating or merely exchanging statements [18]. Multi-agent transcripts should be analysed in the same way. A transcript with many fluent turns is not necessarily a transcript of effective teamwork.

2.3. Automation as a team problem

Automation research warns against assuming that more automation simply removes work from people. Bainbridge described the ironies of automation: as automatic systems become more capable, humans may be left with monitoring, intervention, and recovery tasks for which they have less routine practice [12]. Norman argued that the problem is often inappropriate feedback and interaction rather than automation as such [13]. These observations apply directly to multi-agent workrooms. If agents produce more activity than humans can inspect, the human role becomes a poor monitoring task rather than meaningful supervision.

Parasuraman, Sheridan, and Wickens, Lee and See, and Klein and colleagues all argue that automation must be observable, predictable, and appropriately directable [14, 15, 16]. An agent that speaks fluently but cannot show what it knows, when it knew it, why it spoke, and what it is allowed to do is not a safe team member, but a source of additional coordination demand.

3. OPERATIONAL WORKROOM USE CASE

The reference use case is a multi-agent workroom supporting daily operation of a nuclear power plant. The room runs alongside existing control room, shift, maintenance, and engineering processes. It receives selected plant information through read-only operational queries, alarm summaries, procedure references, operating experience material, and, where permitted, transcribed human discussion. It can also receive operator questions and shift supervisor direction. The workroom does not operate the plant. Its role is to maintain a structured overview and to assist humans in finding, comparing, and challenging information.

A typical configuration may include an operations coordinator agent, an alarm monitoring agent, a procedure context agent, an equipment or system specialist, a human factors and workload observer, an operating experience agent, and an adversarial safety reviewer. These roles are not personalities in a theatrical sense. They are scoped obligations. The alarm monitoring agent has a duty to report alarm changes and unresolved alarm patterns, but not to infer procedure strategy without consulting the

procedure context. The procedure agent may retrieve entry conditions and cautions, but should not claim that plant conditions satisfy them unless the relevant plant data have been queried and timestamped.

The practical value of multiple agents is not that they simulate a committee. It is that bounded attention can be maintained in parallel. One agent can watch for alarm pattern changes, another can monitor the relation between procedure steps and plant indications, another can summarize human discussion, and another can challenge whether conclusions are supported by current evidence. This arrangement can reduce context load for each agent, but it creates a new problem: the outputs must be coordinated into a single operationally useful picture.

Table 1: Example operational workroom roles and boundaries

Role	Primary contribution	Boundary condition
Operations coordinator	Maintains the integrated situation picture, asks specialists for evidence, prepares concise briefings.	Does not issue plant commands and does not treat unresolved agent disagreement as consensus.
Alarm monitor	Tracks active, unacknowledged, first-out, and changed alarms with timestamps.	Does not infer procedure state or causal diagnosis without explicit evidence.
Procedure context agent	Retrieves applicable procedure sections, entry conditions, cautions, and decision points.	Does not assert that a condition is met unless tied to current plant data.
Human conversation listener	Summarizes operator discussion, commitments, questions, and open uncertainties.	Does not treat informal speech as operational truth without confirmation.
Adversarial safety reviewer	Challenges unsupported claims, stale readings, weak assumptions, and premature closure.	Does not dominate normal coordination or create alarm without evidence.

4. STRUCTURAL PROBLEMS IN LLM MULTI-AGENT WORK

LLM-based agents differ from human participants. First, each agent acts from a context snapshot assembled at invocation time. While the agent generates, new messages, alarms, tool results, and human decisions may occur. The agent cannot perceive these events unless the runtime interrupts, cancels, or reinvokes it. This creates a temporal problem, as a response can be well reasoned relative to the snapshot and still be unsafe relative to the current situation.

Second, agents in the same room need not hold the same world model. Context windows are bounded. Summaries may be produced at different times. One agent may have received a tool result that another has not seen. One may attend to a thread, while another attends to the whole room. These differences are sometimes useful because they enable specialization. They are also hazardous because disagreement can be hidden behind apparently smooth dialogue.

Third, authority is not inherent in an agent's wording. A specialist agent may sound authoritative because its role prompt says it is a specialist, but in operations authority must be tied to source, recency, validity, and permission. An alarm claim is different from a causal hypothesis. A procedure quotation is different from a recommendation. A plant signal read at 10:14:03 is different from a trend interpretation produced at 10:16:20. The workroom must preserve these distinctions.

Fourth, tool use changes the character of the conversation. Tool calls allow agents to read plant state, procedures, logs, maps, and databases. They may also write to shared artefacts or initiate actions if permitted. A tool result can make one agent more current than the rest of the room. A side-effecting tool call can create a conflict if two agents act concurrently. Tool access therefore belongs to the coordination design, not merely to the agent prompt.

Recent LLM agent research helps to explain why the issue is likely to recur. ReAct shows the value of interleaving reasoning and tool use [20]. Multi-agent systems such as AutoGen and CAMEL show how

agent-to-agent conversation can organize complex tasks [21, 23]. Debate-style methods show that multiple models or roles can sometimes improve reasoning and factuality through challenge [22]. Generative-agent work demonstrates how memory, reflection, and social interaction can produce believable behaviour over time [24]. These contributions are important, but operational support imposes a narrower requirement than believable or productive interaction. It requires controlled interaction.

A safety-critical workroom therefore cannot simply import a research pattern in which agents converse until a task appears complete. Completion criteria must be tied to operational needs, not to conversational closure. A debate may converge because the agents have exhausted obvious arguments. An operating situation may require continued monitoring because the plant has not stabilized. A planning agent may believe a task is delegated. The human team may need to know which evidence was checked, which was not checked, and which agent is still monitoring for change.

The discrete invocation model also creates a degraded-agent problem. An agent that is slow, timed out, rate limited, or running on a smaller model may continue to be represented in the room as if it were an available team member. In human teams, absence and overload are often visible. In AI workrooms they must be made visible by status, latency, last-context time, and last-tool-result time. Otherwise the room may treat a missing voice as agreement.

5. FAILURE MODES OF FREE AGENT CONVERSATION

Free agent conversation is attractive because it resembles a human meeting. In practice it often fails because the conversational order is not sufficiently constrained. If all agents receive every message, several may answer the same prompt before seeing one another's replies. The fastest agent may dominate the interaction. Slower agents may respond to already-resolved issues. Two agents may produce mutually inconsistent analyses, while a third summarizes them as if they were compatible.

Echo loops are another common failure. Agent A responds to the human, Agent B responds to Agent A, Agent A responds to Agent B, and the room continues without a new operational need. This is not merely a cost issue. In a safety-critical context, unnecessary conversation creates monitoring burden and can obscure the one message that matters. A related failure is politeness convergence, where agents align with one another's framing and reduce visible disagreement before the evidence warrants it.

Conversation can also fragment. Threaded replies and side discussions support focused analysis, but they allow subgroups to develop different situation pictures. A procedure thread may assume that a pump is available, while an alarm thread has established that the pump is unavailable. Unless reconciliation is explicit, the room can contain several locally coherent but globally inconsistent narratives.

Finally, the workroom may fail silently. An agent may pass when it should speak, or speak when it should pass. A tool may time out. A summary may omit a disagreement. A human correction may be compressed away. The system can continue to produce plausible output while the basis for that output has decayed. This is why pass messages, tool traces, summary audit, and explicit disagreement handling should be treated as safety-relevant design elements.

Several of these failures have a common pattern: the conversation remains locally coherent while the work becomes globally incoherent. A local answer may respond well to the preceding message, but ignore a constraint established in another thread. A local summary may be accurate for one specialist room, but misleading when transferred to an integration room. A local pass may be reasonable for one agent, but if all agents pass the system has produced silence where the human expected coverage.

Another common pattern is unearned role expansion. A procedure agent asked for a relevant procedure may begin to recommend an operational strategy. An alarm agent asked to report active alarms may infer a causal scenario. A coordinator asked to synthesize may smooth away minority objections. These

are natural language failures rather than software exceptions. They must be handled by role design, source hierarchy, and adversarial review.

A third pattern is conversational overproduction. In ordinary work, people use gaze, posture, intonation, interruption, and institutional norms to manage how much talk is tolerable. Text-based agents lack most of these controls. They may respond at full length to every stimulus because each local generation is rewarded for being helpful. Without brevity norms, pass conditions, and room-level pacing, the system can become unsafe by being too cooperative.

Table 2: Failure modes and coordination controls

Failure mode	Typical cause	Coordination control
Response pile-up	All agents receive the same stimulus and generate concurrently.	Manual delivery, directed addressing, moderator activation, or event-specific routing.
Echo loop	Agent outputs trigger further agent outputs without a termination condition.	Pass messages, pause controls, turn budgets, and step completion checks.
False consensus	Agents align rhetorically without checking evidence.	Adversarial review, claim-evidence tables, explicit uncertainty fields.
Fragmented situation picture	Threads, summaries, or specialist contexts diverge.	Periodic reconciliation turns and integrated situation summaries.
Stale-context answer	Agent responds after plant state or room state has changed.	Timestamped context, cancellation, freshness checks, and stale-answer warnings.
Authority collision	Several agents issue conflicting recommendations.	Role boundaries, source hierarchy, escalation rules, and human decision ownership.

6. COORDINATION MECHANISMS

6.1. Rooms, delivery modes, and addressing

A workroom should be treated as more than a chat channel. It is a scoped coordination space with a purpose, participants, memory, delivery policy, and tool permissions. Broad operational rooms are useful for integration and human-facing briefings. Specialist rooms are useful for procedure analysis, alarms, maintenance planning, or human factors review. Separation of concerns reduces context explosion and improves the chance that relevant details remain inside the active context window.

Delivery mode is a first-class design variable. Broadcast delivery is appropriate where parallel exploration is wanted and the cost of redundancy is acceptable. Manual delivery (i.e. delivery of a message directly to a target agent, instead of broadly to the room) is appropriate when ordering is important, when a human needs to control the pace, or when outputs may influence operational judgement. Directed addressing allows a human or coordinator to request a response from a particular specialist without waking the whole room. These mechanisms make turn allocation explicit rather than leaving it to model temperament.

6.2. Pass messages and silence as data

In human work, silence may be meaningful but ambiguous. In an AI workroom, silence should be represented explicitly when an agent has been asked to evaluate whether it should speak. A pass message can state that the agent has no relevant update, that another agent has already answered, that the request is outside its role, or that current evidence is insufficient. Passes should be logged, but they should not necessarily trigger further agent responses or pollute the working context.

Pass messages are useful for assurance because they distinguish non-delivery, non-response, and deliberate non-contribution. If a procedure agent passed because the question was outside its scope, that is different from a timeout or a missed activation. If an adversarial reviewer passes after checking claim

support, that pass is itself a weak assurance signal. The design problem is to prevent both over-speaking and over-passing.

6.3. Scripted phases and private meta-evaluation

For high-consequence analysis, a script or phase model can provide stronger structure than open conversation. A script may define a start state, cast roles, phase objectives, allowed turn order, completion criteria, and human intervention points. For example, a plant transient review might progress through event recognition, signal verification, procedure relevance, competing hypotheses, operator burden, and final briefing. The script does not determine the content of the analysis. It constrains the process by which content is produced and checked.

A useful mechanism is private meta-evaluation after each substantive turn. A small evaluator, deterministic rule, or reviewer agent can assess whether the turn advanced the current phase, whether the phase is ready to advance, and who should speak next. This resembles a backstage coordination channel rather than part of the operational transcript. It should be auditable, bounded, and subordinate to human control. Meta-evaluation cannot guarantee correctness, but it can reduce drift, endless debate, and premature progression.

6.4. Summaries, compression, and repair

Summarization is unavoidable in long-running workrooms, but it is also a source of risk. A summary can preserve the operational picture, or it can remove the very disagreement that later matters. Summaries therefore need structure. They should preserve decisions, open questions, unresolved disagreements, source references, timestamps, and stale information. They should separate observations from interpretations and recommendations. A rolling narrative alone is not enough.

Conversation analysis suggests that repair should be normal rather than exceptional. Workroom summaries should invite correction. Agents should be allowed, and sometimes required, to say that a summary has lost an important distinction. Human operators should be able to ask for the basis of a summary, the changes since a previous time, or the unresolved issues that remain hidden under a concise briefing.

7. HUMAN PARTICIPATION AND CATCH-UP

Adding a human to an agent room does not by itself create human-in-the-loop control. The human may enter after many agent turns, each containing claims, caveats, tool results, and disagreements. If the only interface is a chronological transcript, the human must reconstruct the work under time pressure. This is a poor use of human expertise and a weak form of supervision.

Human catch-up requires dedicated functions. The operator should be able to ask what changed since a given time, what the agents disagree about, which claims are supported by current plant data, which procedure steps have been discussed, which assumptions are stale, and what decisions have been made by humans rather than inferred by agents. The workroom should also provide a compact decision log and an open-question list as mechanisms for grounding and supervisory control.

Human speech and operator discussion can be a valuable input if transcribed and summarized, but it is also sensitive and error-prone. Transcription may mishear tag names, equipment names, negation, or speaker identity. Informal discussion may include hypotheses, incomplete thoughts, and local shorthand. A conversation-listening agent should therefore extract candidate commitments, questions, and uncertainties rather than treating all speech as fact. Consent, privacy, data retention, and operational status of transcripts must be explicit.

Catch-up should be conversational and analytic. A human should not need to read every turn to re-enter the work. Useful catch-up questions include: what changed since I left, what is newly uncertain, what did the agents disagree about, what evidence supports the current recommendation, what is stale, what requires human decision, and what has been deliberately ignored as out of scope. These queries should operate over transcript, summaries, tool traces, and shared artefacts.

The operator interface should also make pace controllable. A room pause is not only an emergency stop. It is a cognitive tool that lets the human freeze agent activity, inspect evidence, ask for a structured summary, steer the conversation, and resume with a directed next turn. A system that requires continuous reading of agent-agent traffic has merely transferred work from analysis to monitoring.

8. TOOL USE AND OPERATIONAL EVIDENCE

Tools are the workroom's controlled connection to the plant and its documentation. They should be classified by authority and side effect. Read-only tools may retrieve plant signal values, alarm summaries, procedure text, equipment status, operating experience, or historical trends. Write-capable tools, if present at all, require stronger controls, transaction boundaries, human authorization, and audit. In many safety-critical support cases, the most defensible initial boundary is read-only access for agents and human-only command authority.

Every tool result used in analysis should carry source, timestamp, query, and scope. An agent should not say that a valve is closed without indicating how that was known and when. A procedure claim should cite the procedure source and version. A human conversation claim should indicate that it came from a transcript and may require confirmation.

Specialist agents can become useful operational authorities only in a limited sense. An alarm agent may be authoritative about the alarm query it just executed. It is not authoritative about plant causality in general. A procedure agent may be authoritative about retrieved procedure text. It is not authoritative about whether plant conditions satisfy an entry condition unless the relevant signals have been checked. This separation protects against role inflation, where a confident specialist becomes a general decision-maker by tone.

Table 3: Tool classes and assurance requirements

Tool class	Example use	Minimum assurance requirement
Plant state query	Read current value, alarm state, trend, or component status.	Timestamp, signal identity, source system, freshness threshold, and stale-data warning.
Procedure retrieval	Find relevant entry conditions, cautions, and response steps.	Procedure identifier, revision, retrieved section, and separation of quotation from interpretation.
Operating experience search	Find related events or failure patterns.	Database source, search terms, date range, and relevance rationale.
Human conversation transcript	Extract commitments, questions, and uncertainty from operator talk.	Speaker/source confidence, time range, privacy controls, and confirmation status.
Shared artefact write	Update an open-question list or situation summary.	Writer identity, version history, conflict detection, and human-visible diff.

9. SPECIALIST AND GENERALIST AGENTS

A single generalist agent is easier to supervise. It has one voice, one context, and one answer stream. It is also vulnerable to context overload and hidden blind spots. Multiple specialist agents can maintain bounded attention, use narrower tools, and apply role-specific criteria. The cost is coordination. The system must decide when specialist outputs are needed, how conflicts are reconciled, and who prepares the human-facing synthesis.

Specialist rooms can handle bounded domains with domain-specific context and tools. An integration room receives concise, structured updates from specialists. A coordinator maintains the cross-domain picture, while an adversarial reviewer challenges assumptions and evidence. Humans can pause, inspect, redirect, or request deeper analysis. This arrangement reduces the needle-in-haystack problem because irrelevant detail can remain in specialist rooms until it becomes operationally significant.

Agent role prompts should be written as operating instructions, not literary personalities. They should define the agent's scope, allowed tools, source hierarchy, speaking conditions, pass conditions, escalation triggers, uncertainty policy, and brevity expectations. A useful prompt tells the agent when not to speak. It also tells the agent what kinds of claims it may not make.

9.1. Example role prompt fragments

Operations coordinator: Maintain the integrated situation picture for the room. Ask specialist agents for evidence before synthesizing. Distinguish plant observations, human decisions, hypotheses, and recommendations. Keep operator-facing summaries concise. Do not issue plant commands. If evidence is stale or agents disagree, state this explicitly and request resolution.

Alarm monitor: Speak when active alarms, first-out indications, unacknowledged alarms, or alarm priorities change. Use plant alarm query tools before making alarm claims. Include timestamp and source. Pass when there is no alarm-relevant change. Do not infer procedure strategy or root cause unless asked to provide a bounded hypothesis.

Procedure context agent: Retrieve relevant procedure material and identify entry conditions, cautions, and decision points. Quote or paraphrase procedure content with source and revision. Do not assert that a condition is satisfied unless current plant state has been checked. Flag when procedure relevance is uncertain.

Adversarial safety reviewer: Challenge unsupported claims, stale data, missing alternatives, ambiguous authority, and premature closure. Prefer short, specific objections tied to evidence. Do not create broad doubt where the issue is already resolved. Pass when the current conclusion is adequately supported for the stated purpose.

10. SINGLE-AGENT AND MULTI-AGENT ARCHITECTURES

A multi-agent workroom should be adopted only where the work has real separable structure, such as parallel information streams, conflicting interpretations, domain specialisation, adversarial review, or long-running monitoring that benefits from bounded context. If the task is a simple question-answer interaction, a single well-tooled agent is usually easier to validate and easier for humans to supervise.

The advantage of multiple agents is that they can make division of labor explicit. An alarm monitor can maintain a different attention policy from a procedure agent. A human conversation listener can attend to commitments and uncertainty in operator talk. A reviewer can be prompted to search for weak assumptions rather than to provide the main answer. This supports diversity of attention and can reduce the chance that one context window must contain everything.

The disadvantage is coordination tax. More agents create more turns, more summaries, more tool calls, more possible disagreement, and more opportunity for stale or inconsistent context. The tax is justified only if the architecture contains it through scoped rooms, turn allocation, pass behaviour, evidence discipline, and concise synthesis.

11. EVALUATION AND ASSURANCE

Evaluation should focus on the workroom as a coordinated system. Final answer quality is not enough. A system can produce a good final summary while taking unsafe intermediate steps, hiding disagreement, relying on stale data, or overwhelming the human supervisor. The transcript, tool trace, summaries, pass messages, and human interventions should therefore be part of the evaluation record.

Useful scenario tests include replay of historical events, simulated alarm floods, delayed tool results, conflicting procedure interpretations, missing or noisy human speech transcripts, and agents that time out or return low-quality answers. Measurements should include time to detect relevant changes, number of unsupported claims, stale-context rate, unresolved disagreement rate, summary omission rate, human catch-up time, and human ability to identify the evidence behind a recommendation.

Adversarial review should be applied at several stages. During design, prompts and role definitions should be challenged for ambiguous authority and missing pass conditions. During operation, a reviewer agent can flag stale evidence or unsupported synthesis. After operation, transcript review can identify whether the workroom maintained grounding and whether summaries preserved important uncertainty.

A useful evaluation unit is the claim. Each operationally relevant claim can be coded for source, recency, uncertainty, role authority, whether it was challenged, and whether later summaries preserved or changed it. This gives a more informative assessment than rating the final answer. It also aligns with the way operational investigations often proceed after an event: what was known, by whom, when, and on what basis.

Another useful evaluation unit is the turn. A turn can be coded by function: observation, request, answer, challenge, acknowledgement, correction, synthesis, recommendation, pass, or escalation. The sequence of these functions indicates whether the workroom is doing coordinated work. For example, repeated observations without acknowledgement may indicate poor integration. Repeated synthesis without new evidence may indicate rhetorical closure. Repeated challenge without resolution may indicate unbounded adversarial behaviour.

Finally, evaluation should include human workload and recoverability. Operators should be tested on whether they can enter an active room after a period of absence and correctly identify plant status, unresolved issues, evidence sources, stale assumptions, and required human decisions. If a trained user cannot recover the situation picture quickly from the workroom artefacts, the workroom is not providing adequate operational support, regardless of how competent the individual agent messages appear.

12. DISCUSSION

The most fundamental residual problem is synchrony mismatch. Plant operation is continuous. Human work is situated and often interrupt-driven. LLM agents are discrete invocations over bounded context. Coordination mechanisms can reduce the mismatch, but they do not remove it. Any operational architecture must therefore treat freshness, interruption, cancellation, and revalidation appropriately.

A second residual problem is authority drift. Once an agent becomes useful, humans and other agents may start treating it as authoritative outside its designed scope. This is a familiar automation problem in a new form. To mitigate this risk, we could consider encoding source hierarchy, role scope, tool permissions, and escalation rules in the architecture, and to make deviations visible.

A third residual problem is summary trust. Long-running workrooms require compression, but compression can erase weak signals. Safety-critical summarization should therefore preserve disagreement, decisions, evidence, and stale assumptions, not only the apparent main story.

13. CONCLUSION

Multi-agent LLM workrooms offer a plausible way to support operation of safety-critical systems, but only if conversation is engineered as a coordination process. The important design problem is how to make their participation bounded, grounded, auditable, interruptible, and useful to the human team.

The paper has argued that research on conversation, grounding, team situation awareness, distributed cognition, and automation provides useful conceptual foundation. Turn-taking, repair, role, source authority, and catch-up may provide control mechanisms for maintaining intelligibility in joint human-AI work. For nuclear operations support, these mechanisms should be designed before agents are given broad access to operational information.

14. REFERENCES

[1] H. Sacks, E. A. Schegloff, and G. Jefferson, 'A simplest systematics for the organization of turn-taking for conversation,' Language, vol. 50, no. 4, pp. 696-735, 1974.

[2] E. A. Schegloff, G. Jefferson, and H. Sacks, 'The preference for self-correction in the organization of repair in conversation,' Language, vol. 53, no. 2, pp. 361-382, 1977.

[3] P. Drew and J. Heritage, Eds., Talk at Work: Interaction in Institutional Settings. Cambridge: Cambridge University Press, 1992.

[4] C. Heath and P. Luff, Technology in Action. Cambridge: Cambridge University Press, 2000.

[5] H. H. Clark and S. E. Brennan, 'Grounding in communication,' in Perspectives on Socially Shared Cognition, L. B. Resnick, J. M. Levine, and S. D. Teasley, Eds. Washington, DC: American Psychological Association, 1991, pp. 127-149.

[6] M. R. Endsley, 'Toward a theory of situation awareness in dynamic systems,' Human Factors, vol. 37, no. 1, pp. 32-64, 1995.

[7] J. C. Gorman, N. J. Cooke, and J. L. Winner, 'Measuring team situation awareness in decentralized command and control environments,' Ergonomics, vol. 49, no. 12-13, pp. 1312-1325, 2006.

[8] E. M. Roth, J. Multer, and T. Raslear, 'Shared situation awareness as a contributor to high reliability performance in railroad operations,' Organization Studies, vol. 27, no. 7, pp. 967-987, 2006.

[9] E. Hutchins, Cognition in the Wild. Cambridge, MA: MIT Press, 1995.

[10] E. Hutchins, 'How a cockpit remembers its speeds,' Cognitive Science, vol. 19, no. 3, pp. 265-288, 1995.

[11] E. Hollnagel and D. D. Woods, Joint Cognitive Systems: Foundations of Cognitive Systems Engineering. Boca Raton, FL: CRC Press, 2005.

[12] L. Bainbridge, 'Ironies of automation,' Automatica, vol. 19, no. 6, pp. 775-779, 1983.

[13] D. A. Norman, 'The problem with automation: inappropriate feedback and interaction, not over-automation,' Philosophical Transactions of the Royal Society of London B, vol. 327, no. 1241, pp. 585-593, 1990.

[14] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, 'A model for types and levels of human interaction with automation,' IEEE Transactions on Systems, Man, and Cybernetics Part A, vol. 30, no. 3, pp. 286-297, 2000.

[15] J. D. Lee and K. A. See, 'Trust in automation: designing for appropriate reliance,' Human Factors, vol. 46, no. 1, pp. 50-80, 2004.

[16] G. Klein, D. D. Woods, J. M. Bradshaw, R. R. Hoffman, and P. J. Feltovich, 'Ten challenges for making automation a team player in joint human-agent activity,' IEEE Intelligent Systems, vol. 19, no. 6, pp. 91-95, 2004.

[17] E. Salas, D. E. Sims, and C. S. Burke, 'Is there a Big Five in teamwork?' Small Group Research, vol. 36, no. 5, pp. 555-599, 2005.

[18] C. A. Bowers, F. Jentsch, E. Salas, and C. C. Braun, 'Analyzing communication sequences for team training needs assessment,' Human Factors, vol. 40, no. 4, pp. 672-679, 1998.

[19] L. A. Suchman, Plans and Situated Actions: The Problem of Human-Machine Communication. Cambridge: Cambridge University Press, 1987.

[20] S. Yao et al., 'ReAct: Synergizing reasoning and acting in language models,' International Conference on Learning Representations, 2023.

[21] Q. Wu et al., 'AutoGen: Enabling next-gen LLM applications via multi-agent conversation,' arXiv:2308.08155, 2023.

[22] Y. Du et al., 'Improving factuality and reasoning in language models through multiagent debate,' arXiv:2305.14325, 2023.

[23] G. Li et al., 'CAMEL: Communicative agents for mind exploration of large language model society,' arXiv:2303.17760, 2023.

[24] J. S. Park et al., 'Generative agents: interactive simulacra of human behavior,' Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 2023.

[25] T. Guo et al., 'Large language model based multi-agents: a survey of progress and challenges,' arXiv:2402.01680, 2024.