Borghoff *et al. Discover Computing* (2025) 28:138
https://doi.org/10.1007/s10791-025-09667-2

Discover Computing

RESEARCH Open Access

An organizational theory for multi-agent interactions integrating human agents, LLMs, and specialized AI

Uwe M. Borghoff[1*], Paolo Bottoni[2†] and Remo Pareschi[3†]

†Paolo Bottoni and Remo Pareschi have contributed equally to this work.

*Correspondence:
Uwe M. Borghoff
uwe.borghoff@unibw.de
[1]Computer Science, University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
[2]Computer Science, Sapienza University of Rome, Viale Regina Elena 295, 00161 Rome, Italy
[3]Stake Lab, University of Molise, C. da Fonte Lappone, 86090 Pesche (IS), Italy

Abstract

Purpose Recent advances in AI, especially in large language models, have created new opportunities to integrate human and artificial agents through shared linguistic capabilities. This paper presents a multi-agent organizational framework in which human agents, LLMs, and specialized agents (narrow AIs) collaborate via dynamic, topic-based group formation. Topic-driven interactions enable agents to coalesce around evolving interests, supported by threshold-based protocols for temporal adaptation, topic emergence, and participation.

Methods Within our framework, human agents guide the overall system objectives, while consultant agents (LLMs) provide semantic analysis and mediation, and specialized agents perform focused domain tasks. By leveraging automated topic modeling, the approach eschews rigid ontologies and instead supports adaptive and interpretable content management. Mathematical properties ensure system coherence—across roles, tasks, and timescales—while allowing the natural evolution of interests and groups.

Results We illustrate the framework's versatility with example scenarios in emergency response, healthcare research, and financial decision-making, emphasizing how human decision-makers, LLM-based consultants, and specialized worker agents jointly fulfill complex goals through transparent topic alignment and threshold-driven coordination. This formalization advances human-computer interaction as a multi-agent phenomenon that integrates human insight with the strengths of next-generation AI models in a cohesive, evolving system.

Keywords Human-Computer Interaction, Large Language Models, Topic-Based Group Formation, Multi-Agent Systems, Natural Language Processing, Adaptive Interaction Protocols

1 Introduction

The accelerating evolution of artificial intelligence (AI), particularly through *Large Language Models* (LLMs), has enabled unprecedented collaboration between humans and machines. Yet the most transformative potential lies not in standalone agents, but in orchestrating heterogeneous collectives—human actors, LLM-based advisors, and specialized AI workers—toward common objectives. This paper introduces a novel framework combining topic-based group formation with threshold-driven protocols, yielding a unified model for human-centered multi-agent coordination.

Existing approaches confront three fundamental challenges, each with broad implications across diverse domains. First, semantic fragmentation arises from incompatible vocabularies, which hinder effective communication across agent types. This is particularly detrimental in contexts such as emergency response, where dynamically formed teams must adapt in real time to the evolving nature of a crisis, and healthcare research, which demands interpretable topic discovery that bridges the gap between clinical expertise and AI systems. Second, static coordination remains a persistent obstacle. Fixed hierarchical structures reduce the responsiveness of systems, especially in domains like business operations or financial decision-making, where the ability to reconfigure workflows adaptively is essential for maintaining efficiency and competitiveness. Third, the issue of opaque decision-making looms large. Black-box AI models obscure the rationale behind choices, thereby undermining accountability—a critical concern in high-stakes environments where transparency and trust are paramount.

To overcome these challenges, our proposed framework unifies symbolic and subsymbolic reasoning through the introduction of *Topic-Based Communication-Space Petri Nets* (TB-CSPNs). This approach enables: (i) self-organization of agents into dynamic groups based on semantic proximity, (ii) interpretable coordination mechanisms via threshold-based activation, and (iii) executive traceability designed to support human or centauric supervision [1].

Our approach is grounded in three foundational pillars that together define the architecture of our system. First, we emphasize *agent specialization*, distinguishing between human or centauric supervisors, LLM consultants, and specialized AI workers—each with distinct roles and capabilities. Second, we adopt a principle of *topic-driven coordination*, where the formation and dissolution of agent groups are governed by semantic thresholds, enabling context-aware, adaptive collaboration. Third, we formalize these dynamics through the lens of TB-CSPNs, providing mathematical guarantees for system structure, process flow, and emergent behavior. To demonstrate our framework, we pursue two complementary tracks. The first focuses on formal use cases, presenting detailed instantiations in domains such as emergency response, healthcare, and finance. The second adopts a more pedagogical perspective, using hospital-based analogies in the Appendices to convey key intuitions and principles in an accessible manner.

This work contributes to the advancement of the design of collaborative and interpretable *Multi-Agent Systems* (MAS). The following outlines the structure of the paper, highlights the novelty of the TB-CSPN model, and explains the implementation details to demonstrate the broad applicability of our approach.

Section 2 grounds the TB-CSPN framework in key concepts from MAS, topic modeling, and Petri net theory. It introduces agentic AI and centauric agents, outlines the layered roles of Supervisors, Consultants, and Workers, and explains how topic modeling enables semantic coordination. Colored Petri nets provide the formal basis for modeling concurrent, semantically driven interactions across agent types. Section 3 reviews key areas of related work to contextualize the TB-CSPN framework, including Petri net extensions, MAS coordination strategies, topic-driven interaction models, and the emerging role of LLMs as agents. It highlights how TB-CSPN builds on and extends these foundations through formal semantics, dynamic topic modeling, and integrated human-AI collaboration. The discussion also contrasts TB-CSPN with existing implementations, identifying gaps in semantic adaptability and coordination that TB-CSPN addresses, and relates it to complementary approaches focused on anomaly detection and ensemble learning. Section 4 presents the formalism of the TB-CSPN framework, which integrates topic modeling with multi-agent communication across layered Petri nets. Agents operate in Surface, Observation, and Computation layers and interact via tokens that carry semantic content. Transitions process these tokens to support semantic refinement and coordination. The framework enables dynamic group formation based on topic thresholds and supports flexible implementation through modular architectures. Section 5 outlines how TB-CSPN implements its three core agent types. Supervisor agents guide strategy and system interaction, often as human-AI hybrids. Consultant agents, typically LLM-based, manage semantic analysis and topic transformation. Worker agents carry out reliable, domain-specific tasks. Each operates through role-specific transitions aligned with the framework's layered structure. Section 6 outlines the dynamic features that support TB-CSPN's adaptability and coherence. It details role-based topic evolution, structured information flow, and threshold-driven group formation. The framework also enables validated topic emergence and uses weighted similarity measures to guide semantic alignment, ensuring both flexibility and coordinated system behavior. Section 7 outlines the parameters that govern TB-CSPN's coordination logic, from transition thresholds to topic management and similarity weighting. These parameters can be tailored to different application needs—whether prioritizing speed, precision, or strategic balance—and support runtime adaptation. This flexibility reflects the framework's ability to handle diverse operational contexts, as demonstrated across the three application domains. Section 8 presents three illustrative scenarios—emergency response, healthcare research, and financial strategy—to demonstrate TB-CSPN's versatility. Each example shows how Supervisor, Consultant, and Worker agents collaborate using topic-tagged tokens, threshold-based transitions, and structured group formation. The cases highlight how TB-CSPN supports both rapid decision-making and long-term coordination across diverse domains, using a unified, semantically grounded architecture. Section 9 explains plans to extend TB-CSPN with features like decider transitions, formal verification, and temporal logic. It also proposes a phased implementation strategy—including middleware, agent interfaces, and domain-specific prototypes—to bridge theory and practice. All components will be open source to support iterative refinement and real-world adoption. Section 10 concludes the paper. The appendices A–E formally define the TB-CSPN framework, detailing its components, agent roles, and dynamic behaviors. It provides mathematical representations of Petri net structures, token semantics, and agent transitions using real-world analogies. The dynamics of topic evolution, similarity measures, and group formation are rigorously specified, alongside the parameter configurations that govern strategic, semantic, and operational behavior. Practical guidance is also given for tuning these parameters across different domains and adapting them in real time to maintain system responsiveness and coherence. An implementation that serves as a proof of concept is available in an external repository for others to learn from, experiment with, and build upon.

2 Background and scope

Before presenting our formal framework, we establish here its conceptual foundations and position it within the broader landscape of MAS and semantic coordination. This section introduces the key concepts and technologies underlying our approach and establishes the terminology that will be used throughout the paper.

2.1 Agentic AI and multi-agent coordination

Although the term "agentic AI" has recently become popular, it is fundamentally grounded in the study of MAS. While public discourse often portrays LLMs as stand-alone general-purpose intelligences, real-world applications more commonly rely on coordinated networks of specialized agents. As noted in [2, 3], multi-agent coordination is essential for combining distinct cognitive and operational capabilities, regulating emergent dynamics in distributed systems, and ensuring that decisions remain accountable and auditable. Further, the integration of AI with human cognition—explored in [1]—blurs the line between purely human actors and AI-augmented ones. When AI support enhances human judgment without distorting its semantic integrity, we refer to such hybrids as *monotonic centauric agents*. These agents enforce monotonicity in human-AI collaboration, meaning AI augmentations can refine but never contradict existing, human-defined semantics, such as emergency protocols or clinical guidelines. In the TB-CSPN framework, topic evolution is preserved through the use of threshold guards and Supervisor oversight (see Sect. 4). Hence, these agents inherit human values and intentions while benefiting from computational enhancements.

Our framework has two distinct organizational structures, as shown in Fig. 1. First, there are the agents. Human and monotonic centauric agents fall under the *Supervisor agent* category and are supported by two types of agents: *Consultant agents*, which

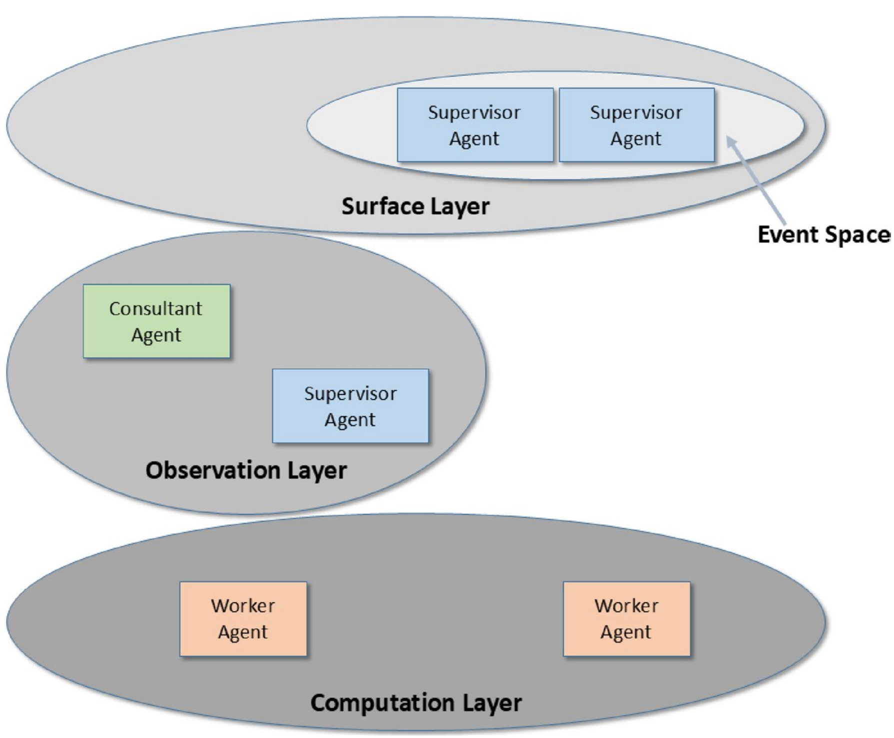

Fig. 1 Organizational structures in TB-CSPN

are typically realized through LLM-based systems that provide semantic interpretation, abstraction, and linguistic synthesis, as well as *Worker agents*, which carry out narrow, domain-specific tasks with deterministic behavior. Second are the communication layers for *Surface*, *Observation*, and *Computation*, which organize places in the TB-CSPN. Each transition belongs to exactly one layer. Although agents are typically associated with specific categories of transitions (e.g., Supervisor agents with transitions in the Surface layer), an agent can control transitions distributed across various layers. This enables agents to primarily operate within their designated layer while maintaining interaction capabilities throughout the system architecture.

While each agent type has primary responsibilities in certain layers (Supervisor agents in Surface, Consultant agents in Observation, and Worker agents in Computation), the matrix above demonstrates how all agent types can operate across multiple layers with different roles. The Event Space highlights the interface between external stimuli and the internal dynamics of the system. This flexible agent-layer architecture enables coherent coordination while maintaining specialized responsibilities.

2.2 Topic modeling as a semantic bridge

To integrate symbolic and sub-symbolic forms of intelligence, we adopt *topic modeling* as a mediating layer. Though relatively shallow as a learning method, topic modeling plays a key role as a semantic interface. It condenses large amounts of data into interpretable topic vectors, enabling human or centauric agents to inspect, revise, or curate the resulting topics. LLM-based Consultant agents can elaborate on or recombine these topics to generate deeper insights, while Worker agents subscribe to the most relevant topics to filter and act on task-specific information.

Compared to opaque end-to-end deep learning systems, topic modeling enables explicit alignment between human understanding and machine inference. Its rich history in *Natural Language Processing* (NLP) sees *Latent Dirichlet Allocation* (LDA) by Blei et al. [4] as a foundational approach. LDA models documents as mixtures of latent topics and models topics as distributions over words, creating an interpretable semantic structure. Its various extensions (see, e.g., [5, 6]) include neural topic models that leverage deep learning representations while maintaining interpretability.

Applications of topic modeling span diverse domains, including discovering latent hypertexts [7] and developing financial indexes from social media analyses [8]. Our framework extends these applications specifically to multi-agent coordination, using topic distributions as a universal semantic representation across agent types.

2.3 Petri nets for process modeling

Petri nets provide the formal foundation for our TB-CSPN framework, offering a mathematically rigorous approach to modeling concurrent and distributed systems [9, 10]. Originally developed for process modeling, Petri nets employ a bipartite graph structure with places (representing states or conditions) and transitions (representing events or actions), connected by directed arcs.

We build specifically upon colored Petri nets [11], which extend basic Petri nets by allowing tokens to carry data values (colors). This extension is essential to our approach because it enables tokens to carry complex topic distributions that influence transition behavior throughout the network.

Compared to other modeling frameworks, Petri nets offer several distinct advantages when modeling complex agentic systems that involve heterogeneous human and artificial agents, including LLMs and centauric supervisors. Graph-based agent models—such as those found in standard MAS frameworks [12, 13]—typically represent interactions via state diagrams or message-passing graphs, but they often lack the formal semantics necessary to model concurrency, token-based flow, or semantically enriched coordination. Petri nets, by contrast, are expressly designed to model concurrent processes with causal dependencies and asynchronous communication, which are core characteristics of modern agentic systems.

Unlike purely mathematical approaches such as process algebras (e.g., π-calculus) [14, 15], which excel in specifying communication patterns at a logical level, Petri nets offer a more intuitive and implementation-ready representation of system state, data flow, and organizational structure. This makes them especially suited for agent orchestration in real-world, semantically layered environments such as those modeled by TB-CSPN. Furthermore, Petri nets support key formal properties—such as reachability, liveness, and boundedness—that enable both theoretical analysis and practical system verification. These features are essential for safety-critical domains, such as healthcare, financial systems, emergency response, and military intelligence analysis [16], where TB-CSPN applications must ensure consistent, dependable, and traceable coordination. Finally, Petri nets support a natural integration of centauric human-AI collaboration: by associating transitions with high-level supervisory decisions (strategic, ethical, contextual), we can capture the *effect* of human judgment and its augmentation through AI—without needing to fully model internal cognitive processes. This sets Petri nets apart from most MAS formalisms, which often treat human agents as external to the formal structure.

In summary, Petri nets provide an expressive and extensible framework that bridges symbolic process modeling with sub-symbolic semantic processing, offering a uniquely powerful tool for modeling the hybrid, adaptive, and semantically rich dynamics of agentic human-AI systems.

3 Related work

Having established the technological foundations of our approach, we now position our research within the broader landscape of related work, highlighting how our framework extends or complements existing approaches to multi-agent coordination, LLM agency, and human-AI integration.

3.1 Related petri net extensions

Several specialized Petri net extensions share conceptual similarities with aspects of our TB-CSPN framework and may inform future developments: *Hierarchical Petri Nets* [17] provide mechanisms for abstraction and modular design, with conceptual parallels to our communication layers. *Petri Nets with Inhibitor Arcs* [18] allow transitions to be conditionally enabled or disabled based on token presence, similar to our threshold-based activation mechanisms. *Contextual Petri Nets* [19] enhance the formalism with context-dependent conditions for transition firing, which conceptually relates to our topic-dependent activation.

While our current TB-CSPN realization does not directly incorporate these extensions, they represent valuable directions for future refinement as discussed in Sect. 9.

3.2 MAS and coordination

MAS research provides essential concepts and methodologies for modeling distributed interactions among heterogeneous computational entities. The field has a rich history of addressing coordination challenges through various mechanisms and protocols [12, 13, 20]. We specifically build upon the comprehensive survey by Dorri et al. [21], which covers core MAS concepts, agent taxonomies, and coordination strategies.

Traditional MAS research has explored various organizational structures for agent coordination. Horling and Lesser [22] survey organizational paradigms including hierarchies, holarchies, coalitions, teams, and federations—concepts that inform our layered approach. While MAS frameworks offer valuable abstractions for agent behavior and interaction, they often lack the formal machinery required for modeling detailed token-based semantics or verifying cross-layer coordination at scale. In contrast, our TB-CSPN framework builds on the formal foundations of Petri nets to offer both a high-level view of agent roles and a rigorous execution model grounded in concurrency theory. Moreover, by explicitly incorporating humans—and centauric agents—as formal participants, TB-CSPN extends traditional MAS concepts into the domain of human-agent integration, positioning our work alongside recent efforts to reconceptualize human-computer interaction as human-agent interaction [2, 3].

For formal modeling of agent interactions, we also draw from frameworks such as *Concurrent Game Structures* and *Alternating-Time Temporal Logic* [23–25]. Our approach complements these by integrating explicit topic-based semantics and layered communication across agent types within a formally verifiable structure.

3.3 Topic-driven interaction systems

TB-CSPN's emphasis on topic-based coordination relates to emerging research on topic-driven systems across various domains. Lu et al. [26] integrate multimodal emotion recognition with topic-aware models for social robotics, enabling context-sensitive interactions. Similarly, Wang et al. [27] develop topic-driven knowledge selection mechanisms for dialogue generation that improve the relevance of system responses.

The concept of implicit interactions based on shared topics has been explored by Hanteer and Rossi [28], who analyzed how users in social networks cluster around thematic interests. Our approach extends this concept to explicit coordination among heterogeneous agents, using topic distributions as the primary mechanism for dynamic group formation across human, LLM, and specialized AI agents.

3.4 LLMs as agents

The emergence of LLMs as potential agents marks a significant development in AI research. Recent work by Singh et al. [29] explores the shift from linear interactions to dynamic, iterative workflows with LLMs, incorporating reflection, planning, and multi-agent collaboration.

The role of LLMs as semantic mediators is further explored by Ghisellini et al. [30], who demonstrate how LLMs can integrate strategic frameworks with decision heuristics through semantic analysis. Their approach uses vector space representations and similarity calculations to bridge previously separate knowledge traditions, similar to how our Consultant agents synthesize information across domains. Their work highlights the potential of LLMs as bridges between formal analytical structures and experiential knowledge, a function analogous to the role of our Consultant agents in connecting Supervisor strategic intent with Worker operational execution.

Our use of LLMs as Consultant agents relates to emerging research on LLMs for planning and reasoning. Capitanelli and Mastrogiovanni [31] demonstrate LLMs' capabilities in planning actions for human-robot interactions—conceptually aligned with our consultant agents' role in bridging strategic intent and operational execution.

For the specific challenge of integrating symbolic and subsymbolic reasoning, we draw from Tenenbaum's work [32] on bridging these approaches, which aligns with our use of topic modeling as a mediating layer between agent types.

3.5 Human-AI integration and collaboration

Integrating humans and AI systems constitutes a central concern of our framework. The concept of centauric agents (human-AI hybrids) builds upon work in augmented intelligence [1] and human-AI teaming [33].

For evaluation of human responses to AI systems, we acknowledge work by Giudici et al. [34], who analyze user reactions to AI within interactive contexts, and Kim and Im [35], who investigate attributions of human traits to AI systems. These human factors influence how our supervisor agents interact with the broader system.

The emotional and social dimensions of human–machine interaction are further explored in Ahmadi and Haddadi's survey on Artificial Emotional Intelligence [36], which aligns with our concern for effective communication across agent types.

Our approach differs from many human-AI collaboration systems by treating humans as first-class agents within the formal model rather than external controllers or supervisors of an AI system. This perspective enables more natural integration of human decision-making within the overall system dynamics, while preserving appropriate roles and responsibilities across agent types. While TB-CSPN emphasizes semantic mediation and dynamic task orchestration, Zhang et al. [37] focus on multi-dimensional feature fusion and ensemble learning for robust and accurate anomaly detection, offering a complementary perspective on intelligent system responsiveness.

3.6 Agentic AI implementations and formal gaps

Several recent frameworks incorporate aspects of agentic behavior, often within multi-agent configurations. While these systems provide practical utility, they generally lack cohesive semantic coordination and a formal foundation. As shown in Table 1, many of these implementations capture parts of agentic coordination but do not reach the level of dynamic, semantically grounded interaction enabled by TB-CSPN.

Systems like LangChain's ReAct [38] and AutoGen MAD [39] use turn-based loops and procedural message passing to model multi-agent interaction, mimicking Surface and Consultant-Worker transitions, but lacking semantic thresholding or adaptive

Table 1 Mapping Practical Implementations to TB-CSPN Components

System	TB-CSPN Analog	Limitations
LangChain ReAct	Consultant-Worker interactions	Procedural, lacks semantic modeling
AutoGen MAD	Multi-agent Surface transitions	No layered semantics
TogetherAI Orchestrator	Supervisor-Worker split	Static role assignment, no adaptation
Agentic RAG	Topic tokens as Surface signals	Manual topic curation
ReWOO	Sequential Worker flows	Hardcoded logic, no dynamic roles

coordination. Frameworks such as TogetherAI's Orchestrator [40] impose a fixed Supervisor-Worker division that limits role flexibility and semantic mediation, whereas TB-CSPN enables fluid specialization across Supervisor, Consultant, and Worker roles. In terms of semantic grounding, Agentic RAG [41] introduces topic-like tokens but depends on manual curation and static structures, while TB-CSPN dynamically generates topic semantics through token flow and agent interaction. Finally, unlike rigid systems such as ReWOO [42], which follow predefined agent flows, TB-CSPN supports real-time adaptation through token-driven transitions and layered semantics. This allows for flexible and emergent process management.

These implementations collectively illustrate the growing demand for sophisticated agentic coordination. However, they reveal key limitations in semantic modeling, adaptability, and structural formalism. TB-CSPN addresses these through a mathematically grounded, multi-layered architecture that unifies Petri net dynamics with emergent topic-based semantics.

4 A framework for TB-CSPNs

Our framework combines topic modeling with multi-agent communication to provide a structured yet flexible way of representing interactions among diverse agent types. These interactions are guided by dynamically evolving topics of interest, enabling context-aware coordination and adaptation.

At its core, the TB-CSPN framework models a MAS as a structured network. In this network, places (shown as circles) store topic-labeled tokens, while transitions (shown as rectangles) process these tokens, modifying their content and topic distributions. The tokens themselves carry both data and semantic topic distributions. Information flows through three distinct layers—Surface, Observation, and Computation—each supporting a different aspect of agent communication and coordination.

This structure creates a natural hierarchy that mirrors organizational roles. Supervisor agents—purely human or centauric (human-AI hybrids) [1, 3]—primarily operate in the Surface layer, setting strategic direction. The centauric nature of these agents is particularly valuable when processing massive influxes of raw data (such as IoT broadcasts or real-time sensor networks) that would overwhelm purely human cognitive capabilities. By augmenting human strategic thinking with AI-driven data processing, centauric supervisor agents can effectively manage the Surface layer despite increasing data complexity. Consultant agents bridge the Surface and Observation layers, analyzing and transforming topic distributions. Worker agents function mostly in the Computation layer, performing specialized domain tasks.

We summarize these matrix-like interactions in Table 2, which shows typical agent functions across communication layers.

Table 2 Mapping of agent functions across communication layers

LAYER/ROLE	Supervisor	Consultant	Worker
Surface	Strategic direction; external mediation	Optional preprocessing (e.g., early filtering)	*Rare*: direct interaction with raw data
Observation	Guidance of semantic interpretation	Semantic analysis; topic refinement; heuristic reasoning	Support for translation into computation
Computation	Policy enforcement; value alignment	Transformation into operational tokens	Execution of domain-specific tasks

4.1 Petri net constituents across layers

In TB-CSPN, agents are modeled as bound to transitions, though they may also own places or store tokens for private use. Places function as shared communication channels that support asynchronous interaction via token flow. To increase expressiveness, the framework distinguishes between two types of places: communication places, which are accessible to transitions from different agents and facilitate inter-agent coordination; and agent-owned places, which act as private memory areas accessible only by transitions of the owning agent. The interaction between layers in TB-CSPN is governed by the type of transitions. Intra-layer transitions occur within a single communication layer, enabling coordination at the same semantic level. Cross-layer transitions allow for vertical semantic refinement by moving tokens between layers. These transitions are core mechanisms of computation and transformation. Agent cooperation is modeled through the production and consumption of tokens.

Regarding token semantics, TB-CSPN extends the principles of colored Petri nets: tokens are structured data objects with the following components:

- *Payload*: The primary content or data being processed (e.g., text, numerical values, or structured records)
- *Topic Distribution*: A mapping from topics to weight values, representing the token's semantic characterization (e.g., $\{\mathrm{topic}_1 : 0.8, \mathrm{topic}_2 : 0.5, \mathrm{topic}_3 : 0.3\}$)
- *Metadata*: Additional attributes such as timestamps, provenance information, or processing directives

When a transition "refines" a token in TB-CSPN, it creates a new token instead of altering the original. Token flow across layers follows two main patterns: persistence via transformation, where each step generates a semantically related token that retains part of the original payload while refining its topic distribution; and replacement, where the new token contains entirely new information, altering both payload and topic. This approach maintains clarity and analytical rigor while enabling meaningful semantic evolution across the system.

4.2 Communication layers and topic flow

As shown in Fig. 1, TB-CSPN divides the network into three communication layers, each corresponding to a distinct level of information processing. The Surface layer is the interface between the system and external entities (users, sensors, or other systems). Tokens entering here typically contain raw information without fully developed topic distributions. Semantic analysis and transformation occur in the Observation layer. The Consultant agents operating at this level enrich tokens with topic distributions based on content analysis, identifying patterns and relationships. In the Computation layer, Worker agents perform domain-specific operations based on well-structured topic information. Figure 2 shows some examples.

4.3 Threshold-based dynamic group formation

A significant feature of the TB-CSPN framework is its ability to dynamically form agent groups based on shared topical interests. Individual agents maintain interest thresholds relative to specific topics. When a token surpasses an agent's threshold for topic relevance, the agent becomes a candidate for processing it. Groups organically emerge

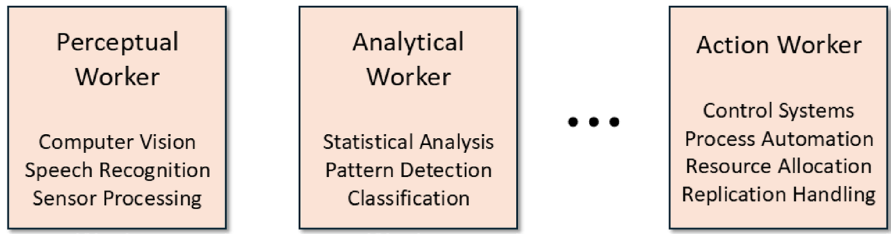

Fig. 2 Worker agent domain-specific operation examples

from multiple agents sharing common topical interests, facilitating collaborative efforts without rigid hierarchies. Importantly, as the semantic content evolves over time, agent memberships in these ad hoc groups adjust dynamically, ensuring fluidity and responsiveness to changing informational contexts.

The TB-CSPN formalism is adaptable to a range of implementation architectures, including message-passing systems and microservices. In practice, tokens are defined as semantically enriched messages with detailed metadata, while transitions are implemented as discrete computational services that handle token transformation, enrichment, and routing. Subscription mechanisms enable agents to process only messages relevant to their topic thresholds, and communication layers are structured as distinct processing layers to ensure clarity in information flow. For the full mathematical specification and additional formal details, see Appendix A.

5 Agent implementations

Now that the TB-CSPN framework has been defined, we turn to the practical implementation of its different agent types.

5.1 Supervisor agent

Supervisor agents form the strategic core of the system. They orchestrate high-level decision-making and mediate crucial interactions between external environments and internal system processes. Their responsibilities are twofold. First, they provide strategic direction by establishing overarching objectives, authorizing the formation of agent groups, and shaping the system's long-term evolution. Second, they serve as mediators between the system and the outside world. They evaluate incoming information and determine how to semantically frame and introduce it into the token space. These functions are carried out through specialized processes that guide internal collaboration and system-environment interaction.

In contemporary implementations, supervisor agents increasingly operate as centauric entities—human-AI hybrids—in response to the growing complexity and scale of information beyond the capabilities of the human mind alone. This hybrid approach is especially effective in three key scenarios. First, when data volumes exceed human processing capacity, AI augmentation enables supervisors to manage high-throughput data streams from sources such as IoT devices, multimodal inputs, and real-time telemetry. In contexts where complex decision-making requires computational insight, AI supports intricate reasoning, optimization, and uncertainty analysis, while human oversight ensures ethical coherence. Finally, in high-stakes environments where a rapid response is essential, such as in crisis response or financial markets, AI provides immediate analytical support. This allows humans to focus on supervisory control and value alignment.

This integration is consistent with the principle of monotonic centaurs [1], which aim to augment human cognition without compromising the semantic clarity or strategic coherence of the decision-making process.

Within TB-CSPN, Supervisor agents fulfill their roles through purpose-built transitions. *Initiative transitions* create tokens that define new strategic directions and catalyze the formation of collaborative agent groups. *Authorization transitions* evaluate proposals generated by Consultant agents and serve as control points for high-impact adaptations and structural changes. Through *delegation transitions*, Supervisor agents assign tasks to Consultant or Worker agents, orchestrating operational workflows that align with strategic goals. Finally, *external interface transitions* process inputs from the environment—whether originating from users, sensors, or external systems—and convert them into structured tokens with initial topic distributions. These transitions serve as the gateway through which the outside world enters the semantic fabric of the system. Through these transitions, Supervisor agents provide top-down guidance while maintaining a responsive, semantically grounded interface with both internal agent layers and external realities.

5.2 Consultant agent

The crucial role of Consultant agents is to bridge the strategic intent set by Supervisor agents and the operational execution carried out by Worker agents. They specialize in semantic analysis, topic modeling, and strategic recommendation, making them indispensable within the TB-CSPN framework.

Consultant agents play a pivotal role within the TB-CSPN architecture, operating primarily in the Observation layer to mediate between surface-level interactions and deeper semantic processes. Their responsibilities span several critical functions. They begin with semantic analysis, transforming unstructured or loosely structured content into rich topic distributions that capture underlying meaning. Through pattern recognition, they identify emerging trends and latent relationships across tokens, enabling them to support strategic foresight and decision-making. These insights feed into strategic recommendations, where consultants advise on optimal group configurations, topic refinements, and resource allocations. Finally, they undertake operational translation, bridging the gap between high-level directives issued by Supervisor agents and the concrete task representations needed by Worker agents. Thanks to their semantic fluency, Consultant agents are particularly well-suited for implementation using advanced LLMs. The inherent capabilities of LLMs—contextual reasoning, natural language understanding, and adaptive semantic processing—enable them to fulfill these roles with precision and flexibility.

Within a TB-CSPN, Consultant agents operate through specialized transitions, each designed to perform a distinct semantic function. *Content enhancement transitions* enrich individual tokens by refining their semantic granularity without compromising the integrity of their original payload. *Topic aggregation transitions* synthesize multiple information sources into high-level summary tokens, helping to distill strategic insights from distributed inputs. *Memory and temporal analysis transitions* ensure that historical context is preserved and leveraged, enabling the system to adapt to ongoing developments over time. Finally, *group formation recommendation transitions* actively suggest collaborative configurations by analyzing topical alignments and interaction patterns, thereby fostering responsive and relevant team dynamics.

Incorporating LLMs into Consultant agents significantly enhances their capacity for semantic precision. These models allow agents to extract implicit meaning from multimodal inputs, discover subtle connections between different concepts, and generate context-sensitive, historically informed recommendations. LLMs also facilitate communication across specialized technical vocabularies and general linguistic registers, effectively mediating heterogeneous agent environments.

In TB-CSPN, Consultant agents occupy a central intermediary position, orchestrating the flow of semantically enriched information across agent layers. At the Surface-Observation interface, they translate external stimuli into semantically structured token representations. Through the topic transformation pipeline, they refine these tokens iteratively for clarity and relevance. They communicate actionable insights back to supervisor agents via robust feedback mechanisms, thereby strengthening strategic decision-making with grounded analytical input. They also play a key role in group coordination by aligning topical interests with organizational priorities to support dynamic, goal-oriented collaboration.

5.3 Worker agent

Worker agents comprise the operational core of the TB-CSPN framework. They translate high-level strategic objectives into precise, reliable, executable tasks within specific domains. They serve as the backbone of TB-CSPN execution, primarily operating within the Computation layer to deliver deterministic, domain-specific functionality. These agents are responsible for domain-specific execution and perform well-defined operations within specialized fields—ranging from automated data processing to mechanical control—with a high degree of reliability. Their behavior is characterized by deterministic processing, whereby consistent inputs yield reproducible outputs. This supports system predictability and formal verifiability. By doing so, Worker agents optimize resource efficiency while complying with strict performance constraints and minimizing computational overhead. They also contribute to higher-level reasoning layers by reporting their status and results. They transmit structured outputs and performance indicators to inform Supervisor and Consultant agents.

Worker agents' functionality is expressed through a set of domain-aligned transitions, each of which enables a specific class of operational behavior. *Perceptual transitions* handle the conversion of raw sensory data—such as visual streams, audio inputs, or environmental readings—into structured, actionable tokens. These transitions are often the first stage in translating physical-world observations into computationally tractable forms. *Analytical transitions* focus on interpretive tasks, such as statistical inference, pattern recognition, and specialized classification routines based on the agent's domain expertise. Finally, *action transitions* are responsible for executing specific tasks, including dispatching control signals, automating tasks, allocating resources dynamically, and managing workflows.

These transitions work together to provide the stable, localized functionality necessary to support the broader emergent behaviors of the TB-CSPN system. By ensuring deterministic execution and tightly scoped operations, Worker agents contribute critical structure and reliability to the overall architecture.

Appendix B provides comprehensive mathematical and formal descriptions of these transitions, along with their structural roles in TB-CSPN.

6 Framework properties and dynamics

We now examine the dynamic properties that support the framework's adaptability and coherence. These properties influence how the system evolves, detects new topics, manages information flow, and maintains semantic consistency. These properties also help TB-CSPN balance stability with responsiveness.

6.1 Temporal evolution and interaction flow patterns

TB-CSPN manages temporal dynamics through hierarchical topic evolution, structured by agent roles:

- *Role-Based Evolution Rates*: Supervisor transitions induce substantial topic changes, Consultant transitions introduce moderate semantic adjustments, and Worker transitions produce minimal, precise refinements. This ranking of semantic influence ($\delta_S \geq \delta_C \geq \delta_W$) ensures ordered cascading of updates.
- *Transition-Specific Patterns*: Each agent type modifies token topics per its role—Supervisors drive strategic priorities, Consultants refine semantic clarity, and Workers ensure operational fidelity.
- *Organizational Stability*: Ongoing Supervisor oversight and multi-layered topic interest regulation maintain coherence and prevent semantic fragmentation.
- *Interest Inertia*: Newly introduced topics initially receive constrained weights, preventing abrupt semantic shifts and supporting stable adaptation.

TB-CSPN supports structured information flows that reflect the organizational architecture. Figure 3 illustrates the canonical processing pathway through all three

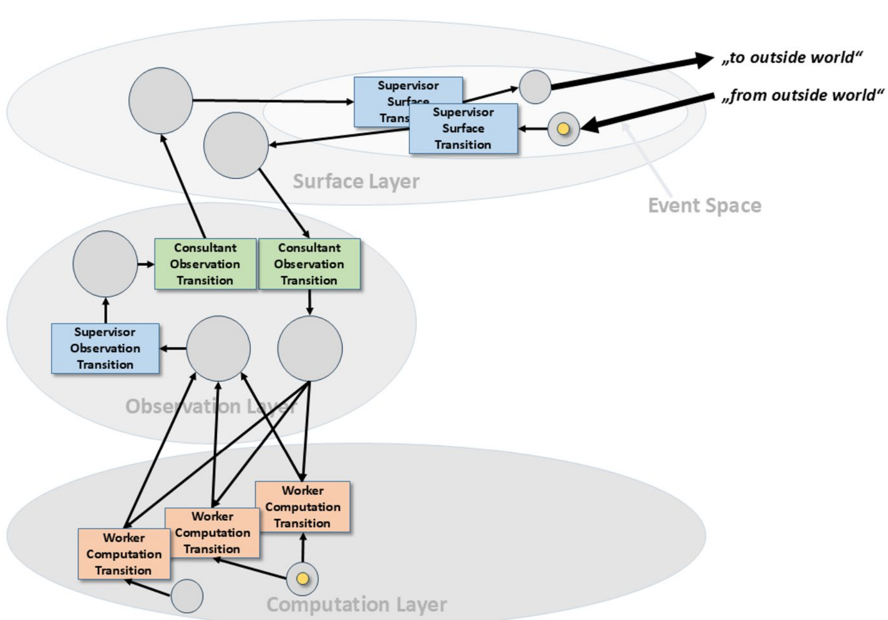

Fig. 3 Standard token flow in TB-CSPN showing the canonical pathway through Surface, Observation, and Computation layers. In this typical operational pattern, token processing follows the layered architecture, with each agent type operating primarily within its designated layer

communication layers. This pattern is consistent with the notion of a *Workflow Petri net* [43]. A Supervisor agent performs a transition fired by external input at the Surface layer, thus establishing high-level goals for the system. A Consultant transition in the Observation layer models the analysis and refinement of the resulting tokens so they can be mapped to semantically meaningful topics aligned with current objectives. One or more available Worker agents then execute the required transition at the Computation layer according to the analysis, realizing a domain-specific action that implements the strategic intent. The token describing the result of this execution is then made available to fire a Consultant (complex) transition again at the Observation layer. Through this transition, outcomes are interpreted, results are summarized, and possible relevant patterns or anomalies are identified. Finally, a Supervisor transition at the Surface layer is triggered by the complex token that describes the result of this execution. This evaluation may result in adjustments to strategic plans, which determine the next cycle of coordinated action.[1]

6.2 Group Formation and Threshold Dynamics

While agents primarily operate within their designated layers as shown in Fig. 1, the TB-CSPN framework enables dynamic group formation that can span across layers when topic distributions trigger interest across different agent types. This cross-layer group formation facilitates coordinated responses to topics requiring multi-faceted expertise. It enables Supervisor, Consultant, and Worker agents to collaborate despite their different operational contexts.

Dynamic group formation is a core feature of the TB-CSPN framework, governed by topic-based thresholds. As Fig. 4 demonstrates, threshold mechanisms enable context-sensitive coordination:

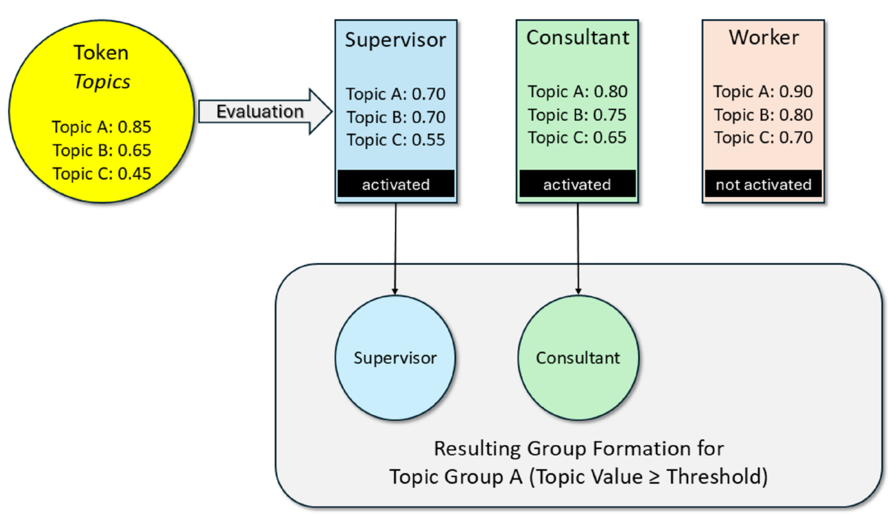

Fig. 4 Threshold-based group formation in TB-CSPN showing how varying topic weights trigger agent participation. As tokens evolve across layers, different agents engage based on their threshold settings, creating emergent collaboration patterns that adapt to changing semantic contexts

[1] In the stylized representation of Fig. 3, we have collapsed possibly sophisticated activities into single transitions. These transitions could be expanded into subnetworks using *Hierarchical Petri nets* [17]. Additionally, we are not considering the specific assignment of Supervisors or Consultants to specific agents. While we assume that different Worker agents would perform the required tasks in different ways, this introduces a degree of non-determinism to the actual workflow.

- *Threshold-Based Activation*: Agents join collaborative groups when token topic weights exceed their individual interest thresholds.
- *Contextual Participation*: As topic distributions evolve, group membership dynamically adjusts without requiring explicit reconfiguration.
- *Hysteresis Effects*: Small threshold margins (ϵ values) prevent oscillatory group membership, ensuring operational stability.
- *Role-Appropriate Thresholds*: Threshold values typically follow $\tau_S \leq \tau_C \leq \tau_W$ to ensure hierarchical activation aligned with organizational responsibilities.

This configuration ensures that strategic oversight is triggered early, while execution-level actions only occur when the topic has sufficient semantic strength. The threshold-based approach enables flexible self-organization while maintaining coordinated system behavior aligned with strategic objectives.

6.3 Topic emergence and similarity measures

The TB-CSPN framework enables the dynamic emergence of new topics through a structured, multi-stage validation process that respects the hierarchical roles of agents. For a topic to be formally integrated into the system, it must first undergo cross-layer validation: Supervisor agents must authorize its strategic relevance, Consultant agents must confirm its semantic coherence, and Worker agents must verify its practical applicability. Beyond this layered approval, the topic must meet specific stability requirements, demonstrating persistence over time, coherence across semantic layers, and broad endorsement from relevant agent types. Additionally, semantic differentiation is essential—new topics must introduce conceptually distinct content that adds meaningful value to the system, avoiding overlap or redundancy with existing topics. After passing these checks, the TB-CSPN framework incorporates a topic into the active semantic landscape. This involves updating key structural components, such as color functions, guard conditions, token patterns, and similarity matrices. This allows the system to adapt seamlessly to new knowledge.

Figure 5 illustrates a specific subprocess within our TB-CSPN framework focused on the *topic emergence and validation* workflow. Although most operations follow a

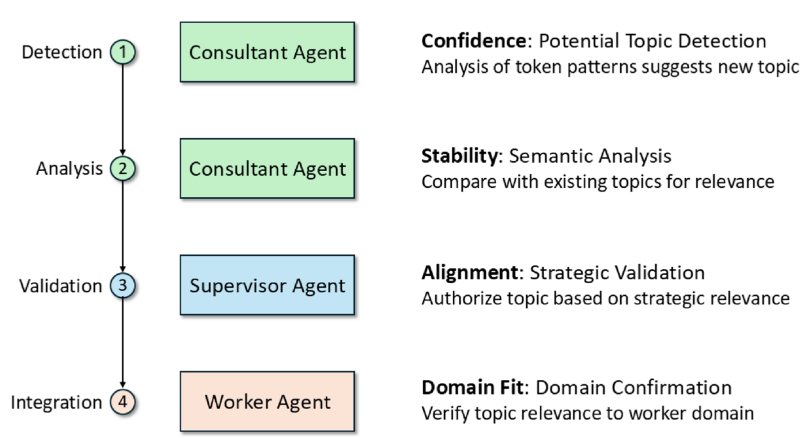

Fig. 5 Topic emergence and validation process showing the multi-stage evaluation of potential new topics across agent types. This specialized workflow illustrates one of several interaction patterns supported by the TB-CSPN framework, with direct connections representing its adaptive capabilities

Table 3 Summary of TB-CSPN Application Scenarios

Feature	Emergency Response	Healthcare Research	Financial Strategy
Key Topics	Evacuation, medical, logistics	Genomic markers, drug trials	AI stocks, market volatility
Time Scale	Seconds-minutes	Weeks-months	Hours-days
Threshold Priority	Speed ($\tau_S \leq 0.5$)	Precision ($\tau_C \geq 0.85$)	Balance ($\tau_W \approx 0.9$)
Group Dynamics	Rapid formation/ dissolution	Emergent topic validation	Continuous portfolio updates
Agent Focus	Coordination	Discovery	Optimization

standard flow pattern, specialized processes such as topic emergence may involve direct connections between different agent types when semantically appropriate.

Consultant agents initiate the topic detection and analysis phases. Supervisor agents provide strategic validation, and Worker agents confirm domain relevance. These direct connections serve as exception paths for topic validation rather than typical operational flows. This flexibility enables specialized workflows while preserving the framework's architectural integrity.

In TB-CSPN, topic similarity is evaluated across different agent layers, each reflecting distinct semantic priorities based on the agent's role. At the Supervisor level, similarity is assessed using high-level topic co-occurrence patterns aligned with strategic objectives. Consultant agents use statistical modeling, NLP techniques, and domain-informed reasoning to compute similarity and capture nuanced semantic relationships. At the Worker level, similarity is evaluated using task-specific metrics, which are often based on execution requirements or data compatibility. This ensures that operational actions align with the relevant semantic context. With this layered approach, similarity assessments can be adapted to the needs and perspectives of each agent type. A weighted combination of these layer-specific similarities—parameterized by (w_S, w_C, w_W)—allows for the fine-tuning of system behavior to emphasize strategic alignment, semantic refinement, or operational execution.

Formal definitions and update semantics are provided in Appendix C.

7 Parameter configuration and tuning in TB-CSPN

The TB-CSPN framework relies on a set of configuration parameters that govern how transitions, places, and tokens behave. These parameters determine how topics evolve, how transitions are activated, and how the system adapts to changing semantic and operational contexts. Each parameter group plays a distinct role in shaping system dynamics. Supervisor agents are modeled as human or hybrid (centauric) actors, but the framework does not attempt to simulate internal human reasoning. Instead, Supervisor transitions represent the production of strategic decisions—such as initiating groups, adjusting thresholds, or approving topics—while abstracting away the cognitive processes behind them. This design follows established practices in human-in-the-loop and business process modeling, where human actions are formalized to maintain coherence in complex systems.

For centauric agents, which combine human judgment with AI capabilities, this abstraction is particularly well suited. It allows the system to encode collaborative outcomes while preserving a clear, structured representation. In contrast, Worker and Consultant transitions are modeled as synthetic behaviors. Supervisor transitions serve as formal proxies for human or hybrid interventions, enabling the system to reflect strategic decision-making and coordinate activity across layers.

7.1 Parameter categories and their roles

The parameters of TB-CSPN are organized into six major categories, each playing a distinct role in determining how the system evolves:

Supervisor Transition Parameters orchestrate the system's strategic behavior. These include the maximum variation rates (δ_S), the initial influence of new topics introduced by Supervisor transitions (α_S), and the thresholds that determine when these transitions become active (τ_S). Together, they modulate how Supervisor nodes prioritize novelty and responsiveness: higher values of α_S amplify the impact of emergent topics, while elevated τ_S values restrict Supervisor activation to situations of pronounced topic relevance.

Consultant Transition Parameters manage the analytical layer of the system. Governed by parameters such as δ_C, which restricts the rate of topic transformation, Consultant transitions ensure that token modifications maintain semantic continuity while remaining sensitive to contextual changes. This configuration promotes a balance between adaptability and analytical consistency.

Worker Transition Parameters are tailored to operational stability. Here, parameters like δ_W play a crucial role: by limiting topic weight variation during Worker transitions, they preserve dependable behavior in task-specific contexts, minimizing fluctuations in the system's execution layer.

Group Formation Parameters—denoted by ϵ_S, ϵ_C, and ϵ_W—introduce hysteresis effects in transition enabling/disabling mechanisms. These parameters are designed to prevent erratic toggling caused by marginal fluctuations in topic relevance. By doing so, they stabilize group memberships over time and contribute to a more coherent dynamic organization of the MAS.

Topic Management Parameters ($\sigma_{\min}$, $\sigma_{\max}$, $c_{\text{stability}}$) control how topics are kept, merged, or removed during system operation. $\sigma_{\min}$ and $\sigma_{\max}$ set the bounds for topic similarity, while $c_{\text{stability}}$ defines how much evidence is needed to keep or update a topic. These parameters ensure that the system remains semantically consistent while enabling the emergence of new ideas.

Similarity Function Weights (w_S, w_C, w_W) adjust how much influence Supervisor, Consultant, and Worker agents have when measuring topic similarity. They let the system shift focus—for example, favoring Supervisor input in emergencies or Consultant analysis during uncertain situations.

For a detailed table of all parameters, their ranges, and behavioral implications, please refer to Appendix D.

7.2 Threshold dynamics example

To illustrate how threshold parameters dynamically govern transition enabling and group formation, consider the following scenario derived from an emergency response domain within the TB-CSPN model:

1. *Initial transition enabling:* At time *k*, a Worker transition associated with a rescue team has an evacuation topic threshold of 0.7. A token with evacuation weight of 0.85 arrives at the relevant place. Since 0.85 exceeds the activation threshold, the transition becomes enabled and the rescue team joins the active response group.
2. *Topic weight decrease:* At time $k + 1$, a new token arrives, now with an evacuation weight of 0.65. Falling below the 0.7 threshold, this causes the transition to become

disabled, leading to the temporary exclusion of the rescue team from coordinated action.

3. *Consultant transition mediation:* The system reacts by activating a Consultant transition, which diagnoses the context change and produces a token for Supervisor review, highlighting the modification in group composition.
4. *Threshold adjustment by Supervisor:* In response, the Supervisor transition recalibrates the threshold, lowering it to 0.6. At time $k+2$, the previously excluded transition becomes active again, since the current topic weight (0.65) now exceeds the updated threshold, restoring the rescue team's participation.

This scenario illustrates how thresholds enable dynamic, context-sensitive regulation of group participation. Higher-level transitions monitor and adapt to fluctuations in topic relevance.

7.3 Parameter configuration for different domains

The TB-CSPN framework is built to adapt flexibly across a range of application domains by adjusting key parameters. To illustrate this adaptability, we propose example configurations for three representative contexts.

In emergency response scenarios, the system must prioritize speed and responsiveness. In this setting, parameters are adjusted to enable rapid topic integration, low activation thresholds, and swift topic evolution. This allows for fast group formation and timely action under urgent conditions.

- High α_S(0.7$-$0.8): fast integration of novel topics
- Low τ_C, τ_W(0.5$-$0.6): minimal thresholds for analytical and operational activation
- Elevated δ_S, δ_C(0.2$-$0.3): promotes swift topic evolution
- Low $c_{\text{stability}}$(0.4): accelerates topic emergence
- High w_S(0.5$-$0.6): prioritizes Supervisor perspective in similarity assessments

In collaborative research contexts, greater selectivity and topic stability are preferable.

- Moderate α_S(0.5$-$0.6): balanced topic introduction
- High τ_C, τ_W(0.6$-$0.7): stronger evidence required for activation
- Low δ_S, δ_C(0.1$-$0.2): supports semantic consistency
- Higher $c_{\text{stability}}$(0.5$-$0.6): ensures topic persistence
- Balanced weights w_S, w_C, w_W(0.4, 0.4, 0.2): maintains multiple viewpoints

In structured business and financial environments, a balance between operational reliability and strategic responsiveness is critical:

- Moderate α_S(0.6), low α_W(0.2): prioritizes strategy without sacrificing stability
- Stratified thresholds: τ_S(0.5), τ_C(0.6), τ_W(0.8): enforces role hierarchy
- Low δ_W(0.05$-$0.1), moderate δ_C(0.15$-$0.2): operational consistency with analytical flexibility
- Larger ϵ_W(0.1): prevents frequent Worker deactivation
- $\sigma_{\min}$(0.4), $\sigma_{\max}$(0.8): maintains clear category boundaries

7.4 Runtime parameter adaptation

The TB-CSPN framework includes built-in mechanisms that adapt key parameters at runtime. This maintains responsiveness in dynamic and evolving environments.

Threshold learning adjusts the activation criteria for Supervisor transitions based on performance feedback, enabling the system to optimize task allocation in real time. Similarity weight adjustment shifts the influence of Supervisor, Consultant, or Worker transitions depending on topic stability—emphasizing Consultant input during periods of uncertainty or Supervisor control during strategic shifts. Finally, interest evolution rate control dynamically tunes the δ parameters to match system activity, speeding up topic evolution when rapid adaptation is needed and reinforcing stability during routine operations.

In summary, the TB-CSPN parameter system offers detailed and flexible control over the coordination of multiple agents. By tuning parameters to specific domains and adjusting them at runtime, the framework strikes a balance between responsiveness and coherence, as well as autonomy and structure.

8 Examples of framework application

To illustrate TB-CSPN's versatility, we present three domain examples where Supervisor, Consultant, and Worker agents collaborate through topic-labeled tokens and threshold-based coordination. These demonstrate how the formal properties in Sect. 6 enable both rapid response and long-term collaboration.

It is important to note that the Emergency Response scenario provided is intended only as a conceptual illustration of the dynamics and capabilities of the TB-CSPN framework; it does not reflect any deployment in real-world emergency systems or operational environments. Similarly, the Healthcare Research scenario presented here is purely illustrative, designed to demonstrate the formal and operational properties of the TB-CSPN framework; it does not involve or report on any clinical trials, patient data, or empirical medical studies. Also, the Financial Strategy scenario discussed is a hypothetical scenario designed to illustrate the adaptability of the TB-CSPN framework; it does not reflect any live trading systems or actual financial operations.[2]

8.1 Emergency response coordination as TB-CSPN

Our first example models an emergency response system using the TB-CSPN framework. In this scenario, rapid group coordination is critical, requiring fast token flow, strategic topic routing, and threshold-based enabling of transitions. Group formation is represented through the manipulation of structured tokens encoding group identity, membership, and coordination scope.

We start with an initial TB-CSPN configuration.

[2] While the appendices use hospital management as an accessible analogy to explain key concepts, the real-world scenarios presented in the main text go further by demonstrating the full formalization of the TB-CSPN framework, including all system constraints. These scenarios also showcase the use of finely tuned operational parameters—beyond the illustrative values used in the appendix—and validate system behavior through end-to-end token flows, providing a more complete and realistic view of the model in action.

$$
\begin{aligned}
T_{\mathrm{E}} =& \{t_{\textsf{emergency\_coordinator}}\} \quad \text{(Supervisor Transitions)} \\
T_{\mathrm{C}} =& \{t_{\textsf{situation\_analyst}}, t_{\textsf{resource\_coordinator}}\} \quad \text{(Consultant Transitions)} \\
T_{\mathrm{W}} =& \{t_{\textsf{rescue\_team1}}, t_{\textsf{medical\_team1}}, t_{\textsf{logistics\_team1}}, t_{\textsf{weather\_bot}}, t_{\textsf{traffic\_bot}}\} \\
& \text{(Worker Transitions)} \\
P_{\mathrm{surf}} =& \{p_{\mathrm{alerts}}, p_{\mathrm{directives}}, p_{\mathrm{status}}\} \\
P_{\mathrm{obs}} =& \{p_{\mathrm{analysis}}, p_{\mathrm{coordination}}, p_{\mathrm{resources}}\} \\
P_{\mathrm{comp}} =& \{p_{\textsf{rescue\_ops}}, p_{\textsf{medical\_ops}}, p_{\textsf{logistics\_ops}}, p_{\textsf{weather\_data}}, p_{\textsf{traffic\_data}}\} \\
P_{\mathrm{groups}} =& \{p_{\textsf{group\_registry}}\} \\
\mathrm{T} =& \{\text{evacuation}, \textsf{medical\_emergency}, \textsf{supply\_chain}, \textsf{weather\_alert}, \\
& \textsf{traffic\_condition}\}
\end{aligned}
$$

At time *k*, the TB-CSPN system receives an emergency alert, which is introduced as a token in the place p_{alerts}. This token represents an initial, unstructured input—such as a message reporting a critical event—that will be processed and semantically enriched as it flows through the network. This marks the starting point for coordinated agent activation and response based on the alert's topic content.

$$
\begin{aligned}
&\tau_{\mathrm{alert}} = (\text{“Severe flooding reported in the downtown area, multiple injuries,} \\
&\text{roads blocked.”}, \{\text{timestamp} : k, \text{source} : \text{“emergency\_hotline”}\})
\end{aligned}
$$

The token is processed by a *Consultant* transition $t_{\textsf{situation\_analyst}}$ that processes τ_{alert} and produces:

$$
\begin{aligned}
&\tau_{\mathrm{analyzed}} = (\ldots, \{(\text{evacuation}, 0.85), (\text{medical\_emergency}, 0.75), \\
&\quad (\text{traffic\_condition}, 0.70), (\text{weather\_alert}, 0.60), (\text{supply\_chain}, 0.45)\}, \ldots)
\end{aligned}
$$

A recommendation token is produced:

$$
\begin{aligned}
\tau_{\mathrm{recommendation}} = (&\text{“Recommend immediate evacuation and medical response”}, \\
&\{(\text{evacuation}, 0.9), (\text{medical\_emergency}, 0.8)\}, \\
&\{\text{timestamp} : k, \text{type} : \text{“recommendation”}\})
\end{aligned}
$$

It is placed in $p_{\mathrm{coordination}}$ for review by the *Supervisor*. At time *k*, the group formation is encoded in a structured *group token* inserted into $p_{\textsf{group\_registry}}$:

$$
\begin{aligned}
\tau_{\mathrm{group}} = (&\textsf{group\_id} = \text{“flood\_response”}, \\
&\text{members} = \{\text{rescue\_team1}, \text{medical\_team1}, \text{traffic\_bot}\}, \\
&\text{topics} = \{(\text{evacuation}, 1.0), (\text{medical\_emergency}, 0.8)\}, \\
&\text{metadata} = \{\text{timestamp} : k, \text{priority} : \text{“high”}\})
\end{aligned}
$$

Transition guards consult this registry to check topic thresholds and group inclusion.

$$
\begin{aligned}
&\text{enabled}(t_{\textsf{rescue\_team1}}, p_{\textsf{rescue\_ops}}) = \\
&\quad \textsf{is\_member}(t_{\textsf{rescue\_team1}}, \text{“flood\_response”}) \wedge \text{topic[evacuation]} \geq 0.7 \\
&\text{enabled}(t_{\textsf{medical\_team1}}, p_{\textsf{medical\_ops}}) = \\
&\quad \textsf{is\_member}(t_{\textsf{medical\_team1}}, \text{“flood\_response”}) \wedge \text{topic[medical\_emergency]} \geq 0.7
\end{aligned}
$$

At time *k*, topic proximity shapes routing and inference

$$\begin{aligned}
&\mathrm{sim}_E(\text{evacuation}, \text{medical_emergency}) = 0.85\\
&\mathrm{sim}_C(\text{evacuation}, \text{medical_emergency}, k) = 0.80\\
&\mathrm{sim}_W(\text{evacuation}, \text{medical_emergency}) = 0.75\\
&\mathrm{sim}(\cdot) = 0.82 \quad \text{with weights } w_E = 0.5, w_C = 0.3, w_W = 0.2
\end{aligned}$$

This example illustrates how TB-CSPN effectively supports emergency response coordination by combining topic-driven semantics with structured agent interactions. Topic-tagged tokens are routed through transitions that are semantically relevant to their content, and transitions are enabled based on dynamic topic interest thresholds. Structured tokens in a registry maintain persistent group configurations, while semantic and structural guards coordinate activation across agent roles. In practice, the Supervisor (acting as the emergency coordinator) defines strategic directives and encodes them into group structures. Consultant agents, such as situation analysts, interpret incoming data and generate actionable recommendations. Worker agents—including rescue teams, medical units, and traffic controllers—are activated based on group membership and topic relevance, ensuring a rapid and organized response.

8.2 Healthcare research collaboration in TB-CSPN

This example models research collaboration using the TB-CSPN framework, emphasizing longer-term interaction and emerging new topics.

The initial TB-CSPN Configuration is described as follows:

$$\begin{aligned}
T_{\mathrm{E}} =& \{t_{\text{research_director}}\} \quad \text{(Supervisor Transitions)}\\
T_{\mathrm{C}} =& \{t_{\text{research_analyst}}, t_{\text{domain_coordinator}}\} \quad \text{(Consultant Transitions)}\\
T_{\mathrm{W}} =& \{t_{\text{genomics_analyzer}}, t_{\text{clinical_trials_bot}}, t_{\text{statistics_bot}},\\
& t_{\text{drug_interaction_analyzer}}\} \quad \text{(Worker Transitions)}\\
P_{\mathrm{surf}} =& \{p_{\mathrm{findings}}, p_{\mathrm{directives}}, p_{\mathrm{publications}}\}\\
P_{\mathrm{obs}} =& \{p_{\mathrm{analysis}}, p_{\mathrm{correlations}}, p_{\text{research_topics}}\}\\
P_{\mathrm{comp}} =& \{p_{\text{genomics_data}}, p_{\text{clinical_data}}, p_{\text{statistical_analysis}}, p_{\text{drug_data}}\}\\
P_{\mathrm{groups}} =& \{p_{\text{group_registry}}\}\\
\mathrm{T} =& \{\text{rare_diseases}, \text{drug_trials}, \text{genomic_markers},\\
& \text{patient_outcomes}, \text{statistical_analysis}\}
\end{aligned}$$

At time *k*, a research finding enters the system as a token in p_{findings}:

$$\begin{aligned}
\tau_{\mathrm{finding}} = (&\text{“New correlation discovered between genetic marker XYZ}\\
&\text{and treatment response in rare autoimmune conditions.”},\\
&\{\}, \{\text{timestamp} : k, \text{source} : \text{“lab_report_128”}\})
\end{aligned}$$

The token is processed by a *Consultant* transition $t_{\text{research\_analyst}}$ that produces:

$$\begin{aligned}
\tau_{\mathrm{analyzed}} = (&\text{“New correlation discovered between genetic marker XYZ}\\
&\text{and treatment response in rare autoimmune conditions.”},\\
&\{(\text{genomic_markers}, 0.90), (\text{rare_diseases}, 0.85),\\
&(\text{drug_trials}, 0.65), (\text{patient_outcomes}, 0.60),\\
&(\text{statistical_analysis}, 0.55)\},\\
&\{\text{timestamp} : k, \text{source} : \text{“lab_report_128”}, \text{analyzed_by} :\\
&\text{“research_analyst”}\})
\end{aligned}$$

The analyzed token is placed in p_{analysis}, where it enables *Consultant* transitions to process it further, i.e., $t_{\mathsf{research\_analyst}}$ produces a recommendation token

$$\begin{aligned}\tau_{\text{recommendation}} = (&\text{“Recommend focused research on XYZ marker's mechanism”},\\ &\{(\text{genomic\_markers}, 0.95), (\text{rare\_diseases}, 0.90)\},\\ &\{\text{timestamp} : k, \text{type} : \text{“research\_direction”}\})\end{aligned}$$

The recommendation token is placed in $p_{\mathsf{research\_topics}}$ for *Supervisor* consideration. At time *k*, as in the emergency coordination case, group formation is managed via colored tokens, storing group identity, membership, and associated topics, placed in the static group registry. Let transition $t_{\mathsf{research\_director}}$ process $\tau_{\text{recommendation}}$ and produce:

$$\begin{aligned}\tau_{\text{group}} = (&\mathsf{group\_id} = \text{“autoimmune\_study”},\\ &\text{members} = \{\text{research\_analyst}, \text{genomics\_analyzer},\\ &\text{drug\_interaction\_analyzer}\},\\ &\text{topics} = \{(\text{genomic\_markers}, 1.0), (\text{rare\_diseases}, 0.9)\},\\ &\text{metadata} = \{\text{timestamp} : k, \text{priority} : \text{“high”}\})\end{aligned}$$

This token is inserted into $p_{\mathsf{group\_registry}}$. Transition guards later check group membership via this registry. After several firing sequences, a new potential topic emerges:

$$\begin{aligned}&(\text{autoimmune\_genetics}, 0.75, \{\text{genomic\_markers}, \text{rare\_diseases}\}) \in\\ &\quad \text{emerge}(\{\tau_1, \tau_2, \ldots, \tau_n\}, k+5)\end{aligned}$$

This topic undergoes multi-level validation, and upon confirmation, is integrated into the TB-CSPN structure.

This example illustrates how TB-CSPN enables collaborative research via topic-tagged tokens, static yet semantically dynamic group formation through structured group tokens, threshold-based transition enabling, and topic emergence mechanisms.

8.3 Financial scenario in TB-CSPN

Our third example introduces a financial decision-making scenario, modeling the strategic management of a media-sensitive investment portfolio. The system monitors information flows, assesses investment opportunities, and dynamically reallocates resources based on topic-driven signals from the financial ecosystem.

$$\begin{aligned}T_{\text{S}} &= \{t_{\mathsf{market\_director}}\} \quad \text{(Supervisor Transitions)}\\ T_{\text{C}} &= \{t_{\mathsf{risk\_analyst}}, t_{\mathsf{opportunity\_scanner}}\} \quad \text{(Consultant Transitions)}\\ T_{\text{W}} &= \{t_{\text{simulator}}, t_{\mathsf{forecast\_engine}}, t_{\mathsf{portfolio\_optimizer}}\} \quad \text{(Worker Transitions)}\\ P_{\text{surf}} &= \{p_{\text{news}}, p_{\text{directives}}, p_{\text{signals}}\}\\ P_{\text{obs}} &= \{p_{\text{analysis}}, p_{\text{strategies}}\}\\ P_{\text{comp}} &= \{p_{\text{simulations}}, p_{\text{metrics}}, p_{\text{portfolios}}\}\\ \text{T} &= \{\text{AI\_stocks}, \text{climate\_finance}, \text{market\_volatility}, \text{regulation}, \text{publ\_sentiment}\}\end{aligned}$$

A token from news feeds is inserted into p_{news}: τ_{input} =(“Tech sector sees surge amid AI breakthroughs”,, {source: “Reuters”, timestamp: k}). The transition *risk_analyst* enriches it by τ_{analyzed} =(..., {(AI_stocks, 0.9), (market_volatility, 0.6)},...). A *Supervisor* transition then issues the directive $\tau_{\text{directive}}$ =(“Consider increased exposure to AI sector”, {(AI_stocks, 1.0)},...) which is placed into $p_{\text{directives}}$ and read by a *Worker* transitions $t_{\text{simulator}}$

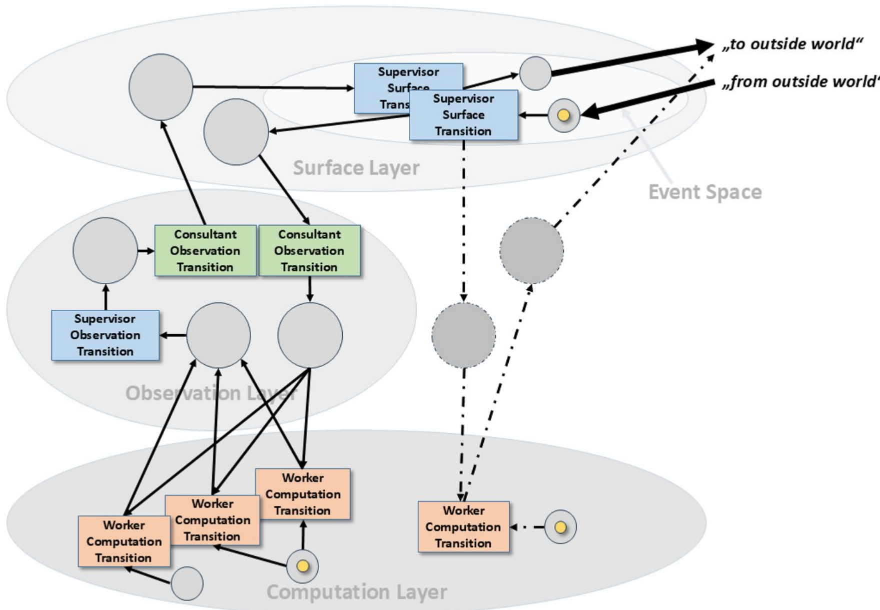

Fig. 6 In addition to the standard token flow in TB-CSPN shown in Fig. 3, presenting a canonical path through the Surface, Observation, and Computation layers, decider transitions (dashed lines) can bypass a layer to address time-critical scenarios or obtain secondary validations

that generates projected outcomes based on AI market conditions, and $t_{\text{portfolio\_optimizer}}$ proposes allocation changes across assets. Enabling is governed by thresholds, e.g.: $\text{guard}(t_{\text{portfolio\_optimizer}}, p_{\text{portfolios}}) = \text{AI\_stocks} \geq 0.8$. As in prior examples, topic similarity is computed: $\text{sim}(AI\_stocks, market\_volatility, k) = 0.72$

This financial scenario illustrates how TB-CSPN enables responsive, data-driven portfolio management through coordinated semantic processing. Consultant agents, such as LLM-based analysts, proactively intercept and analyze emerging topics from financial news and market signals. Based on this analysis, Supervisors redirect strategy by issuing new directives aligned with the most relevant topics. Worker agents then execute simulations and optimize portfolio allocations accordingly. Together, these components enable real-time adjustments and strategic coherence in a complex, fast-moving financial environment.

9 Future work and implementation pathways

While TB-CSPN offers a solid formal foundation for multi-agent coordination, there is still room for further development. This section presents plans to refine the theory and create practical implementations connecting the formal model to real-world systems.

9.1 Theoretical extensions

As hinted at in Sect. 3.1, we plan to explore possible refinements of the model, such as those offered by *Hierarchical Petri Nets* [17], *Petri Nets with Inhibitor Arcs* [18], or *Contextual Petri Nets* [19], with a view to modeling sophisticated forms of control over task execution and cooperative construction of results targeted at specific objectives, possibly under adversarial contextual conditions.

One specific extension we intend to develop is a TB-CSPN structure with so-called *decider transitions*. Figure 6 shows how these transitions (represented by dashed lines) can bypass conventional layer progression. These decider transitions serve two important purposes:

- *Real-time responsiveness*: Accelerating decision processes when external constraints demand rapid responses, allowing direct Supervisor-Worker interactions in time-critical scenarios.
- *Validation redundancy*: Enabling secondary computations from additional Worker agents to verify results. This is particularly useful for reducing the effects of hallucinations or errors in LLM-based Consultant agents.

To minimize the risk of LLM hallucinations, we plan to integrate *Retrieval-Augmented Generation* (RAG) into future versions of the TB-CSPN model [44, 45]. This will allow Consultant agents to base their results on external knowledge using decider transitions or other techniques. It will also allow for real-time validation between agents and secondary verification by Worker agents. These redundancies and controls will be implemented before LLM-generated content is officially integrated.

Further theoretical work includes designing formal verification methods tailored to TB-CSPN and integrating temporal logic to capture time-sensitive threshold behavior.

9.2 Implementation strategy

Translating the TB-CSPN framework into practical systems requires a phased development approach that balances theoretical integrity with integration feasibility.

- As a first step in implementation, we will develop a middleware platform that encodes the core semantics of TB-CSPN using modular, interoperable components. This includes a *token orchestration layer* that integrates message queues, such as Kafka[3] or RabbitMQ,[4] with a token state tracker, such as Neo4j[5] or RedisGraph.[6] This integration enables asynchronous, semantically structured transitions between agents. A *topic modeling service* will support this layer by leveraging existing libraries like Gensim[7] for LDA or HuggingFace[8] for neural models, with enhancements for online learning and incremental updates. Finally, a *threshold registry* will manage agent-specific activation conditions, offering potential interfaces for external control via policy APIs or machine learning-based tuning mechanisms.
- Next, we will define a set of standardized interfaces for each agent type to ensure consistent interaction across the system. *Supervisor APIs* will support secure, authenticated communication with human or centauric agents. *Consultant connectors* will provide abstraction layers for integrating with various LLM backends—including OpenAI, Anthropic, and open-source models—while enforcing a consistent structure for prompts and responses. *Worker wrappers* will encapsulate domain-specific microservices or toolchains, presenting them through a unified interface. To facilitate interoperability across components, we will also integrate

[3] https://kafka.apache.org/documentation/

[4] https://www.rabbitmq.com/

[5] https://neo4j.com/

[6] https://github.com/RedisGraph.

[7] https://radimrehurek.com/gensim/

[8] https://huggingface.co/

schema adapters compatible with frameworks like LangChain [38] and AutoGen [39].

- Finally, we will develop use-case-specific prototypes to validate the TB-CSPN framework in real-world scenarios. These prototypes will include an *emergency response coordination system* that integrates real-time data feeds, such as weather updates, social media alerts, and logistics APIs; a *healthcare research collaboration hub* that connects clinicians with LLM-powered analytical tools; and a *financial decision support module* that enables dynamic portfolio adjustments and risk alerts based on evolving market signals. These reference implementations will serve as practical testbeds to refine the framework and demonstrate its versatility across domains.

All components will be released as open-source modules to enhance existing agentic frameworks with a formal coordination layer based on token semantics and topic-driven Petri nets. Feedback from implementation will help refine the theory, creating a two-way link between practical design and formal modeling.

10 Conclusion

We have proposed TB-CSPN as a comprehensive formal framework for agent orchestration in the emerging era of agentic AI. Our contribution is significant in two fundamental respects:

First, we establish a mathematically rigorous model for integrating heterogeneous agent types—human, LLM-based, and specialized AI—within a unified coordination structure. This formalization provides theoretical guarantees while supporting the practical flexibility demanded by complex sociotechnical systems. By developing a formal approach to agent orchestration that accommodates both human and artificial participants, we create a foundation for systems that can evolve alongside advances in AI without losing coherence or human centrality.

Second, our work bridges two traditions with deep foundations in computer science that have historically traveled separate paths: the theory of concurrent processes and natural language processing. By integrating Petri nets (a classical formalism for modeling concurrent systems) with topic modeling and LLMs (key technologies in modern NLP), we create a synergy that addresses the fundamental coordination challenges of mixed-initiative systems. This integration enables semantic mediation across traditionally siloed AI paradigms, offering a principled approach to the agent heterogeneity that characterizes effective real-world systems.

This bridging of traditions is particularly evident in how our framework builds upon our earlier work [2, 3]. In those models, the Surface layer mediated environmental interaction, the Observation layer interpreted and routed content, and the Computation layer housed logic and decision strategies—an approach particularly suited to interactive software systems where computation concentrated in the Computation layer and observation acted as a bridge to the surface.

The TB-CSPN framework evolves this foundation in two important respects:

First, it explicitly introduces distinct agent types for each communication layer—*Supervisors*, *Consultants*, and *Workers*—rather than assuming that computation alone accounts for the system's intelligence and coordination. This is more than a terminological refinement; it reflects a conceptual expansion. In this model, agents in the Surface and Observation layers are not passive conduits but active participants in behavior, interpretation, and strategic regulation.

Consultant agents, typically instantiated as LLMs, exemplify this shift. While LLMs are popularly associated with chatbots, that is only one of their possible uses. In our framework, Consultant agents serve as proactive semantic mediators. They interpret input tokens, refine topic distributions, detect latent structures, and recommend actions—behaviors enabled through API-based interactions rather than user-facing dialogs. As illustrated in our examples, these agents often exhibit foresight and top-down influence, intercepting and expanding topics in ways that contribute directly to organizational intelligence.

Second, the TB-CSPN framework distinguishes decision-making from computation. Supervisor agents, situated at the Surface layer, are responsible for strategic orientation and external mediation. While this may appear to contradict the earlier model, which housed decision-making within computation, it reflects a broader understanding: decision processes include not only algorithmic inference but also bounded rationality, contextual heuristics, and value-guided prioritization. These capacities align more naturally with strategic oversight than with low-level logic.

Although the examples in the paper focus on cross-layer interactions, the TB-CSPN framework can also handle intra-layer exchanges. For instance, it can accommodate interactions among Supervisor agents at the Surface layer or among Worker agents at the Computation layer. These lateral transitions can result in subnetworks that can be modeled using hierarchical Petri nets or by explicitly defining peer-level arcs.

Overall, the TB-CSPN framework offers a significant advance in formalizing multiagent orchestration for the age of agentic AI. By combining Petri nets' mathematical rigor with the semantic power of topic modeling and LLMs, we provide both theoretical foundations and practical mechanisms for systems where human and artificial agents collaborate effectively. This approach addresses the growing diversity of intelligent agents in real-world applications—human, symbolic, and neural—and establishes principled structures for their adaptive coordination across differentiated roles.

A Formal Specification of TB-CSPN

How to Use This Appendix: This section provides the complete formal specification of the TB-CSPN framework. For readers seeking a more conceptual understanding, intuitive explanations of key concepts are woven throughout the text, and the core formal requirements of the model are clearly identified. To help contextualize the formalism, we also include healthcare-based examples that illustrate how TB-CSPN components operate in real-world scenarios. Before diving into the technical definitions, we begin with an illustrative case drawn from a hospital coordination system.

In this context, *places* represent communication channels—such as an emergency room dashboard or a lab results queue. *Transitions* correspond to processing units, including roles like a triage nurse or an AI-based diagnostic agent. Meanwhile, *tokens* carry patient cases enriched with topic distributions, for example: [cardiac: 0.9, urgent: 0.7]. This scenario provides an intuitive mapping between TB-CSPN components and real-world healthcare operations, grounding the formalism in a concrete domain. To build an intuitive understanding, imagine a Petri net as a type of workflow system. In this system, *places* act like containers or queues that hold individual work items, while *transitions* represent the processing steps that consume those items and produce new ones. The *tokens* are the work items themselves, moving through the network as tasks are carried out. *Arcs* define the paths along which these tokens travel, connecting places to transitions and structuring the flow of activity.

Formally, a *colored* Petri net is defined as:

$$\mathcal{N} = (P, T, A, \Sigma, \text{color}, \text{guard}, M_0)$$

Each component of the Petri net maps to a concept relevant to real-world domains. P represents work queues—for example, a place labeled "Unprocessed X-rays" where incoming cases wait to be reviewed. T defines the processing units, such as an AI-powered radiology system that interprets imaging data. The function color specifies data format rules, ensuring that tokens conform to standards like DICOM for medical imaging.[9] Finally, guard defines entry conditions for transitions—for instance, a rule that only chest X-rays may be processed by a particular diagnostic module.

The process begins when meaningful topics emerge within the system. For example, based on incoming patient reports, the system may detect a topic such as cardiac_emergency. This triggers relevant agent activation and group formation. In response to the detected topic, the system activates the appropriate agents to form a coordinated response. A human cardiologist joins the supervisory group, denoted as $\mathcal{AG}_S$, to provide strategic oversight. An LLM-powered triage assistant is assigned to the consultant group, $\mathcal{AG}_C$, to support semantic interpretation and decision-making. Meanwhile, an ECG analyzer is activated within the worker group, $\mathcal{AG}_W$, to perform specialized diagnostic tasks.

Formally, this structure is described as follows:

$$\begin{aligned} \mathcal{TP} &= \{\text{stroke}, \text{sepsis}, \ldots\} \\ \mathcal{AG} &= \mathcal{AG}_S \cup \mathcal{AG}_C \cup \mathcal{AG}_W \\ \text{dist}(\mathcal{TP}) &\in \mathbb{P}(\mathcal{TP}) \end{aligned}$$

Consider a sample token representing a patient case. The payload contains the main clinical information, such as "68-year-old male, chest pain." This is enriched with topic annotations that capture semantic relevance—for instance, cardiac: 0.9, male: 0.5—indicating a strong association with cardiac issues and moderate relevance to gender. The token also includes metadata, such as priority: high, source: EMT, which provides contextual details about the urgency of the case and its origin.

Because the weight of the topic cardiac has exceeded the 0.8 threshold, signaling high relevance, agents respond selectively based on predefined thresholds. The cardiologist, whose threshold for cardiac topics is 0.7, is activated and joins the group. The general MD, however, has a higher threshold of 0.9 and does not participate, as the topic weight falls below the required level. Meanwhile, the ECG bot, with a lower threshold of 0.6, is also activated, recognizing the topic's significance and engaging accordingly.

Mathematically, the group formation process is captured by the function:

$$g(t) = \{a \in \mathcal{AG} \mid \exists \tau \in \mathcal{TP} : \text{interest}_a(\tau, t) \geq \theta_a(\tau)\}$$

This defines a group $g(t)$ at time t consisting of agents a whose interest in some topic τ meets or exceeds their individual threshold $\theta_a(\tau)$.

[9] https://www.dicomstandard.org/

Different implementation strategies within TB-CSPN are suited to varying organizational contexts. A centralized approach—such as a hospital command center—is effective for small, tightly coordinated teams where oversight is concentrated. In contrast, a distributed setup, like a telemedicine network, is better suited for cross-facility coordination where agents operate semi-autonomously across locations. For scenarios requiring both high-level coordination and local autonomy, a hybrid model—such as a regional trauma system—offers a balanced solution that can flexibly accommodate both centralized directives and distributed execution.

B Formal specification of agent transitions

This section presents the formal roles of Supervisor, Consultant, and Worker agents within the TB-CSPN framework, using parallels from hospital operations to clarify abstract computational mechanisms. These roles are introduced through practical analogies to familiar medical processes, while mathematical definitions ground their behavior within the system's formal semantics.

In the context of a hospital, supervisors—such as chief medical officers or department heads—are responsible for high-level decision-making, including the allocation of resources, the implementation of treatment protocols, and staff coordination. These real-world roles closely mirror those of Supervisor agents in TB-CSPN, who provide strategic oversight and guide the system's overall structure. While these agents may be implemented as humans or human-AI hybrids, the transitions they govern are formal representations of strategic outcomes. Such transitions are not concerned with internal cognition but rather with external decisions that have structural implications for the system, including group formation and topic validation.

These abstract operations are directly relatable to administrative decisions in hospital management. For example, opening a new ICU wing corresponds to the creation of a new agent group; authorizing an experimental protocol reflects the validation of a new topic; and assigning patients to specific teams maps to the delegation of tasks. Within TB-CSPN, such groups are formalized as structured tokens comprising a group identifier, a list of agent members, associated semantic topics, and supplementary metadata. These tokens reside in a persistent registry and are queried by guard conditions, enabling transitions to dynamically reason about group composition and coordination context.

The Supervisor agent's functionality is governed by a trio of transition functions. The initiative function creates new group tokens; the authorization function determines whether proposed changes are accepted; and the delegation function routes strategic tasks to appropriate Consultant or Worker agents. This is expressed as follows:

$$\begin{aligned} \mathrm{initiative} &: \mathrm{S} \times \mathbb{N} \times \mathrm{T} \to \wp(\mathrm{G}) \\ \mathrm{authorize} &: \mathrm{S} \times \mathrm{G} \times \mathbb{N} \to \mathrm{BOOL} \\ \mathrm{delegate} &: \mathrm{S} \times \mathrm{Task} \times \mathbb{N} \to \mathrm{C} \cup \mathrm{W} \end{aligned}$$

For instance, a transition t_S may model an administrative workflow; changes to a guard condition $\mathrm{guard}(t_g)$ could represent updates to protocol approval criteria; and the creation of a token τ_{out} might correspond to a triage event assigning a patient to a new care pathway.

Consultant agents act as semantic intermediaries between strategic oversight and operational execution. Their real-world counterparts include medical AI systems that interpret clinical data, recommend diagnostic procedures, or propose treatment options. When a diagnostic system outputs a confidence level above a given threshold, Consultant components are selectively activated. For example, a treatment recommender may engage immediately, while a second-opinion module might remain idle. A test selector might also be triggered to suggest complementary diagnostics.

These activation decisions are driven by enhanced threshold logic and semantic similarity assessments. The enhanced threshold θ_{enhanced} is derived from a combination of a base diagnostic threshold and a context-dependent adjustment:

$$\theta_{\text{enhanced}} = \text{combine}(\theta(d), \theta_C(d,t))$$

Similarly, the semantic similarity score used for activation blends global topic similarity with a Consultant-specific evaluation:

$$\text{sim}_{\text{enhanced}} = \lambda \cdot \text{sim}(t_1, t_2, t) + (1-\lambda) \cdot \text{sim}_C(t_1, t_2)$$

These blended functions allow Consultant agents to respond with nuance by adjusting their actions based on both global system knowledge and local task relevance.

Worker agents perform specialized, deterministic tasks and are analogous to automated clinical instruments like MRI machines or lab robots. These agents operate within tightly scoped domains, and their behavior is governed by three core guarantees. First, domain containment ensures that each Worker interacts only with relevant places in the net, captured formally as $\forall p \in P_{\mathcal{D}_w}$. Second, deterministic processing guarantees consistency in output, expressed as $\text{color}_w(\tau, t) = \text{color}_w(\tau, t')$ for repeated invocations under identical conditions. Third, execution timing is bounded within a performance window: $\mu_{\min} \leq \text{fire\_time} \leq \mu_{\max}$, which allows for predictable scheduling within the network's computational fabric.

A typical patient workflow illustrates the coordinated actions of the three agent types. A Supervisor admits a patient and creates a token for the new case. A Consultant agent analyzes the intake report, identifies a likely cardiac issue, and assigns a high relevance score to the topic. A Worker agent performs an ECG test, producing a structured token as output. The Consultant then reviews the result, refines the topic distribution, and submits findings. Finally, the Supervisor evaluates the complete case and approves a course of treatment. This sequence demonstrates how TB-CSPN agents operate across strategic, semantic, and operational layers in a coordinated loop.

C Formal specification of dynamic properties

Navigating Dynamic Properties: This section explores how TB-CSPN handles temporal evolution, topic emergence, and role-specific interaction patterns. Conceptual explanations are interwoven with formal definitions, and real-world medical workflows are used to illustrate each concept.

In hospital operations, different roles adapt at different rates. Supervisors, such as administrators, may revise policies on a quarterly basis. Consultants—diagnosticians and clinical advisors—tend to update methods more frequently, such as monthly. Worker agents, like lab instruments or automated systems, maintain consistent procedures over longer periods.

To formalize this, each agent type is associated with a change bound that defines how quickly their behavior or topic distributions may evolve. Supervisors have a bound of

$\delta_S \leq 0.3$, reflecting the relatively slow pace of institutional policy change. Consultants are bounded by $\delta_C \leq 0.2$, modeling moderate adaptation, such as shifts in diagnostic practices. Workers are the most stable, with $\delta_W \leq 0.1$, indicating consistent execution of standardized protocols. These constraints are encoded as:

$$
\begin{aligned}
&|\text{topics}(\tau)[t] - \text{topics}(\tau')[t]| \leq \delta_S \quad \text{(Supervisor)}\\
&|\text{topics}(\tau)[t] - \text{topics}(\tau')[t]| \leq \delta_C \quad \text{(Consultant)}\\
&|\text{topics}(\tau)[t] - \text{topics}(\tau')[t]| \leq \delta_W \quad \text{(Worker)}
\end{aligned}
$$

Topic emergence within TB-CSPN often mirrors how new medical protocols are introduced in clinical settings. For instance, the detection of an unusual cluster of patient cases may prompt a Supervisor—such as a chief physician—to validate a new topic. A Consultant then analyzes its correlation with existing topics, and if it passes similarity thresholds, Worker agents may be reconfigured or activated to address it. This process is governed by formal validation requirements, ensuring that newly surfaced topics are semantically relevant and operationally actionable.

$$
\begin{aligned}
&\exists t_S \in T_\text{S} : \text{authorize}(t_S, \text{topic}, k) = \text{TRUE}\\
&\qquad\qquad\quad \text{sim}_C(\text{topic}, \text{topic}', k) > \sigma_{\min}\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{topic} \in \mathcal{D}_{t_W} \text{ for relevant } t_W \in T_\text{W}
\end{aligned}
$$

To understand the interaction dynamics across agent roles, consider a typical patient diagnosis flow. A Supervisor initiates the process by admitting the patient. A Consultant then orders appropriate diagnostic tests. The Worker agent performs the lab analysis, after which the Consultant interprets the results. Finally, the Supervisor reviews the output and approves a treatment plan. This sequence is formalized as:

$$
\begin{aligned}
\text{Initiative} &: t_S \rightarrow p_\text{surf}\\
\text{Analysis} &: p_\text{surf} \rightarrow t_C \rightarrow p_\text{obs}\\
\text{Execution} &: p_\text{obs} \rightarrow t_W \rightarrow p_\text{comp}
\end{aligned}
$$

Each phase of this flow corresponds to a distinct layer of the TB-CSPN architecture: Surface, Observation, and Computation. This allows for clear separation of concerns and facilitates traceable transitions.

Underpinning this entire process is a semantic similarity model that determines how agents align with topics and tasks. At the strategic level, sim_S captures alignment with organizational priorities. For Consultants, sim_C measures diagnostic correlation, often informed by semantic models such as LLMs. For Worker agents, sim_W captures compatibility with operational capabilities, such as whether a particular lab test is supported. These components are integrated into a weighted similarity function:

$$
\begin{aligned}
&\qquad \text{sim}(t_1, t_2, k) = w_S \text{sim}_S + w_C \text{sim}_C + w_W \text{sim}_W\\
&\text{where} \quad w_S + w_C + w_W = 1
\end{aligned}
$$

This formulation ensures that alignment is context-sensitive and can be tuned based on the relative importance of strategic, diagnostic, or operational considerations.

D Formal specification of parameter configuration

How to Navigate This Section: This section introduces key TB-CSPN parameters through analogies to hospital management. Intuitive medical parallels are used to motivate each parameter, while essential mathematical rules formalize their behavior. Throughout, real-world configurations and parameter effects are illustrated using representative process flows.

Before exploring formal definitions, consider several familiar hospital scenarios. The parameter δ_S reflects how frequently hospital policies are updated—quarterly in fast-paced environments, or annually in more stable ones. The threshold τ_C represents the minimum confidence required for a test result to support a diagnosis, such as 80%. The weight w_S describes the influence of a chief physician relative to clinical specialists in treatment decisions. Finally, ϵ_W can represent a short grace period before surgical equipment is reassigned, ensuring operational stability.

These parameters can be grouped by the type of control they represent. Supervisor agents, similar to hospital administrators, manage strategic parameters such as policy change frequency (δ_S), the strength with which new protocols are initially introduced (α_S), and urgency thresholds that trigger intervention (τ_S). Formally:

$$\begin{aligned} |\mathsf{new\_policy} - \text{current}| &\leq \delta_S \\ \mathsf{init\_weight}(\mathsf{new\_protocol}) &\leq \alpha_S \\ \text{approve} \iff \text{urgency} &\geq \tau_S \end{aligned}$$

In this case, δ_S limits the pace of policy adjustments, α_S bounds the initial influence of untested protocols, and τ_S defines the urgency required for executive approval.

On the clinical side, parameters affect how quickly diagnostic and procedural standards can evolve, and the confidence level required before decisions are made. For example, δ_C restricts how often Consultant agents update diagnostic guidelines and δ_W governs the stability of Worker-level procedures. Thresholds like τ_C(for diagnoses) and τ_W(for equipment activation) ensure that actions are taken only when warranted by sufficient evidence. Parameters like ϵ_C add stability by requiring a grace period before a diagnostic decision can be reversed, reducing volatility in borderline cases.

Consider a case in which emergency room severity reaches 0.8. If the trauma team has a threshold $\theta = 0.7$, it activates immediately. Cardiology, with a higher threshold of 0.85, does not engage. Meanwhile, the laboratory, operating under a relaxed threshold of $\theta = 0.6 + \epsilon_W$, also activates. This behavior is captured by the group activation rule:

$$\text{activate}(team) \iff \mathsf{case\_severity} \geq \theta_{team} - \epsilon_{type}$$

Topic adoption follows a similar pattern. A new treatment protocol may emerge upon detection of a number of similar cases. If it satisfies minimum similarity ($\sigma_{\min} = 0.4$) and recurrence ($c_{\text{stability}} = 0.5$) requirements, and its similarity to existing treatments exceeds 0.7, the system integrates it into formal guidelines. These dynamics are governed by:

$$\begin{aligned} \text{adopt} &\iff \text{sim}(new, existing) \in [\sigma_{\min}, \sigma_{\max}] \\ \text{persist} &\iff \mathsf{recurrence\_rate} \geq c_{\text{stability}} \end{aligned}$$

Different hospital types require different parameter configurations. In trauma centers, high responsiveness is essential; hence, a large $\delta_S = 0.3$ and a low $\tau_W = 0.5$. Research hospitals balance supervisory and clinical decision-making, with weights such as $w_S = 0.4, w_C = 0.4$. In contrast, routine care clinics prioritize stability, with a low $\delta_S = 0.1$ and a conservative diagnosis threshold of $\tau_C = 0.8$.

At runtime, TB-CSPN supports continuous feedback-driven adaptation. For example, hospital systems monitor patient outcomes over time. Consultant agents analyze these trends and suggest parameter changes. A Supervisor (a medical director) evaluates the suggestions and authorizes updates. Once approved, new parameters are propagated to all relevant departments. This process is formalized through the following equations:

$$\begin{aligned} \text{new\_threshold} &= \text{current} + \lambda \cdot (\text{success\_rate} - \text{target}) \\ \text{update\_rate} &= f(\text{case\_volume}, \text{urgency}) \end{aligned}$$

Here, $\lambda = 0.1$ represents a conservative adjustment rate, preventing overreaction to short-term fluctuations. The function $f()$ modulates update frequency based on factors such as case volume and urgency, ensuring system adaptation to shifting conditions.

E Proof of concept implementation

To support the practical viability of the TB-CSPN framework, we developed a lightweight proof of concept (PoC) implementation centered on the financial decision-making scenario in Sect. 8.3. The prototype demonstrates token propagation across the three communication layers—Surface, Observation, and Computation—and models semantic coordination among three functional agents. A Consultant agent analyzes textual inputs (e.g., news headlines) and enriches them with topic distributions. A Supervisor agent evaluates whether the token content meets topic-specific thresholds and, if so, issues actionable directives. A Worker agent executes a domain-specific task (portfolio reallocation) in response to supervisor directives.

The implementation is written in Python, using modular components to allow easy extension or substitution of agents. For example, the Worker agent can be replaced by a full-fledged optimization agent based on G-learning, as envisioned in ongoing thesis work associated with this project. Each token is assigned a unique identifier and dynamically updated metadata indicating its position in the semantic coordination flow (e.g., layer=observation). Execution traces provide transparency and reproducibility of the agent interactions. An excerpt from a typical run is shown below:

```
[Input] Processing: Tech sector sees surge amid AI breakthroughs
[Consultant] Token created.
[Trace] Received by Consultant | ID=...
   | Layer=observation | Topics={'AI_stocks': 0.9, ...}
[Supervisor] Evaluating token...
[Trace] Supervisor Evaluation | ID=...
   | Layer=surface | Topics={...}
[Supervisor] Directive triggered.
[Trace] Issued by Supervisor | ID=...
   | Layer=surface | Topics={'AI_stocks': 1.0}
[Worker] Executing portfolio reallocation.
[Trace] Executed by Worker | ID=...
   | Layer=computation | Topics={'AI_stocks': 1.0}

[Input] Processing: Uncertainty rises after unexpected Fed decision
[Consultant] Token created.
...
[Supervisor] No directive issued.
[Computation] No task sent to Worker agent.
```

The codebase is organized to facilitate the inclusion of real-world large language models (LLMs) for topic analysis (e.g., OpenAI or HuggingFace), integration of asynchronous execution pipelines, and persistent token logging.

This implementation confirms that TB-CSPN's semantics-driven coordination logic, threshold-based transitions, and modular agent architecture are not only theoretically sound but also practically executable in a minimal yet extensible system.

Acknowledgements
The authors would like to acknowledge the use of DeepL Translator, DeepL Write and ChatGPT in the preparation of this paper. These tools assisted in enhancing the language quality and translating certain sections originally written in German or Italian. Naturally, the content remains a faithful reflection of the authors' original research and intellectual contributions.

Author contributions
All authors listed have made equal, substantial, direct, and intellectual contributions to the work.

Funding
Open Access funding enabled and organized by Projekt DEAL. Remo Pareschi has been funded by the European Union—NextGenerationEU under the Italian Ministry of University and Research (MUR) National Innovation Ecosystem grant ECS00000041-VITALITY—CUP E13C22001060006.

Data availability
All relevant data are included in the article. The sample datasets used in this study can be found in the attached Appendices A–D. A repository containing the proof-of-concept implementation described in Appendix E is available at https://github.com/Aribertus/tb-cspn-poc.

Materials availability

Not applicable.

Code availability

Not applicable.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
All authors have approved the article for publication.

Competing interests
The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential Conflict of interest.

Received: 15 April 2025 / Accepted: 24 June 2025
Published online: 03 July 2025

References

1. Pareschi R. Beyond human and machine: an architecture and methodology guideline for centaurian design. Science. 2024;6:71. https://doi.org/10.3390/sci6040071.
2. Borghoff UM, Bottoni P, Pareschi R. A system-theoretical multi-agent approach to human-computer interaction In: Arencibia AQ, Affenzeller M, Moreno-Díaz R. editors Computer Aided Systems Theory - EUROCAST 2024 - 19th International Conference, Las Palmas de Gran Canaria, Spain, February 25 - March 1, 2024, Revised Selected Papers, Part I, Vol. 15172 of LNCS 23–32, Springer, 2024. https://doi.org/10.1007/978-3-031-82949-9_3.
3. Borghoff UM, Bottoni P, Pareschi R. Human-artificial interaction in the age of agentic AI: a system-theoretical approach. Front Hum Dyn. 2025;7:1579166. https://doi.org/10.3389/fhumd.2025.1579166.
4. Blei DM, Ng AY, Jordan MI. Latent Dirichlet allocation. J Mach Learn Res. 2003;3:993–1022.
5. Blei DM. Probabilistic topic models. Commun ACM. 2012;55:77–84. https://doi.org/10.1145/2133806.2133826.
6. Abdelrazek A, Eid Y, Gawish E, Medhat W, Hassan A. Topic modeling algorithms and applications: a survey. Inf Syst. 2023;112: 102131. https://doi.org/10.1016/j.is.2022.102131.
7. Rossetti M, Pareschi R, Stella F, Fontana FA. Integrating concepts and knowledge in large content networks. New Gener Comput. 2014;32:309–30. https://doi.org/10.1007/S00354-014-0407-4.
8. Pinelli N, Pareschi R. Social news aggregations and the bitcoin: mapping their correlation through the three dimensions of content, sentiment and time In: Tonelli R, Ortu M. editors Proceedings of the 5th International Workshop on Emerging Trends in Software Engineering for Blockchain, WETSEB@ICSE 2022, Pittsburgh, Pennsylvania, 19 May 2022 33–38 (ACM, 2022). https://doi.org/10.1145/3528226.3528379.
9. Murata T. Petri nets: properties, analysis and applications. Proc IEEE. 1989;77:541–80. https://doi.org/10.1109/5.24143.
10. Reisig W. Elements of distributed algorithms: modeling and analysis with Petri nets. Berlin: Springer; 2013.
11. Jensen K. Coloured Petri Nets. Berlin: Springer; 1997.
12. Wooldridge MJ. Introduction to multiagent systems. Hoboken: Wiley; 2002.
13. Weiss G, editor. Multiagent Systems. 2nd ed. Cambridge: MIT Press; 2013.
14. Milner R. Communicating and mobile systems - the Pi-calculus. Cambridge: Cambridge University Press; 1999.
15. Sangiorgi D, Walker D. The Pi-Calculus: A Theory of Mobile Processes. Cambridge: Cambridge University Press; 2001. https://doi.org/10.1017/CBO9780511606432.
16. Nitzl C, Cyran A, Krstanovic S, Borghoff UM. The use of artificial intelligence in military intelligence: An experimental investigation of added value in the analysis process. Front Hum Dyn. 2025;7:1540450. https://doi.org/10.3389/fhumd.2025.1540450.
17. Fehling, R. A concept of hierarchical petri nets with building blocks In: Rozenberg G. editors Advances in Petri Nets 1993, Vol. 674 of LNCS 148–168, Springer, 1991. https://doi.org/10.1007/3-540-56689-9_43.
18. Agerwala T, Flynn M. Comments on capabilities, limitations and "correctness" of petri nets. SIGARCH Comput Archit News. 1973;2:81–6. https://doi.org/10.1145/633642.803973.
19. Montanari U, Rossi F. Contextual nets. Acta Inf. 1995;32:545–96. https://doi.org/10.1007/BF01178907.
20. Borghoff UM, Pareschi R, Fontana FA, Formato F. Constraint-based protocols for distributed problem solving. Sci Comput Program. 1998;30:201–25. https://doi.org/10.1016/S0167-6423(97)00011-7.
21. Dorri A, Kanhere SS, Jurdak R. Multi-agent systems: a survey. IEEE Access. 2018;6:28573–93. https://doi.org/10.1109/ACCESS.2018.2831228.
22. Horling B, Lesser V. A survey of multi-agent organizational paradigms. Knowl Eng Rev. 2004;19:281–316.
23. Alur R, Henzinger T, Kupferman O. Alternating-time temporal logic. J ACM. 2002;49:672–713. https://doi.org/10.1145/585265.585270.
24. Alechina N, Logan B, Nguyen HN, Raimondi F. Model-checking for resource-bounded ATL with production and consumption of resources. J Comput Syst Sci. 2017;88:126–44. https://doi.org/10.1016/J.JCSS.2017.03.008.
25. Mogavero F, Murano A, Perelli G, Vardi M. Reasoning about strategies: on the satisfiability problem. *Log. Meth. Comp. Sci.* 2017;13. https://doi.org/10.23638/LMCS-13(1:9)2017.
26. Lu I, Huang J, Lee W. An emotion-driven and topic-aware dialogue framework for human-robot interaction. Adv Robotics. 2024;38:267–81. https://doi.org/10.1080/01691864.2023.2297902.
27. Wang S, et al. TopicKS: Topic-driven knowledge selection for knowledge-grounded dialogue generation In: Ko H, Hansen JHL editors. 23rd Annual Conference of the International Speech Communication Association, Interspeech 2022, Incheon, Korea, September 18-22, 2022 1121–1125, ISCA, 2022. https://doi.org/10.21437/INTERSPEECH.2022-11188.
28. Hanteer O, Rossi L. An innovative way to model twitter topic-driven interactions using multiplex networks. Front Big Data. 2019;2:9. https://doi.org/10.3389/FDATA.2019.00009.
29. Singh A, Ehtesham A, Kumar S, Khoei TT. Enhancing AI systems with agentic workflows patterns in large language model (ed.aiiot2024.org) IEEE World AI IoT Congress, AIIoT 2024, Seattle, WA, USA, May 29-31, 2024 527–532, IEEE, 2024. https://doi.org/10.1109/AIIOT61789.2024.10578990.
30. Ghisellini R, Pareschi R, Pedroni M, Raggi GB. Recommending actionable strategies: A semantic approach to integrating analytical frameworks with decision heuristics. *Information* 2025;16. https://www.mdpi.com/2078-2489/16/3/192.
31. Capitanelli A, Mastrogiovanni F. A framework for neurosymbolic robot action planning using large language models. Front Neurorobot. 2024;18:1342786. https://doi.org/10.3389/FNBOT.2024.1342786.
32. Tenenbaum JB, Kemp C, Griffiths TL, Goodman ND. How to grow a mind: statistics, structure, and abstraction. Science. 2011;331:1279–85. https://doi.org/10.1126/science.1192788.
33. Seeber I, et al. Machines as teammates: a research agenda on AI in team collaboration. Inf Manag. 2020;57: 103174. https://doi.org/10.1016/j.im.2019.103174.
34. Giudici M, Liguori F, Tocchetti A, Brambilla M. Unveiling human-ai interaction and subjective perceptions about artificial intelligent agents In: Stefanidis K. et al. editors. Web Engineering - 24th International Conference, ICWE 2024, Tampere,

Finland, June 17-20, 2024, Proceedings, Vol. 14629 of LNCS 414–418, Springer, 2024. https://doi.org/10.1007/978-3-031-62362-2_36.

35. Kim J, Im I. Anthropomorphic response: understanding interactions between humans and artificial intelligence agents. Comput Hum Behav. 2023;139: 107512. https://doi.org/10.1016/J.CHB.2022.107512.
36. Ahmadi N, Hammond T. Recognizing and responding to human emotions: A survey of artificial emotional intelligence for cooperative social human-machine interactions. In: Hammond T, Hogan H. editors. IEEE Frontiers in Education Conference, FIE 2023, College Station, TX, USA, October 18-21, 2023 1–5 (IEEE, 2023). https://doi.org/10.1109/FIE58773.2023.10343328.
37. Zhang H, Li J, Liu X, Dong C. Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection. Fut Generat Comput Syst. 2021;122:130–43. https://doi.org/10.1016/J.FUTURE.2021.03.024.
38. LangChain. Langgraph. https://langchain-ai.github.io/langgraph/ (2025). Accessed: 2025-03-15.
39. LangChain. Agentic rag. https://blog.langchain.dev/agentic-rag-with-langgraph/ 2023. Accessed: 2025-03-15.
40. LangChain. Reflexion agent. https://blog.langchain.dev/reflection-agents/ 2023. Accessed: 2025-03-15.
41. Hu S, Lu C, Clune J. Automated design of agentic systems 2025. https://arxiv.org/abs/2408.08435. arXiv:2408.08435.
42. Xu B et al. Rewoo: Decoupling reasoning from observations for efficient augmented language models. *CoRR* 2023. arXiv:2305.18323.
43. Van Der Aalst WMP. The application of petri nets to workflow management. J Circ Syst Comput. 1998;08:21–66. https://doi.org/10.1142/S0218126698000043.
44. Ahn Y, Lee S, Shim J, Park J. Retrieval-augmented response generation for knowledge-grounded conversation in the wild. IEEE Access. 2022;10:131374–85. https://doi.org/10.1109/ACCESS.2022.3228964.
45. Zhang R, et al. Interactive AI with retrieval-augmented generation for next generation networking. IEEE Netw. 2024;38:414–24. https://doi.org/10.1109/MNET.2024.3401159.

Publisher's Note