# Human-artificial interaction in the age of agentic AI: a system-theoretical approach

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

[1]Department of Computer Science, University of the Bundeswehr Munich, Neubiberg, Germany,
[2]Department of Computer Science, Sapienza University of Rome, Rome, Italy, [3]Stake Lab, Università
degli Studi del Molise, Campobasso, Italy

This paper presents a novel perspective on human-computer interaction (HCI), framing it as a dynamic interplay between human and computational agents within a networked system. Going beyond traditional interface-based approaches, we emphasize the importance of coordination and communication among heterogeneous agents with different capabilities, roles, and goals. The paper distinguishes between Multi-Agent Systems (MAS)—where agents maintain autonomy through structured cooperation—and Centaurian systems, which integrate human and AI capabilities for unified decision making. To formalize these interactions, we introduce a framework for communication spaces, structured into surface, observation, and computation layers, ensuring seamless integration between MAS and Centaurian architectures, where colored Petri nets effectively represent structured Centaurian systems and high-level reconfigurable networks address the dynamic nature of MAS. We recognize that elements such as task recommendation, feedback loops, and natural language interfaces are common in contemporary adaptive HCI. What distinguishes our framework is not the introduction of these elements *per se*, but the synthesis of architectural principles that systematically accommodate both autonomy-preserving and integration-seeking configurations within a shared formal foundation. Our research has practical applications in autonomous robotics, human-in-the-loop decision making, and AI-driven cognitive architectures, and provides a foundation for next-generation hybrid intelligence systems that balance structured coordination with emergent behavior.

## 1 Introduction

Agentic AI systems—capable of iterative planning, autonomous task decomposition, and continuous learning—are rapidly reshaping the landscape of human-computer interaction (HCI). Recent advances in Large Language Models (LLMs) and advanced conversational agents have revitalized the field of multi-agent systems, whose roots in Artificial Intelligence predate the current rise of generative AI. Historically, multi-agent systems have relied on agents with relatively limited capabilities; however, the emergence of powerful, conversational LLMs greatly expands the range of possible multi-agent interactions. In this new paradigm, humans themselves can participate as full-fledged agents, thanks to their innate conversational and decision-making abilities. The result is a complex, interconnected ecosystem (Hopf et al., 2024).

Simultaneously, novel and more direct forms of human-AI integration are emerging (Pareschi, 2024; Saghafian and Idan, 2024), in which advanced AI cognition is harnessed to create "Centaurian" intelligence—a seamless fusion of human and machine-driven capabilities.

These developments push beyond traditional HCI in two distinct directions that mirror fundamental patterns found in biological systems (Borghoff et al., 1995; von Bertalanffy, 1968; Miller, 1978). Multi-agent systems, like biological ecosystems, emphasize how collections of autonomous entities can coordinate effectively while preserving distinct boundaries and roles. In contrast, "Centaurian" systems pursue deeper integration—analogous to the symbiotic relationships in nature—fusing human and artificial competencies in tightly knit partnerships that often blur the lines between human decision making and AI-driven processes. Living systems theory (Maturana and Varela, 1980) helps us understand how both approaches must address a core challenge: maintaining system identity through regulated boundaries and feedback loops (Wiener, 1948), whether in loosely coupled collectives or tightly integrated hybrid intelligences.

From an ethical and regulatory perspective, the blurring of boundaries between human and artificial agents raises critical issues of accountability and transparency. As hybrid systems increasingly share decision-making authority between human and AI components, especially in domains such as healthcare, defense, or finance, it becomes essential to ensure traceable paths of action and define clear loci of control. By preserving observable communication layers and explicit transitions, our architectural model supports auditability and human oversight even in deeply integrated Centaurian configurations. While our goal is primarily analytical rather than normative, we recognize that formal clarity in modeling is a prerequisite for ethical deployment.

Herbert Simon's influential work on cognitive architectures (Simon, 1996) provides additional insight into how such human-AI integration might be structured. His tripartite model—comprising an external interface, a coding mechanism, and an internal processing system—was originally conceived to explain human problem-solving but extends naturally to hybrid systems. When human abilities merge with artificial intelligence, the resulting configuration can evolve from traditional tool use (*Homo Faber*) to a more profound integration (*Centaurus Faber*) where each component may be instantiated by either human cognition or AI subsystems (Pareschi, 2024). This perspective suggests how system boundaries might be organized to support both autonomous operation and deep integration.

The framework proposed in this paper serves a dual purpose: first, as a **design methodology** for structuring hybrid human-AI systems, and second, as an **analytical lens** for understanding their coordination dynamics. While not prescriptive in a regulatory sense, it offers a normative foundation for evaluating the degree and quality of integration between agents. Its modular structure also allows it to be adapted for formal verification and implementation in real-world systems.

To address these architectural challenges, we introduce a formal framework based on Petri nets (Petri, 1962; Murata, 1989) that can model both multi-agent and Centaurian paradigms while supporting the key requirements suggested by living systems theory and Simon's architecture: clear boundaries, regulated interactions, and adaptive feedback loops. Unlike graph-based agent frameworks

(Wooldridge, 2002; Weiss, 2013), which prioritize topological relationships but struggle with dynamic concurrency, Petri nets offer native support for parallel processes and state transitions via token semantics, critical for hybrid human-AI systems. Colored Petri nets (Jensen, 1995), in particular, allow us to encode heterogeneous data, agent roles, and interaction semantics—key to modeling learning, adaptation, and reasoning in hybrid agent configurations.

Our approach, grounded in rigorous process modeling, provides a unified way to specify how heterogeneous agents—human and artificial—can coordinate their activities while either maintaining autonomy or achieving deeper integration as needed.

The remainder of this paper is organized as follows. Section 2 analyzes two fundamental paradigms in human-machine collaboration—the multi-agent approach and the Centaurian one—in light of the recent resurgence of agentic AI. Section 3 provides foundational background on Petri nets and their extensions. Section 4 introduces the concept of communication spaces as a unifying framework for these paradigms, grounded in formal architectures and coordination mechanisms. Sections 5 and 6 illustrate these ideas with two use cases, demonstrating how communication spaces can be applied in practical HCI scenarios. Section 7 discusses related work, and Section 8 concludes with implications and future directions.

# 2  Paradigms in human-AI integration

Agentic AI systems have reinvigorated two distinct approaches to human-AI collaboration: multi-agent architectures and Centaurian integration. This section examines these paradigms in detail, highlighting the distinct challenges each presents for system design and coordination.

Indeed, although both paradigms involve interactions between intelligent entities, they represent fundamentally different approaches to human-artificial collaboration (Pareschi, 2024). Multi-agent systems maintain distinct boundaries between components while enabling complex interactions, much like natural ecosystems. In contrast, Centaurian systems mirror the tight integration seen in biological organisms, where components merge functionally to create new capabilities (Saghafian and Idan, 2024).

## 2.1  Architectural differences

The primary architectural distinction lies in how these systems organize and maintain their identity (Pareschi, 2024).

Multi-agent systems (MAS) emphasize functional independence, where agents—human or artificial—operate as distinct entities with decision-making capabilities and dynamic collaboration. This design has allowed MAS implementations to thrive even with relatively simple AI agents, focusing on coordination rather than deep integration.

Centaurian systems, by contrast, create unified composite entities where human and artificial components become functionally interdependent. Theoretical foundations for such systems trace back to Licklider (1960) and Engelbart (1962), but

their practical feasibility has emerged only recently with generative AI systems capable of complementing human cognitive functions.

A key concept in both systems is that of *communication spaces*—regions of interaction where information exchange and coordination occur. In biological systems, these are signaling pathways or neural networks; in human-machine systems, they manifest as protocols, interfaces, and shared representational spaces (Bradshaw et al., 2011). These spaces enable "joint activity", characterized by inter-predictability, common ground, and directability.

## 2.2 Integration paradigms, operational dynamics, and convergence points

Each paradigm embodies different principles from living systems theory: In *MAS*, collaboration occurs through well-defined protocols between autonomous agents, resembling distinct organisms in an ecosystem. Effectiveness emerges from collective behavior while maintaining individual identities. In *Centaurian systems*, integration blurs the boundaries between human and artificial components, forming functionally unified entities similar to biological organisms.

The ecosystem versus organism analogy extends to operational dynamics:

- **Identity maintenance**: *MAS* preserve distinct agent identities while allowing interaction while, *Centaurian systems* create new composite identities through integration.
- **Adaptation mechanisms**: *MAS* adapt via reconfiguration of agent relationships, while *Centaurian systems* evolve through internal transformation of integrated components.
- **System boundaries**: *MAS* maintain clear boundaries with well-defined interfaces, while *Centaurian systems* develop permeable boundaries for deep functional integration.

Despite these distinctions, some scenarios blur the boundaries between paradigms, especially in adaptive complex systems. Consider large action models (LAMs), where human operators provide training feedback to artificial systems. Depending on the context, these interactions can manifest as multi-agent collaboration or Centaurian integration. Such convergence points suggest that while the paradigms are conceptually distinct, they can complement each other in practice.

## 2.3 Implications for system design

Viewing these paradigms through the lens of living systems theory has key design implications: The *MAS* design focus is on preserving agent autonomy, developing protocols for effective coordination, and ensuring system resilience through independent agents. *Centaurian systems*, on the other hand, have a design focus on emphasizing functional integration mechanisms, developing shared representational spaces, and encouraging emergent capabilities.

Section 4 will introduce communication spaces as a unifying framework supporting both paradigms, thus enabling both the loose coupling of MAS and the tight integration of Centaurian systems while preserving their core attributes.

## 2.4 Situating human-machine collaboration in the context of agentic AI

Advances in agentic AI heighten the relevance of multi-agent and Centaurian systems. Unlike traditional AI chatbots, agentic AI systems engage in sophisticated reasoning and iterative planning to solve complex, multi-step problems autonomously. These systems follow a four-step cycle of *perception*, *reasoning*, *action*, and *learning*, continuously improving over time.

Our framework provides a theoretical foundation for integrating agentic AI into human-machine collaboration. It is intended to be used primarily for system design, analytical modeling, and normative assessment. The multi-agent paradigm aligns with contexts where agentic AI must maintain autonomy while coordinating with human and artificial agents via defined protocols—such as in distributed customer service or collaborative software development.

Conversely, the Centaurian paradigm is critical for cases that require deep integration with human operators, such as healthcare or video analytics, where AI seamlessly blends with human expertise and decision making. The tight coupling characteristic of Centaurian systems enables fluid interactions for complex tasks that require human judgment alongside AI-driven analysis.
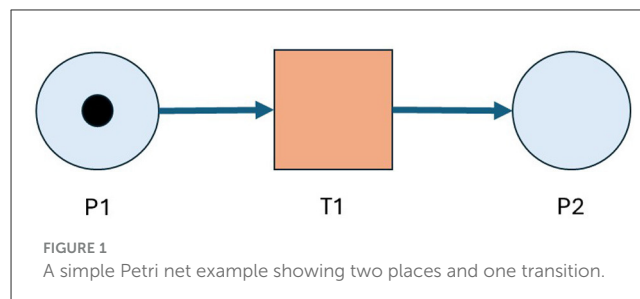
As agentic AI systems grow more sophisticated, the interplay between these paradigms becomes more dynamic. For instance, an enterprise AI system might operate independently (multi-agent mode) for routine tasks while tightly integrating with human experts (Centaurian mode) for complex decisions. Our framework supports the understanding and design of such interactions.

Additionally, the "data flywheel" effect—continuous learning and adaptation—aligns with our model's emphasis on system evolution. Learning occurs both in the improvement of independent agents (multi-agent paradigm) and in the refinement of human-AI integration patterns (Centaurian paradigm), demonstrating how our framework accommodates various aspects of agentic AI development.

## 3 Technical background: Petri nets

To provide a rigorous foundation for modeling the complex interactions in both multi-agent and Centaurian systems, we employ Petri nets as our core formal framework. Petri nets present several advantages for our purpose: they offer a clear graphical representation while maintaining mathematical precision, they naturally capture concurrent processes and synchronization requirements, and they can be extended to handle sophisticated data types and conditions through their colored variants. These properties make them especially well-suited for modeling interactions among heterogeneous agents, whether in loosely coupled multi-agent configurations or tightly integrated Centaurian systems. This section introduces the key concepts of Petri nets and their extensions, which will serve as the formal basis for our communication spaces framework.

Petri nets are a well-established formalism for describing and analyzing the flow of information and control in concurrent systems. Originally introduced by Petri (1962), they have evolved into a family of models widely applied in computer science, systems engineering, and workflow management. A Petri net consists of *places*, *transitions*, and *arcs* connecting places to transitions (or vice versa), with the system's state captured by *tokens* that reside in places. As transitions fire, tokens move among places, thus modeling the progress of a process or distributed computation (Murata, 1989).



FIGURE 1
A simple Petri net example showing two places and one transition.

## 3.1 Basic concepts and advantages

- **Graphical clarity**: One of the key strengths of Petri nets is their visual nature. Processes, resource constraints, and synchronization points are represented in a diagram, offering an intuitive view of concurrent interactions. This makes Petri nets well-suited for communicating system designs to both technical and non-technical stakeholders.
- **Explicit concurrency**: Petri nets explicitly capture concurrency by allowing multiple transitions to fire independently if they have sufficient tokens in their input places. This feature provides a natural way to represent multi-agent or multi-component systems where actions may proceed in parallel.
- **Formal analysis tools**: A rich ecosystem of analysis techniques (e.g., reachability, liveness, boundedness) and software tools (e.g., colored Petri nets Tools[1]) exists for verifying properties of Petri net models (Jensen, 1995). This is particularly useful for systems requiring rigorous guarantees, such as safety-critical applications or complex human-AI workflows.
- **Modularity and extensibility**: Petri nets provide a modular framework that can be extended to various specialized forms (e.g., timed, stochastic, colored), allowing the designer to incorporate specific features such as timing constraints, probabilistic behavior, or typed tokens.

## 3.2 Colored Petri nets and typed tokens

While classical Petri nets use indistinguishable tokens to mark places, *colored* Petri nets extend this model by assigning *types* or *colors* to tokens. Places become typed containers, and transitions can incorporate guard functions that enable more complex behaviors (Jensen, 1995). This extension allows designers to encode complex data structures and interaction protocols directly in the net, significantly increasing expressiveness. In the context of hybrid agent systems, colored Petri nets are especially advantageous for:

- **Heterogeneous agents**: Different types of tokens can represent messages, tasks, or capabilities unique to human and synthetic agents.

- **State and context tracking**: Color sets can capture the internal state of agents or system contexts, allowing transitions to fire only when certain conditions (guards) on those states are met.
- **Protocol definition and enforcement**: Complex interaction sequences, including the dynamic creation and distribution of tasks, can be succinctly specified and analyzed.
- **Integration with communication spaces**: As we will see in Section 4, communication spaces—surface, observation, and computation—can be mapped onto distinct parts of a colored Petri net, leveraging types to separate or coordinate different communication and processing domains.

## 3.3 Illustrative example

Figure 1 presents a simplified Petri net modeling a small concurrent process. Places are shown as circles, transitions as rectangles, and arcs indicate the flow of tokens.

In a colored Petri net version (Figure 2), tokens carry additional data types that reflect the roles or tasks assigned to different agents.

## 3.4 Relevance to hybrid agent systems

Petri nets, and particularly their colored variants, are ideally suited to represent the parallel and event-driven nature of interactions among humans and AI components. Their token-based semantics naturally handles concurrency, while the ability to extend tokens with data types makes them powerful enough to capture the complexities of heterogeneous, hybrid agent systems.

As we will see in the next section, Petri nets' properties—particularly their ability to model concurrent processes, handle heterogeneous data types, and support both loose and tight coupling—make them ideal for implementing the communication spaces framework, which addresses the coordination challenges identified in our discussion of multi-agent and Centaurian paradigms.

# 4 Communication spaces and agent architectures

The contrasting requirements of multi-agent and Centaurian paradigms—maintaining clear boundaries in one case while enabling deep integration in the other—call for a unified framework
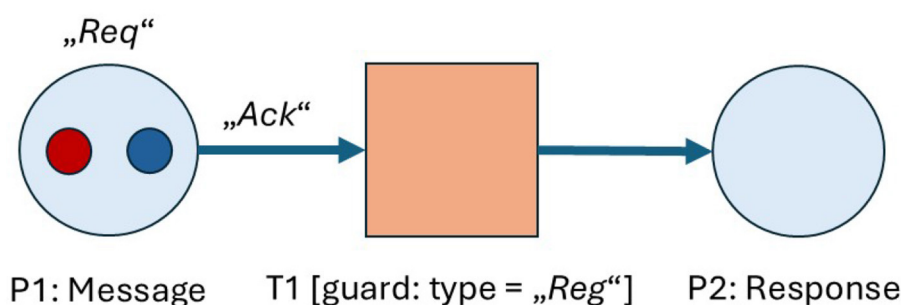
---

1   https://cpnide.org/

FIGURE 2
A colored Petri net where tokens carry types or data, allowing conditional transitions.

that can support both approaches. While Petri nets provide a strong formal foundation for modeling processes and synchronization, we need to extend their capabilities to capture the complexities of human-AI collaboration fully. We introduce *communication spaces* as this unifying framework, building on Petri nets while adding specialized structures for handling varied communication styles, heterogeneous capabilities, and adaptive protocols.

## 4.1 Foundations and rationale

Petri nets excel at modeling token flows among well-defined places, but in hybrid human-AI systems, we face additional challenges. Agents differ widely in perceiving, processing, and sharing information: humans communicate through natural language, AI systems through structured data, and physical devices through sensor signals. Moreover, these communication patterns must adapt dynamically as the system shifts between loose coupling (characteristic of multi-agent systems) and tight integration (needed in Centaurian configurations). These requirements call for a framework that can accommodate asymmetric interaction capabilities while supporting both autonomous operation and deep cognitive fusion.

### A mini-example of heterogeneous agents

Consider a team as shown in Figure 3 with a human operator, an LLM-based conversational agent, and a swarm of simple robotic drones. The human and LLM exchange strategic decisions via text-based conversation, while the drones communicate real-time state updates in a lightweight sensor-data format. Additionally, the human observes a video feed of drone activity. Each of these "channels" imposes different synchronization rules, data formats, and reliability guarantees—precisely the nuances communication spaces help formalize.

## 4.2 Core ideas of communication spaces

Communication spaces group interactions into three conceptual layers (Borghoff et al., 2024a): *surface*, *observation*, and *computation*. Whether in a multi-agent (MAS) or a Centaurian system, each space encapsulates a coherent set of interaction rules and constraints. The formation of these spaces typically involves either point-to-point or broadcast communication[2], or communication through the environment[3]. Several examples illustrate this multi-agent organization of communication spaces:

- In Mobile *Ad-hoc* NETworks (MANETs), mobile devices communicate with one another via wireless links without relying on an underlying infrastructure: each device acts as both an endpoint and a router forwarding messages to devices within radio range. Software support for MANETs typically presents a coordination layer that sends specific messages to individual agents to guarantee the execution of cooperative processes while considering the need for connectivity continuance. In Bottoni et al. (2006), a formal model of the coordination mechanisms was given in terms of High-Level Petri nets.
- Management of transportation networks requires maintaining complete information on the whereabouts of vehicles and the state of roads to be mapped on the topology of the transportation network (see Ciancia et al. (2018) for a concrete example). Then, the overall state of the transportation system is given by the information of which vehicle is at which node of the network, information which has to be maintained consistently for each vehicle.

---

2  Point-to-point or broadcast communication often uses message passing as a crucial concept in many system interactions, especially in distributed systems, concurrent programming, or event-driven architectures like ours. It allows different components (agents, large action models, etc.) to communicate without internal knowledge of the other system (decoupling), making it easy to add more components to a messaging-coupled system (scalability). Even different messaging protocols can be easily implemented and exchanged spontaneously (flexibility), with some protocols allowing asynchronous interaction in which the sender and receiver do not have to work simultaneously (asynchrony). Xiao et al. (2020) presents a comprehensive overview and analysis of the most advanced blockchain consensus protocols based on message passing.

3  Shared memory architectures can form such an environmental common ground in the Bradshaw et al. (2011) sense.
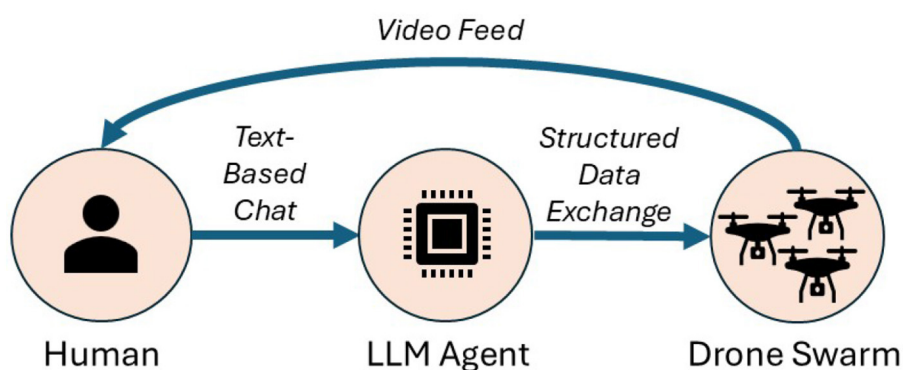
**FIGURE 3**
Conceptual illustration of a human operator, an LLM agent, and a swarm of drones, each using distinct communication channels (text-based, structured data, and video feed).
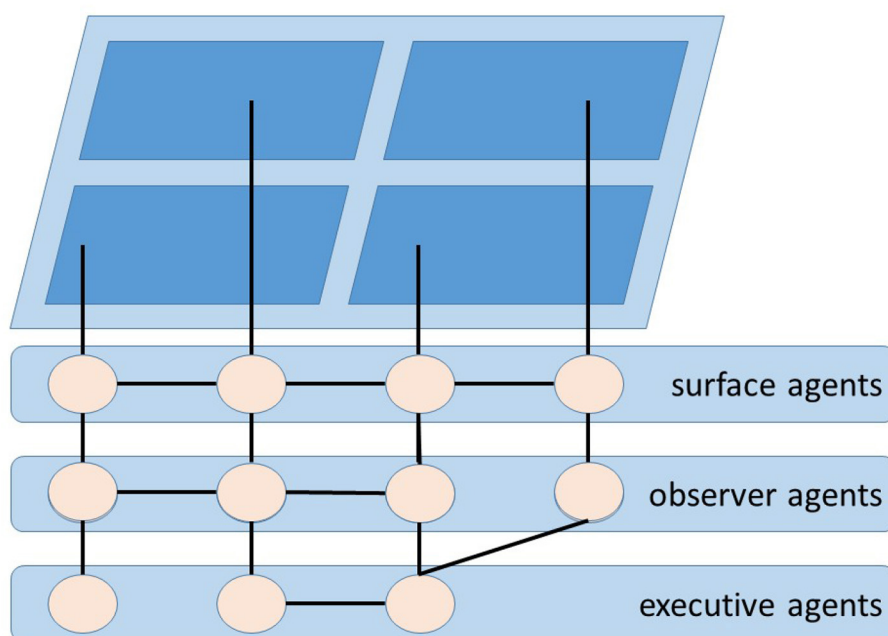


**FIGURE 4**
A layered architecture for HCI.

- In the classical work by Dorigo et al. (2006), a computational model was derived from how ants communicate information on trails to reach food sources by marking them through pheromones recognized by other ants. The computational model was, therefore, based on associating weights to paths to a result and strengthening those associated with a successful one.
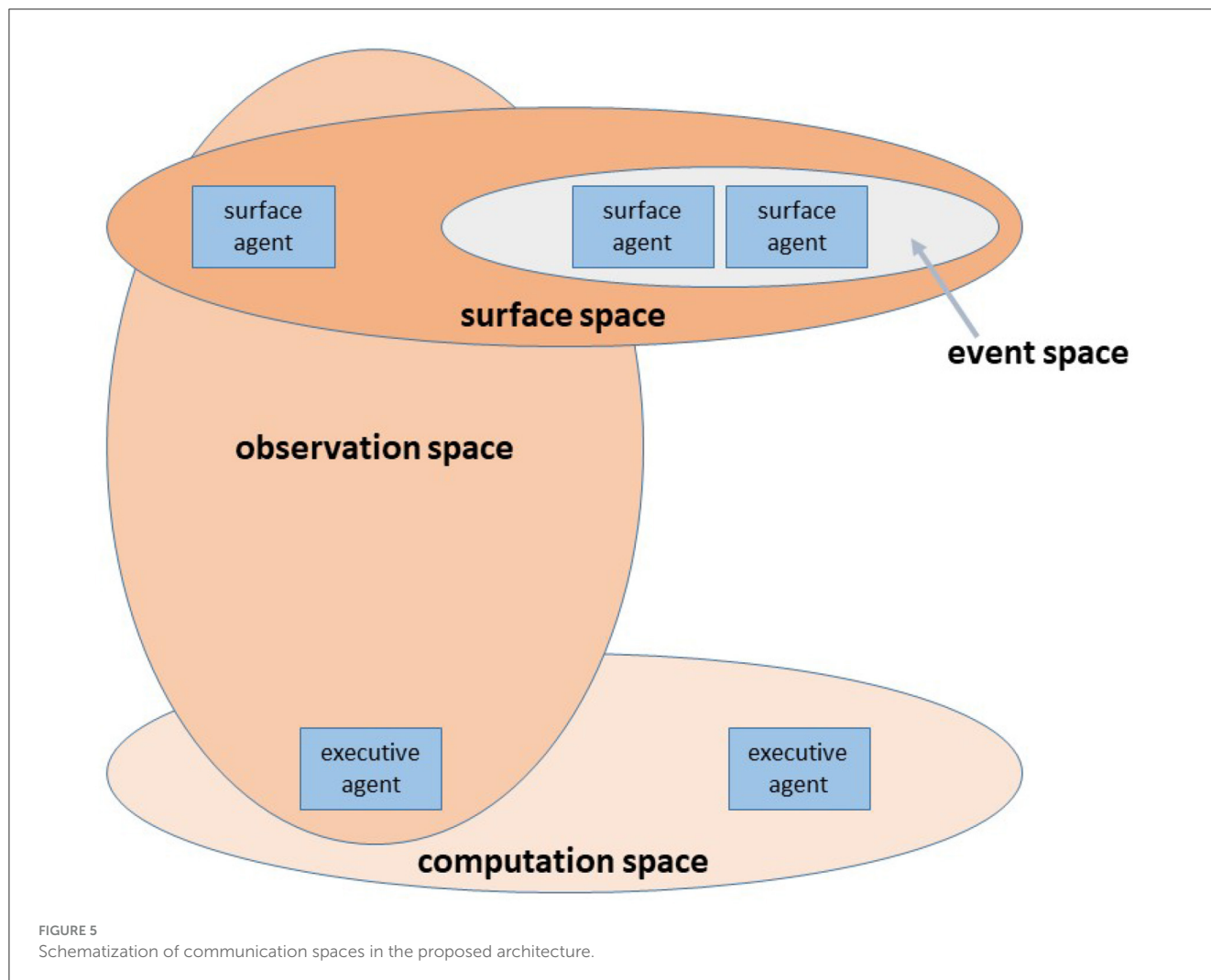
Figure 4 shows a layered architecture similar to the *Model-View-Controller* (MVC) paradigm, where *executive agents* implement the actual computations, *observer agents* realize a bridge between the computations, thereby managing the formatting of data for presentation and decoding user interactive commands and data entry, and *surface agents* (or mediators) are responsible for maintaining the presentation of the overall state to the users, thereby managing the materialization of the formatted data and the presentation layout, and providing support for capturing user interactions.

Note that in this architecture, the actions performed by the executive agents, and not just their results, can be subject to observation. Vertical structuring is also shown, relating surface agents and executive agents through an observation space (see Figure 5).

## Surface space

This space mediates all contact with the outside environment—user interfaces, sensors, and external APIs. In MAS, surface space typically involves message-passing protocols or event listeners. In Centaurian systems, it can reflect a direct blending of human sensory input and AI-driven data capture.

**FIGURE 5**
Schematization of communication spaces in the proposed architecture.

### Observation space

Bridging the surface interface with internal processing, observation space handles message transformations, routing, and light coordination. In MAS, protocols here ensure agents remain autonomous yet cooperative. In Centaurian systems, observation may feature continuous feedback loops that unify human perception with AI analysis.

### Computation space

Serving as the "core" of the system, the computational space makes decisions, allocates resources, and generates final outputs. MAS solutions often involve multiple autonomous modules, each coordinating a portion of the computation. In contrast, Centaurian architectures could fuse human insight with AI algorithms in a shared decision-making environment.

### Flows and feedback

Information travels *vertically* through these spaces—from raw input at the surface to internal processing and back. There may also be *horizontal* flows where agents collaborate within a given space.

Feedback loops enable adaptive behavior, such as refining protocols or redistributing tasks in real time based on performance.

Each of these spaces can be seen as managed by an associated *group-agent*, facilitating distributed problem-solving among a group of users within a networked system or application. The structure of a group-agent is schematized in Figure 6, where the ID and ST compartment are common to all agents. Then, the COMPOSITION compartment is specific to a group-agent, while other agents have specific additional members for defining their state, and are endowed with a set of behaviors.

Group-agents are relevant in collaborative systems where multiple users and agents work together to solve complex problems or tasks and proper delivery of messages is crucial to ensure that each participant is correctly informed of the overall progress toward task achievement. A group-agent can be created at any time with respect to a topic and its register behavior regulates composition modification through a protocol which respects the localization constraints. The following pseudo-code provides an abstract specification of the construction of group-agents for communication spaces.

```
register(self, agent) {
        if agent is concerned with self.topic then
```
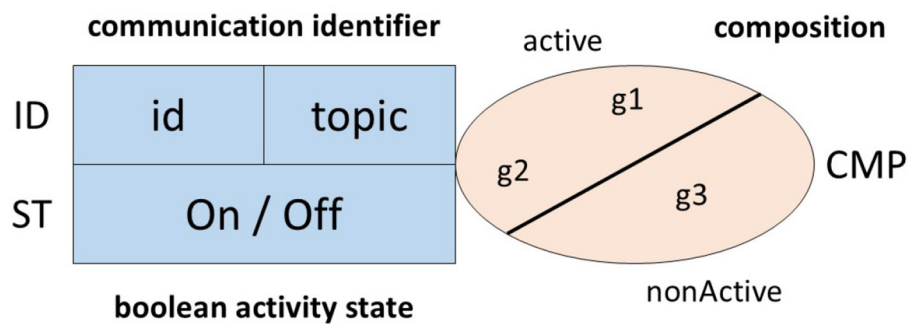
**FIGURE 6**
A schematization of the basic structure of a group agent.

```
    if agent.ST = ON then
        self.CMP[active].Add(agent)
    else self.CMP[nonActive].Add(agent)
    }
}
```

Once created, a group-agent has to guarantee that any message from a member of the group can be read by all other (active) members. In particular, a message sent by an agent registered to a group and concerning the group's topic is delivered to all active agents registered to that group, as for the group's `deliver` behavior, to be invoked on message generation.

```
deliver(m){
    if self.ST == ON and self.isConcernedWith(m.topic) then
        for each agent in self.CMP[active]
            agent.addToBuffer(m)

}
```

Similarly, deregistration can occur, when a component is no longer relevant to a group topic, or due to an explicit decision by the coordination mechanism. Deregistration can only occur for active agents.

```
deregister(self, agent) {
    if agent ∈ self.CMP[active] then
        self.CMP[active].Remove(agent)
}
```

The information about the activity status of an agent registered to a group is updated through the `switch` behavior

```
switchCMP(self, agent) {
    if agent.ST == OFF and agent ∈ self.CMP[active]
then
        self.CMP[active].Remove(agent)
        self.CMP[nonactive].Add(agent)
    if agent.ST == ON and agent ∈ self.
CMP[nonactive] then
        self.CMP[nonactive].Remove(agent)
        self.CMP[active].Add(agent)
}
```

## 4.3 Implementing communication spaces

Each of the three spaces can be mapped onto subsets of Petri net places (or subnets) with typed tokens indicating relevant data, tasks, or events. This approach partitions a net into self-contained "zones", each governed by specific interaction rules and constraints.

### Multi-agent architecture

In a multi-agent system, *surface agents* manage user interactions and detect events from the environment, acting as the system's interface with external inputs. *Observer agents* ensure smooth communication by transforming and routing messages, enabling coherent collaboration among autonomous components. Meanwhile, *executive agents* operate within the computational space, executing specialized tasks and maintaining a partial system state to support overall functionality.

A key benefit is clear boundary maintenance and scalability: each agent type can be added or removed with minimal impact on the others, provided communication protocols remain consistent.

### Centaurian architecture

In Centaurian systems, a *unified interface layer* merges human and AI sensory input within the surface space, creating a seamless interaction point. A *shared observation layer* acts as a bridge between human and AI representations, facilitating smooth cognitive fusion. Meanwhile, a *hybrid processing layer* integrates human reasoning with algorithmic methods, allowing for emergent collective problem-solving.

Such architectures thrive on adaptability, where boundaries are permeable and agents can fluidly exchange roles or data to optimize performance.

## 4.4 Formal representation with colored Petri nets: communication space Petri net

We extend colored Petri nets to create a *Communication Space Petri net*, where places are partitioned into *surface*, *observation*, and *computation* categories. Tokens carry types or "colors"
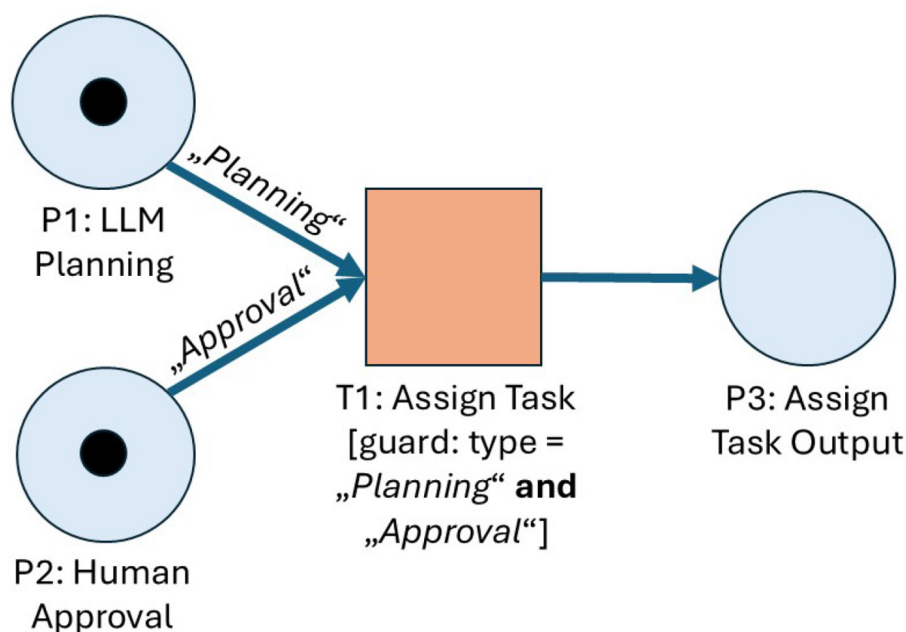
**FIGURE 7**
A schematic colored Petri net snippet illustrating how a transition fires only if both an LLM planning token and a human approval token are present, capturing Centaurian synergy.

corresponding to data formats or message semantics relevant to each space. Transition guards incorporate rules for agent-specific capabilities (e.g., a color set may denote "human text command" vs. "drone sensor data").

This enriched formalism preserves the tractability of standard Petri nets while accommodating the specialized interactions of heterogeneous agents. For example, transitions in the observation space can fire only when tokens of type "parsed command" are present, ensuring that raw data never bypasses the necessary transformation steps.

### Example snippet

Imagine a transition as shown in Figure 7 in the computation space labeled *Assign Task*, which fires only when it receives both a "planning token" from the LLM agent and an "approval token" from the human. Such a rule encapsulates the synergy sought in a Centaurian design, formalizing how human sign-off triggers an AI-based planning process.

### 4.5  Implementation considerations

Implementing communication spaces requires careful attention to the following aspects: *State management* ensures consistent updates across distributed agents or fused human-AI modules, especially in time-sensitive environments. *Resource allocation* deals with balancing computational load, bandwidth, and potential real-time constraints. *Protocol adaptation* supports switching between agent autonomy (MAS) and deeper integration (Centaurian) based on changing system goals or unexpected conditions.

A system may operate predominantly in MAS mode for routine tasks, yet temporarily shift to a more Centaurian-like strategy when it encounters novel or ambiguous conditions. This capacity to blend paradigms on demand underscores the flexibility of communication spaces.

The benefits can be summarized as follows: By demarcating surface, observation, and computation spaces within a unified Petri net model, designers gain a clear blueprint for structuring agent interactions. This approach enables them to capitalize on the strengths of both multi-agent and Centaurian paradigms—maintaining well-defined boundaries when necessary, but also allowing for deep integration and emergent capabilities where advantageous. The next sections illustrate how communication spaces operate in real-world scenarios, further emphasizing their role in building robust human-AI systems.

## 5  Use case 1: multi-agent interaction with satellite and swarm robots

This use case demonstrates how our theoretical framework accommodates both multi-agent and Centaurian paradigms within a complex system. While predominantly exhibiting multi-agent characteristics through its distributed architecture, the system also incorporates Centaurian elements in specific human-AI interaction points. Figure 8 illustrates the data flow in an experiment with a semi-centralized coordinated swarm of robots, using both "rigid" optimization algorithms and "flexible" intervention through a large language model (LLM). This setup encapsulates a true multi-agent
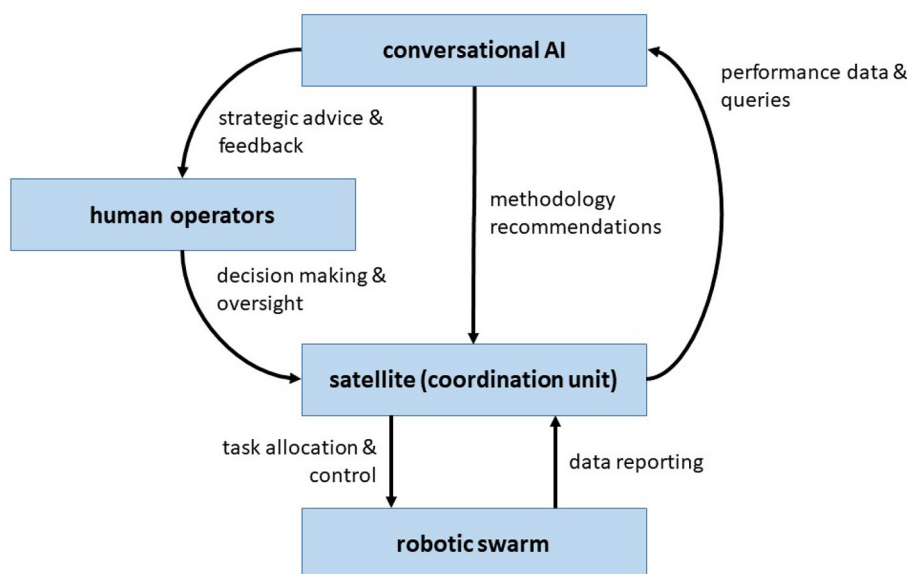
**FIGURE 8**
Interaction of human operators, conversational AI, satellite, and a robotic swarm.

HCI interaction, integrating human operators, conversational AI, the satellite control unit, and swarm robots.

This use case extends the application of a semi-centralized control strategy, originally enhanced with blockchain technology for security and transparency (Carovilla et al., 2023). The multi-agent paradigm is evident in how distinct entities—the satellite control unit, swarm robots, LLM, and human operators—maintain their functional independence while collaborating through well-defined protocols. However, the system also exhibits Centaurian characteristics at certain points of interaction, particularly at the human-LLM interface. The overall information and decision flow, described in detail in the following sections, is shown in Figure 9.

## 5.1 System overview

The system exemplifies the multi-agent paradigm through its composition of distinct, autonomous components: a satellite control unit, a swarm of robotic agents, an LLM, and human operators. Each agent maintains its own decision-making capabilities while participating in the collective process. The satellite provides centralized coordination and processes the data collected by the swarm, while the LLM offers strategic advice on algorithm selection and parameter tuning. The human operators are not just supervisors, but autonomous agents who monitor these interactions and intervene to guide the system through complex decision processes.

## 5.2 Integration with LLM and human operators

The integration layer demonstrates a hybrid approach that combines both paradigms. Following the multi-agent paradigm, the
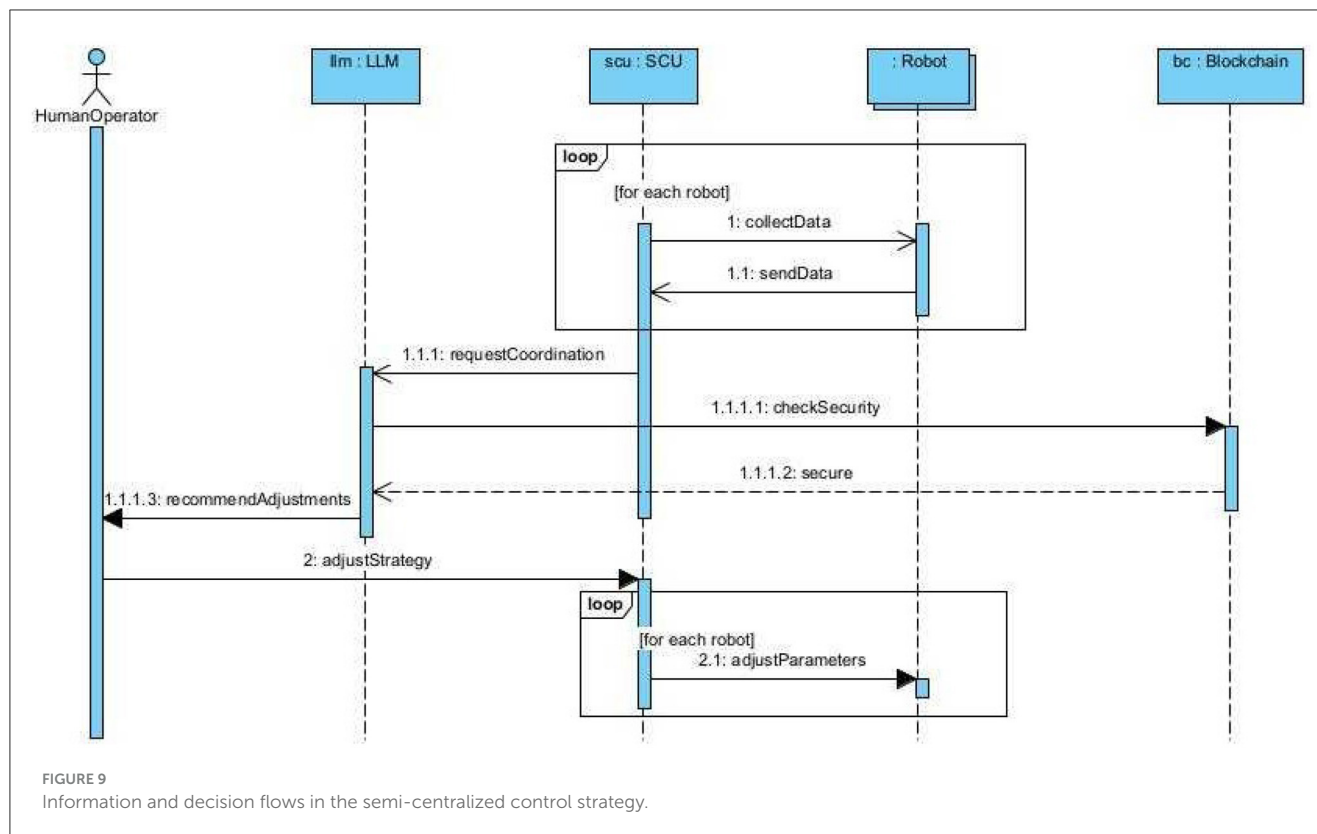
LLM interacts with the blockchain and control algorithms as an independent agent, providing a flexible intelligence layer. However, the interaction between human operators and the LLM exhibits Centaurian characteristics, creating a tightly coupled decision unit where human judgment and AI recommendations are seamlessly integrated. This dual-nature system remains robust and adaptive through both the independent operation of its components and the synergistic human-AI integration points.

## 5.3 Practical applications

The practical applications show how the two paradigms complement each other. For path optimization and adaptive problem solving, the system primarily operates in a multi-agent mode, with the LLM acting as an independent advisor. However, when human operators review and implement suggestions, the interaction temporarily shifts to a Centaurian mode, creating a tightly integrated human-AI decision unit. For example, in path planning, the LLM can suggest changes to curve parameters based on detected obstacles or changes in terrain, maintaining its agent independence, while human operators review these suggestions through a Centaurian-style deep integration process.

## 5.4 Human-agent collaboration

The collaboration model in this system demonstrates the value of supporting both paradigms. While the overall architecture follows multi-agent principles, with human operators functioning as independent agents within the system, their interaction with the LLM and control systems often exhibits Centaurian characteristics. This flexible approach allows the system to leverage both the independence and coordination capabilities of the multi-agent

**FIGURE 9**
Information and decision flows in the semi-centralized control strategy.

paradigm and the deep integration benefits of the Centaurian approach. The result is a robust system that protects against the risks of automation while actively leveraging the unique strengths of both humans and machines, pushing the boundaries of what technology can achieve through this dual-paradigm approach.

## 6 Use case 2: large action models (LAMs) through feedback loops on HCI interactions

While our first use case emphasized the multi-agent paradigm, this second use case demonstrates a stronger inclination toward the Centaurian approach, while still maintaining some multi-agent characteristics. Figure 10 shows the framework of RABBIT TECH's Large Action Model (LAM)[4], which integrates advanced computational agents to effectively model and predict human actions on computer applications. The system exemplifies the Centaurian paradigm's emphasis on tight integration between human and artificial components, while incorporating multi-agent elements in its distributed architecture.

This use case illustrates the transformative potential of LAMs in creating a new paradigm for human-computer interaction. Unlike traditional multi-agent systems where components maintain clear boundaries, Rabbit LAM demonstrates the Centaurian ideal of deep functional integration, continuously learning from user input and refining its predictive capabilities. It embodies

a sophisticated feedback loop system where the boundaries between human and artificial intelligence become increasingly fluid, reflecting the core principle of Centaurian systems where components form an integrated whole rather than remaining distinct entities.
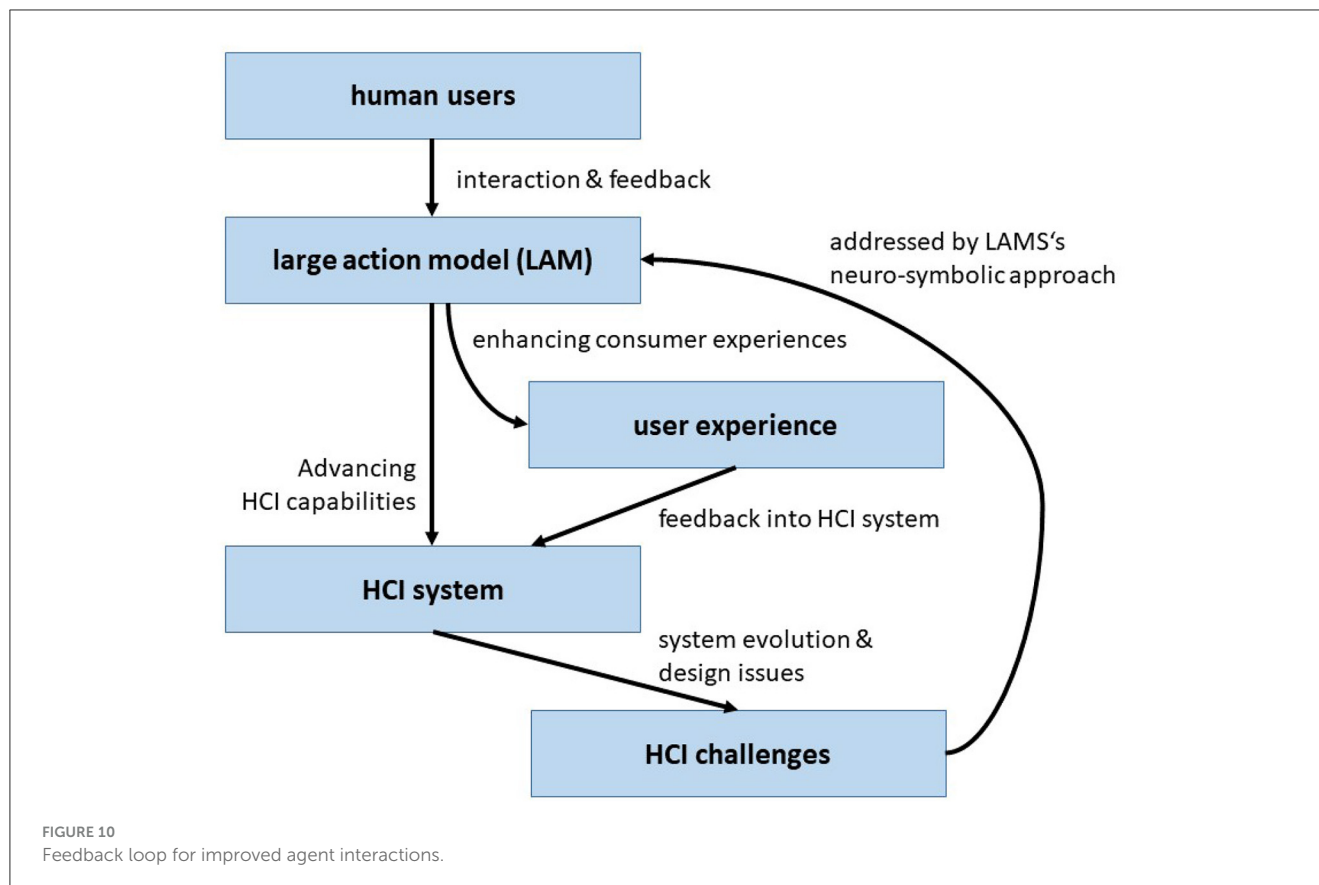
### 6.1 System overview

The LAM framework in the RABBIT API exemplifies the Centaurian paradigm through its core components: the LAM node, human users, and the HCI system, along with representations of the challenges this integration addresses and the user experience. Rather than maintaining strict boundaries between these components, they evolve together through their interactions. The LAM node, while functioning as a computational agent, achieves a level of integration with human users that goes beyond traditional multi-agent relationships, using neuro-symbolic programming to create a unified decision-making entity.

### 6.2 Neuro-symbolic integration

The neuro-symbolic approach of LAM represents a key Centaurian characteristic: the seamless integration of different processing paradigms. By combining neural network capabilities for pattern recognition with symbolic AI for rule-based processing, LAM demonstrates how apparently distinct computational approaches can be unified into a coherent whole, much like the human-AI integration central to Centaurian

---

4   https://www.rabbit.tech/research

**FIGURE 10**
Feedback loop for improved agent interactions.

systems. This integration enables the execution of complex tasks across multiple applications with high accuracy and minimal latency.

For example, consider a LAM processing an ambiguous user command such as "Schedule a meeting with the team soon." The LLM (stochastic) might probabilistically interpret "soon" as "within 2 days" based on training data, while the symbolic scheduler (deterministic) enforces a rule that "soon" must map to a fixed 24-h window. To resolve this, the observation space (see Figure 5) mediates between the two layers: the LLM's proposal is marked as a "suggestion" token, which triggers a human-AI negotiation protocol in the computational space. Here, the system either accepts the LLM's output with a confidence threshold ($> 80\%$) or defaults to the symbolic rule, logging the conflict for later refinement. This ensures adaptability while maintaining traceability—a key requirement for Centaurian systems in high-stakes domains.

## 6.3 Human-agent interaction

The human-LAM interaction epitomizes the Centaurian ideal of symbiotic integration. Through a natural language-based user interface that processes spoken or typed commands, the system achieves a level of interaction where the boundaries between human intent and machine execution become increasingly blurred. The feedback system allows LAM to learn from each interaction, adapting its models to better match user preferences—a hallmark of

Centaurian systems where components evolve together rather than merely coordinating their actions.

## 6.4 Practical applications and challenges

LAM's approach to task management and problem solving demonstrates the advantages of Centaurian integration over mere multi-agent coordination. By anticipating user needs and providing proactive solutions, it shows how deep human-AI integration can surpass the capabilities of systems in which components merely coordinate their actions. However, this tight integration poses unique challenges, including ensuring the transparency of AI decisions and preserving privacy—challenges that are characteristic of Centaurian systems, where the boundaries between human and artificial components become blurred.

## 6.5 Key takeaways

The feedback loops in LAM exemplify the evolutionary nature of Centaurian systems. Unlike multi-agent systems where components maintain fixed identities and interact through stable protocols, LAM's feedback loops create a dynamic system where human and artificial components grow together, continuously refining their interaction patterns. This approach to HCI demonstrates how Centaurian principles can create more intuitive and responsive computing environments, while

still benefiting from certain multi-agent characteristics in its distributed architecture.

# 7  Related work

The implementation of agent interactions has been explored through various computational approaches. Basic interaction protocols have been formalized for distributed computation (Aspnes and Ruppert, 2009), while more complex patterns have been developed for knowledge combination (Andreoli et al., 1994; Borghoff and Schlichter, 1996) and distributed problem-solving (Borghoff et al., 1998). Recent work has focused on higher-level interactions, particularly in intention recognition (de Greef et al., 2007; Inga et al., 2023), which is becoming increasingly important as artificial agents grow more sophisticated in their ability to understand and respond to human goals.

Guamán et al. (2021) presents an innovative unsupervised clustering theory using self-organizing maps to classify *MVC* patterns based on software quality metrics in a continuous interaction process. Their goal is to identify quality features that determine the similarity of MVC applications without architectural bias. Lawless and Sofge (2021) use interdependence theory to determine whether convergence has a positive or negative effect in a competition between human and artificial agents, as in system dynamics models.

Mobile applications, as well as satellite and swarm robotics, have grown significantly in recent years, with architectural patterns playing a critical role in their success. To ensure the correct implementation of an MVC pattern, Dobrean and Diosan (2022) propose an automated technique that analyzes and detects architectural issues using data from *Software Development Kits* (SDKs) specifically targeted at mobile code bases. To enhance coordination and security in swarm robotic systems, Carovilla et al. (2023) introduce a semi-centralized framework that integrates blockchain technology, featuring a centralized control unit that coordinates the swarm, while blockchain technology ensures secure and decentralized data storage and communication. Kumar and Choppella (2023) take the opposite approach, extending pattern descriptions with a *System of Systems* (SoS) model to apply interactive dynamics to an MVC pattern.

Using the *Internet of Things* (IoT) paradigm as a complex distributed system that shares a 3-tier architecture consisting of embedded nodes, gateways that connect an embedded network to the wider internet, and data services in servers or the cloud, Riliskis et al. (2015) propose a novel approach for programming applications across 3-tiers using a distributed extension of the MVC architecture. Also in the IoT context, with a focus on the gaming/sports arena, Chen (2025) sees potential for entertainment robots to recognize human posture through artificial intelligence. By detecting and monitoring users' movements in real time, these robots provide a personalized and interactive entertainment experience. Caci and Dhou (2020) also discuss the interplay between games with personality and artificial intelligence. They use the term "virtual human" to describe a computer program that simulates a human in some aspects.

With the ubiquity of AI applications, HCI research is increasingly integrating these approaches, demonstrating that AI and HCI are mutually beneficial when they collaborate (Panda and Roy, 2024). However, criticism and bad experiences have also been reported. For example, Choudhuri et al. (2024) state that there were no statistical differences in participants' productivity or self-efficacy when using ChatGPT compared to traditional resources. Instead, they found significantly higher frustration levels, identifying five distinct errors resulting from violations of human-AI interaction guidelines, leading to various (negative) consequences for the participants.

LLMs are increasingly being used in studies to investigate the interaction between humans and artificial agents. Borghoff et al. (2024b) explore human-artificial interaction with generative AIs in a software engineering project course. In the same domain, Nascimento et al. (2023) evaluates ChatGPT-generated code against developer-generated code to determine which tasks are better suited for engineers and which are better handled by AI. This could lead to more efficient interaction (e.g., AI as a tutor for SE developers) and provide new insights into innovative AI strategies that include the involvement of *humans-in-the-loop* to support the tasks of software engineering. Capitanelli and Mastrogiovanni (2024) demonstrate that LLMs can play an important role in planning actions in human-robot interactions.

Giudici et al. (2024) present and evaluate a method for analyzing user reactions to AI using a live-streaming platform where human streamers conduct interviews that are transmitted to a specially developed GPT voice interface using a crowd-based approach. Kim and Im (2023) study the attribution of human characteristics to artificial intelligence. They developed a tool to measure how users form anthropomorphic reactions to interactions with AI chatbots in a banking service setting. For a survey of *Artificial Emotional Intelligence* (AEI) for cooperative social human-machine interactions see Ahmadi and Hammond (2023).

Message Passing Neural Networks (MPNNs) (Papillon et al., 2023) are a special class of neural networks that use the aforementioned message-passing paradigm to capture interactions between entities (often represented as nodes in a graph). In MPNNs, nodes in a graph communicate with their neighbors through a series of message-passing steps. This interaction allows the network to learn representations by aggregating information from neighboring nodes; see also the RABBIT use case in Section 6.

A formal model of the global activity of a system of agents is provided by *concurrent game structures*, describing a situation where agents act according to individual *strategies* by which to select individual actions based on knowledge of the global state of the system. Reasoning on the possible evolutions of such systems leads to the definition of some variations of *Alternating-Time Temporal Logic* (see e.g., Alur et al., 2002; Alechina et al., 2017; Mogavero et al., 2017). In contrast, a logic that incorporates spatial constraints on a par with temporal ones has been recently proposed by Bottoni et al. (2024). In general, these models abstract the actual mechanisms through which information about states can be exchanged among agents and assume that all agents act rationally regarding their strategies. The level of non-determinism inherent to considering human agents would, therefore, require some adaptation of these logics in the direction

of identifying classes of agents that can follow several strategies at once.

Although beyond the scope of this paper, there exists a humanistic perspective (Knell and Rüther, 2024), as well as an ethical dimension, to human-artificial interaction. For the latter, the reader is referred to Kumar and Choudhury (2023b), Karlan (2023) and Nieto (2024). In some ways, "virtual superhuman" AI technologies could undermine the value of human achievements (Schaap et al., 2024) or even may pose an existential threat (Kumar and Choudhury, 2023a).

# 8  Conclusion and future work

We have outlined a framework that systematically considers how human and artificial agents interact within a networked ecosystem. By articulating HCI not just as human-computer interactions but as dynamic interactions among different agents within a flexible, multi-agent system, we have highlighted the potential for more complex and effective interactions. The application of our multi-agent model to two different use cases—satellite and swarm robotics, and Large Action Models (LAMs)—has not only validated our theoretical constructs, but also underscored the practical implications of such interactions.

In the first use case, we observed how a semi-centralized system, enabled by a large language model, could improve decision-making processes and adaptability through real-time data-driven adjustments. This setup not only ensures efficiency, but also preserves human oversight, which is critical for managing complex, dynamic environments.

The second use case involving LAMs further highlighted the potential of integrating advanced computational agents to predict and shape human-computer interactions. Using neuro-symbolic programming, the LAM framework demonstrated its ability to refine its operations based on continuous feedback from human interactions, promoting an adaptive and responsive HCI system.

We have thus established a foundational framework for the design and analysis of HCI systems that views these interactions as a complex interplay among multiple agents. This approach allows us to better understand and harness the collective capabilities of different agents, thereby enhancing the system's adaptability and resilience. Importantly, this perspective aligns with emerging technology paradigms that foster a deep understanding and integration of human agents. Indeed, our contribution can be seen as a systemic evolution in design approaches, emphasizing the front-end to enhance usability and effectiveness. This contrasts with, but complements, systemic approaches that prioritize the back-end for security and robustness, as illustrated by Salzano et al. (2023). Together, these perspectives synergistically improve the overall architecture of HCI systems.

The integration of human and artificial agents also raises critical ethical and regulatory considerations. Our framework's layered communication spaces–surface, observation, and computation– provide inherent auditability through explicit transition logs and token-based interaction traces, aligning with AI accountability guidelines (Floridi, 2019; EU, 2024). For high-stakes domains such as healthcare or defense, this enables clear attribution of decisions to human or AI agents, while preserving the fluidity of Centaurian collaboration. Future work will further formalize these mechanisms to ensure compliance with emerging standards for hybrid intelligence systems.

Ultimately, our study enriches the understanding of HCI as a dynamic and evolving field characterized by the complex yet symbiotic relationships between humans and machines. As we delve deeper into these interactions, the possibility of transforming human-computer interactions into more seamless and effective experiences becomes increasingly tangible. This promises a future in which the human and artificial dimensions are not merely aligned but fully integrated in a balanced and complementary manner. Such integration could pave the way for a new era of collective and hybrid intelligence— where humans and machines work closely together—potentially the most significant and immediate outcome of current rapid technological advances.

Future research will focus on improving the adaptability of MAS by addressing the challenge of dynamic team composition. Standard Petri nets, with their static structure, have difficulty accommodating scenarios where agents join or leave while maintaining consensus, as illustrated by the `register` and `deregister` (for active agents) pseudo-code in Section 4. Therefore, we plan to model agents as individual (colored) Petri nets with specialized input and output locations, allowing token-based message passing for seamless registration and deregistration. However, smooth coordination requires high-level reconfiguration mechanisms to dynamically manage agent lifecycles. While these improvements enhance the adaptability of MAS, Centaurian systems, with their stable human-AI integration, remain well-suited to Petri net formalization due to their reliance on fixed coordination structures.

Another important future research direction is the development of a hybrid approach that bridges MAS and Centaurian systems. While Petri nets effectively model structured Centaurian intelligence, where human and AI components collaborate through predefined coordination rules, they fall short in dynamic MAS scenarios that require self-organization. Future research will therefore integrate high-level reconfigurable networks to enable agent fluidity, while using communication spaces as an intermediate layer for seamless coordination between autonomous and tightly coupled agents. This three-tiered approach will enable dynamic transitions between structured Centaurian intelligence and decentralized MAS interactions. Given the increasing importance of Centaurian AI in human-computer collaboration, this hybrid framework will improve efficiency, adaptability, and interoperability.

An alternative set-theoretic approach abstracts agent relationships without rigid network structures, defining interactions through set membership and relational mappings. While this circumvents the limitations of Petri nets in dynamic scenarios, its implementation requires structured protocols for message flow and consistency. Our future research will therefore explore how *set-theoretic models* can complement Petri

nets to ensure both formal rigor and practical applicability in human-AI collaboration.

## Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

## Author contributions

UB: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. PB: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. RP: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

## Funding

## Acknowledgments

## Conflict of interest

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.

The author(s) declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

## Generative AI statement

The author(s) declare that Gen AI was used in the creation of this manuscript. We used the gratuitous (free) versions of DeepL Translator and DeepL Write as of 2025, and ChatGPT 4.0 to verify American English language, syntax and grammar.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

Ahmadi, N., and Hammond, T. (2023). "Recognizing and responding to human emotions: a survey of artificial emotional intelligence for cooperative social human-machine interactions," in *IEEE Frontiers in Education Conference, FIE 2023, College Station, TX, USA, October 18–21* (IEEE), 1–5. doi: 10.1109/FIE58773.2023.10343328

Alechina, N., Logan, B., Nguyen, H. N., and Raimondi, F. (2017). Model-checking for resource-bounded ATL with production and consumption of resources. *J. Comput. Syst. Sci.* 88, 126–144. doi: 10.1016/j.jcss.2017.03.008

Alur, R., Henzinger, T., and Kupferman, O. (2002). Alternating-time temporal logic. *J. ACM* 49, 672–713. doi: 10.1145/585265.585270

Andreoli, J., Borghoff, U. M., and Pareschi, R. (1994). "Constraint-based knowledge brokers," in *Proceedings of the First International Symposium on Parallel Symbolic Computation-PASCO'94, Hagenberg/Linz, Austria, September 26–28, volume 5 of Lecture Notes in Computing*, eds. H. Hong (Singapore: World Scientific), 1–11. doi: 10.1142/9789814533584

Aspnes, J., and Ruppert, E. (2009). "An introduction to population protocols," in *Middleware for Network Eccentric and Mobile Applications*, eds. B. Garbinato, H. Miranda, and L. E. T. Rodrigues (Springer: New York), 97–120. doi: 10.1007/978-3-540-89707-1_5

Borghoff, U. M., Bottoni, P., Mussio, P., and Pareschi, R. (1995). "Multi-agent coordination in living systems," in *Proceedings of the 1995 International Workshop on Biologically Inspired Evolutionary Systems, BIES 1995, Sony Corporation Conference Hall, Tokyo, Japan, May 30–31*, eds. C. Numaoka (Japan: Society for Fuzzy Theory and Systems (SOFT)), 93–100.

Borghoff, U. M., Bottoni, P., and Pareschi, R. (2024a). "A system-theoretical multi-agent approach to human-computer interaction," in *Computer Aided Systems Theory - EUROCAST 2024 - 19th International Conference, Las Palmas de Gran Canaria, Spain, February 25 - March 1*, Revised Selected Papers, Part I, number 15172 in Lecture

Notes in Computer Science, eds. A. Q. Arencibia, M. Affenzeller, and R. Moreno-Díaz (Springer: New York), 23–32. doi: 10.1007/978-3-031-82949-9_3

Borghoff, U. M., Bottoni, P., and Pareschi, R. (2025). Human-artificial interaction in the age of agentic AI: a system-theoretical approach. *arXiv Preprint, arXiv:2502.14000.* doi: 10.48550/arXiv.2502.14000

Borghoff, U. M., Minas, M., and Mönch, K. (2024b). "Automatic program assessment, grading and code generation: possible AI-support in a software development course," in *Artificial Intelligence and Soft Computing - 23rd International Conference, ICAISC 2024, Zakopane, Poland, June 16–20, Proceedings, Part III, number 15166 in Lecture Notes in Artificial Intelligence*, eds. L. Rutkowski et al. (Springer: New York), 39–51. doi: 10.1007/978-3-031-81596-6_4

Borghoff, U. M., Pareschi, R., Fontana, F. A., and Formato, F. (1998). Constraint-based protocols for distributed problem solving. *Sci. Comput. Program.* 30, 201–225. doi: 10.1016/S0167-6423(97)00011-7

Borghoff, U. M., and Schlichter, J. H. (1996). On combining the knowledge of heterogeneous information repositories. *J. Univers. Comput. Sci.* 2, 515–532. doi: 10.3217/jucs-002-07-0515

Bottoni, P., De Rosa, F., Hoffmann, K., and Mecella, M. (2006). Applying algebraic approaches for modeling workflows and their transformations in mobile networks. *Mob. Inf. Syst.* 2, 51–76. doi: 10.1155/2006/704187

Bottoni, P., Labella, A., and Perelli, G. (2024). "Strategies in spatio-temporal logics for multi-agent systems," in *Leveraging Applications of Formal Methods, Verification and Validation. REoCAS Colloquium in Honor of Rocco De Nicola - 12th International Symposium, ISoLA 2024, Crete, Greece, October 27–31, Proceedings, Part I, volume 15219 of Lecture Notes in Computer Science*, eds. T. Margaria, and B. Steffen (Springer: New York), 287–305. doi: 10.1007/978-3-031-737 09-1_18

Bradshaw, J. M., Feltovich, P. J., and Johnson, M. (2011). "Human-agent interaction," in *The Handbook of Human-Machine Interaction: A Human-Centered Design Approach*, eds. G. A. Boy (CRC Press), 283–300. doi: 10.1201/9781315557380-14

Caci, B., and Dhou, K. (2020). "The interplay between artificial intelligence and users' personalities: a new scenario for human-computer interaction in gaming," in *HCI International 2020 - Late Breaking Papers: Cognition, Learning and Games - 22nd HCI International Conference, HCII 2020, Copenhagen, Denmark, July 19-24, Proceedings, volume 12425 of Lecture Notes in Computer Science*, eds. C. Stephanidis et al. (Springer: New York), 619–630. doi: 10.1007/978-3-030-60128-7_45

Capitanelli, A., and Mastrogiovanni, F. (2024). A framework for neurosymbolic robot action planning using large language models. *Front. Neurorobotics*. 18:1342786. doi: 10.3389/fnbot.2024.1342786

Carovilla, A., Pareschi, R., and Salzano, F. (2023). "Integrating blockchain for enhanced coordination and security in semi-centralized robotic swarms," in *Proceeding IEEE International Workshop on Technologies for Defense and Security (TechDefense)* (Rome: IEEE), 95–99. doi: 10.1109/TechDefense59795.2023.10380842

Chen, X. (2025). Research on entertainment robots based on artificial intelligence interaction for human posture recognition and sports activity monitoring. *Entertain. Comput.* 52:100761. doi: 10.1016/j.entcom.2024.100761

Choudhuri, R., Liu, D., Steinmacher, I., Gerosa, M., and Sarma, A. (2024). "How far are we? the triumphs and trials of generative AI in learning software engineering," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 1–13. doi: 10.1145/3597503.3639201

Ciancia, V., Gilmore, S., Grilletti, G., Latella, D., Loreti, M., and Massink, M. (2018). Spatio-temporal model checking of vehicular movement in public transport systems. *Int. Jour. on Soft. Tools for Tech. Transfer* 29, 289–311. doi: 10.1007/s10009-018-0483-8

de Greef, T., van Dongen, K., Grootjen, M., and Lindenberg, J. (2007). "Augmenting cognition: reviewing the symbiotic relation between man and machine," in *Foundations of Augmented Cognition, Third International Conference, FAC 2007, Held as Part of HCI International 2007, Beijing, China, July 22-27, Proceedings, volume 4565 of LNCS*, eds. D. Schmorrow, and L. Reeves (Springer: New York), 439–448. doi: 10.1007/978-3-540-73216-7_51

Dobrean, D., and Diosan, L. (2022). Pathways for statically mining the model-view-controller software architecture on mobile applications. *Soft Comput.* 26, 10493–10511. doi: 10.1007/s00500-022-06908-0

Dorigo, M., Birattari, M., and Stutzle, T. (2006). Ant colony optimization. *IEEE Comput. Intell. Mag.* 1, 28–39. doi: 10.1109/CI-M.2006.248054

Engelbart, D. C. (1962). *Augmenting Human Intellect: A Conceptual Framework*. Technical Report AFOSR-3223, Stanford Research Institute, Menlo Park, CA. doi: 10.21236/AD0289565

EU. (2024). Regulation (EU) 2024/… of the European Parliament and of the Council on Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act). OJ L, 2024. Final text pending publication in the Official Journal of the European Union. Available online at: https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng (accessed May 1, 2025).

Floridi, L. (2019). Translating principles into practices of digital ethics. *Philos. Technol.* 32, 185–193. doi: 10.1007/s13347-019-00354-x

Giudici, M., Liguori, F., Tocchetti, A., and Brambilla, M. (2024). "Unveiling human-AI interaction and subjective perceptions about artificial intelligent agents," in *Web Engineering - 24th International Conference, ICWE 2024, Tampere, Finland, June 17–20, 2024, Proceedings, volume 14629 of Lecture Notes in Computer Science*, eds. K. Stefanidis et al. (Springer: New York), 414–418. doi: 10.1007/978-3-031-62362-2_36

Guamán, D., Delgado, S., and Pérez, J. (2021). Classifying model-view-controller software applications using self-organizing maps. *IEEE Access* 9, 45201–45229. doi: 10.1109/ACCESS.2021.3066348

Hopf, K., Nahr, N., Staake, T., and Lehner, F. (2024). The group mind of hybrid teams with humans and intelligent agents in knowledge-intense work. *J. Inf. Technol.* 40, 9–34. doi: 10.1177/02683962241296883

Inga, J., Ruess, M., Robens, J. H., Nelius, T., Rothfuß, S., Kille, S., et al. (2023). Human-machine symbiosis: a multivariate perspective for physically coupled human-machine systems. *Int. J. Hum. Comput. Stud.* 170:102926. doi: 10.1016/j.ijhcs.2022.102926

Jensen, K. (1995). *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, Volume 2 of Monographs in Theoretical Computer Science*. Springer-Verlag, Berlin, Heidelberg.

Karlan, B. (2023). Human achievement and artificial intelligence. *Ethics Inf. Technol.* 25:40. doi: 10.1007/s10676-023-09713-x

Kim, J., and Im, I. (2023). Anthropomorphic response: understanding interactions between humans and artificial intelligence agents. *Comput. Hum. Behav.* 139:107512. doi: 10.1016/j.chb.2022.107512

Knell, S., and Rüther, M. (2024). Artificial intelligence, superefficiency and the end of work: a humanistic perspective on meaning in life. *AI Ethics* 4, 363–373. doi: 10.1007/s43681-023-00273-w

Kumar, M., and Choppella, V. (2023). "Modeling model-view-controller (MVC) architecture pattern as system of systems (SOS) to aid comprehension and reasoning," in *Proceedings of the ACM Conference on Global Computing Education Vol 2, CompEd 2023, Hyderabad, India, December 5–9*, eds. V. Choppella, D. B. Phatak, A. Luxton-Reilly, and M. Craig (Hyderabad: ACM), 197. doi: 10.1145/3617650.3624934

Kumar, S., and Choudhury, S. (2023a). Humans, super humans, and super humanoids: debating stephen hawking's doomsday AI forecast. *AI Ethics* 3, 975–984. doi: 10.1007/s43681-022-00213-0

Kumar, S., and Choudhury, S. (2023b). Normative ethics, human rights, and artificial intelligence. *AI Ethics* 3, 441–450. doi: 10.1007/s43681-022-00170-8

Lawless, W. F., and Sofge, D. A. (2021). "Interdependence and vulnerability in systems: a review of theory for autonomous human-machine teams," in *Engineering Artificially Intelligent Systems - A Systems Engineering Approach to Realizing Synergistic Capabilities, volume 13000 of Lecture Notes in Computer Science*, eds. W. F. Lawless et al. (Springer: New York), 247–259. doi: 10.1007/978-3-030-89385-9_15

Licklider, J. (1960). Man-computer symbiosis. *IRE Trans. Hum. Factors Electron* 1, 4–11. doi: 10.1109/THFE2.1960.4503259

Maturana, H. R., and Varela, F. J. (1980). *Autopoiesis and Cognition: The Realization of the Living, Volume 42 of Boston Studies in the Philosophy of Science*. D. Reidel Publishing Company, Dordrecht, Holland.

Miller, J. G. (1978). *Living Systems*. McGraw-Hill: New York.

Mogavero, F., Murano, A., Perelli, G., and Vardi, M. (2017). Reasoning about strategies: on the satisfiability problem. *Log. Meth. Comp. Sci.* 13, 1–37. doi: 10.23638/LMCS-13(1:9)2017

Murata, T. (1989). Petri nets: properties, analysis and applications. *Proc. IEEE* 77, 541–580. doi: 10.1109/5.24143

Nascimento, N., Alencar, P., and Cowan, D. (2023). Comparing software developers with ChatGPT: an empirical investigation. *CoRR abs*/2305.11837. doi: 10.48550/arXiv.2305.11837

Nieto, B. F. (2024). "Toward a human-centered framework for trustworthy, safe and ethical generative artificial intelligence: a multi-level analysis of large language models social impact," in *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering, EASE 2024, Salerno, Italy, June 18–21* (ACM), 505–509. doi: 10.1145/3661167.3661177

Panda, S., and Roy, S. T. (2024). Reflections on emerging HCI-AI research. *AI Soc.* 39, 407–409. doi: 10.1007/s00146-022-01409-y

Papillon, M., Sanborn, S., Hajij, M., and Miolane, N. (2023). Architectures of topological deep learning: A survey of message-passing topological neural networks. *arXiv preprint arXiv:2304.10031*. doi: 10.48550/arXiv.2304.10031

Pareschi, R. (2024). Beyond human and machine: an architecture and methodology guideline for centaurian design. *Science* 6:71. doi: 10.3390/sci6040071

Petri, C. A. (1962). Kommunikation mit Automaten. Dissertation, Schriften des IIM 2, Rheinisch-Westfälisches Institut für Instrumentelle Mathematik an der Universität Bonn, Bonn.

Riliskis, L., Hong, J., and Levis, P. A. (2015). "Ravel: programming iot applications as distributed models, views, and controllers," in *Proceedings of the 2015 International Workshop on Internet of Things towards Applications, IoT-App 2015, Seoul, South Korea, November 1*, eds. C. Xu, P. Zhang, and S. Sigg (Seoul: ACM), 1–6. doi: 10.1145/2820975.2820977

Saghafian, S., and Idan, L. (2024). Effective generative AI: the human-algorithm centaur. *arXiv preprint arXiv:2406.10942*. doi: 10.2139/ssrn.4594780

Salzano, F., Pareschi, R., Marchesi, L., and Tonelli, R. (2023). "Blockchain-based information ecosystems," in *Proceedings of the Fifth Distributed Ledger Technology Workshop (DLT 2023), Bologna, Italy, May 25–26, volume 3460 of CEUR Workshop Proceedings*, eds. P. Mori, I. Visconti, and S. Bistarelli. Available online at: CEUR-WS.org

Schaap, G., van de Sande, Y., and Schraffenberger, H. (2024). "Outperformed by AI: interacting with superhuman AI changes the way we perceive ourselves," in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, CHI EA 2024, Honolulu, HI, USA, May 11–16*, (ACM), 1–7. doi: 10.1145/3613905.3650961

Simon, H. A. (1996). *The Sciences of the Artificial*. MIT Press, Cambridge, MA, 3 edition.

von Bertalanffy, L. (1968). *General System Theory: Foundations, Development, Applications*. George Braziller, New York.

Weiss, G. ed. (2013). *Multiagent Systems*. MIT Press, 2nd edition.

Wiener, N. (1948). *Cybernetics: Or Control and Communication in the Animal and the Machine*. MIT Press, Cambridge, MA.

Wooldridge, M. J. (2002). *Introduction to Multiagent Systems*. Wiley.

Xiao, Y., Zhang, N., Lou, W., and Hou, Y. T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Commun. Surv. Tutor.* 22, 1432–1465. doi: 10.1109/COMST.2020.2969706