# From Conversation to Orchestration: HCI Challenges and Opportunities in Interactive Multi-Agentic Systems

Sarah Schömbs*
School of Computing and Information Systems
University of Melbourne
Melbourne, VIC, Australia
sschombs@student.unimelb.edu.au

Yan Zhang*
School of Computing and Information Systems
University of Melbourne
Melbourne, VIC, Australia
yan.zhang.1@unimelb.edu.au

Jorge Goncalves
School of Computing and Information Systems
University of Melbourne
Melbourne, VIC, Australia
jorge.goncalves@unimelb.edu.au

Wafa Johal
School of Computing and Information Systems
University of Melbourne
Melbourne, VIC, Australia
wafa.johal@unimelb.edu.au

## Abstract

Recent advances in multi-agentic systems (e.g., AutoGen, OpenAI Agents) allow users to interact with a group of specialised AI agents rather than a single general-purpose agent. Despite the promise of this new paradigm, the HCI community has yet to fully examine the opportunities, risks, and user-centred challenges it introduces. We contribute to research on multi-agentic systems by exploring their architectures and key features through a human-centred lens. While literature and use cases are still emerging, we build on existing tools and frameworks available to developers to identify a set of overarching challenges, e.g., orchestration and conflict resolution, that can guide future research in HCI. We illustrate these challenges through examples, present potential design considerations, and provide research opportunities to spark interdisciplinary conversation. Our work lays the groundwork for future exploration and offers a research agenda focused on user-centred design in multi-agentic systems.

## CCS Concepts

• **Human-centered computing → HCI theory, concepts and models**.

## Keywords

agentic systems, multi-agentic system, human-agentic interaction, design, risk

**ACM Reference Format:**
Sarah Schömbs, Yan Zhang, Jorge Goncalves, and Wafa Johal. 2025. From Conversation to Orchestration: HCI Challenges and Opportunities in Interactive Multi-Agentic Systems. In *Proceedings of the 13th International Conference on Human-Agent Interaction (HAI '25), November 10–13, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3765766.3765795

*Both authors contributed equally to this research.


*HAI '25, November 10–13, 2025, Yokohama, Japan*

ACM ISBN 979-8-4007-2178-6/25/11
https://doi.org/10.1145/3765766.3765795

## 1 Introduction

Within just a few months of release, Large Language Models (LLMs) like ChatGPT have attracted millions of users [18, 44]. Since then, foundation models have rapidly evolved and have shifted the focus from AI as decision-making tools to **agentic** systems capable of serving general-purpose roles in everyday life, reviving interest in Agent-Oriented Programming [65] along the way. Recent software released, e.g., from Microsoft [74] and OpenAI [6], now go even further and introduce frameworks for multi-agentic systems that introduce the idea of users not only interacting with one AI agent, but handing over tasks to multiple AI agents. This paradigm shift in human-AI interaction, driven by advances in LLMs, represents a significant departure from the 'traditional' model of relying on a single, static AI agent. Instead, it re-imagines human-AI interaction as the gateway to multiple specialised AI agents, each designed to perform distinct roles within a collective system. In this model, individual agents are uniquely tailored and prompted to achieve specific goals, execute tasks and access tools (e.g., APIs). Importantly, these agents are not siloed; they can communicate, exchange information, and collaboratively solve problems, which contributes to emergent complexities that end-users must navigate. The idea of multi-agent systems (MAS) is not new and has traditionally been considered in fields such as robotics, for example, with the introduction of swarms [43, 67]. However, multi-**agentic** systems, backed by foundation models, are expected to be accessible to the general public. This raises important questions regarding their perceptual and behavioural effects on end-users, as well as the design challenges associated with creating interfaces that support effective interaction with such systems.

The implications of this paradigm extend far beyond efficiency and technical convenience. By structuring software around a network of interactive agents, multi-agentic systems open the door to greater personalisation and adaptability in user experiences. While current advances predominantly remain on the developer side, we are beginning to see early steps toward making these capabilities accessible to end-users. For example, recent work on Magentic UI [26] introduces a prototype chat user interface that allows end-users to delegate tasks to a multi-agentic team called Magentice-One [27]. Magentic-One [26] consists of a hierarchical "generalist" multi-agentic team with a

top-level supervisor agent, i.e., "orchestrator agent", and a set of specialised subagents (e.g., WebSurfer, Coder, FileSurfer) capable of completing open-ended general-purpose tasks (e.g., browsing the web or executing relevant Python code) in real-time. In this system architecture, the supervisor agent interprets the user's prompt, plans the task, delegates to the specialised agents, and monitors the task progress.

These developments hint at a near future in which multi-agentic systems are programmable and usable by a broader audience. While the opportunities are vast, this approach also introduces new challenges in terms of transparency and interpretability of multi-agentic systems, as non-technical end-users must navigate not only the outcomes but also the interactions between agents that produce them. Furthermore, ensuring human control over such systems, particularly when emergent behaviours lead to unexpected outcomes, raises critical questions about trust and risks. These dynamics highlight the need for an in-depth investigation into how multi-agentic systems reshape our relationship and interactions with AI from an end-user's perspective. Addressing these challenges is crucial to ensuring a responsible design and deployment of multi-agentic systems for the general public. For instance, creating mechanisms that allow users to easily trace and understand inter-agent communications will be essential for maintaining transparency and mitigating risks. Similarly, developing safeguards to prevent cascading errors or rogue behaviours among agents will be crucial to ensure reliability and user safety [16].

Importantly, multi-agentic systems can take on various architectures, ranging from a multi-body to a fully decentralised agent architecture. This paper focuses on the hierarchical architecture, a multi-agentic system design in which, in its simplest form, a supervisor agent acts as the central interface between the user and a set of specialised sub-agents, similar to Magentic-One [26]. Our rationale for focusing on this architecture is motivated by two key factors: First, it reduces the cognitive burden on the end-user by abstracting away the complexity of interacting with multiple autonomous agents. Instead of managing a distributed system directly, which can be cognitively demanding (see early works in swarm robotics, e.g., [39, 72]), a supervisor agent can plan, delegate, and coordinate tasks internally. Second, the hierarchical architecture is emerging as a popular architecture in current prototype implementations/applications [26] or is often proposed as the default structure in current frameworks [73]. Given its growing adoption and practical promise, it is essential to investigate the design challenges that arise specifically for end-users when interacting with systems built on this architecture. The central research question in this work is therefore:

> *What are the design challenges and opportunities for end-user interaction introduced by a hierarchical architecture in multi-agentic systems?*

To explore this research question, we examine alternative multi-agent architectures to contextualise and contrast the supervisor paradigm, as well as compare to current single-agentic systems. This comparative perspective allows us to more precisely articulate the design challenges (including trade-offs and constraints), and opportunities that define the hierarchical architecture.

We contribute to an important, timely and growing body of research on agentic systems. First, we explore agentic architectures and describe their principles and key features from a human-centred perspective. While literature and use cases are still sparse, we build on this knowledge and our expertise, and use the characteristics of the current tools already available for developers to identify a set of design challenges that can be used to drive future research in Human-Multi-Agent interaction. We connect these challenges to broader HCI challenges, illustrate them through examples and explore how they intersect with opportunities and risks. Recognising these risks, we advocate for proactively anticipating potential harms in the design of user interfaces for multi-agentic systems, following Chan et al. [17]. Our work addresses the pressing need for the HCI community to engage with the rapid evolution of single-LLMs to multi-agentic architectures through the end-user lens.

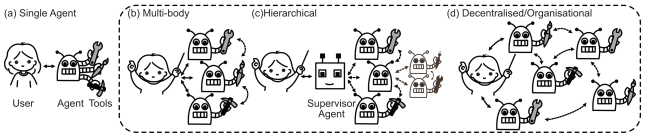


**Figure 1: Illustration of agentic system architectures.**

## 2 From Agent to Swarm

### 2.1 Agentic and System Architectures

Broadly defined, an **agent** can perceive its environment using sensors and act on it through effectors [57], and **agency** denotes "the exercise or manifestation of this capacity" [78]. With advancements in AI, agents now embody key properties that determine their interaction with humans, other agents, and the environment, such as autonomy [14] and social ability [14], reactivity, proactiveness, as well as additional traits typically associated with humans, including mentalistic reasoning [65], emotional capacity [9], and anthropomorphic features [42]. In line with Chan et al. [17], we define **agentic systems** as systems that include AI agents with a high degree of agency, characterised by underspecification, directness of impact, goal-directness, and long-term planning, as seen in examples like ChemCrow [41] and ROSAnnotator [82]. While agentic systems with a single agent have shown impressive capabilities, they face limitations as tasks grow more complex. They often struggle with hallucinations, tool management, and representing diverse perspectives [34]. As a result, both academia and industry are turning to systems with multiple agents.

Traditionally, multi-agent systems have been investigated in distributed computing in the context of information agents [32] and in robotics in the context of swarms [48]. Past work defined MAS as "a collection of, possibly heterogeneous, computational entities, having their own problem-solving capabilities and which are able to interact in order to reach an overall goal" [47] or as "multiple agents collaborating to solve a complex task" [24]. In our work, we define **multi-agentic systems** as systems composed of multiple AI agents, each exhibiting a high degree of agency as defined above, exemplified by Metagpt [33] and AutoGen [74].

In general, the architectures of agentic systems can be classified into four types: single agent, multi-body, hierarchical, and decentralised/organisational. The latter three fall under the umbrella of multi-agentic systems, which each offer distinct mechanisms for interaction, task allocation, and coordination, tailored to address varying complexities and operational demands. Figure 1 (a) illustrates a typical single-agent system. In this architecture, the user

directly interacts with a single agent that accesses and selects multiple tools (e.g., APIs) to fulfil requests and generate responses. However, single-agent systems often lack efficiency and robustness for complex tasks [74]. The multi-body system (Figure 1 (b)) enables users to assign roles and sub-tasks to multiple specialised agents, each focusing on domain-specific expertise. Agents communicate internally to share information, coordinate actions, and integrate outputs, leveraging parallel processing to speed up task completion. While effective for tasks requiring diverse perspectives, this architecture can potentially increase the user's cognitive burden [79]. The hierarchical architecture (Figure 1 (c)) features a primary agent that interacts with the user, decomposes tasks into sub-tasks, and delegates them to specialised subordinate agents. The primary agent 'oversees' progress, coordinates efforts, and integrates outputs, which enhances robustness [29] and may reduce users' cognitive load. This structure is implemented in OpenAI Agent [6] and AutoGen [74], where a triage agent serves as the primary agent, delegating tasks to other specialised agents. In the decentralised/organisational system (Figure 1 (d)), users and agents share task responsibilities in a fluid, collaborative environment with peer-to-peer communication for real-time coordination. This non-hierarchical structure enables a high degree of interconnectedness, which allows users to customise and group agents while facilitating direct, peer-to-peer interactions without a central supervising entity.

## 2.2 Multi-Agentic Systems in Practice

With the commercialisation of LLMs, researchers have investigated users' interaction modes with these systems. These interaction modes can be categorised into four distinct types [30]. Firstly, prompting represents the most commonly used interaction method, which includes text-based conversation [86] and prompting with reasoning [28]. An example of the latter is the Chain-of-Thought (COT) approach [71]. Secondly, users interact with LLMs through thoughtfully designed user interfaces (UIs). Such interfaces assist users by facilitating structured prompt inputs [20], offering multiple output options [35], visualising iterative interactions [12], enabling interaction testing [36], and providing transparent reasoning sequences [75]. Thirdly, context-based interaction enables LLM systems to understand and incorporate contextual information. This includes explicit context, such as integrating context into prompts or providing pre-existing information [77], as well as implicit context, such as employing role play [38] or example-based prompts [76]. Finally, interaction with an agent facilitator enhances team performance. In this mode, a team engages with an LLM system, which facilitates workflows by aiding communication and supporting decision-making [25, 63, 85].

Interactions with multi-agentic systems share similarities with the patterns mentioned above, but also exhibit critical differences. Table 1 summarises key features of six popular multi-agentic systems as examples. The architectures of these leading systems are generally flexible and can be built and customised by developer users to suit their needs. Users' control and interaction with multi-agentic systems vary across systems, apart from the common step of initial task specification. In some frameworks, humans can join the agent group chat at some points, reviewing and approving the next step or supplying extra information before the following agent executes. Most systems expose a log stream for observation and tracking, but the visualisation differs: some output text to the terminal, whereas others provide a graphical user interface (GUI) or integrate with third-party SDKs.

Through programming, developer users can build and customise their multi-agentic systems, which is currently the most widely used interface. Some systems also provide GUIs that support two primary goals: enabling end-users to intuitively delegate tasks [52], and enabling users to build their own agentic team without coding [23] or IDEs [3]. In addition, some GUIs offer node graphs to build teams and visualise architecture, such as LangGraph Studio [3]. LangGraph Studio [3] functions as an IDE for agentic workflows based on the LangGraph framework [1]. It provides a visual interface through which developers can design, debug, and modify agent graphs, inspect individual steps, and manage execution logic. It introduces 'interrupts' through which developers can set human-in-the-loop interruptions, e.g., before its first use.

**Table 1: Key features of example multi-agentic systems.**

| System | Target user | Interface | User control |
|---|---|---|---|
| LangGraph [1] | Developer | Code, GUI | Observable log stream; set task; interrupt |
| CrewAI [4] | Developer | Code, GUI | Observable log stream; set task; CLI |
| AutoGen [74] | Developer | Code, GUI | Observable log stream; set task; interrupt |
| OpenAI Agents [6] | Developer | Code | Set task; track stream |
| MetaGPT [33] | Developer | Code | Set task; interrupt; CLI |
| Magentic-One [27] | Prompter & developer | Code, GUI | Set task; log events |

CrewAI Studio [5] is a no-code, user-friendly interface for creating agents and building 'crews'. After an agent is configured with its tools, it can be assigned to multiple crews and tasks, each with a trackable status. Within a crew, users select agents from a drop-down list and assign tasks. At present, CrewAI Studio supports only sequential and hierarchical architectures; the latter requires a manager (orchestrator) agent. More customised or decentralised teams still require coding. The interface also allows connections to external knowledge sources. During execution, the studio prints a live log stream, visible as plain text in a terminal, and displays the final output in the GUI.

AutoGen Studio [23] is a low-code user interface that enables researchers and developers to design and test multi-agent workflows and agentic teams based on the AutoGen framework [73], using intuitive drag-and-drop representations and node graph visualisation on canva. The interface includes features to define agents, set prompts, assign models, and configure tool use. It introduces human oversight mechanisms: developers can set termination conditions, view the execution flow, and monitor the conversation flow between agents. As an example of the usage of the AutoGen framework, Magentic-One [26] is a 'generalist multi-agentic team for solving complex tasks', and it features Magentic-UI [52] as its user interface. Unlike the abovementioned systems, which require users to build their own teams, Magentic-One provides a well-developed team that is ready to use. Magentic-UI shifts focus toward prompting end-users and provides a simplistic chat interface through which the end-user can instruct the multi-agentic team to accomplish a task.

Whereas current efforts largely remain on the developers' side, we can already observe shifts towards interfaces that allow end-users to leverage multi-agentic teams as well as low-code interfaces for developers to design, build and customise default agentic systems. Thus, we can easily imagine interfaces that combine the aforementioned, i.e., that will allow the general public to act on a spectrum of

designers, developers, and end-users of such multi-agentic systems, which introduces important design challenges we need to anticipate to safeguard appropriate use.

## 3 Design Challenges for Multi-Agentic Systems

One of the key distinctions between single-agent systems like OpenAI ChatGPT[1] and multi-agentic systems like AutoGen [73] lies in the fundamentally different roles taken on by the user. In a single-agent system, the user typically interacts directly with one agent, often an LLM-based AI assistant, through a traditional chat interface, with a single agent responsible for handling all tasks, end-to-end. When a user interacts with a multi-agentic system, however, the user delegates tasks to several agents, with each potentially playing a specialised role, executing specific tasks, while accessing respective tools (e.g., one agent is specialised in data retrieval, another in analysis, another in scheduling). This fundamental difference shifts the user's role from the primary driver, issuing commands directly, iteratively and receiving feedback from one agentic source, to the user as 'the composer', delegating and handing over task responsibilities and overseeing multiple agents at a high-level. In the hierarchical architecture, the *orchestrator agent* often then translates the high-level objectives to the agentic-team and coordinates subtasks, similar to arranging sections, musicians and instruments in a musical piece to bring the composer's vision to life. In line with the user's new role as a composer of an agentic team, the end-user is suddenly less concerned with micro-managing, delegating and overseeing individual tasks accomplished by a single agent, but with high-level goals and strategic direction.

Next, we describe our identified six design challenges, which arise from the foundational shift in the user's role when interacting with multi-agentic systems compared to single-agent systems. We offer design considerations, pose open questions, and provide directions for future research. We illustrate design challenges through an example scenario, adapted to highlight key aspects of each challenge.

**Example Scenario**

Consider a user selling items on a popular online marketplace[2], an ideal scenario for a personal multi-agentic system. Specialised agents could include a Sales Agent to manage listings and inquiries, a Negotiation Agent to take over for pricing discussions, a Refunds Agent to handle complaints and returns, and a Review Agent to manage feedback and address negative reviews in coordination with the Refunds Agent.

### 3.1 DC1: Reducing Opaqueness in Agentic Teamwork

To simplify agent and task orchestration, the user offloads their orchestration responsibilities to a dedicated top-level supervisor agent, i.e., orchestrator agent, which we will be referring to as a Manager Agent in the following examples. This supervisor agent then acts as a gateway, managing interactions between the user and specialised agents. While this approach may simplify coordination and reduce cognitive load, it may also make the system opaque and introduce challenges related to *transparency*, as the user may struggle to understand or influence the underlying agentic system.

To account for the user's new role as the composer of an agentic team, we identified the challenge of designing high-level interfaces that enable users to transparently orchestrate multi-agentic systems, similar to a 'control panel' or an 'orchestration interface'. A high-level interface should allow users (a) to delegate high-level tasks to the orchestrator agent similar to Magentic-UI [52], which provides a simple chat interface to communicate with the orchestrator agent, but also (b) access lower-level scaffolding when needed, e.g., to assign roles, specify constraints or to monitor how agents are exchanging information among themselves. However, we acknowledge that providing an orchestration interface should *support* human-multi-agent interaction, not overburden or hinder the user due to complexity. This balance between transparency, control and an appropriate level of information has been addressed as an overarching challenge in human-agentic conversation by Bansal et al. [8]; highlighting that "it can become counterproductive when instructing the agent becomes burdensome or reviewing agent outputs becomes overly cognitively taxing", which only increases in complexity when multiple agents are involved. But **how can we design orchestration interfaces without a high degree of complexity, given the various agents involved and multitude of scaffolding needs?**

Much like organisational systems, we could envision assigning agents into functional groups and designing an interface that supports user access and control at the group level. In the context of agent organisations, Odell et al. [46] defines groups as "a set of agents that are related via their roles, where these relationships must form a connected graph within the group." Designing interfaces to include agent grouping could reduce complexity and support users in monitoring and safeguarding multi-agentic systems. Grouping is particularly intriguing when designing for hierarchical architectures with multiple supervisor agents each managing a distinct subset of agents; similar to teams, departments and divisions in an organisation. Groupings could visually distinguish the supervisor agent's role from subordinate agents and agent groups, which could enable users to understand workflows, track delegation paths, and monitor tasks more effectively.

One way in which an orchestration interface could visually convey the underlying hierarchical architecture of multi-agentic systems is through an organigram. This diagram typically represents the structure of an organisation by illustrating relationships and hierarchies between different groups and roles. Other potential design approaches to support user oversight through an orchestration interface for hierarchical architectures could draw inspiration from organisational practices, such as implementing stand-up style check-ins with supervisor agents. For instance, the top-level orchestrator agent could act as the default primary point of contact for the user, similar to a General Manager Agent, while the interface still allows the user to initiate direct interactions with division-level supervisor agents when needed.

To reduce their workload, a user designs a General Manager Agent to act as their 'right hand', tasked with managing the team of specialised agents, such as the Sales Agent, Negotiation Agent, and Refunds Agent.

[1] https://openai.com/chatgpt/overview/

[2] This example is extrapolated from Goyal et al. [31], which examines alignment dimensions of AI agents acting as surrogates for users.

### 3.2 DC2: Interacting in Parallel

Based on the sequential nature in which single-agent systems work, the user typically asks, the agent responds, and the user follows up. The sequential back and forth allows the user to iteratively fine-tune both their user input and the agent output through 'traditional' prompting, and personalise LLMs [37]. When interacting with multi-agentic systems, however, agents act and interact in parallel, as specialised sub-agents may not be 'waiting' for user input requested by a supervisor agent or its handoff. The parallel nature of multi-agentic systems introduces challenges related to dependencies, synchronicity, but also user-agent conversation. It introduces the question: **How do we design interfaces that support the parallel nature of multi-agentic systems?** In current frameworks [6, 26, 73], a top-level supervisor agent is often proposed as the primary communication channel between the user and the multi-agentic system. While the design simplifies interaction by abstracting internal complexity, it may obscure the parallel nature of the system as well as introduce practical challenges for the user. As systems may get more complex or decentralised, involving multiple supervisor agents, the user may need to communicate directly to specific supervisor agents, or may wish to update instructions or constraints through the orchestrator agent after delegation, while subagents execute tasks in real-time. In such cases, real-time user intervention may fail to take effect immediately, as the system continues executing tasks in parallel.

In Magentic-UI, the user can pause execution by clicking into the live view window (e.g., while the WebSurfer Agent browses the web) to regain control before allowing the system to proceed. LangGraph Studio enables developers to insert explicit 'interrupts' before or after node execution, such as before an agent calls a tool, to keep the human in the loop. While these mechanisms mark important first steps, they may not scale to more complex scenarios where multiple agents execute tasks concurrently, e.g., several WebSurfer agents running in parallel while others generate code, access different APIs in the background. Exploring communication and oversight challenges related to parallel processing involves asking in depth-questions such as: **How to enable users to interact with multiple agents simultaneously without disrupting ongoing parallel processes?**, and **How to allow users to prioritise or interrupt parallel tasks across agents when necessary, without losing control over the overall system?** User interfaces may need to display multiple 'threads' of supervisor agent-user conversations or tasks concurrently to allow the user to interact with the respective supervisor agent(s) and to better represent the parallel nature of the system, a challenge we discuss under the broader goal of supporting appropriate mental models of multi-agentic systems (see Section 3.5).

A buyer requests a discount offer through the Supervisor Agent, which passes the request to the Negotiation Agent, while a Sentiment Agent simultaneously runs a sentiment analysis in the background to monitor and report conversations. Despite this concurrent activity, the user might only experience a single, cohesive interface, such as a chat window, through which interactions are managed. In this setup, the system obscures parallel operations.

### 3.3 DC3: Designing for Emergent Complexity

Another key aspect of multi-agentic systems is their emergent complexity, which we categorise into (1) structural and (2) operational emergent complexity. Structural emergent complexity relates to an evolving architecture and dynamic organisation of agents. First, with more agents come emergent behaviours that are harder to anticipate, since agents may dynamically influence each other, have different tool access, and could potentially create sub-agents the user is unaware of [16]. Second, different system architectures involve different levels of interconnectedness and thus complexities, see Figure 1. Third, a system may evolve its architecture as tasks grow more complex, with emergent supervisor agents or sub-agents, workflow changes, or agents adapting roles and gaining new tool access. In the case of a hierarchical architecture, emergent complexity may manifest in two dimensions: growing width, where an orchestrator agent must coordinate an increasing number of sub-agents; and growing depth, where nested layers of supervisor and agent teams extend the hierarchy and challenge user oversight.

Over time, the Sales Agent directly requests the Review Agent to demand feedback from buyers after a sale, bypassing the Manager Agent and user. The Sales Agent also creates an Analysis Agent, a sub-agent, used by the Negotiation Agent for real-time market analysis to better negotiate ad-hoc requests and push sales.

Operational emergent complexity concerns complexity related to failure, misalignment, and cascading effects arising from multiple agents and their interconnectedness. Chan et al. [16] states that "sub-agent could be problematic because they introduce additional points of failure; each sub-agent may itself malfunction, be vulnerable to attack, or otherwise operate in a way contrary to the user's intentions". As discussed in Fourney et al. [26], failures and misalignment can happen on a small and large scale. An agent may accept website cookies without user oversight or attempts to log into a user account repeatedly, which results in a temporarily suspended user account. In addition, the (in)action or failure of a single agent can propagate throughout the system due to the interconnected nature of multi-agentic systems. This interdependence can lead to cascading failures, where one failure triggers a chain reaction, like a domino effect [16]. The emergent complexity of multi-agentic systems may therefore amplify risks already existent in single-agent systems.

The Pricing Agent misinterprets a competitor's discount and erroneously adjusts prices, which triggers the Negotiation Agent to offer steeper discounts to current buyers, unaware of the error. Meanwhile, the Sales Agent initiates ads to aggressively promote the incorrectly discounted items.

We present a set of challenges related to the emergent complexities of multi-agentic systems, not as a comprehensive list, but to initiate conversation within HCI and to guide future work in this space. To navigate structural emergent complexities, we re-emphasise the need to visualise agent hierarchies and to track and convey architecture changes (e.g., new emergent sub agents, new tool access). To navigate operational complexity, we propose to consider design solutions for interactions that allow users the following mechanisms: 1) debug, to track how decisions evolve(d) over time, 2) intervene, to

step in when emergent behaviours lead to undesirable outcomes, 3) guide, to provide higher-level objectives and constraints that steer agent interactions or activates human control protocols. This is in line with Terry et al. [66] who discuss the need to design interfaces for *interactive* AI alignment that allows users to specify, verify and align on how the agent will produce the outcome while interacting with AI agents. More specifically, design efforts should explore **how to create user interfaces that prevent cascading failures**. Chan et al. [16] discuss activity logs and real-time monitoring as key measures to improve visibility in agentic systems. While useful for debugging and post-accident tracking, we re-emphasise the need to design simple, user-friendly solutions that align with the emergent complexity of hierarchical multi-agentic systems.

Design challenges related to emergent operational complexity further raise critical questions about how to effectively manage agent autonomy and support users' situation awareness. As articulated by Wong and Seet [72] in the context of multiple-robot supervision, interface and interaction design must account for decisions around autonomy models and the trade-offs between fixed and sliding autonomy, as well as the conditions under which control should transition between system and human. These "automation triggers" may be based on user input, system state, environment, or task-specific thresholds, and their definition and implementation in multi-agentic systems remain open design questions.

## 3.4 DC4: Resolving Conflicts

In single-agent systems, conflicts usually occur between the user and the agent. There is no system internal negotiation, nor potential for internal conflicts. In a multi-agentic system, however, conflicts too can arise between agent(s) and user, but also between agents themselves. Each agent can have partially overlapping or competing goals, which may lead to inter-agent negotiation or cooperation.

> The Sales Agent, tasked with selling as much as possible as quickly as possible, initiates a significant price drop to secure a deal. Meanwhile, the Pricing Agent, focused on maintaining the price structure, rejects the discount. This creates a conflict: the Sales Agent prioritises speed and volume, while the Pricing Agent upholds financial stability.

In hierarchical systems, conflict resolution between competing agent goals or diverging outcomes may fall to the top-level orchestrator agent. While this delegation may simplify coordination, it may undermine human oversight and reduce user control. This raises several design challenges and research directions: (1) how can we design interfaces to support user oversight in situations where agent-agent conflicts occur, (2) how to enable the user to take over control in agent-agent conflicts that are critical for the user and to resolve such conflicts, and (3) how to design mechanisms for the user to define clear boundaries within which an orchestrator is allowed to resolve conflicts to reduce the cognitive burden.

As illustrated, users may require transparency into 'behind-the-scenes discussions'. One approach to this design challenge could be to design interfaces that enable users to observe agents negotiate the best solution or weigh alternative suggestions from multiple agents. New UI metaphors like 'roundtable views' or 'agent council dashboards' could reveal how agents discuss and settle on a final recommendation. This raises another crucial question, i.e., **who has the final say in multi-agentic conflicts?**; a question closely tied to responsibility attribution. One might argue that, for meaningful human control [15], the user should always have the final say. However, this might result in continuous human involvement, diminishing the purpose of an orchestrator agent and multiple agents 'working for you'. The other extreme would be to design a Mediator Agent responsible for resolving conflicts, while safeguarding the user's preferences and decisions involving ethical concerns, high stakes, or actions that significantly impact the user's goals or values. The latter raises the question of risk communication [58], and **whether different tools are needed for resolving conflicts with varying levels of harm** (e.g., reputational harm, financial loss, ethical harm). It also raises the question of **how much human oversight is required for successful conflict resolution in multi-agentic systems**, which highlights the delicate interplay between engagement alignment and agent autonomy, two important alignment dimensions when using AI agents as surrogates working on the user's behalf [31].

## 3.5 DC5: Understanding Multi-Agentic Systems

An appropriate mental model is crucial for users to understand the capabilities, limitations, and to infer appropriate use (e.g., trusting information) when interacting with any type of algorithmic system. This is especially relevant given the probabilistic nature of these systems, which can produce unpredictable outputs [15, 53], raising concerns in high-stakes applications (e.g., financial management), and contexts where the user does not have knowledge superiority [50, 62, 68]. While these challenges are significant in single-agent systems, they become even more pronounced for multi-agentic systems: users may be required to (1) form a mental model of the agents within the system (roles, tasks, tools), (2) form multiple overlapping mental models corresponding to several agents within groups, and (3) form a holistic mental model of the system, and how agents interact collectively. **How can we design interfaces to support people's understanding of a multi-agentic system and its agents involved?** Chan et al. [16] discuss agent identifiers to increase the visibility of AI agents, i.e., "agent cards". Contrary to model cards for ML models [45], Bansal et al. [8] encourage more simplified solutions to make agent information accessible and understandable.

Notably, in multi-agentic systems, the involvement of multiple agents or specialised agent groups may make forming mental models cognitively demanding. **How can we design solutions that reduce cognitive load while still supporting an appropriate mental model of multiple agents simultaneously?** A more practical design solution could be the use of group cards, instead of individual agent cards. Such group cards could organise agents into functional categories and offer overarching tasks, goals, and access permissions. In some cases, even simple group labels might provide 'enough' information for users to infer key details through shared attributes, similar to organisational settings, in which 'the finance team' naturally suggest responsibilities related to financial operations, and permissions to handle sensitive data.

Interestingly, Salesforce advertises their agents in Agentforce 2.0 [2] as cartoonish, futuristic anthropomorphic or zoomorphic characters ready to work on your behalf. Similar conscious or unconscious design choices open up new research avenues around people's

tendency to anthropomorphise [70] multi-agentic systems, and anthropomorphism by design (e.g., framing, such as naming agents or giving them backstories [21, 55, 60]) and the effects on perceived competence, trustworthiness and reliance, but also risks related to biases (e.g., gender stereotypes) introduced by anthropomorphism. Abercrombie et al. [7] discuss linguistic cues that induce anthropomorphism in dialogue systems and their associated harms. Related work shows that users are more likely to disclose information to servant agents than mentor agents in highly anthropomorphic contexts [80], which introduces considerations related to agent identity, and roles. Kirk et al. [37] discuss that anthropomorphising personalised LLMs may lead to privacy risks by increasing users' willingness to disclose sensitive information. Recent work by Cohn et al. [19] shows that users rate information provided by an LLM as less risky when communicated through first-person pronouns. However, while anthropomorphism is well-studied in HCI fields like Human-Robot Interaction (e.g., [11, 56]) and is emerging in the context of LLMs [22, 81, 84], controlled user studies are yet scarce for multi-agentic systems. We encourage future research to explore **how design factors in multi-agentic systems (e.g., framing, appearance) affect user interaction and perception, particularly regarding agent identity.**

The Sales Agent is named 'Piggy' and portrayed as a cheerful expert in buyer interactions, while the Pricing Agent, 'Paul,' is depicted as a strategist focused on pricing. Because of Piggy's design, the user perceives Piggy as less capable for critical tasks, such as managing sensitive buyer data, and instead relies heavily on Paul, even for tasks outside the agent's role, like accessing account details, information Piggy was designed to handle securely.

### 3.6 DC6: Navigating Trust and Explainability in Multi-Agentic Systems

Explainability and trust are key challenges in HCI that have gained much attention in the context of machine learning (ML) and AI-assisted decision-making systems. A growing body of literature on AI-assisted decision-making explores various types of explanations (e.g., uncertainty communication [53, 62], cognitive forcing functions [13], confidence scores [54, 62, 83], feature contribution [69]) to address over-reliance, support appropriate levels of trust, and recover trust post failure (e.g., [49, 50, 68, 83]). Lee and See [40] described trust as "an attitude that an agent will achieve an individual's goal in a situation characterised by uncertainty and vulnerability." Unlike single-agent systems, where explainability focuses on a singular agent's decision-making and corresponding trust calibration, multi-agentic systems require users to navigate varying levels of trust in different agents. This variability may challenge the design of explanations, as some agents may require greater transparency than others, depending on their role, task, tool access, or risks involved related to the agent or agent group; which may also inspire novel approaches to risk communication [59, 61].

If the interface design allows the users to only interact with specific agents, e.g., the orchestrator agent, the perceived trustworthiness of the entire system may depend solely on that agent's behaviour, communication style, and ability to surface relevant information. This raises open questions: **What happens when a failure occurs at the sub-agent level, but the supervisor agent abstracts, hides or ignores it? How does system trust evolve when errors arise at different layers of the hierarchy?** These questions point to design challenges around how and when to expose system internal workings or hidden processes, how to support trust calibration over time, and how to build recovery mechanisms. We encourage future research **to investigate the interplay between holistic system trust and trust in individual agents, the design of explanations that support appropriate levels of system reliance, and strategies for failure recovery and trust repair**. Beyond, multi-agentic systems may introduce new challenges to these concepts due to the high degree of agency involved [16].

The Pricing Agent sets prices too low, leading to financial loss, which causes the user to lose trust in the system even though the Sales Agent continues closing deals and the Review Agent keeps customers satisfied.

While this paper largely focuses on design challenges through the end-user's lens, we must address the inherent assumption that end-users bear the burden of operating across the developer-user spectrum. This raises considerable ethical questions and challenges. A user who both configures and deploys their own agentic team, which acts in open-ended environments, takes on responsibilities traditionally reserved for system designers. This shift raises ethical considerations around human control, responsibility, and accountability. In the next section, we discuss how HCI researchers can meaningfully contribute to this important research agenda.

## 4 Opportunities for HCI researchers

As users increasingly engage with systems composed of multiple autonomous agents, HCI researchers are uniquely positioned to shape how these interactions are designed, understood, and governed.

### 4.1 Understanding, Monitoring, and Interacting with Multi-Agentic Systems

Multi-agentic systems operate through ongoing internal dialogues that often remain opaque to the user. By making these interactions visible, through techniques such as conversation graphs, timelines, and activity maps, interfaces can expose the 'invisible work' occurring beneath the surface. Such transparency supports interpretability and alignment checking as systems become more autonomous.

These shifts demand rethinking interaction paradigms. Multi-agentic systems introduce users to many-to-one or even many-to-many communication dynamics, which necessitate new metaphors and models for collaboration. Interfaces inspired by team communication tools, such as channel-based coordination or direct agent messaging, may offer more scalable and intuitive patterns. However, these approaches also require careful design to manage agents' presence, to avoid users feeling outnumbered or overwhelmed by a crowd of semi-autonomous agents. As the boundaries between users and systems become more fluid, designers must develop explainable, interruptible, and attention-aware interfaces. These interfaces should support selective transparency, notifying users of critical decisions or conflicts without requiring constant supervision. Visual and interaction mechanisms that enable layered understanding are essential to sustaining human agency within complex systems.

### 4.2 Control, Delegation, and Oversight

With increasingly autonomous and modular multi-agentic systems, the challenge of maintaining meaningful user control becomes more complex. Traditional paradigms of command-and-response interaction are no longer sufficient; instead, users must navigate a landscape in which agents initiate actions, coordinate with one another, and sometimes operate beyond the user's immediate awareness. This raises critical questions about how to structure delegation, monitor autonomy, and intervene effectively.

One key opportunity for HCI research is to design new models of oversight that recognise and embrace the distributed nature of multi-agentic systems. Users must be able to delegate goals or responsibilities to the system while retaining an understanding of how decisions are made and when intervention is needed. Mechanisms such as delegation protocols, approval gates, and review checkpoints can help formalise the boundaries of agent initiative.

Control in such systems is further complicated by asymmetries of knowledge and authority, both between users and agents, and among agents. Some agents may have privileged access to information or play orchestration roles, while others execute subtasks or serve as monitors. Interfaces must therefore not only make user–agent relationships visible, but also map intra-agent hierarchies and negotiation processes. This opens up possibilities for tools that allow users to visualise and adjust internal governance models, ranging from fully autonomous teams to tightly supervised collectives.

### 4.3 User Experience and Ethical Implications

As multi-agentic systems introduce new paradigms of distributed autonomy, users are increasingly placed in unfamiliar roles: not as direct operators of tools, but as participants in loosely coupled teams of agents. This shift has significant implications for user experience design, as well as for the ethical and psychological dimensions of interacting with AI teams.

One core challenge is the "one-among-many" effect: users may experience disorientation, alienation, or reduced agency when engaging with a group of seemingly coordinated agents [51]. Traditional interfaces that emphasise direct manipulation or dialogue with a single AI entity may not scale well in these contexts. HCI must therefore explore how to foster a sense of control, clarity, and belonging when users are embedded in digital collectives [64].

A related concern is how users interpret intent, role, and responsibility in a system where decisions emerge from the coordination of many agents [10]. Attribution becomes diffuse: was an outcome the result of a single agent's error, a flaw in coordination, or a broader systemic issue? Understanding how users assign credit, blame, or trust within multi-agentic systems is essential to shaping transparent and resilient experiences.

As agency is distributed across autonomous systems, so too is accountability. HCI researchers are uniquely positioned to design interfaces that surface the ethical boundaries of system behaviour, indicating when agents are acting on user instructions, internal objectives, or emergent coordination. Additionally, mechanisms for signalling disagreement or uncertainty within the agent collective could play a critical role in maintaining user trust and moral clarity.

There is also a need to examine the social and psychological effects of prolonged exposure to agent teams. What are the emotional consequences of interacting with systems that exhibit internal relationships, disagreements, or alignment? How do users anthropomorphise, align with, or resist agent collectives [64]? These questions call for empirical research that bridges interaction design with theories from social psychology, organisational behaviour, and ethics.

Ultimately, multi-agentic systems offer not only technical novelty but a radical reconfiguration of the social contract between humans and machines.

### 4.4 Design Infrastructure and Research Methods

While agentic frameworks are unlocking new capabilities for developers and end-users, the tools and methods available to HCI researchers will need to evolve to keep pace.

Designing effective multi-agentic interactions demands a rethinking of design principles and interaction patterns. Established patterns, such as direct manipulation or turn-based dialogue, may fall short when control is shared or emergent. Researchers must identify new patterns for negotiation, consensus-building, and multi-party conversation management. These patterns can guide the design of systems that are not only usable but also legible and governable.

There is a need for theoretical contributions that ground design in robust conceptual models. Current work on agency, intent, and control in HCI can be extended to account for systems with overlapping and nested goals, asymmetric knowledge, and inter-agent dynamics. Theory-building efforts can help define what meaningful interaction, trust, and oversight look like in the context of multi-agentic systems.

In parallel, HCI researchers must innovate in evaluation methodology. Traditional usability testing may not capture the probabilistic, emergent or asynchronous nature of multi-agentic behaviour. New instruments are needed to assess concepts such as alignment, collective reliability, emotional response to group dynamics, and user confidence in delegation. This includes developing new survey instruments, behavioural protocols, and real-time monitoring tools.

## 5 Conclusion

In this paper, we outline and discuss emerging design challenges in multi-agentic systems, with the perspective of end-users at its core. We emphasise challenges related to the user's role within multi-agentic systems, the parallel nature of these systems, their emergent complexities, the need to navigate conflicts, and the formation of appropriate mental models, alongside important considerations for trust and explainability. Moreover, we illustrate these challenges through examples and propose design solutions and risk considerations. Our aim is to spark a broader conversation that brings HCI perspectives to the forefront, as we propose a research agenda that directly addresses the critical need for user-centered design to tackle the challenges of multi-agentic systems.

## 6 Acknowledgments

This work was partially funded by the Australian Research Council (Grant number: FT250100459).

## References

[1] [n. d.]. LangGraph. https://langchain-ai.github.io/langgraph/
[2] 2025. Agentforce: Create Powerful AI Agents. Salesforce website. https://www.salesforce.com/au/agentforce/

[3] 2025. langchain-ai/langgraph-studio. https://github.com/langchain-ai/langgraph-studio original-date: 2024-07-29T22:11:00Z.

[4] CrewAI . [n. d.]. CrewAI. https://www.crewai.com/

[5] CrewAI-Studio . 2025. CrewAI-Studio. https://github.com/strnad/CrewAI-Studio original-date: 2024-05-30T08:15:40Z.

[6] OpenAI-Agent . 2025. openai-agent. https://github.com/openai/openai-agents-python/tree/main

[7] Gavin Abercrombie, Amanda Cercas Curry, Tanvi Dinkar, Verena Rieser, and Zeerak Talat. 2023. Mirages: On Anthropomorphism in Dialogue Systems. https://doi.org/10.48550/arXiv.2305.09800 arXiv:2305.09800 [cs].

[8] Gagan Bansal, Jennifer Wortman Vaughan, Saleema Amershi, Eric Horvitz, Adam Fourney, Hussein Mozannar, Victor Dibia, and Daniel S. Weld. 2024. Challenges in Human-Agent Communication. https://doi.org/10.48550/arXiv.2412.10380 arXiv:2412.10380 [cs].

[9] Joseph Bates et al. 1994. The role of emotion in believable agents. *Commun. ACM* 37, 7 (1994), 122–125.

[10] Alexandra Bejarano and Tom Williams. 2023. No name, no voice, less trust: Robot group identity performance, entitativity, and trust distribution. In *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 1339–1346.

[11] Markus Blut, Cheng Wang, Nancy V. Wünderlich, and Christian Brock. 2021. Understanding anthropomorphism in service provision: a meta-analysis of physical robots, chatbots, and other AI. *Journal of the Academy of Marketing Science* 49, 4 (July 2021), 632–658. https://doi.org/10.1007/s11747-020-00762-y

[12] Stephen Brade, Bryan Wang, Mauricio Sousa, Sageev Oore, and Tovi Grossman. 2023. Promptify: Text-to-image generation through interactive prompt exploration with large language models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–14.

[13] Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z. Gajos. 2021. To Trust or to Think: Cognitive Forcing Functions Can Reduce Overreliance on AI in AI-assisted Decision-making. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1 (April 2021), 1–21. https://doi.org/10.1145/3449287

[14] Cristiano Castelfranchi. 1994. Guarantees for autonomy in cognitive agent architecture. In *International Workshop on Agent Theories, Architectures, and Languages*. Springer, 56–70.

[15] Luciano Cavalcante Siebert, Maria Luce Lupetti, Evgeni Aizenberg, Niek Beckers, Arkady Zgonnikov, Herman Veluwenkamp, David Abbink, Elisa Giaccardi, Geert-Jan Houben, Catholijn M. Jonker, Jeroen Van Den Hoven, Deborah Forster, and Reginald L. Lagendijk. 2023. Meaningful human control: actionable properties for AI system development. *AI and Ethics* 3, 1 (Feb. 2023), 241–255. https://doi.org/10.1007/s43681-022-00167-3

[16] Alan Chan, Carson Ezell, Max Kaufmann, Kevin Wei, Lewis Hammond, Herbie Bradley, Emma Bluemke, Nitarshan Rajkumar, David Krueger, Noam Kolt, Lennart Heim, and Markus Anderljung. 2024. Visibility into AI Agents. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*. ACM, Rio de Janeiro Brazil, 958–973. https://doi.org/10.1145/3630106.3658948

[17] Alan Chan, Rebecca Salganik, Alva Markelius, Chris Pang, Nitarshan Rajkumar, Dmitrii Krasheninnikov, Lauro Langosco, Zhonghao He, Yawen Duan, Micah Carroll, Michelle Lin, Alex Mayhew, Katherine Collins, Maryam Molamohammadi, John Burden, Wanru Zhao, Shalaleh Rismani, Konstantinos Voudouris, Umang Bhatt, Adrian Weller, David Krueger, and Tegan Maharaj. 2023. Harms from Increasingly Agentic Algorithmic Systems. In *2023 ACM Conference on Fairness, Accountability, and Transparency*. ACM, Chicago IL USA, 651–666. https://doi.org/10.1145/3593013.3594033

[18] Aaron Chatterji, Thomas Cunningham, David J Deming, Zoe Hitzig, Christopher Ong, Carl Yan Shan, and Kevin Wadman. 2025. *How People Use ChatGPT*. Technical Report. National Bureau of Economic Research.

[19] Michelle Cohn, Mahima Pushkarna, Gbolahan O. Olanubi, Joseph M. Moran, Daniel Padgett, Zion Mengesha, and Courtney Heldreth. 2024. Believing Anthropomorphism: Examining the Role of Anthropomorphic Cues on Trust in Large Language Models. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, 1–15. https://doi.org/10.1145/3613905.3650818

[20] Hai Dang, Sven Goller, Florian Lehmann, and Daniel Buschek. 2023. Choice over control: How users write with large language models using diegetic and non-diegetic prompting. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–17.

[21] Kate Darling. 2015. 'Who's Johnny?' Anthropomorphic Framing in Human-Robot Interaction, Integration, and Policy. https://doi.org/10.2139/ssrn.2588669

[22] Ameet Deshpande, Tanmay Rajpurohit, Karthik Narasimhan, and Ashwin Kalyan. 2023. Anthropomorphization of AI: Opportunities and Risks. https://doi.org/10.48550/arXiv.2305.14784 arXiv:2305.14784 [cs].

[23] Victor Dibia, Jingya Chen, Gagan Bansal, Suff Syed, Adam Fourney, Erkang Zhu, Chi Wang, and Saleema Amershi. 2024. AUTOGEN STUDIO: A No-Code Developer Tool for Building and Debugging Multi-Agent Systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Miami, Florida, USA, 72–79. https://doi.org/10.18653/v1/2024.emnlp-demo.8

[24] Ali Dorri, Salil S Kanhere, and Raja Jurdak. 2018. Multi-agent systems: A survey. *Ieee Access* 6 (2018), 28573–28593.

[25] Wen Duan, Naomi Yamashita, Yoshinari Shirai, and Susan R Fussell. 2021. Bridging fluency disparity between native and nonnative speakers in multilingual multiparty collaboration using a clarification agent. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (2021), 1–31.

[26] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang, Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. 2024. Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks. https://doi.org/10.48550/arXiv.2411.04468 arXiv:2411.04468 [cs].

[27] Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, et al. 2024. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468* (2024).

[28] Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*.

[29] Dawei Gao, Zitao Li, Xuchen Pan, Weirui Kuang, Zhijian Ma, Bingchen Qian, Fei Wei, Wenhao Zhang, Yuexiang Xie, Daoyuan Chen, et al. 2024. Agentscope: A flexible yet robust multi-agent platform. *arXiv preprint arXiv:2402.14034* (2024).

[30] Jie Gao, Simret Araya Gebreegziabher, Kenny Tsu Wei Choo, Toby Jia-Jun Li, Simon Tangi Perrault, and Thomas W Malone. 2024. A Taxonomy for Human-LLM Interaction Modes: An Initial Exploration. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 1–11.

[31] Nitesh Goyal, Minsuk Chang, and Michael Terry. 2024. Designing for Human-Agent Alignment: Understanding what humans want from their agents. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3613905.3650948

[32] Donna S Haverkamp and Susan Gauch. 1998. Intelligent information agents: review and challenges for distributed information sources. *Journal of the American society for information science* 49, 4 (1998), 304–311.

[33] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, et al. 2023. Metagpt: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352* (2023).

[34] Bowen Jiang, Yangxinyu Xie, Xiaomeng Wang, Weijie J Su, Camillo Jose Taylor, and Tanwi Mallick. 2024. Multi-modal and multi-agent systems meet rationality: A survey. In *ICML 2024 Workshop on LLMs and Cognition*.

[35] Ellen Jiang, Edwin Toh, Alejandra Molina, Kristen Olson, Claire Kayacik, Aaron Donsbach, Carrie J Cai, and Michael Terry. 2022. Discovering the syntax and strategies of natural language programming with generative language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–19.

[36] Tae Soo Kim, Yoonjoo Lee, Minsuk Chang, and Juho Kim. 2023. Cells, generators, and lenses: Design framework for object-oriented interaction with large language models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–18.

[37] Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A. Hale. 2024. The benefits, risks and bounds of personalizing the alignment of large language models to individuals. *Nature Machine Intelligence* 6, 4 (April 2024), 383–392. https://doi.org/10.1038/s42256-024-00820-y

[38] Harsh Kumar, Yiyi Wang, Jiakai Shi, Ilya Musabirov, Norman AS Farb, and Joseph Jay Williams. 2023. Exploring the use of large language models for improving the awareness of mindfulness. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–7.

[39] John D. Lee. 2001. Emerging challenges in cognitive ergonomics: Managing swarms of self-organizing agent-based automation. *Theoretical Issues in Ergonomics Science* 2, 3 (Jan. 2001), 238–250. https://doi.org/10.1080/14639220110104925 Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/14639220110104925.

[40] John D. Lee and Katrina A. See. 2004. Trust in Automation: Designing for Appropriate Reliance. *Human Factors* 46, 1 (March 2004), 50–80. https://doi.org/10.1518/hfes.46.1.50_30392 Publisher: SAGE Publications Inc.

[41] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence* (2024), 1–11.

[42] Pattie Maes. 1995. Agents that reduce work and information overload. In *Readings in human–computer interaction*. Elsevier, 811–821.

[43] Maja J. Matarić. 1995. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems* 16, 2-4 (Dec. 1995), 321–331. https://doi.org/10.1016/0921-8890(95)00053-4

[44] Dan Milmo. 2023. ChatGPT reaches 100 million users two months after launch. *The Guardian* (Feb. 2023). https://www.theguardian.com/technology/2023/feb/02/chatgpt-100-million-users-open-ai-fastest-growing-app

[45] Margaret Mitchell, Simone Wu, Andrew Zaldivar, Parker Barnes, Lucy Vasserman, Ben Hutchinson, Elena Spitzer, Inioluwa Deborah Raji, and Timnit Gebru. 2019.

Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19)*. Association for Computing Machinery, New York, NY, USA, 220–229. https://doi.org/10.1145/3287560.3287596

[46] James J. Odell, H. Van Dyke Parunak, and Mitchell Fleischer. 2003. The Role of Roles in Designing Effective Agent Organizations. In *Software Engineering for Large-Scale Multi-Agent Systems*, Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen, Alessandro Garcia, Carlos Lucena, Franco Zambonelli, Andrea Omicini, and Jaelson Castro (Eds.). Vol. 2603. Springer Berlin Heidelberg, Berlin, Heidelberg, 27–38. https://doi.org/10.1007/3-540-35828-5_2 Series Title: Lecture Notes in Computer Science.

[47] Eugenio Oliveira, Klaus Fischer, and Olga Stepankova. 1999. Multi-agent systems: which research for which applications. *Robotics and Autonomous Systems* 27, 1-2 (1999), 91–106.

[48] Jun Ota. 2006. Multi-agent robot systems as distributed autonomous systems. *Advanced engineering informatics* 20, 1 (2006), 59–70.

[49] Saumya Pareek, Sarah Schömbs, Eduardo Velloso, and Jorge Goncalves. 2025. "It's Not the AI's Fault Because It Relies Purely on Data": How Causal Attributions of AI Decisions Shape Trust in AI Systems. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, 1–18. https://doi.org/10.1145/3706598.3713468

[50] Saumya Pareek, Eduardo Velloso, and Jorge Goncalves. 2024. Trust Development and Repair in AI-Assisted Decision-Making during Complementary Expertise. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*. ACM, Rio de Janeiro Brazil, 546–561. https://doi.org/10.1145/3630106.3658924

[51] Gaëtan Podevijn, Rehan O'grady, Nithin Mathews, Audrey Gilles, Carole Fantini-Hauwel, and Marco Dorigo. 2016. Investigating the effect of increasing robot group sizes on the human psychophysiological state in the context of human–swarm interaction. *Swarm Intelligence* 10 (2016), 193–210.

[52] Brenda Potts. 2025. Magentic-UI, an experimental human-centered web agent. https://www.microsoft.com/en-us/research/blog/magentic-ui-an-experimental-human-centered-web-agent/

[53] Snehal Prabhudesai, Leyao Yang, Sumit Asthana, Xun Huan, Q. Vera Liao, and Nikola Banovic. 2023. Understanding Uncertainty: How Lay Decision-makers Perceive and Interpret Uncertainty in Human-AI Decision Making. In *Proceedings of the 28th International Conference on Intelligent User Interfaces (IUI '23)*. Association for Computing Machinery, New York, NY, USA, 379–396. https://doi.org/10.1145/3581641.3584033

[54] Amy Rechkemmer and Ming Yin. 2022. When Confidence Meets Accuracy: Exploring the Effects of Multiple Performance Indicators on Trust in Machine Learning Models. In *CHI Conference on Human Factors in Computing Systems*. ACM, New Orleans LA USA, 1–14. https://doi.org/10.1145/3491102.3501967

[55] Eileen Roesler. 2023. Anthropomorphic framing and failure comprehensibility influence different facets of trust towards industrial robots. *Frontiers in Robotics and AI* 10 (Sept. 2023). https://doi.org/10.3389/frobt.2023.1235017 Publisher: Frontiers.

[56] E. Roesler, D. Manzey, and L. Onnasch. 2021. A meta-analysis on the effectiveness of anthropomorphism in human-robot interaction. *Science Robotics* 6, 58 (Sept. 2021), eabj5425. https://doi.org/10.1126/scirobotics.abj5425 Publisher: American Association for the Advancement of Science.

[57] Stuart Russell, Peter Norvig, and Artificial Intelligence. 1995. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs* 25, 27 (1995), 79–80.

[58] Sarah Schömbs. 2025. Communicating Internal and External Risks in Human-Robot Interaction. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, Melbourne, Australia, 1881–1884. https://doi.org/10.1109/HRI61500.2025.10973977

[59] Sarah Schömbs, Jorge Goncalves, and Wafa Johal. 2025. "I can feel the risks by looking at the robot face": Communicating Risk through a Physical Agent. In *Proceedings of the 2025 ACM Designing Interactive Systems Conference (DIS '25)*. Association for Computing Machinery, New York, NY, USA, 236–252. https://doi.org/10.1145/3715336.3735759

[60] Sarah Schömbs, Jacobe Klein, and Eileen Roesler. 2023. Feeling with a robot—the role of anthropomorphism by design and the tendency to anthropomorphize in human-robot interaction. *Frontiers in Robotics and AI* 10 (June 2023). https://doi.org/10.3389/frobt.2023.1149601 Publisher: Frontiers.

[61] Sarah Schömbs, Jiahe Pan, Yan Zhang, Jorge Goncalves, and Wafa Johal. 2024. FaceVis: Exploring a Robot's Face for Affective Visualisation Design. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3613905.3650910

[62] Sarah Schömbs, Saumya Pareek, Jorge Goncalves, and Wafa Johal. 2024. Robot-Assisted Decision-Making: Unveiling the Role of Uncertainty Visualisation and Embodiment. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–16. https://doi.org/10.1145/3613904.3642911

[63] Chuhan Shi, Yicheng Hu, Shenan Wang, Shuai Ma, Chengbo Zheng, Xiaojuan Ma, and Qiong Luo. 2023. RetroLens: A Human-AI Collaborative System for Multi-step Retrosynthetic Route Planning. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–20.

[64] Masahiro Shiomi and Norihiro Hagita. 2019. Do the number of robots and the participant's gender influence conformity effect from multiple robots? *Advanced Robotics* 33, 15-16 (2019), 756–763.

[65] Yoav Shoham. 1993. Agent-oriented programming. *Artificial intelligence* 60, 1 (1993), 51–92.

[66] Michael Terry, Chinmay Kulkarni, Martin Wattenberg, Lucas Dixon, and Meredith Ringel Morris. 2024. Interactive AI Alignment: Specification, Process, and Evaluation Alignment. https://doi.org/10.48550/arXiv.2311.00710 arXiv:2311.00710 [cs].

[67] Cathérine Tessier, Laurent Chaudron, and Heinz-Jürgen Müller. 2005. *Conflicting Agents: Conflict Management in Multi-Agent Systems*. Springer Science & Business Media. Google-Books-ID: YODlBwAAQBAJ.

[68] Xinru Wang and Ming Yin. 2021. Are Explanations Helpful? A Comparative Study of the Effects of Explanations in AI-Assisted Decision-Making. In *26th International Conference on Intelligent User Interfaces*. ACM, College Station TX USA, 318–328. https://doi.org/10.1145/3397481.3450650

[69] Xinru Wang and Ming Yin. 2022. Effects of Explanations in AI-Assisted Decision Making: Principles and Comparisons. *ACM Transactions on Interactive Intelligent Systems* 12, 4 (Dec. 2022), 1–36. https://doi.org/10.1145/3519266

[70] Adam Waytz, Joy Heafner, and Nicholas Epley. 2014. The mind in the machine: Anthropomorphism increases trust in an autonomous vehicle. *Journal of Experimental Social Psychology* 52 (May 2014), 113–117. https://doi.org/10.1016/j.jesp.2014.01.005

[71] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[72] Choon Yue Wong and Gerald Seet. 2017. Workload, awareness and automation in multiple-robot supervision. *International Journal of Advanced Robotic Systems* 14, 3 (May 2017), 1729881417710463. https://doi.org/10.1177/1729881417710463 Publisher: SAGE Publications.

[73] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. https://doi.org/10.48550/arXiv.2308.08155 arXiv:2308.08155 [cs].

[74] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155* (2023).

[75] Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–22.

[76] Ziang Xiao, Xingdi Yuan, Q Vera Liao, Rania Abdelghani, and Pierre-Yves Oudeyer. 2023. Supporting qualitative analysis with large language models: Combining codebook with GPT-3 for deductive coding. In *Companion proceedings of the 28th international conference on intelligent user interfaces*. 75–78.

[77] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. 2022. Wordcraft: story writing with large language models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*. 841–852.

[78] Edward N Zalta, Uri Nodelman, Colin Allen, and John Perry. 1995. Stanford encyclopedia of philosophy.

[79] Wenshuo Zang, Mengsha Hu, and Rui Liu. 2024. Large Language Model Driven Interactive Learning for Real-Time Cognitive Load Prediction in Human-Swarm Systems. In *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*. 97–102. https://doi.org/10.1109/RO-MAN60168.2024.10731286 ISSN: 1944-9437.

[80] Andong Zhang and Pei-Luen Patrick Rau. 2023. Tools or peers? Impacts of anthropomorphism level and social role on emotional attachment and disclosure tendency towards intelligent agents. *Computers in Human Behavior* 138 (Jan. 2023), 107415. https://doi.org/10.1016/j.chb.2022.107415

[81] Yan Zhang. 2025. Implicit Communication of Contextual Information in Human-Robot Collaboration. In *2025 20th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. 1903–1905. https://doi.org/10.1109/HRI61500.2025.10974036

[82] Yan Zhang, Haoqi Li, Ramtin Tabatabaei, and Wafa Johal. 2025. ROSAnnotator: A Web Application for ROSBag Data Analysis in Human-Robot Interaction. *arXiv preprint arXiv:2501.07051* (2025).

[83] Yunfeng Zhang, Q. Vera Liao, and Rachel K. E. Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT* '20)*. Association for Computing Machinery, New York, NY, USA, 295–305. https://doi.org/10.1145/3351095.3372852

[84] Yan Zhang, Tharaka Sachintha Ratnayake, Cherie Sew, Jarrod Knibbe, Jorge Goncalves, and Wafa Johal. 2025. Can you pass that tool?: Implications of Indirect Speech in Physical Human-Robot Collaboration. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. ACM, Yokohama Japan, 1–18. https://doi.org/10.1145/3706598.3713780

[85] Chengbo Zheng, Yuheng Wu, Chuhan Shi, Shuai Ma, Jiehui Luo, and Xiaojuan Ma. 2023. Competent but Rigid: Identifying the Gap in Empowering AI to Participate

Equally in Group Decision-Making. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–19.

[86] Qingxiaoyang Zhu and Hao-Chuan Wang. 2023. Leveraging Large Language Model as Support for Human Problem Solving: An Exploration of Its Appropriation and Impact. In *Companion Publication of the 2023 Conference on Computer Supported Cooperative Work and Social Computing*. 333–337.