

A Taxonomy for Autonomous LLM-Powered Multi-Agent Architectures

Thorsten Händler^a

Ferdinand Porsche Mobile University of Applied Sciences (FERNFH), Austria

Keywords: Taxonomy, Autonomous Agents, Multi-Agent Collaboration, Large Language Models (LLMs), AI System Classification, Alignment, Software Architecture, Architectural Viewpoints, Software-Design Rationale, Context Interaction, Artificial Intelligence, Domain-Ontology Diagram, Feature Diagram, Radar Chart.

Abstract: Large language models (LLMs) have revolutionized the field of artificial intelligence, endowing it with sophisticated language understanding and generation capabilities. However, when faced with more complex and interconnected tasks that demand a profound and iterative thought process, LLMs reveal their inherent limitations. Autonomous LLM-powered multi-agent systems represent a strategic response to these challenges. While these architectures hold promising potential in amplifying AI capabilities, striking the right balance between different levels of autonomy and alignment remains the crucial challenge for their effective operation. This paper proposes a comprehensive multi-dimensional taxonomy, engineered to analyze how autonomous LLM-powered multi-agent systems balance the dynamic interplay between autonomy and alignment across various aspects inherent to architectural viewpoints such as goal-driven task management, agent composition, multi-agent collaboration, and context interaction. Our taxonomy aims to empower researchers, engineers, and AI practitioners to systematically analyze the architectural dynamics and balancing strategies employed by these increasingly prevalent AI systems. The exploratory taxonomic classification of selected representative LLM-powered multi-agent systems illustrates its practical utility and reveals potential for future research and development. An extended version of this paper is available on arXiv (Händler, 2023).


1 INTRODUCTION

In recent years, the emergence and the technological feasibility of large language models (LLMs) have revolutionized the field of artificial intelligence (Brown et al., 2020; Ouyang et al., 2022; Thoppilan et al., 2022; Chowdhery et al., 2022; Zhang et al., 2022). Pre-trained on vast amounts of text data, these models have catalyzed significant advancements by enabling sophisticated language understanding and generation capabilities, opening doors to a broad range of applications (Bommasani et al., 2021; Bubeck et al., 2023; Kaddour et al., 2023). Yet, despite their remarkable capabilities, LLMs also have inherent limitations.

While LLMs excel at generating outputs based on patterns identified in their training data, they lack a genuine understanding of the real world. Consequently, their outputs might seem plausible on the surface, but can be factually incorrect or even *hallucinated* (Maynez et al., 2020; Ji et al., 2023). Moreover, despite their proficiency in handling vast amounts of textual information and their rapid processing and

pattern recognition capabilities, LLMs struggle with maintaining consistent logic across extended chains of reasoning. This deficiency hinders their ability to engage in a deliberate, in-depth, and iterative thought process (aka *slow thinking*) (Sloman, 1996; Kahneman, 2011; Fabiano et al., 2023; Lin et al., 2023). As a result, LLMs encounter difficulties when it comes to handling more complex and interconnected tasks (Kojima et al., 2022; Wei et al., 2022).

These limitations of individual LLMs have led to the exploration of more sophisticated and flexible AI architectures including multi-agent systems that aim at accomplishing complex tasks, goals, or problems with the *cognitive synergy* of multiple autonomous LLM-powered agents (Torantulino et al., 2023; Nakajima, 2023; TransformerOptimus et al., 2023; Park et al., 2023; Shen et al., 2023; Li et al., 2023; Shrestha et al., 2023; Hong et al., 2023). Such systems tackle user-prompted goals by employing a *divide & conquer* strategy, by breaking them down into smaller manageable tasks. These tasks are then assigned to specialized agents, each equipped with a dedicated role and the reasoning capabilities of an LLM, as well as further competencies by utilizing

^a  <https://orcid.org/0000-0002-0589-204X>

contextual resources like data sets, tools, or further foundation models. Taking a cue from Minsky’s *society of mind* theory (Minsky, 1988), the key to the systems’ problem-solving capability lies in orchestrating the iterative collaboration and mutual feedback between these more or less ‘mindless’ agents during task execution and result synthesis.

One of the central challenges for the effective operation of LLM-powered multi-agent architectures (as with many AI systems) lies in finding the optimal *balance between autonomy and alignment* (Yudkowsky, 2016; Bostrom, 2017; Russell, 2022; Wolf et al., 2023; Hong et al., 2023). On the one hand, the systems should be aligned to the goals and intentions of human users; on the other hand, the systems should accomplish the user-prompted goal in a self-organizing manner. However, a system with high autonomy may handle complex tasks efficiently, but risks straying from its intended purpose if not sufficiently aligned, resulting in unexpected consequences and uncontrollable side effects. Conversely, a highly aligned system may adhere closely to its intended purpose but may lack the flexibility and initiative to respond adequately to novel situations. Current systems exhibit diverse approaches and mechanisms to intertwine these *cross-cutting concerns* (Kiczales et al., 1997) throughout their architectural infrastructure and dynamics.

However, existing taxonomies and analysis frameworks for autonomous systems and multi-agent systems (see Section 2.1) fall short in providing means to categorize and understand these challenges and involved architectural complexities posed by LLM-powered multi-agent systems.

This paper¹ aims to bridge this gap by introducing a systematic approach in terms of a comprehensive multi-dimensional taxonomy. This taxonomy is engineered to analyze and classify how autonomous LLM-powered multi-agent systems balance the interplay between autonomy and alignment across their system architectures.

A simplified overview of the dimensions and levels applied in our taxonomy is represented by the cuboid shown in Fig. 1. First, the synergy between autonomy and alignment manifests as a two-dimensional matrix with multiple hierarchical levels. This matrix captures a spectrum of nine distinct system configurations, ranging from systems that strictly adhere to predefined mechanisms (*rule-driven automation*, L0/L0) to those that dynamically adapt in real-time, guided by evolving conditions and user feedback (*user-responsive autonomy*, L2/L2).

¹An **extended paper version** (Händler, 2023) is available at <https://doi.org/10.48550/arXiv.2310.03659>.

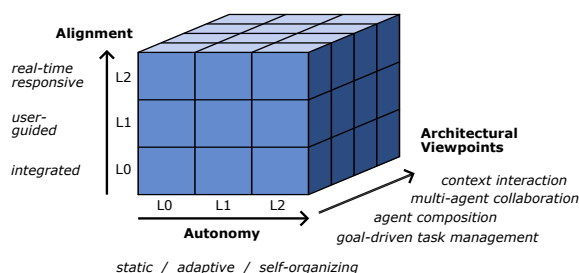


Figure 1: A simplified representation of the proposed multi-dimensional taxonomy for autonomous LLM-powered multi-agent systems. The x-axis represents the level of autonomy, the y-axis the level of alignment, and the z-axis the four applied architectural viewpoints.

Second, these configuration options are applied to multiple distinct architectural viewpoints (Kruchten, 1995), such as the system’s functionality (*goal-driven task management*), its internal structure (*agent composition*), its dynamic interactions (*multi-agent collaboration*) as well as the involvement of contextual resources such as tools and data (*context interaction*). Stemming from these four viewpoints, we have discerned 12 architectural aspects, each with distinct autonomy and alignment levels. This granularity facilitates a nuanced analysis of the system’s architectural dynamics resulting from the interplay between autonomy and alignment across the system architecture, laying the foundations for further analysis and reasoning about design decisions. The contributions of this paper can be categorized as follows:

- 1. Multi-Dimensional Taxonomy.** We introduce a comprehensive multi-dimensional taxonomy tailored to analyze and understand how autonomous LLM-powered multi-agent architectures balance the dynamic interplay between autonomy and alignment across different architectural aspects. For this purpose, our taxonomy provides hierarchical levels for both autonomy and alignment, which are applied to distinct architectural viewpoints and aspects, thus incorporating a third dimension.
- 2. Taxonomic Classification of Selected Systems.** We demonstrate the utility of our taxonomy by classifying a selection of seven autonomous LLM-powered multi-agent systems, which provides insights into the architectural dynamics of the analyzed systems and identifies challenges and development potentials. The taxonomic application also serves as a first empirical validation.

Structure of the Paper. The remainder of this paper is structured as follows. Section 2 gives a short overview of related background. In Section 3, we introduce our multi-dimensional taxonomy, incorporat-

ing specifications of autonomy and alignment levels and their application to the system architecture. By analyzing selected multi-agent systems, Section 4 illustrates the utility of our taxonomy. Finally, Section 5 discusses key insights and concludes the paper.

2 BACKGROUND

2.1 Related Work

Existing Taxonomies. Taxonomies represent structured classification schemes employed to categorize objects in a hierarchical manner according to specific criteria. They find applications in a wide range of disciplines and domains. The field of agent systems spans a variety of configurations and operational structures, with some systems operating as individual entities and others involving multiple interacting agents.

- **Taxonomies for Autonomous Systems** mainly categorize systems based on the level and type of autonomy, intelligence, learning capabilities, and ability to interact with their environment (Wooldridge and Jennings, 1995; Brustoloni, 1991; Maes, 1995; Franklin and Graesser, 1996; Tosic and Agha, 2004).
- **Taxonomies for Multi-Agent Systems** extend beyond the confines of individual agent characteristics, integrating the dynamics of interactions and collaborations among multiple agents (Bird, 1993; Dudek et al., 1996; Van Dyke Parunak et al., 2004; Moya and Tolks, 2007).

While these taxonomies have contributed significantly to our understanding of autonomous agents and multi-agent systems, they were developed prior to the advent of large language models (LLMs), and thus fall short in providing means to categorize and understand the specific challenges and involved architectural complexities posed by autonomous LLM-powered multi-agent systems. Moreover, while the concepts of autonomy and alignment are often discussed in AI literature (Narendra and Annaswamy, 2012; Russell, 2019) and also the system’s architecture plays a fundamental role in software engineering (Bass et al., 2003), none of these existing taxonomies has so far applied a systematic approach to either investigate architectural aspects or combine the concepts of autonomy and alignment.

Current LLM-powered Multi-Agent Systems. In response to limitations of large language models (LLMs) handling task complexity (Kaddour et al., 2023), autonomous multi-agent systems utilizing the

reasoning abilities of LLMs have emerged (see Section 2.2). Currently, several projects are established that aim at realizing such autonomous AI architectures for accomplishing complex tasks based on multiple interacting agents and powered by large language models (LLMs). Exemplary but representative autonomous multi-agent systems are AUTO-GPT (Torantulino et al., 2023), BABYAGI (Nakajima, 2023), SUPERAGI (TransformerOptimus et al., 2023), HUGGINGGPT (Shen et al., 2023), CAMEL (Li et al., 2023), AGENTGPT (Shrestha et al., 2023) and METAGPT (Hong et al., 2023). A recent survey is provided by (Wang et al., 2023), which focuses on investigating and comparing the agents’ characteristics and capabilities in terms of profile generation, memory operations and structures, planning, tool integration and learning strategies. Complementing this, another recent survey (Xi et al., 2023) offers an overview of existing approaches, contextualizing them with foundational technical, methodical, and conceptual paradigms. However, as we dive into the specifics of current autonomous LLM-powered multi-agent systems, striking the right balance between autonomy and alignment emerges as a central challenge. Given the exploratory state of the field, current systems exhibit a wide range of architectures, each with its unique strategy for balancing autonomy and alignment dispersed across various architectural components and mechanisms. The complexity of these systems underscores the importance of a taxonomy that can provide a structured understanding and comparison of these systems.

2.2 System Characteristics

In the following, we shortly outline the main architectural characteristics of autonomous LLM-powered multi-agent systems, as illustrated in Fig. 2.

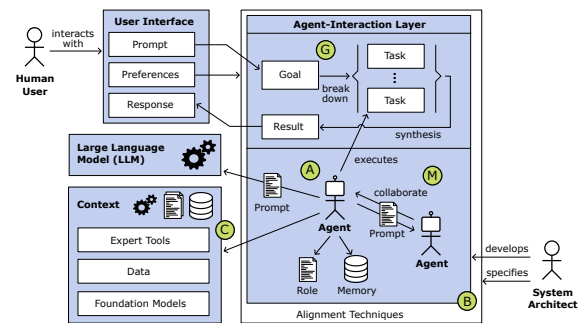


Figure 2: Overview of the primary characteristics of autonomous LLM-powered multi-agent systems, enhanced by contextual resources like tools and data.

- **Goal-Driven Task Management.** Autonomous LLM-powered multi-agent systems are designed

to accomplish user-prompted goals or complex tasks. For this purpose, the system employs an interactive and multi-perspective strategy, by breaking down complex tasks into smaller, manageable tasks, which are subsequently distributed among various agents, each equipped with specific competencies. A crucial aspect of this *divide & conquer* strategy lies in the effective orchestration and the subsequent synthesis of partial results.

- A LLM-Powered Intelligent Agents.** Intelligent agents structure the system as the foundational components. Each agent is endowed with a unique set of competencies, which include a clearly defined role and an individual memory. The backbone of their reasoning and interpretative capabilities is rooted in the incorporation of large language models (LLMs). This enables the agents not only to reflect upon the tasks or to plan and process the assigned tasks efficiently, but also to access and utilize contextual resources, as well as to communicate with other agents.
- M Multi-Agent Collaboration.** The interaction layer provides the workspace for a network of LLM-powered agents. While executing the assigned tasks, these agents collaborate with each other via prompt-driven message exchanges to delegate responsibilities, seek assistance, or evaluate task results. Key to the agents' collaboration is to effectively combine the strengths of each agent (*cognitive synergy*). The power of these systems emerges from the coordinated efforts of the collective (*society of mind* (Minsky, 1988)).
- C Context Interaction.** Some tasks require the utilization of contextual resources, such as expert tools, data, further specialized foundation models, or other applications. These resources extend the agents' ability to gather environmental information, create or modify artefacts, or initiate external processes, thus enables the agents to effectively execute complex tasks.
- B Balancing Autonomy and Alignment.** The dynamics of LLM-powered multi-agent systems are characterized by a complex interplay between autonomy and alignment. As captured in Fig. 3, this complexity can be traced back to the triadic interplay and inherent tensions among human users, LLM-powered agents, and governing mechanisms or rules integrated into the system. *Alignment*, in this context, ensures that the system's actions are in sync with human intentions and values. On the other side of the spectrum, *autonomy* denotes the agents' inherent capacity for self-organized strategy and operation, allow-

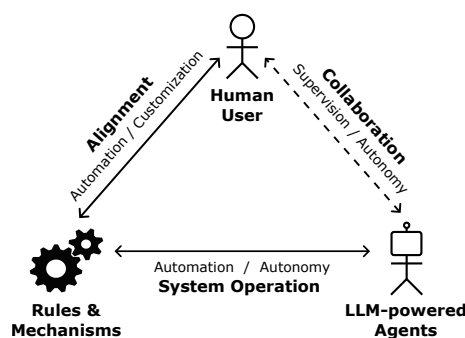


Figure 3: Triadic interplay and dynamic tensions between the decision-making entities in LLM-powered multi-agent systems.

ing them to function independent of predefined *rules and mechanism* and without human *supervision*. Moreover, in systems steered by user-prompted goals, it becomes pivotal to distinct between generic alignment aspects, in terms of *mechanisms* predefined by system architects, and user-specific preferences *customized* by system users. However, from an architectural perspective, autonomy and alignment transform into *cross-cutting concerns* (Kiczales et al., 1997). They traverse components and mechanisms across the architectural infrastructure and dynamics. Achieving a balanced configuration of autonomy and alignment is a crucial challenge, which directly impacts the system's efficiency and effectiveness.

For a comprehensive discussion of related work as well as for a detailed specification and analysis of relevant architectural concepts, please refer to extended paper version (Händler, 2023).

3 MULTI-DIMENSIONAL TAXONOMY

In this section, we introduce the system of our multi-dimensional taxonomy, engineered to methodically analyze the interplay between autonomy and alignment across architectures of autonomous LLM-powered multi-agent systems. The taxonomy weaves three crucial dimensions, i.e. hierarchical levels of autonomy and alignment as well as architectural viewpoints. Together, they form a three-dimensional matrix for classifying system configurations (see Fig. 1).

Section 3.1 delves into the complexities of the interplay between autonomy and alignment. Subsequently, Section 3.2, underscores the importance of incorporating architectural viewpoints into the taxonomic system. Rather than applying the autonomy-alignment matrix flatly, we propose analyzing each

architectural viewpoint as well as further inherent architectural characteristics individually. Such a viewpoint-focused approach allows for a deeper and more nuanced understanding of the systems, reflecting the architectural dynamics and complexities. Finally, in Section 3.3, we unify these components, mapping the autonomy-alignment dimensions and levels onto aspects inherent to the viewpoints.

3.1 Interplay Between Autonomy and Alignment

Autonomy and alignment, as interdependent and interplaying concepts, have their roots in management sciences and organizational behavior, playing integral roles in the ways teams and systems function (Mintzberg, 1989; O’reilly Iii and Tushman, 2008). In these fields, autonomy typically refers to the degree of discretion employees or teams possess over their tasks, while alignment denotes the degree to which these tasks correspond to the organization’s overall objectives. In the field of AI, the interplay between autonomy and alignment remains pivotal (Russell et al., 2015; Bostrom, 2017). AI systems, by nature, operate with varying degrees of independence and are often designed to accomplish complex tasks that are potentially beyond human capabilities. However, uncontrolled autonomy can pose risks. If the goals of an AI system deviate from those of its human supervisors, it could lead to unforeseen consequences or uncontrollable side effects. As such, understanding and defining the bounds of autonomy and alignment becomes essential for effective system operation. For this purpose, we adopt a pragmatic perspective on both autonomy and alignment.

Table 1: Matrix showcasing the interplay between gradations of alignment (*vertical*) and autonomy (*horizontal*) in the context of LLM-powered multi-agent architectures.

Levels of Autonomy & Alignment	L0: Static	L1: Adaptive	L2: Self-Organizing
L2: Real-time Responsive	User-Supervised Automation	User-Collaborative Adaptation	User-Responsive Autonomy
L1: User-Guided	User-Guided Automation	User-Guided Adaptation	User-Guided Autonomy
L0: Integrated	Rule-Driven Automation	Pre-Configured Adaptation	Bounded Autonomy

3.1.1 Autonomy

The degree of autonomy refers to the extent to which an AI system can make decisions and act independently of rules and mechanisms defined by humans.

For LLM-powered multi-agent systems, this translates to a system’s proficiency in addressing the goals or tasks specified by the user in a self-organizing manner, adapting and re-calibrating to the complexities of a given situation. Autonomous multi-agent systems are *by nature* striving for this end-to-end automatic goal completion and task management from a user perspective. Automation pertains to tasks being carried out without human input (Brustoloni, 1991; SAE International, 2016), while autonomy pertains to *decisions about tasks* being made without human intervention (Franklin and Graesser, 1996; Parasuraman et al., 2000; Beer et al., 2014). In the domain of LLM-powered multi-agent systems, we look beyond mere task automation, focusing on how these systems internally manage their dynamics to fulfill user objectives. Our taxonomy, therefore, distinguishes systems on a spectrum of autonomy. Drawing from the triadic interplay (Fig. 3), on the one end of the spectrum, we see systems that heavily rely on predefined rules, set by system architects. While they may execute tasks autonomously, their decision-making process is constrained within a fixed set of parameters (*low autonomy*). On the other hand, we encounter systems characterized by their ability for self-organisation and dynamic self-adaptation. Rather than relying on hard-coded mechanisms, they harness the power of LLMs to interpret, decide, and act, making them more adaptable to changing situations (*high autonomy*).

Autonomy Levels. The levels of autonomy, represented on the x-axis in our matrix (see Fig. 1 and Table 1), articulate the degree of agency of the LLM-powered agents in making decisions regarding the system operation, independently from predefined mechanisms.

L0: Static Autonomy - At this foundational level, systems are primarily automated, relying heavily on the rules, conditions, and mechanisms embedded by system architects. The systems follow defined rules and predetermined mechanisms. While the agents are not empowered to modify these rules, some degree of flexibility remains resulting from rule-based options and alternatives.

L1: Adaptive Autonomy - Evolving from the static level, systems at this stage possess the capability to adapt their behavior within a structure and procedural guidelines established by the system architects. The LLM-powered agents are capable of adjusting the system’s operations within this provided framework (such as flexible infrastructures and protocols) due to the needs of the given application scenarios, but not beyond.

L2: Self-Organizing Autonomy - At this high-

est level of autonomy, LLM-powered agents emerge as the principal actors, capable of self-organization, actively learning and dynamically tailoring their operations in real-time based on environmental cues and experiences. However, this might also include highly generic infrastructures that are modifiable by the LLM-powered agents and thus allow self organisation.

3.1.2 Alignment

In the context of AI, the term alignment traditionally refers to the challenge of ensuring that an AI system’s behavior aligns with human intentions, values or goals. This intricate problem, often framed as the *control problem*, is a cornerstone of AI safety discourse (Bostrom, 2017; Russell, 2019). However, when viewed through a practical lens, especially in the context of autonomous LLM-powered multi-agent systems, the alignment paradigm acquires a more interactive, user-centric perspective (Amodei et al., 2016), as it can be seen as a calibration of conditions tied to user-prompted goals. This includes preferences, policies, constraints, and boundaries which collectively steer or regulate the system’s trajectory towards achieving its set targets. Importantly, within this framework, alignment is not seen as counter to autonomy. Instead, it acts to complement and refine it, being applicable across various levels of autonomy.

For our taxonomy, we combine two important dimensions of alignment: its origin and timing, reflecting the dynamic tension between automated alignment mechanisms and human customization, as illustrated in Fig. 3. The origin delves into who dictates the alignment, the system architect or the system user. Meanwhile, timing refers to when the alignment is specified, encompassing phases like pre-deployment, post-deployment but prior to runtime, or even during runtime. Furthermore, we’ve categorized alignment into levels. The base level, or *low alignment level*, signifies alignment that’s already embedded into the system’s design by the system architects. This intrinsic alignment sets broad behavioral boundaries without focusing on specific user preferences. On the other hand, the *high alignment levels* are more adaptable and centered around user-specified alignment. Here, users have the flexibility to set their preferences either before the system enters its runtime or, ultimately, during its active operation.

Alignment Levels. The levels of alignment, represented on the y-axis in our matrix (see Fig. 1 and Table 1), measure the degree to which users of the system can influence or adjust the system’s behavior.

L0: Integrated Alignment - At this foundational level, the alignment techniques are built directly

into the system’s architecture. In such system, alignment mechanisms are static and rule-driven, and cannot be altered by the users.

L1: User-Guided Alignment - Evolving from the previous level, the User-Guided Alignment offers a degree of customization. This level empowers users by allowing them to set or adjust specific alignment parameters, such as conditions, rules, or boundaries, before the system starts its operation. These interactions are primarily facilitated via user interfaces designed to capture user preferences in a structured manner.

L2: Real-Time Responsive Alignment - The highest level of alignment is represented by means to adjust the system’s behavior in real-time. Thanks to integrated real-time monitoring mechanisms, the system can actively solicit user feedback user decisions at critical junctures or decision points. This responsiveness enables a high level of collaboration in terms of ongoing feedback between the user and the system.

3.1.3 Combinations of Autonomy and Alignment

By combining these two dimensions in our matrix, we provide a comprehensive view of the interplay between diverse gradations of autonomy and alignment within LLM-powered multi-agent systems. Table 1 gives an overview of the employed levels and the resulting spectrum of potential combinations.

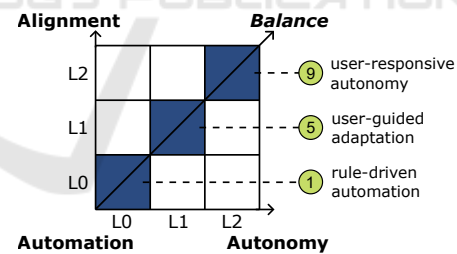


Figure 4: Interplay between autonomy and alignment: balancing evolving levels of dynamism and responsibilities of both LLM-powered agents (*autonomy*) and human users (*alignment*).

As illustrated in Fig. 4, departing from static and rule-driven system configurations (*automation*), this autonomy-alignment matrix captures the progression of dynamism and responsibilities as we move along the axes. On the y-axis, alignment levels represent the gradation of human users’ involvement—from integrated systems where the user’s role is passive (L0), to real-time responsive setups demanding active participation (L2). On the x-axis, the autonomy levels signify the evolving capabilities of LLM-powered agents, progressing from static behaviors

(L0) to adaptive (L1) and, ultimately, self-organizing mechanisms (L2). This matrix structure reflects the triadic interplay and dynamic tensions illustrated in Fig. 3. As we delve deeper into the matrix, the challenge becomes evident: ensuring balance between the evolving responsibilities of LLM-powered agents and the goals and intentions by the human users, ultimately resulting in a dynamic collaboration between agents and humans.

3.2 Architectural Viewpoints

Architectural viewpoints are a structured means to analyze and assess complex systems from diverse perspectives focusing on selected aspects and layers of an architecture (Bass et al., 2003; Clements et al., 2003). Central to these viewpoints is the consideration of stakeholder concerns, which inform and determine the highlighted aspects and their interrelations in each viewpoint. Providing a combined multi-perspective analysis, viewpoints serve as an effective framework to examine the structures and dynamics of software architectures. For our taxonomy, we leverage viewpoints on autonomous LLM-powered multi-agent systems. Rather than mapping the autonomy-alignment taxonomy flatly onto the system, which oversimplifies the multi-faceted nature of these systems, analyzing each architectural viewpoint individually offers a tailored lens, enabling to comprehend the role and impact of autonomy and alignment within the system. Each viewpoint reveals distinct insights into the system’s behavior, internal interactions, composition, and context interaction, leading to a more nuanced and comprehensive classification (Rozanski and Woods, 2012).

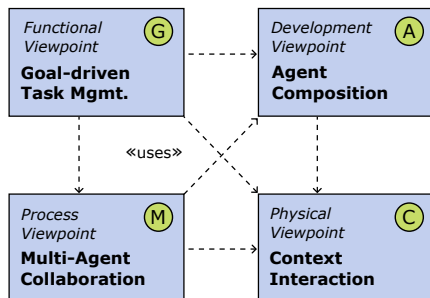


Figure 5: Architectural viewpoints oriented to the 4+1 view model of software architecture (Kruchten, 1995) applied to autonomous LLM-powered multi-agent systems.

3.2.1 Applied Viewpoints

For our taxonomy, we orient to Kruchten’s renowned 4+1 view model of software architecture (Kruchten, 1995), an established standard viewpoint model for software architecture, adapting it to suit the archi-

tectural characteristics of LLM-powered multi-agent systems (see Section 2.2). Our taxonomy encompasses the following four architectural viewpoints on these systems (refer to Fig. 5 and 2):

- G Goal-Driven Task Management (Functional Viewpoint):** Kruchten’s functional viewpoint refers to the system’s visible functionalities as experienced by its users (Kruchten, 1995). In the context of autonomous LLM-powered multi-agent systems, we see Goal-driven Task Management as a manifestation of this functional viewpoint. It entails the system’s capabilities and mechanisms to decompose user-prompted goals or complex tasks into manageable tasks, and subsequently, orchestrate task execution, combine the results, and deliver the final result forming the response.
- A Agent Composition (Development Viewpoint):** According to Kruchten, the development viewpoint is primarily focusing on the system’s software architecture, the breakdown into components, and their organization (Kruchten, 1995). In our context, we interpret this as Agent Composition, focusing on the system’s internal composition, particularly the assembly and constellation of agents. It includes the types and roles of agents, their memory usage, the relationships between agents.
- M Multi-Agent Collaboration (Process Viewpoint):** Kruchten’s process viewpoint concerns the dynamic aspects of a system, specifically the system procedures and interactions between components (Kruchten, 1995). We apply this to the Multi-Agent Collaboration in our model, emphasizing the collaborative task execution and interactions among agents. This encompasses the application of communication protocols, the dynamics of actions management, such as the actual task execution, mutual task delegation, as well as the evaluation and merging of task results on agent level, as well as the management of communication components and prompt engineering.
- C Context Interaction (Physical Viewpoint):** According to Kruchten, the physical viewpoint involves the system’s mapping to physical resources (Kruchten, 1995). We extend this to Context Interaction, focusing on the system’s interaction with the external environment. It includes how the system acquires, integrates, and utilizes contextual resources such as external data, expert tools, and further foundation models as well as the organized distribution and utilization of contextual resources within the agent network.

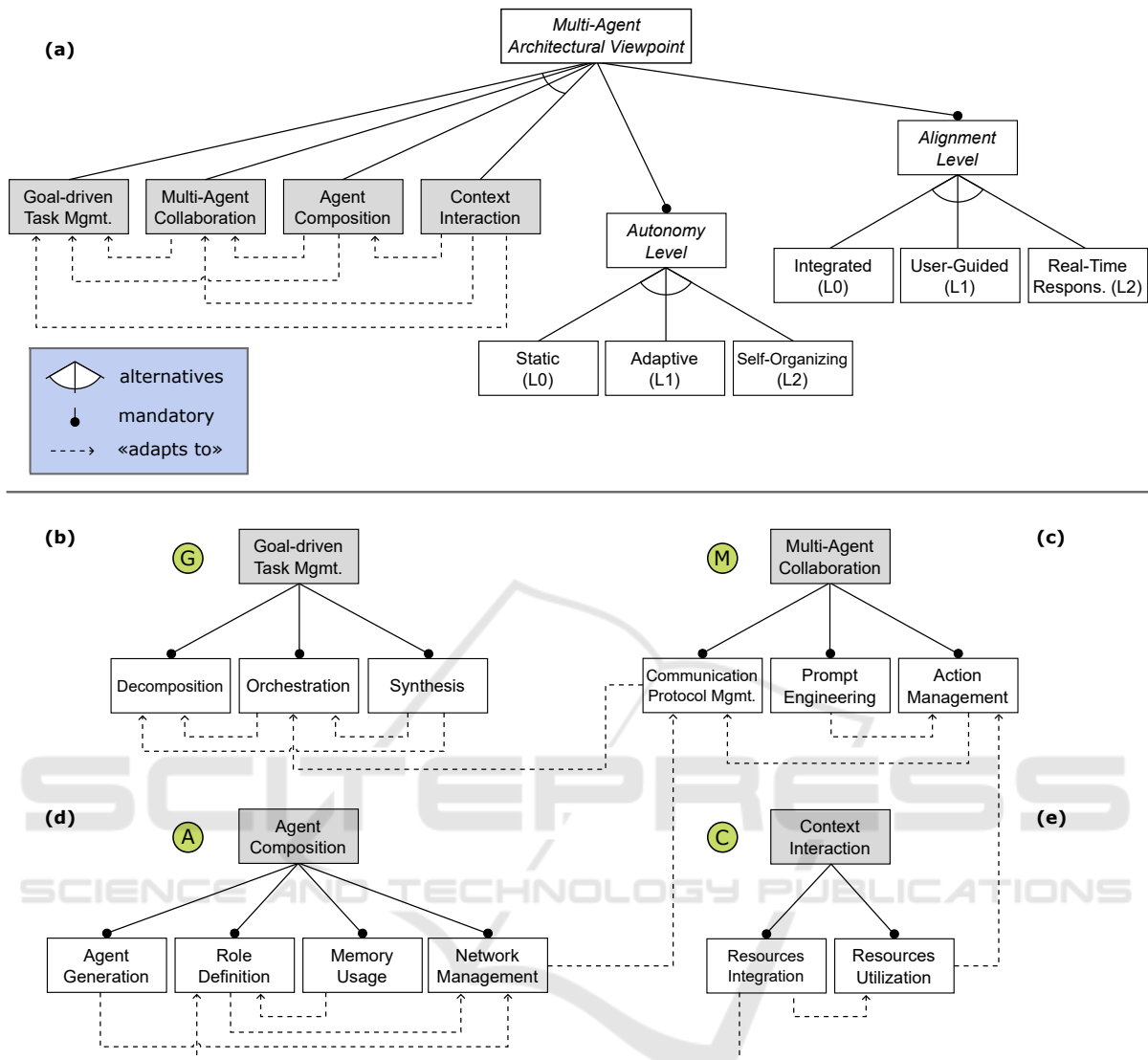


Figure 6: Feature diagram showcasing the taxonomic structure. Each viewpoint integrates autonomy and alignment levels (a). The diagram further illustrates viewpoint-specific aspects and mechanisms (b–e).

3.3 Interplay of Autonomy and Alignment in the System Architecture

As already illustrated, both autonomy and alignment serve as *cross-cutting concerns* (Kiczales et al., 1997) impacting the operational efficiency of various architectural aspects across LLM-powered multi-agent systems. Thus, in the following, we map our matrix of autonomy and alignment levels onto the architectural viewpoints. This projection crafts a three-dimensional matrix, offering a prism through which these systems can be analyzed and categorized (also see Fig. 1). (Händler, 2023) provides a systematic

overview of the resulting viewpoint-specific combinations of autonomy and alignment levels.

Viewpoint-Specific Aspects and Level Criteria. As outlined above, architectural viewpoints provide means to analyze certain aspects and aspect relations of the system’s architecture in a multi-perspective manner (Rozanski and Woods, 2012). We now systematize the viewpoint-specific aspects employed in our taxonomy.

Fig. 6 gives an overview of our taxonomy’s characteristics, structured through a feature diagram (Batory, 2005; Schobbens et al., 2007). In particular, Fig. 6 (a) structures the viewpoint-specific taxonomic structure. Each of the four integrated viewpoints provides a certain combination of autonomy

and alignment levels. As illustrated in Figs. 6 (b–e), this structure is refined by viewpoint-specific aspects and their interdependencies in terms of requirements-driven dependencies (*adapts-to*), presuming a high-autonomy system configuration (Händler, 2023). These dependencies suggest that the capabilities of a dependent aspect evolve in line with the needs and stipulations of the aspect it points to. In turn, also these viewpoint-specific aspects can be assessed by the autonomy and alignment levels, resulting in a more nuanced taxonomic classification.

For a comprehensive specification of level criteria for both autonomy and alignment related to each aspect, as well as for a discussion of the interdependencies among these aspects, please refer to (Händler, 2023).

4 CLASSIFICATION OF SELECTED SYSTEMS

In order to demonstrate the practical utility of our taxonomy, we analyze and classify selected existing autonomous LLM-powered multi-agent systems. We have chosen a set of seven state-of-the-art multi-agent systems for this assessment: AUTOGPT (Torantulino et al., 2023), BABYAGI (Nakajima, 2023), SUPERAGI (TransformerOptimus et al., 2023), HUGGINGGPT (Shen et al., 2023), METAGPT (Hong et al., 2023), CAMEL (Li et al., 2023), and AGENTGPT (Shrestha et al., 2023). Each of these systems is maintained and available as open-source project. For each selected system, we gathered relevant information by examining the technical documentation and research papers, where available, as well as reviewing the code base. We further engaged with each system to explore its real-time functionalities, with emphasis on alignment mechanisms available before and during runtime.

Taxonomic Classification. The taxonomic classification relies on a detailed assessment of autonomy and alignment levels for viewpoint-specific aspects of the systems. Table 2 reports on the results of assessing these levels of autonomy (AU) and alignment (AL) for aspects characterizing the four architectural viewpoints applied by our taxonomy. In particular, for Goal-driven Task Management, the aspects of decomposition (Decom), orchestration (Orch), and synthesis (Synth); for Multi-Agent Collaboration, the aspects of communication-protocol management (CommP), prompt engineering (PrEng), and action management (ActM); for Agent Composition, the aspects of agent generation (AGen), role definition (RoleD), memory usage (MemU), and network manage-

ment (NetM); for Context Interaction, the aspects of resource integration (Integ), and resource utilization Util are distinguished. The viewpoint-specific aspects with corresponding level criteria applied for the assessment are detailed in (Händler, 2023).

Fig. 7 displays the derived autonomy and alignment levels per multi-agent system using radar (or spider) charts (Tufté, 2001). In particular, architectural aspects form the multiple axes. The level scheme (L0, L1, L2) for autonomy and alignment is depicted by grey circles linking these axes. The blue graph then represents the assessed autonomy levels, the green dashed graph the corresponding alignment levels. The extended paper version (Händler, 2023) provides a detailed discussion of analysis results per system as well as a comprehensive comparative analysis.

Strategies Across System Groups. We now explore how different categories of systems balance the interplay between autonomy and alignment. Based on our taxonomic classification and the resulting system profiles as illustrated in Fig. 7, we can categorize the selected 7 systems under analysis into three distinct system groups, which encompass general-purpose systems, central-controller systems, and role-agent systems. It’s important to note that our categorization into these three groups, based on the systems chosen for this exploration, doesn’t capture the entire spectrum of autonomous LLM-powered multi-agent systems. For a comprehensive overview of existing systems, we recommend referring to the recent surveys provided by (Wang et al., 2023; Xi et al., 2023). In the following, the key characteristics as observed from the corresponding system profiles are discussed.

- **General-Purpose Systems** - representing multi-agent systems designed for and adaptable to a broad spectrum of tasks and applications. Within the analyzed set of multi-agent systems, the following fall into this group: AUTO-GPT (Torantulino et al., 2023), BABYAGI (Nakajima, 2023), SUPERAGI (TransformerOptimus et al., 2023), and AGENTGPT (Shrestha et al., 2023). Goals are decomposed autonomously and represented as prioritized task lists (L2 Decom). They employ a multi-cycle process framework performed by dedicated task-management agents represented by certain generic agent types, including a single task-execution agent. Relations and communications between these agents are strictly predefined, and agent conversations express as a monologue of the task-execution agent, resulting in low autonomy levels (L0) for communication protocol (CommP), and network management (NetM). The task-related actions are performed autonomously

Table 2: Assessment of autonomy (AU) and alignment (AL) levels across viewpoint-specific aspects of selected LLM-powered multi-agent systems. Detailed level criteria for viewpoint-specific aspects are discussed in (Händler, 2023). * ZAPIER, a workflow-automation tool, has been included to contrast the results.

LLM-powered Multi-Agent Systems	Goal-driven Task Mgmt.						Multi-Agent Collaboration						Agent Composition						Context Interact.					
	Decom		Orch		Synth		CommP		PrEng		ActM		AGen		RoleD		MemU		NetM		Integ		Util	
	AU	AL	AU	AL	AU	AL	AU	AL	AU	AL	AU	AL	AU	AL	AU	AL	AU	AL	AU	AL	AU	AL	AU	AL
Auto-GPT (Torantulino et al., 2023)	2	0	0	0	1	0	0	0	1	0	2	0	0	0	1	0	0	0	0	0	0	0	2	0
BabyAGI (Nakajima, 2023)	2	0	0	0	1	0	0	0	1	0	2	0	0	0	1	0	0	0	0	0	0	0	2	0
SuperAGI (TransformerOptimus et al., 2023)	2	0	1	0	1	1	0	0	1	0	2	0	1	1	2	1	0	1	0	0	0	1	2	1
HuggingGPT (Shen et al., 2023)	2	0	1	0	2	0	0	0	2	0	2	0	2	0	2	0	1	0	0	0	2	0	2	0
MetaGPT (Hong et al., 2023)	2	0	0	0	2	0	1	0	1	0	2	0	0	0	0	0	0	0	1	0	0	0	2	0
CAMEL (Li et al., 2023)	2	0	0	0	1	0	0	0	1	0	1	0	0	1	1	1	0	0	0	1	0	0	0	0
AgentGPT (Shrestha et al., 2023)	2	1	1	0	1	0	0	0	1	0	2	0	1	1	2	0	0	0	0	0	0	0	2	1
Zapier* (Rahmati et al., 2017)	1	1	0	1	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1

by the task-execution agent (mostly L2 autonomy ActM). While resource integration is based on provided mechanisms (Integ), the resources are selected and utilized by the LLM-powered in a self-organizing manner (L2 autonomy for Util), except for CAMEL; resulting in similar autonomy profiles for the aforementioned aspects. Besides from these commonalities, these systems distinguish in certain characteristics. Both AUTO-GPT and BABYAGI employ generic task-execution agent, and provide no further alignment options at all. Moreover, these systems employ a generic task-execution agent with predefined agent roles and relations, resulting in L0 autonomy for AGen and NetM. In contrast, SUPER-AGI and AGENTGPT employ execution agents with self-organizing agent roles (L2 autonomy for RoleD), an adaptable orchestration process (L1 for Orch), and some alignment options, especially for agent-specific aspects. Moreover, these systems employ execution agents, whose roles can be customized by the user (L1 alignment for AGen).

- **Central LLM Controller** - marks a third group specialized in leveraging and combining contextual resources for accomplishing complex goals. HUGGINGGPT (Shen et al., 2023) serves as an archetype of such systems, utilizing resources especially in terms of existing ML models integrated via HUGGING FACE. HUGGINGGPT is characterized by a single central LLM-powered

control agent with monologue-based reflection and planning. Language serves as generic interface to manage the interplay between multiple specialized foundation models. In comparison to other systems or system groups, we see the highest levels of autonomy granted to this central agent (mostly L2); also see Fig. 7 (d). Furthermore, we see a finite and artefact-oriented process adaptable by the LLM-powered agent for orchestrating the different model-related tasks (L1 autonomy). Beyond prompting the task, there are no further user-centric alignment options (L0 alignment).

- **Role-Agent Systems** - employ an interplay or simulation between multiple dedicated roles agents. This collaboration can serve different purposes, such as simulating a discussion or solving tasks that demand for a multi-perspective collaboration. With defined roles in a certain environment (such as in a software development project), their application is bound to this application domain or special purpose. Among the analyzed systems, METAGPT (Hong et al., 2023) and CAMEL (Li et al., 2023) represent such systems. In contrast to the general-purpose systems, the execution agents play roles with dedicated responsibilities in a certain application domain. Furthermore, these role agents actually collaborate directly with each other. In case of the two exemplary systems, this collaboration is realized by communication pro-

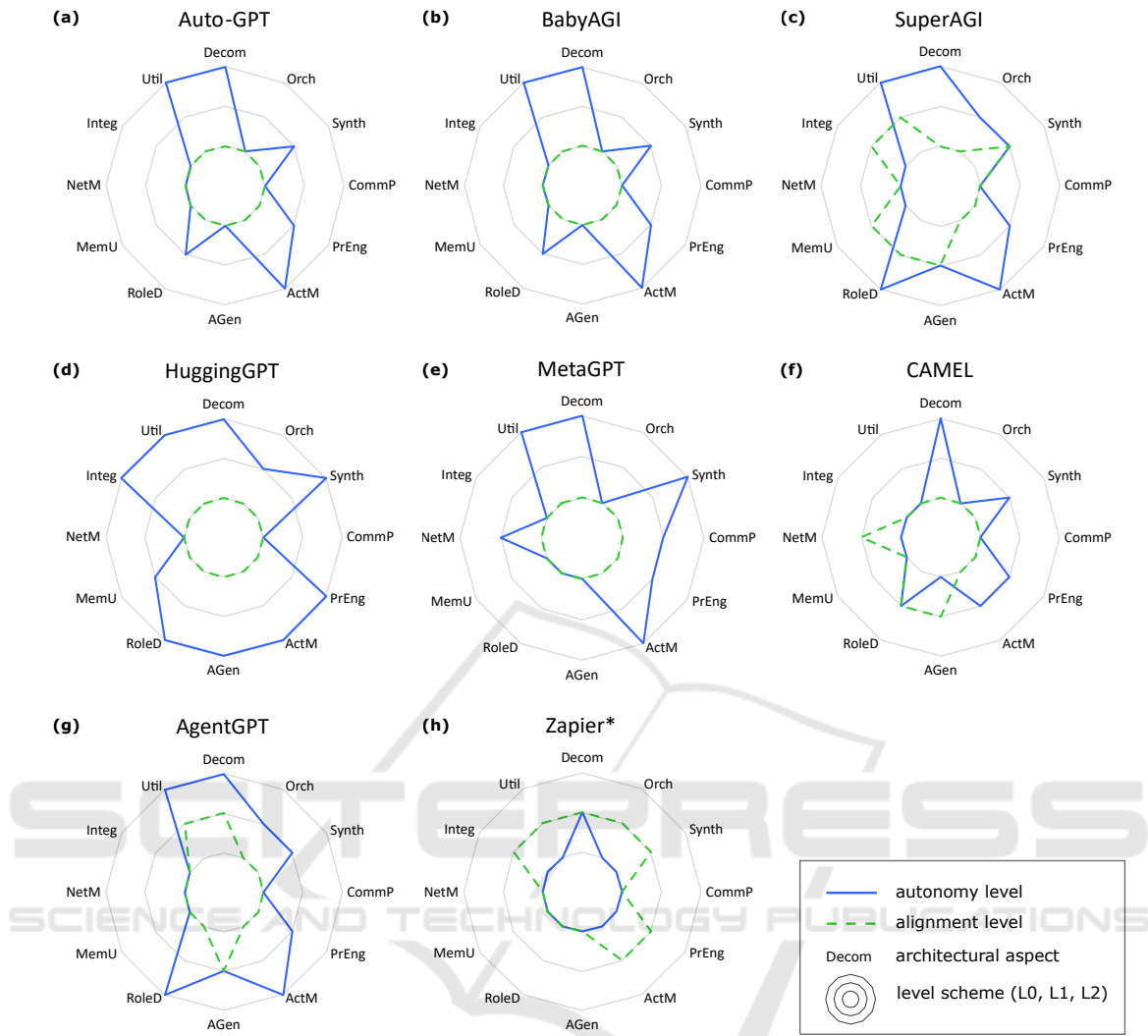


Figure 7: Radar charts illustrating the system profiles based on an assessment of architectural aspects in terms of autonomy (blue graph) and alignment (green dashed graph) levels. Detailed assessment data can be found in Table 2.

tools employing a dynamic exchange between agents with instructor and executor roles. In particular, CAMEL employs two such role agents based on predefined agent types, but adjustable by the user. In ongoing strict dialogue cycles, the AI-user role agent instructs the AI-assistant role agent to execute the tasks (L0 autonomy for CommP). Similar to SUPERAGI, CAMEL requires the user to specify the agents' roles (L1 alignment). METAGPT, in contrast, internally assigns predefined roles with responsibilities alongside a waterfall development process (L0 alignment); thus, also expressing a finite and artefact-oriented process (L0 autonomy for Orch), terminating with the produced and tested software program. However, like in real-world software project, refinement iterations can follow, optional feedback cy-

cles make it adaptable for the agents (L1 autonomy for CommP).

Strategy Assessment. Beyond differences in the applied communication protocols, it is the flexibility of agent roles (in relation to both autonomy and alignment) and further customization options for agent-specific aspects that distinguishes the systems' strategies (see above). However, when examining how the systems deal with autonomy and alignment across further aspects, most systems and system groups show similar strategies. The reasoning capabilities of LLM-powered agents are especially leveraged in areas demanding high autonomy, such as the goal decomposition, the actual execution of task-related actions, and the utilization of contextual resources. Interestingly, these high-autonomy aspects are mostly combined with low alignment levels, resulting in

bounded autonomy aspects (refer to Table 1). A closer look at aspect interdependencies, as depicted in Fig. 6, reveals that these internally *unbalanced* aspects are accompanied by other low-autonomy aspects equipped with limited flexibility. For instance, autonomous action management depends on strict or predefined communication protocol. For further details, refer to (Händler, 2023). In these cases, the predefined and rule-based mechanisms serve as integrated alignment guiding and controlling the accurate operation of the dependent autonomous aspects.

5 CONCLUSION

In this paper, we have introduced a comprehensive multi-dimensional taxonomy engineered to analyze how autonomous LLM-powered multi-agent systems balance the dynamic interplay between autonomy and alignment across their system architectures. For this purpose, the taxonomy employs a matrix that combines hierarchical levels of autonomy and alignment. This matrix is then mapped onto various architectural aspects organized by four architectural viewpoints reflecting different complementary concerns and perspectives. The resulting taxonomic system enables the assessment of interdependent aspect configurations in a wide spectrum, ranging from simple configurations, such as predefined mechanisms combined with system-integrated alignment (*rule-driven automation*), to sophisticated configurations, such as self-organizing agency responsive to user feedback and evolving conditions (*user-responsive autonomy*). Applied to 12 distinct architectural aspects inherent to viewpoints, such as goal-driven task management, multi-agent collaboration, agent composition, and context interaction, this taxonomy allows for a nuanced analysis and understanding of architectural complexities within autonomous LLM-powered multi-agent systems.²

Through our taxonomy's application to seven selected LLM-powered multi-agent systems, its practical relevance and utility has been illustrated. In particular, it has been shown that a combined assessment of autonomy and alignment levels across the architectural aspects of each multi-agent system allows for identifying system profiles that can indicate certain strategies for balancing the dynamic interplay between autonomy and alignment. This exploration of exemplary current systems also revealed several challenges, which are detailed in (Händler, 2023).

²For additional analyses, a more comprehensive discussion and extended results, readers are referred to the extended paper version available on arXiv (Händler, 2023).

Most prominently, we observed a lack of user-centric alignment options across all systems, with little user-guided alignment, but no real-time responsive alignment at all. Moreover, the systems exhibit high autonomy levels mostly for certain aspects, such as the goal decomposition, the action management, or the utilization of contextual resources. In contrast, other key aspects of the system operation show limited autonomy; aspects such as managing the communication protocol, memory usage, or agent network are largely static, leaning heavily on predefined mechanisms.

Based on these and further findings, we especially see two promising avenues for the evolution of autonomous LLM-powered multi-agent systems. Firstly, by employing adaptable and self-organizing communication protocols and agent networks, the systems' role-playing capabilities could be enhanced, which enables them to better simulate complex multi-perspective environments. By reflecting diverse standpoints and strategies, this could also pave the way for more in-depth inter-agent discussions and creativity in problem solving. Secondly, the exploration of real-time responsive systems, which can adapt to evolving conditions as well as to user feedback during runtime, would foster dynamic collaboration and hybrid teamwork between LLM-powered agents and human users.

Departing from an exploratory stage, the field of autonomous LLM-powered multi-agent systems is rapidly evolving, resulting in a growing number of promising approaches and innovative architectures. With their current capabilities and inherent potentials, such as multi-perspective domain simulations or collaborative environments of autonomous agents and human coworkers, these systems could significantly contribute to the progression towards advanced stages of artificial intelligence, such as AGI or ASI. From a pragmatic perspective, there are numerous opportunities for combining LLMs as general purpose technology with the specifics of various application domains. LLM-based multi-agent systems can serve as foundation for developing corresponding domain-specific application layers. The architectural complexities resulting from the dynamic interplay between autonomy and alignment can be seen as one of the key challenges in such systems. By providing a systematic framework for analyzing these complexities, our taxonomy aims to contribute to these ongoing efforts.

For our subsequent endeavors, we aim at developing a comprehensive overview and comparison of existing autonomous LLM-powered multi-agent systems, complementing existing literature reviews in the field (Wang et al., 2023; Xi et al., 2023). To this end,

we intend to analyze and classify available systems using our taxonomy. The identified system profiles and balancing strategies resulting from this analysis will then be combined with further investigations of functional system capabilities.

Building on the foundation of our taxonomy, future initiatives could venture into the following areas: A dedicated exploration and systematization of alignment techniques, particularly tailored for LLM-based interaction and application layers, could serve as reference for future systems. Moreover, the conception of a methodological framework with instruments and benchmarks for measuring the functional capabilities of LLM-powered multi-agent systems could provide a structured template to evaluate key metrics like efficiency, accuracy, and scalability of these systems.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the support from the "Gesellschaft für Forschungsförderung (GFF)" of Lower Austria, as this research was conducted at Ferdinand Porsche Mobile University of Applied Sciences (FERNFH) as part of the "Digital Transformation Hub" project funded by the GFF.

REFERENCES

- Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. (2016). Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.
- Bass, L., Clements, P., and Kazman, R. (2003). *Software architecture in practice*. Addison-Wesley Professional.
- Batory, D. (2005). Feature models, grammars, and propositional formulas. In *9th International Software Product Line Conference*, pages 7–20.
- Beer, J. M., Fisk, A. D., and Rogers, W. A. (2014). Toward a framework for levels of robot autonomy in human-robot interaction. *Journal of human-robot interaction*, 3(2):74.
- Bird, S. D. (1993). Toward a taxonomy of multi-agent systems. *International Journal of Man-Machine Studies*, 39(4):689–704.
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. (2021). On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.
- Bostrom, N. (2017). *Superintelligence*. Dunod.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Brustoloni, J. C. (1991). *Autonomous agents: Characterization and requirements*. Carnegie Mellon University.
- Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*.
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Clements, P., Garlan, D., Little, R., Nord, R., and Stafford, J. (2003). Documenting software architectures: views and beyond. In *25th International Conference on Software Engineering, 2003. Proceedings.*, pages 740–741. IEEE.
- Dudek, G., Jenkin, M. R., Milios, E., and Wilkes, D. (1996). A taxonomy for multi-agent robotics. *Autonomous Robots*, 3:375–397.
- Fabiano, F., Pallagani, V., Ganapini, M. B., Horesh, L., Loreggia, A., Murugesan, K., Rossi, F., and Srivastava, B. (2023). Fast and slow planning. *arXiv preprint arXiv:2303.04283*.
- Franklin, S. and Graesser, A. (1996). Is it an agent, or just a program?: A taxonomy for autonomous agents. In *International workshop on agent theories, architectures, and languages*, pages 21–35. Springer.
- Hong, S., Zheng, X., Chen, J., Cheng, Y., Zhang, C., Wang, Z., Yau, S. K. S., Lin, Z., Zhou, L., Ran, C., et al. (2023). MetaGPT: Meta programming for multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.
- Händler, T. (2023). Balancing autonomy and alignment: A multi-dimensional taxonomy for autonomous LLM-powered multi-agent architectures. *arXiv preprint arXiv:2310.03659*. <https://doi.org/10.48550/arXiv.2310.03659>.
- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.
- Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., and McHardy, R. (2023). Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.
- Kahneman, D. (2011). *Thinking, fast and slow*. Macmillan.
- Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.-M., and Irwin, J. (1997). Aspect-oriented programming. In *ECOOP'97—Object-Oriented Programming: 11th European Conference Jyväskylä, Finland, June 9–13, 1997 Proceedings 11*, pages 220–242. Springer.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.
- Kruchten, P. B. (1995). Architectural blueprints — the “4+1” view model of software architecture. *IEEE software*, 12(6):42–50.

- Li, G., Hammoud, H. A. A. K., Itani, H., Khizbullin, D., and Ghanem, B. (2023). CAMEL: Communicative agents for "mind" exploration of large scale language model society. *arXiv preprint arXiv:2303.17760*.
- Lin, B. Y., Fu, Y., Yang, K., Ammanabrolu, P., Brahman, F., Huang, S., Bhagavatula, C., Choi, Y., and Ren, X. (2023). SwiftSage: A generative agent with fast and slow thinking for complex interactive tasks. *arXiv preprint arXiv:2305.17390*.
- Maes, P. (1995). Artificial life meets entertainment: life-like autonomous agents. *Communications of the ACM*, 38(11):108–114.
- Maynez, J., Narayan, S., Bohnet, B., and McDonald, R. (2020). On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.
- Minsky, M. (1988). *The Society of mind*. Simon and Schuster.
- Mintzberg, H. (1989). *The structuring of organizations*. Springer.
- Moya, L. J. and Tolk, A. (2007). Towards a taxonomy of agents and multi-agent systems. In *SpringSim (2)*, pages 11–18.
- Nakajima, Y. (2023). BabyAGI. <https://github.com/yoheinakajima/babyagi>.
- Narendra, K. S. and Annaswamy, A. M. (2012). *Stable adaptive systems*. Courier Corporation.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.
- O'reilly Iii, C. A. and Tushman, M. L. (2008). Ambidexterity as a dynamic capability: Resolving the innovator's dilemma. *Research in organizational behavior*, 28:185–206.
- Parasuraman, R., Sheridan, T. B., and Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, 30(3):286–297.
- Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*.
- Rahmati, A., Fernandes, E., Jung, J., and Prakash, A. (2017). IFTTT vs. Zapier: A comparative study of trigger-action programming frameworks. *arXiv preprint arXiv:1709.02788*.
- Rozanski, N. and Woods, E. (2012). *Software systems architecture: working with stakeholders using viewpoints and perspectives*. Addison-Wesley.
- Russell, S. (2019). *Human compatible: Artificial intelligence and the problem of control*. Penguin.
- Russell, S. (2022). Artificial intelligence and the problem of control. *Perspectives on Digital Humanism*, page 19.
- Russell, S., Dewey, D., and Tegmark, M. (2015). Research priorities for robust and beneficial artificial intelligence. *AI magazine*, 36(4):105–114.
- SAE International (2016). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles.
- Schobbens, P.-Y., Heymans, P., Trigaux, J.-C., and Bontemp, Y. (2007). Generic semantics of feature diagrams. *Computer networks*, 51(2):456–479.
- Shen, Y., Song, K., Tan, X., Li, D., Lu, W., and Zhuang, Y. (2023). HuggingGPT: Solving AI tasks with ChatGPT and its friends in Hugging Face. *arXiv preprint arXiv:2303.17580*.
- Shrestha, A., Subedi, S., and Watkins, A. (2023). Agent-GPT. <https://github.com/reworkd/AgentGPT>.
- Sloman, S. A. (1996). The empirical case for two systems of reasoning. *Psychological bulletin*, 119(1):3.
- Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al. (2022). Llama: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Torantulino et al. (2023). Auto-GPT. <https://github.com/Significant-Gravitas/Auto-GPT>.
- Tosic, P. T. and Agha, G. A. (2004). Towards a hierarchical taxonomy of autonomous agents. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 4, pages 3421–3426. IEEE.
- TransformerOptimus et al. (2023). SuperAGI. <https://github.com/TransformerOptimus/SuperAGI>.
- Tufte, E. R. (2001). *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT.
- Van Dyke Parunak, H., Brueckner, S., Fleischer, M., and Odell, J. (2004). A design taxonomy of multi-agent interactions. In *Agent-Oriented Software Engineering IV: 4th International Workshop, AOSE 2003, Melbourne, Australia, July 15, 2003. Revised Papers 4*, pages 123–137. Springer.
- Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., et al. (2023). A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.
- Wolf, Y., Wies, N., Levine, Y., and Shashua, A. (2023). Fundamental limitations of alignment in large language models. *arXiv preprint arXiv:2304.11082*.
- Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(2):115–152.
- Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., et al. (2023). The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.
- Yudkowsky, E. (2016). The AI alignment problem: why it is hard, and where to start. *Symbolic Systems Distinguished Speaker*, 4.
- Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X. V., et al. (2022). OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.