The Cognitive Hourglass: Agent Abstractions in the Large Models Era*

Alessandro Ricci[1], Stefano Mariani[2], Franco Zambonelli[2]
Samuele Burattini[1], and Cristiano Castelfranchi[3]

[1] Dipartimento di Informatica - Scienza e Ingegneria,
Alma Mater Studiorum - University of Bologna, Cesena Campus, Italy
`{a.ricci|samuele.burattini}@unibo.it`
[2] University of Modena and Reggio Emilia, Reggio Emilia, Italy
`{stefano.mariani|franco.zambonelli}@unimore.it`
[3] Italian Research Council, Rome, Italy
`cristiano.castelfranci@istc.cnr.it`

Abstract. Recent advances in AI are driving an unprecedented and fast-paced development of myriads of powerful agent tools and applications, mostly based on generative AI technologies such as Large Language/Multi-modal/Agent Models. However, despite many proposals in that direction, the lack of a sound set of engineering abstractions hinders the possibility of methodically engineering agent-based applications fully harnessing such technologies, also due to the gap between cognitive agent-based concepts and LLMs' usage patterns. We argue that such a set of abstractions should constitute the *narrow neck* of an indispensable "cognitive hourglass": a level of abstraction that is meant to be used by humans to understand/design/control agents and MAS, regardless of the specific AI technologies adopted at the implementation level and of the specific application context. Here, we elaborate on the idea of the cognitive hourglass, motivate its need, sketch its envisioned structure, and identify the research challenges for its realisation.

1 Introduction

The hourglass model [2] has been adopted in computer science and engineering as a conceptual metaphor for describing complex systems, like network protocols [1], and as a blueprint to drive their design [42,30]. It provides effective constraints in the design of open systems, enabling the development of an open set of applications on top of an open set of supporting technologies and services. Its key element is a narrow layer as the *neck* of the hourglass, including a selected set of functional abstractions separating and mediating between the upper (application) layers and the bottom (technology, implementation) ones.

As an example, consider the so-called IP Hourglass (Fig. 1, left), which is a model of modern IP-based networked systems. Several network technologies and

* Paper accepted at AAMAS Blue Sky 2024, submitted at EMAS as position paper

protocols (bottom layers) exist that are exploited by many high-level application protocols and systems (upper layers). The Internet Protocol acts as a glue between the upper and the lower layers by making available a simple uniform communication protocol, independent of the actual network technology, that can be used to build any distributed applications.

The hourglass model is useful for understanding, discussing, and governing the recent fast-paced advances in AI that are dramatically increasing the spectrum of technologies that can be exploited to build autonomous agents and Multi-Agent Systems (MAS) and applications—in addition to the many systems and languages already known [51,5]. Large neural models and generative AI technologies such as Large-Language Models (LLMs) [10], Large-Multimodal Models (LMMs) [46], and Foundation Models [54] are witnessing a huge momentum [14] and started to be adopted to build different kinds of "agents" [53,45]: from generalist agents such as Gato [35] to *generative agents* based on LLMs, such as AutoGPT [29]—called Large Agent Models (LAMs).

Such agents show remarkable capabilities in terms of information management and reasoning. However, how to *methodically* exploit such capabilities during the process of designing and building MAS is yet unclear. On the one hand, LLM-based agents rely on properly "prompting" (i.e., triggering) agent behaviours, an activity which is currently far from being methodical and reproducible [49,24]. On the other hand, the current way of building and exploiting LLMs and LAMs is leading to a sort of "eliminativism": deeming higher-level abstractions unnecessary once lower-level ones have been fully understood. In

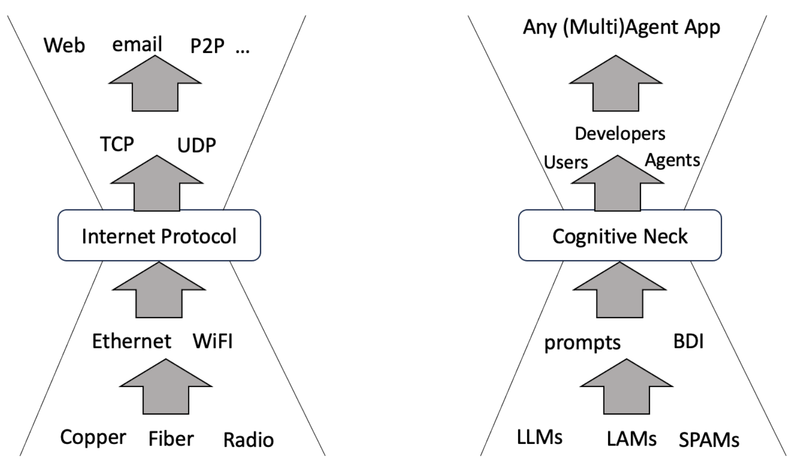

Fig. 1. *(Left)* The IP Hourglass: the narrow neck gives the minimum set of abstractions and mechanisms to build upper layer services on top of the lower layer ones. *(Right)*The Cognitive Hourglass: the narrow neck enables to uniformly exploit the lower layers to build and understand the upper layer systems and applications, and the upper layer, in turn, to model them while governing the lower level models and technologies.

particular, cognitive concepts [31,21,12,11] that are pillars for the understanding and engineering of agent systems [6], seem to be increasingly ignored.

In contrast, in this article, we argue that such concepts constitute the indispensable *neck* of the *cognitive hourglass* (Figure 1, right), that is, the fundamental human-compatible [38] level of abstraction necessary for humans to understand/design/govern agents and MAS at the application level – the top of our hourglass – regardless of the specific AI technologies adopted at the implementation level—the bottom of our hourglass.

2 Cognition and Eliminativism

Some literature about generative AI, including LLMs, LMMs, and LAMs, seem to foster the idea that every conceivable "intelligent" application, and software agents themselves, can be directly built on top of their usage patterns and working mechanisms – such as prompt engineering, instruction-tuning, and generally speaking in-context learning [16,36,26] –, without the need for any scientific and engineering abstraction in between. This perspective nurtures a dangerous trend: *eliminativism* [27], as the attitude of deeming higher-level abstractions unnecessary once lower-level ones have been modelled and understood. This is a degenerate derivation of reductionism, that seeks to explain higher-level phenomena in terms of lower-level ones, but without neglecting the utility of those at the higher level—as they enable the expression of novel properties and interpretations of the lower ones.

In chemistry, for instance, bonding laws must find an "implementation" in terms of the underlying mechanisms of physics (reductionism), but no chemist would then throw away such laws to only think in terms of physics (eliminativism)! Also in computer science, nobody in its right mind would throw away high-level programming abstractions (such as objects, functions, procedures, data structures) just because they are "explained" by (or, can be reduced to) assembly instructions! However, a recent stream of publications about LLM-based agents [45,53] seems inclined to disregard agent-oriented abstractions – such as Allen Newell's Knowledge Level [31], Jennings in [21], as well as Dennett's intentional stance [12] and Castelfranchi's work [33] –, since it can be anyway obtained a set of agent-like capabilities without them being part of the engineering process [35,25,47].

Besides the fact that this claim is still under debate and evaluation [43,4,18], eliminativism is undesirable because the cognitive abstractions developed in the agent community, and more generally within the AI research landscape, have the unquestionable merit of serving us well in building (engineering purpose) and understanding (scientific purpose) systems while using the most suitable abstractions—for our own reasoning processes, as human beings.

Ignoring such abstractions creates a gap between the ones who build and use these systems and their modes of operation: the former typically reason in terms of goals, plans, actions and their consequences, beliefs, knowledge, cause-effect relations [32], etc., whereas the latter require manipulation of prompts, linguistic patterns, examples, and blueprints of desired behaviours. From the engineering viewpoint, a consequence is a limited capability of building systems by composition of simpler parts, a foundational basis of any engineering discipline. From the scientific standpoint, the lack of a layered set of abstractions hinders understanding of a given system in its multiple nuances in terms of properties and behaviours (e.g. focusing on chemical or physical properties of a given material).

The latter ability, in particular, is essential not only under the lens of eXplainable AI [19], safety, and alignment in AI (in one word, "Human-centred AI" [39,38]), but also when considering transferability of concepts across domains, and when shifting attention from an individual agent to a MAS. There, for instance, ascribing to others their own mental states is fundamental to promote collaboration and requires mind-reading [9] others (i.e. prediction of the mental states of other agents) with the proper abstractions.

In the following section, we propose the adoption of a *cognitive hourglass* to align generative technologies and agent-oriented abstractions and stay away from the described eliminativism.

3 The Cognitive Hourglass

To synergistically exploit LLMs and LAMs, and the (necessary) knowledge-based and symbolic (i.e., cognitive) computational tools, it is needed to identify the "neck" of an agent-oriented hourglass. Such neck should enable to uniformly exploit any available enabling technology (bottom layers of the hourglass) to provide services for building, controlling, and understanding, autonomous agents and MAS (upper layers of the hourglass)—Fig. 1, right.

For such concepts to be effectively usable both at the human and at the software level, they should abstract away implementation and technical details. In addition, they should be expressive enough to allow for specifying any kind of structure and behaviour of systems modelled in terms of MAS, being them designed systems/applications, implemented agents systems, or even simulated agents systems. In other words, the neck of the hourglass should be a *cognitive abstraction gate*, with cognitive concepts being the "sand" flowing up and down the hourglass. As such, the neck enables a *bi-directional* flow of symbolic knowledge and concepts between the upper and the lower levels of the hourglass. The "services" of the enabling technologies below the neck can be effectively activated and exploited from above the neck to build and understand systems. The stakeholders, systems, and applications above the neck can be used to govern the lower levels at the most appropriate level of abstraction. Let us now elaborate on our envisioned layered structure of the cognitive hourglass.

The lowest level is where to accommodate any of the available technologies and implementations of agents and MAS. For instance: LLM-based agents [53], LAMs [35], agents built with specific agent platforms like Jade [3] or Jason [7], or whatever other type of Special-Purpose Agent Models (SPAMs). Above it, to be able to exploit all such technologies, one must consider the various types of "mechanisms" such technologies use to interact with the world. E.g., prompts for

LLMs, Beliefs Desires and Intentions (BDI) scripts or alike for the case of cognitive agents. At the upper levels, there are human actors – either as simple users of the below technologies or as engineers in charge of exploiting them to design and develop agent and multi-agent applications – as well as software agents—i.e., agents as components of some multi-agent application and exploiting the below technologies to augment/outsource their capabilities. The cognitive neck, acting as a gateway between the lower and the upper levels, should be flexible and expressive enough to provide access to the lower levels (i.e., to the services provided by the implemented agent technologies) in a simple yet comprehensive and comprehensible way—to effectively support the development of agent systems and their empowerment. In turn, it must also provide for human-compatible [38] interpretations of the lower level mechanisms and usage patterns.

For instance, being generative technologies based on natural language, hence dialogues, the cognitive neck could provide abstractions to re-interpret prompt engineering techniques as *argumentation processes* [44], made up of commitments, expectations, roles, scopes, and similar concepts, that humans exploit (either consciously or not) while engaging in dialogues with each other. Or, *theory of mind* [34] concepts can be used to interact more proficiently with LLMs and other agents by ascribing mental states to them, instead of reasoning in terms of usage patterns, example prompts, reasoning processes blueprints, and similar low-level non-cognitive terms.

Such an hourglass does not impose any specific architecture for software development. For instance, designing a system in which some agents and some generative models interact to carry out a shared goal while providing feedback to each other is not ruled out. The vertical depiction of the hourglass is only meant to convey a sense of layering of abstractions (not architectures), and the role played by the cognitive neck is at the level of abstractions, not technologies. In the example architecture just mentioned, the cognitive hourglass is not "violated" in any way. On the contrary, is can serve to frame generative technologies within human-compatible [38] abstractions, so as to ease the engineering and understanding of the interaction space with the other agents.

How the cognitive neck should be made, what interfaces it should expose (below and above it), and how it could be effectively exploited, is impossible to detail in this paper. This is indeed one of the key research questions this article intends to raise, not answer (yet). Nevertheless, some key concepts include:

- *Wishes* (aka desires, goals) expressed from the upper levels to the cognitive neck. What one wishes is typically the state of affairs that one (agent, human user, or agent developer) wants to achieve. In response to wishes, it is expected that the cognitive neck will reply with.
- *Hows* (aka plans), that is a proposal for actions (and their ordering) to be put to work to achieve wishes, expressed as usage of the below mechanisms and services, possibly in respect of constraints.
- *Constraints* (aka safety and liveness rules). These can be expressed from the upper to the lower levels as specific instructions on how plans should be

built, but also be communicated from the lower to the upper levels, if such constraints emerge during the building of plans.
- *Whys* (aka explanations), mostly for letting the lower levels motivate (whenever needed) the responses provided to the upper levels, e.g., in the form of causal models [32,28]. However, in some cases, whys can be used to let the upper levels motivate their requests to guide/influence the behaviour of the lower levels.
- *Whats* (aka facts or beliefs). Through the cognitive neck, knowledge can transit (typically on request) to the upper level about things known at the lower level. However, one could also think of knowledge transiting from the upper to the lower level to influence its behaviour.

It is worth emphasising that the cognitive hourglass can also play an important role in enabling multi-agent communication and the involvement of agents in interaction and negotiation protocols. Indeed, cognitive approaches to multi-agent interactions assume that messages exchanged in the context of a distributed decision-making process have cognitive context [40,22], and are aimed at transferring knowledge, informing about plans or desires, or proposing courses of action. Again, although tools for developing MAS with LLM-based agents are being proposed [52,55], interactions between agents are limited to prompting conversations and do not account for the specific cognitive meaning of messages.

4 Towards an Integrated Framework

The cognitive hourglass can be a suitable methodological and practical framework for the engineering of agent-based systems, affecting both *design-time* scenarios (Fig. 2, left), involving developers and engineers, and *run-time* ones, involving users and application agents (Fig. 2, right).

At design time, the cognitive hourglass can support developers and engineers in conceiving agent-based architectures that are instrumented to adopt high-level Agent-Oriented Software Engineering (AOSE) methodologies [17] and possibly Agent-Oriented Programming (AOP) languages [5] designed upon the abstractions defined in the cognitive neck. Some existing AOSE methodologies such as TROPOS [8] and the more recent TDF [13] naturally fit in the picture. According to this design-time view, it is interesting to devise – as future research directions – new agent architectures conceived to be compatible with the cognitive neck, eventually integrating different approaches and technologies. An example is *generative BDI architectures*, as agent architectures based on the BDI model and reasoning cycle, but integrating generative AI technologies in key steps of the cycle.

The cognitive hourglass could also affect the overall process of developing agents and MAS, in particular *learning-based* processes. Learning and machine learning techniques – Reinforcement Learning (RL) and Deep RL in particular – are increasingly adopted as reference approaches to develop agents in various domains. In recent works, learning becomes a core ingredient of novel engineering processes based on self-development [23,50], and of adopting developmen-

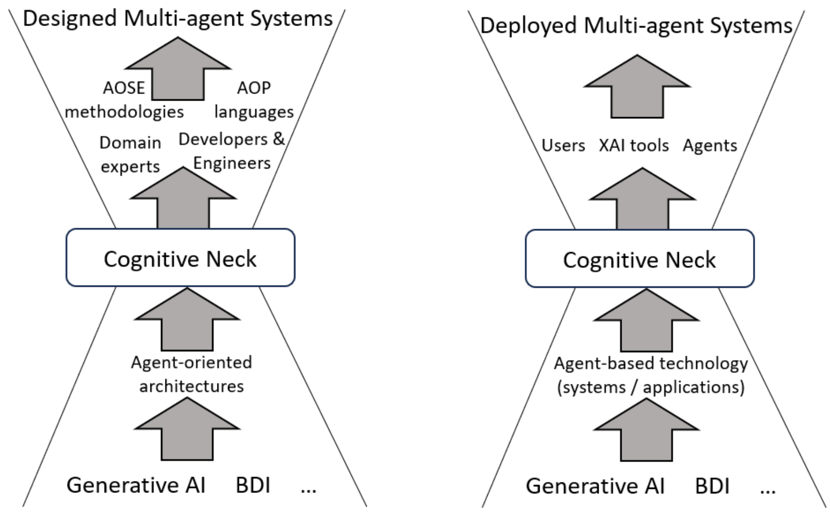

Fig. 2. The Cognitive Hourglass at design-time *(left)* and at run-time *(right)*. At design-time, abstractions in the neck enable to build upper layer tools and systems uniformly, regardless of the ones at the lower layer. At run-time, through the neck, humans and software agents can exploit cognitive abstractions to empower themselves, inspect, and coordinate.

tal learning methods that make agent/MAS engineering similar to an education process [37]. Given the hourglass, the high-level learning-based development processes adopted to "grow up" agents or refine their skills (top) should be based on concepts defined in the cognitive neck, allowing to frame, at the proper level of abstraction, the specific learning techniques and technologies adopted (bottom).

At run-time, the cognitive neck could serve as a practical software tool for enabling users and agents to access a suite of cognitive services. On the one hand, human users can exploit the cognitive neck to interface with the world of existing agent-based systems and tools. Regardless of the specific technology and implementation, the cognitive neck makes sure that the features of the lower layers will be made available in the form of cognitive and human-understandable concepts. On the other hand, agents developed to serve in specific applications and systems will be able to exploit the cognitive neck as a run-time tool to empower their capabilities, for example by dynamically accessing services (agent/LLM-based but not necessarily) to request planning or general information. However, as anticipated in the previous section, the cognitive neck could also become a powerful tool for supporting cognitive inter-agent interactions.

For both human and agent users, the cognitive abstraction level of the neck makes it possible to instrument tools that can make the behaviour of the lower levels transparent and explainable. In other words, the cognitive neck could explicitly define the conceptual interface upon which explainability tools can be designed and exploited. This deeply relates to the scientific viewpoint: the cognitive neck enables a principled understanding of the working mechanisms and usage patterns of what is below the neck, similarly to the layered understanding scientists have of biological, chemical, and physical systems (reductionism, not eliminativism!). But run-time usage of the cognitive neck is not limited to understanding of lower level mechanisms and technologies, as it can be used to compose and govern them so as to obtain the desired behaviour. While doing so, the added value is that agents do not need to switch to reason in terms of prompting, API calls, and similar low-level technical details, but can exploit the cognitive neck abstractions to carry out activities at the most suitable level of abstraction—for them, and for human supervisors.

In all these scenarios the cognitive neck plays the role of an abstraction gate, allowing for the development and integration of different kinds of heterogeneous technologies at the bottom, and the development of proper tools that would allow humans – users, domain experts, engineers – and agents as well to have a full understanding and control of the system.

5 Conclusions and Open Challenges

In this article, we motivated why cognitive abstractions and concepts (from Allen Newell's Knowledge Level to Castelfranchi's work on autonomous goal-directed behaviour) should play a primary role in agent systems engineering. Even, and especially, in the presence of the recent LLMs-enabled agent tools, cognitive abstractions have to constitute the *narrow neck* of a "cognitive hourglass". That is a level of abstraction useful for humans to understand/design/control agents and MAS, regardless of the specific AI technologies adopted at the implementation level and of the specific application context. Yet, for the cognitive hourglass to become a practical and usable tool, there are several key challenges to be faced.

First, the concepts and principles inside the cognitive neck must be identified, to make it both a usable conceptual tool for developers and a software service layer for agents. This includes the possibility of exploiting the cognitive neck to support flexible interactions in multi-agent systems. Second, proper mappings must be developed that – despite the abstraction gate – allow exploitation of the capabilities provided by the bottom/enabling levels while preserving the property of being "human-compatible"—both for users and engineers. Considering the multiple dimensions that are important in the case of MAS – such as the social and organisational dimensions [15], and the environment dimension as well [48] – also demands further studies. Finally, proper forms of integration between cognitive agents and generative AI should be identified. Some relevant efforts in that direction can already be found: [20] explores the use of language models as a source of task knowledge in cognitive agents/systems; [41] proposes a systematic framework called Cognitive Architectures for Language Agents (CoALA), useful for both organizing existing literature on generative agents and identifying directions towards more capable agents, including features as found in cognitive agents and architectures.

References

1. Akhshabi, S., Dovrolis, C.: The evolution of layered protocol stacks leads to an hourglass-shaped architecture. In: Proceedings of the ACM SIGCOMM Conference. pp. 206–217 (2011)
2. Beck, M.: On the hourglass model. Communication of ACM **62**(7), 48–57 (jun 2019). https://doi.org/10.1145/3274770, https://doi.org/10.1145/3274770
3. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing multi-agent systems with JADE. John Wiley & Sons (2007)
4. Berglund, L., Tong, M., Kaufmann, M., Balesni, M., Stickland, A.C., Korbak, T., Evans, O.: The reversal curse: Llms trained on "a is b" fail to learn "b is a" (2023)
5. Bordini, R.H., Braubach, L., Dastani, M., Seghrouchni, A.E.F., Gomez-Sanz, J.J., Leite, J., O'Hare, G., Pokahr, A., Ricci, A.: A survey of programming languages and platforms for multi-agent systems. Informatica **30**(1) (2006)
6. Bordini, R.H., El Fallah Seghrouchni, A., Hindriks, K., Logan, B., Ricci, A.: Agent programming in the cognitive era. Autonomous Agents and Multi-Agent Systems **34**(2) (may 2020). https://doi.org/10.1007/s10458-020-09453-y, https://doi.org/10.1007/s10458-020-09453-y
7. Bordini, R.H., Hübner, J.F., Wooldridge, M.: Programming multi-agent systems in AgentSpeak using Jason. John Wiley & Sons (2007)
8. Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., Mylopoulos, J.: Tropos: An agent-oriented software development methodology. Autonomous Agents and Multi Agent Systems **8**(3), 203–236 (2004), http://dblp.uni-trier.de/db/journals/aamas/aamas8.html#BrescianiPGGM04
9. Castelfranchi, C.: "mind reading": How and for what? Annals of Cognitive Science (2019), https://api.semanticscholar.org/CorpusID:211167867
10. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P.S., Yang, Q., Xie, X.: A survey on evaluation of large language models (2023)
11. Conte, R., Castelfranchi, C.: Cognitive and social action. Psychology Press (1995)
12. Dennett, D.C.: The Intentional Stance. The MIT Press, Cambridge, MA (1987)
13. Evertsz, R., Thangarajah, J., Ly, T.: Practical Modelling of Dynamic Decision Making. Springer Publishing Company, Incorporated, 1st edn. (2019)
14. Fan, L., Li, L., Ma, Z., Lee, S., Yu, H., Hemphill, L.: A bibliometric review of large language models research from 2017 to 2023 (2023)
15. Ferber, J., Gutknecht, O., Michel, F.: From agents to organizations: an organizational view of multi-agent systems. In: International workshop on agent-oriented software engineering. pp. 214–230. Springer (2003)
16. Gao, A.: Prompt engineering for large language models. Available at SSRN 4504303 (2023)
17. Henderson-Sellers, B., Giorgini, P.: Agent-oriented methodologies. Igi Global (2005)
18. Hodel, D., West, J.: Response: Emergent analogical reasoning in large language models. CoRR **abs/2308.16118** (2023), https://doi.org/10.48550/arXiv.2308.16118
19. Holzinger, A., Kieseberg, P., Weippl, E.R., Tjoa, A.M.: Current advances, trends and challenges of machine learning and knowledge extraction: From machine learning to explainable AI. In: Machine Learning and Knowledge Extraction - Second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International Cross-Domain Conference, CD-MAKE, Hamburg, Germany, August 27-30, Proceedings. Lecture

Notes in Computer Science, vol. 11015, pp. 1–8. Springer (2018). https://doi.org/10.1007/978-3-319-99740-7_1, https://doi.org/10.1007/978-3-319-99740-7_1
20. III, R.E.W., Kirk, J.R., Laird, J.E.: Language models as a knowledge source for cognitive agents. CoRR **abs/2109.08270** (2021)
21. Jennings, N.R.: On agent-based software engineering. Artificial Intelligence **117**(2), 277–296 (mar 2000). https://doi.org/10.1016/S0004-3702(99)00107-1, https://doi.org/10.1016/S0004-3702(99)00107-1
22. Labrou, Y., Finin, T., Peng, Y.: Agent communication languages: The current landscape. IEEE Intelligent Systems and Their Applications **14**(2), 45–52 (1999)
23. Lippi, M., Mariani, S., Martinelli, M., Zambonelli, F.: Individual and collective self-development: Concepts and challenges. In: 17th Conference on Computer Science and Intelligence Systems (FedCSIS). pp. 15–21. IEEE (2022)
24. Liu, V., Chilton, L.B.: Design guidelines for prompt engineering text-to-image generative models. In: Proceedings of the CHI Conference on Human Factors in Computing Systems. pp. 1–23 (2022)
25. Liu, X., Yu, H., Zhang, H., Xu, Y., Lei, X., Lai, H., Gu, Y., Ding, H., Men, K., Yang, K., Zhang, S., Deng, X., Zeng, A., Du, Z., Zhang, C., Shen, S., Zhang, T., Su, Y., Sun, H., Huang, M., Dong, Y., Tang, J.: Agentbench: Evaluating llms as agents (2023)
26. Lu, S., Bigoulaeva, I., Sachdeva, R., Madabushi, H.T., Gurevych, I.: Are emergent abilities in large language models just in-context learning? (2023)
27. Mancini, F., Mancini, A., Castelfranchi, C.: Unhealthy mind in a healthy body: A criticism to eliminativism in psychopathology. Frontiers in Psychiatry **13** (2022). https://doi.org/10.3389/fpsyt.2022.889698
28. Mariani, S., Roseti, P., Zambonelli, F.: Multi-agent learning of causal networks in the internet of things. In: Advances in Practical Applications of Agents, Multi-Agent Systems, and Cognitive Mimetics. The PAAMS Collection - 21st International Conference, PAAMS 2023, Guimarães, Portugal, July 12-14, 2023, Proceedings. pp. 163–174 (2023)
29. Mindstreams: Autogpt (2023), https://autogpt.net/
30. Minhas, T.N., Fiedler, M.: Quality of experience hourglass model. In: International Conference on Computing, Management and Telecommunications (ComManTel). pp. 87–92 (2013). https://doi.org/10.1109/ComManTel.2013.6482371
31. Newell, A.: The knowledge level. Artificial Intelligence **18**(1), 87–127 (1982). https://doi.org/10.1016/0004-3702(82)90012-1
32. Pearl, J., Mackenzie, D.: The book of why: The new science of cause and effect (2018)
33. Pezzulo, G., Castelfranchi, C.: Thinking as the control of imagination: a conceptual framework for goal-directed systems. Psychological Research Psychologische Forschung **73**(4), 559–577 (2009). https://doi.org/10.1007/s00426-009-0237-z
34. Rabinowitz, N., Perbet, F., Song, F., Zhang, C., Eslami, S.M.A., Botvinick, M.: Machine theory of mind. In: Dy, J., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 4218–4227. PMLR (10–15 Jul 2018), https://proceedings.mlr.press/v80/rabinowitz18a.html
35. Reed, S.E., Zolna, K., Parisotto, E., Colmenarejo, S.G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J.T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., de Freitas, N.: A generalist agent. Transactions on Machine Learning Research (2023), https://openreview.net/forum?id=1ikK0kHjvj

36. Reynolds, L., McDonell, K.: Prompt programming for large language models: Beyond the few-shot paradigm. In: Extended Abstracts of the CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3411763.3451760
37. Ricci, A.: "go to the children": Rethinking intelligent agent design and programming in a developmental learning perspective. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems. p. 1809–1813. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2022)
38. Russell, S.: Human Compatible: Artificial Intelligence and the Problem of Control. Penguin Publishing Group (2019), https://books.google.it/books?id=M1eFDwAAQBAJ
39. Shneiderman, B.: Bridging the gap between ethics and practice: Guidelines for reliable, safe, and trustworthy human-centered AI systems. ACM Transactions on Interactive Intelligent Systems **10**(4), 26:1–26:31 (2020). https://doi.org/10.1145/3419764, https://doi.org/10.1145/3419764
40. Singh, M.P.: Agent communication languages: Rethinking the principles. Computer **31**(12), 40–47 (1998)
41. Sumers, T.R., Yao, S., Narasimhan, K., Griffiths, T.L.: Cognitive architectures for language agents. arXiv preprint arXiv:2309.02427 (2023)
42. Susanto, Y., Livingstone, A.G., Ng, B.C., Cambria, E.: The hourglass model revisited. IEEE Intelligent Systems **35**(5), 96–102 (2020). https://doi.org/10.1109/MIS.2020.2992799
43. Valmeekam, K., Sreedharan, S., Marquez, M., Olmo, A., Kambhampati, S.: On the planning abilities of large language models (a critical investigation with a proposed benchmark) (2023)
44. Walton, D.N., Krabbe, E.C.W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany, NY, USA (1995)
45. Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W.X., Wei, Z., Wen, J.R.: A survey on large language model based autonomous agents (2023)
46. Wang, X., Chen, G., Qian, G., Gao, P., Wei, X.Y., Wang, Y., Tian, Y., Gao, W.: Large-scale multi-modal pre-trained models: A comprehensive survey. Machine Intelligence Research **20**(4), 447–482 (Jun 2023). https://doi.org/10.1007/s11633-022-1410-8
47. Wang, Z., Cai, S., Liu, A., Jin, Y., Hou, J., Zhang, B., Lin, H., He, Z., Zheng, Z., Yang, Y., Ma, X., Liang, Y.: Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models (2023)
48. Weyns, D., Omicini, A., Odell, J.: Environment as a first class abstraction in multiagent systems. Autonomous agents and multi-agent systems **14**, 5–30 (2007)
49. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382 (2023)
50. Wirsing, M., Belzner, L.: Towards systematically engineering autonomous systems using reinforcement learning and planning. In: Analysis, Verification and Transformation for Declarative Programming and Intelligent Systems: Essays Dedicated to Manuel Hermenegildo on the Occasion of His 60th Birthday, pp. 281–306. Springer (2023)

51. Wrona, Z., Buchwald, W., Ganzha, M., Paprzycki, M., Leon, F., Noor, N., Pal, C.V.: Overview of software agent platforms available in 2023. Information **14**(6) (2023). https://doi.org/10.3390/info14060348
52. Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A.H., White, R.W., Burger, D., Wang, C.: Autogen: Enabling next-gen llm applications via multi-agent conversation framework (2023)
53. Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., Zheng, R., Fan, X., Wang, X., Xiong, L., Zhou, Y., Wang, W., Jiang, C., Zou, Y., Liu, X., Yin, Z., Dou, S., Weng, R., Cheng, W., Zhang, Q., Qin, W., Zheng, Y., Qiu, X., Huang, X., Gui, T.: The rise and potential of large language model based agents: A survey (2023)
54. Zhou, C., Li, Q., Li, C., Yu, J., Liu, Y., Wang, G., Zhang, K., Ji, C., Yan, Q., He, L., Peng, H., Li, J., Wu, J., Liu, Z., Xie, P., Xiong, C., Pei, J., Yu, P.S., Sun, L.: A comprehensive survey on pretrained foundation models: A history from bert to chatgpt (2023)
55. Zhuge, M., Liu, H., Faccio, F., Ashley, D.R., Csordás, R., Gopalakrishnan, A., Hamdi, A., Hammoud, H.A.A.K., Herrmann, V., Irie, K., Kirsch, L., Li, B., Li, G., Liu, S., Mai, J., Piękos, P., Ramesh, A., Schlag, I., Shi, W., Stanić, A., Wang, W., Wang, Y., Xu, M., Fan, D.P., Ghanem, B., Schmidhuber, J.: Mindstorms in natural language-based societies of mind (2023)