On the Historic Roots of Agentic AI in Semantic Web Services

Axel Polleres
Frederik Bauer
Daniil Dobriy
Vienna University of Economics and Business
Vienna, Austria

Tobias Käfer
Lukas Kubelka
Karlsruhe Institute of Technology
Karlsruhe, Germany

Andreas Harth
Thomas Wehr
Friedrich-Alexander-Universität Erlangen-Nürnberg
Nuremberg, Germany

Abstract

In our short talk, we point out that (1) the Semantic Web vision already provided some of the underlying ideas of today's endeavours around agentic AI, and (2) that abstractions, approaches and methods developed in the Web Services and Semantic Web Services communities can be beneficially applied to agentic AI. At the same time, we argue that looking back to history reveals certain still unresolved challenges.

CCS Concepts

• **Information systems → Web services**; • **Computing methodologies → Artificial intelligence**.

Keywords

Semantic Web Services; Agentic AI

ACM Reference Format:
Axel Polleres, Frederik Bauer, Daniil Dobriy, Tobias Käfer, Lukas Kubelka, Andreas Harth, and Thomas Wehr. 2026. On the Historic Roots of Agentic AI in Semantic Web Services. In *Companion Proceedings of the ACM Web Conference 2026 (WWW Companion '26), April 13–17, 2026, Dubai, United Arab Emirates.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3774905.3795076

1 Introduction

The invention of the Web and the emergence of services available online galvanised the idea of *software agents* accessing and combining such services on our behalf, seamlessly achieving our goals and supporting our daily lives. The vision is manifested in the renowned Semantic Web article [5] from 2001. The Semantic Web vision was subsequently backed by the idea of having well-defined, declarative, machine-readable (semantic) descriptions of services and goals that would enable agents to invoke and compose services on the fly. Building on then state-of-the-art Web Services (WS-*) technologies [2], a well-funded and productive research community developed semantic description frameworks [24, 28] that would provide the building blocks for interoperable service ecosystems in which many tasks could be automated. However, the work around these Semantic Web Services ceased around 2008 and the developed approaches have seen little adoption in practice.

Yet, with the advent of Large Language Models (LLMs) and their ability to generate code, interpret data, even use external tools [29] and carry out complex tasks [32], we see old ideas resurface and register renewed interest towards building software agents that carry out tasks for us. Increasingly capable models are being developed; supporting frameworks, such as Anthropic's Model Context Protocol (MCP) [4] and Google's Agent2Agent protocol (A2A) [13] are gaining traction.

Amidst the flurry of activity, we submit that now might be a good time to reflect on whether the roadblocks that prevented the broad adoption of Semantic Web Services have been successfully overcome with these recent developments. Thus, the talk puts the historic efforts towards Semantic Web Services side by side with the recent advances in agentic AI. Our goal is to highlight useful historical conceptualisations, discuss how far these are paralleled or incarnated by current trends around agentic AI, and finally spotlight still unresolved challenges, with potential routes ahead.

2 (Semantic) Web Services

Core specifications around Web Services [2] (WS-*) started to appear around the 2000s and were based on Web standards such as XML, with its first W3C Recommendation published in 1998. A guiding idea was related to standardising message exchange, to connect enterprise systems across organisational boundaries.

The development of Semantic Web Service [11, 21, 24] (SWS) approaches started around the 2000s and ran in parallel to many WS-* developments. A guiding idea in Semantic Web Services was to enable software agents to discover and compose services based on elaborate (semantic) descriptions of capabilities of services.

In the following, we introduce and discuss the WS-* and SWS technology families along several dimensions.

Messaging: Communication between clients and services, and between services via invocation of services, typically requiring specific message formats. We can distinguish between a client that can exchange messages with a service ("service invocation") and a service that can receive but also send messages.

Description: (Machine-readable) specifications of service interfaces and other aspects, such as capabilities. With the specifications of service interfaces including a schema for messages, clients and services can validate whether messages conform to the schema. Descriptions of capabilities and functions can also support clients and services in discovery and composition.

Discovery: Registries with such descriptions of services, which allow for searching for services and matching them with goals. A

WWW Companion '26, Dubai, United Arab Emirates

ACM ISBN 979-8-4007-2308-7/2026/04
https://doi.org/10.1145/3774905.3795076

common pattern for discovery in service-oriented architectures (SOA) is publish-find-invoke [1, 10, 26] (Figure 1).

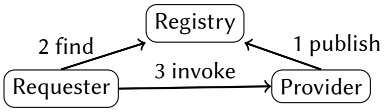

Figure 1: The Publish-Find-Invoke Pattern: A provider publishes service descriptions to a registry; a requester finds a service and then invokes the service.

Composition: Combination of services towards achieving a complex goal. In particular, the *composition* of service interactions within business processes was of interest. The literature distinguishes two ways for the composition of services: first, *orchestration* with centralised control (Figure 2) and second, *choreography* with interactions between peers (Figure 3).

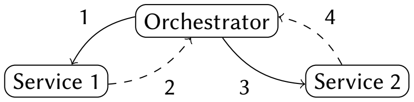

Figure 2: Service Orchestration: An orchestrator makes two sequential service calls (solid) with responses (dashed).

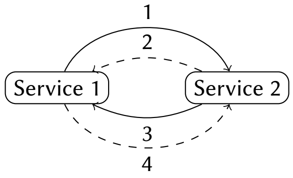

Figure 3: Service Choreography. Two services call (solid) and respond to (dashed) each other sequentially.

Meditation: Resolving differences in message formats, but also other aspects of heterogeneity. That is, mediation is the resolution of mismatches between a client and a service (or between services) such that they can work together in a composition.

Finally, we summarise the state of play of each technology family today and assess how agents are supported.

2.1 WS-* Technologies

The WS-* stack provides standardised messaging protocols and interface descriptions, structured around services with operations that have inputs and outputs. The relevant specifications, SOAP and WSDL, were published by the W3C. Registries allow for discovery of services, and workflow engines allow for the execution of orchestrated business processes. The relevant specifications, UDDI and BPEL, were published by OASIS Open, a pay-for-play standardisation organisation. Major contributors to the WS-* specifications include Microsoft, IBM, HP, Oracle and SAP.

Messaging. SOAP, originally short for Simple Object Access Protocol, has evolved from the work on XML-RPC from the late 1990s [6]. The first version of SOAP with W3C Recommendation status was published in 2003 and the latest version in 2007 [14]. SOAP is a messaging protocol for remote procedure calls between clients and services or between services, that are tunnelled through other transport protocols, most prominently HTTP POST requests, with the service endpoint as target URI of the request. The body of such a request is an XML document with a SOAP envelope element, which is subdivided into header and body elements, to describe the remote procedure call.

Description. WSDL, the Web Services Description Language, has gone through different versions until version 2.0 reached W3C Recommendation status in 2007 [8]. WSDL allows for describing service operations, including the format of (SOAP) messages (typically in terms of XML Schema descriptions) and bindings to a specific transport protocol. As such, the descriptions were, despite human readable comments as part of the description, mostly focused on enlisting API operations, and – borrowing from XML Schema – descriptions of the service input and output messages' grammar. Based on WSDL documents, clients and services can generate conforming SOAP messages or check SOAP messages for conformance.

Discovery. UDDI, Universal Description, Discovery and Integration, graduated around 2002 in version 2 as OASIS Standard [7]. As an early specification to advertise services and implement a registry (with limited search functionality), UDDI distinguished between different levels of description relating to general service description, such as contact information (White pages), business description (in terms of, e.g., USPSC business/sector categories) and technical descriptions, e.g., linking to a WSDL web service description. Although major corporations such as IBM, Microsoft and SAP operated publicly running UDDI registries, all of them seem to have ceased operation before 2010.

Composition. BPEL, the Web Services Business Process Execution Language, was submitted to OASIS in 2003 and published in 2007 in version 2.0 [3] as an OASIS Standard [12]. BPEL is XML-based and allows for describing executable orchestrations of services in a process, i.e., calls to service operations, made by a single service requester, put into order by some control flow (e.g., sequential or parallel ordering). The publication of the BPEL specification was followed by various commercial offerings of software that executes such orchestrations. In contrast, works on choreography, i.e., decentrally governed compositions, merely reached W3C Candidate Recommendation status [19], in 2005.

Mediation. In a WS-* orchestration, mediation is mainly (manually) hard-coded, yet facilitated by the inclusion of corresponding XML standards in BPEL, e.g., functionality to transform messages via XSLT, a language to transform XML documents.

WS-* technologies, in particular SOAP and WSDL, are to some extent still deployed today, especially in regulated industries. A plethora of supporting specifications emerged, like WS-Security [22], which introduced security features, BPEL4People [18] and WS-Human Task [17], which introduced human tasks into service compositions and WS-Policy [30], which allows for describing non-functional, service-level requirements regarding, e.g., security or quality of service.

Overall, the WS-* technology stack provides limited support for automation by agents, and was rather focused on enabling human developers to operate, compose and invoke services or execute pre-defined orchestrated workflows, based on SOAP for messaging and WSDL for syntactic descriptions.

2.2 Semantic Web Services Technologies

While still relying on messaging and syntactic descriptions via SOAP and WSDL, Semantic Web Services attempted to enrich service descriptions related to the functions or capabilities of services, towards automating discovery, composition and mediation. Major funding sources for academic research were the United States’ DARPA via the DAML (DARPA Agent Markup Language) programme, the European Commission via several framework projects and various European national funding bodies.

Description. Frameworks like OWL-S, the Web Ontology Language for Web Services [24], which resulted in a W3C Member Submission in 2004 [23], and WSMO, the Web Service Modeling Ontology [28], which resulted in a W3C Member Submission in 2005 [9], provide semantic extensions of service descriptions. The semantic annotations encompass descriptions of overall capabilities as well as **i**nputs, **o**utputs, **p**re-conditions, and **e**ffects (IOPEs) with declaratively described concepts, e.g., using OWL, the Web Ontology Language, or WSML, the Web Service Modeling Language.

Discovery. Based on semantic annotations, approaches for automated semantic service matchmaking via ontological reasoning, e.g., [20], have been proposed. The approaches leverage ontological reasoning to match (by subsumption) descriptions of goals, or parts thereof, to service capabilities.

Composition. The annotations provide the formal descriptions that symbolic methods of AI planning require. Thus, with regard to automated composition, approaches for generating orchestrations, e.g., [27], have been prototypically implemented. Subsequently, information related to composition, e.g., how a service makes use of other services in an orchestration, can be added to the service descriptions.

Mediation. Especially WSMO put mediation into the focus and defined mediation components to connect and reconcile differences between **o**ntologies, **g**oals, and **w**eb services. The components are modular and reusable (OO, GG, WG, and WW mediators). In OWL-S, mediators are just another service [25]. Similar to discovery, the ambition was that mediation problems were solvable via declarative descriptions and symbolic reasoning.

Semantic Web Services remained largely an academic endeavour, partially because of the high effort for providing the required formal semantic descriptions. The shortage of service descriptions kept repositories and engines – such as WSMO’s reference execution architecture (WSMX, the Web Service Execution Environment) [15] – at a mostly conceptual or prototypical level. Work on OWL-S and WSMO ceased around 2008, when the last releases and deliverables were written[1].

Overall, the SWS technology stack was geared towards supporting agents with automated discovery and composition of services based on formal descriptions. Runtime components would make use of the annotations: planners would automatically construct orchestrations given user goals; logical reasoners would support matchmaking and resolve interoperability problems in mediation.

3 (Semantic) Web Services vs. Agentic AI

Given the historical backdrop, the main part of our talk shall emphasize the striking similarities, but also still missing pieces in comparison of technologies envisioned as part of the (Semantic) Web Services stack with the now emerging technologies in the agentic AI stack, in particular MCP and A2A.

Messaging. In terms of architectural style for network communication, all of WS-*, SWS, MCP and A2A use variants of message-passing or remote procedure calls, such as JSON-RPC.

Description. The MCP specification regarding tool descriptions and the A2A specification regarding cards and skills can be roughly compared to WS-* descriptions. There is some kind of description of what the tool or agent does in natural language and what input/output parameters should look like in JSON Schema or via content types. Such kinds of descriptions are prone to ambiguity as the semantics of the functionality or capability is not precisely captured with natural language descriptions alone.

Discovery. MCP and A2A currently expect agents to access a fixed set of predefined tools, which makes discovery trivial. Building managed, trusted repositories within agent protocols is still in its infancy, below the level of development of, e.g., UDDI. We may view such controlled repositories similar to “app stores”: while first such repositories currently start surfacing without a broadly trusted standard emerging as of yet, respective security concerns have, for instance, already been raised [16]. Moreover, how tools and agents can be discovered and searched dynamically in a safe manner remains an open question. Be it for local or remote tools, how an LLM chooses among a set of similar tools for a given task remains implicit and potentially ambiguous: while language model agents can figure out which tools to use, the underlying decisions remain opaque.

Composition. WS-* and SWS focussed on orchestration. The currently predominant composition approaches around ReAct [32] or Plan-and-Solve [31] in agentic AI, which resemble but also do not fully utilise the potential of known planning algorithms, hardly go beyond, in that a centralised orchestration is assumed. And similar to discovery, the decisions governed by an LLM around composition are difficult to inspect and explain.

Mediation. While automating mediation between services leveraging the capabilities of LLMs to generate structured messages with few examples seems a leap forward, the mediation between the terminology and message structures of the different tools called

[1] For WSMO see https://web.archive.org/web/20090214193307/http://www.wsmo.org/TR/, and for OWL-S see https://web.archive.org/web/20090916012959/https://www.daml.org/services/owl-s/.

is implicitly handled by the LLM, and again lacks declarative control, e.g., through respective contracts, or explicit debugging/renegotiating capabilities in case of resulting failed or redundant tool calls. In addition, for instance WSMO identified other types of mediation apart from message formats alone, namely between goals and between the different communication paradigms and directions of services. Whether and how those more involved types of mediation can be addressed by LLMs is also an open question.

4 Summary and Outlook

In summary, we believe that important lessons can be learnt from revisiting traditional (Semantic) Web Services to shed light on the current state of agentic AI. While the main ideas and principles behind MCP or A2A based agents are not fundamentally new, LLMs and their capabilities have enabled the automation of discovery, composition and mediation not previously achievable. However, as we have argued, automation via LLMs comes at the cost of reduced transparency and control. On the other hand, now possible tradeoffs in terms of the formal details of service descriptions bring new opportunities for research.

Common to the systems that are most developed, from (Semantic) Web Services to recent agentic AI approaches, is that they tend to favour a centralised perspective with (hierarchical) power structures: the registry operator controls access to the registry and the orchestrator controls the composed services. The web, with a permissionless and decentralised architecture (anybody can set hyperlinks without asking; there is no catalogue built into web architecture), potentially provides the means to facilitate coordination among peers. Research questions remain around agents operating in a resource-centred environment, with hypermedia providing decentralised ways for discovery. Also, a full treatment of decentrally governed (multi-agent) choreography – where collaborative agents each pursue potentially distinct goals – remains open. On a commercial level, non-functional aspects and service delivery (in the sense of a business value or product) is currently neither fully observable in the digital realm nor governed by formally represented contracts.

Acknowledgements. This research was funded in part by the Austrian Science Fund (FWF) [10.55776/COE12], the German Research Foundation (DFG) (459291153) and the German Ministry of Research, Technology and Space (BMFTR) (FKZ 16IS23052B). We dedicate this paper to the late Dieter Fensel, who envisioned key principles of agentic AI through his pioneering work on Semantic Web Services decades ago.

References

[1] Marco Aiello. 2018. *The Web Was Done by Amateurs - A Reflection on One of the Largest Collective Systems Ever Engineered.* Springer.

[2] Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. 2004. *Web Services: Concepts, Architectures and Applications.* Springer.

[3] Alexandre Alves, Assaf Arkin, Sid Askary, Charlton Barreto, Ben Bloch, Francisco Curbera, Mark Ford, Yaron Goland, Alejandro Guízar, Neelakantan Kartha, Canyang Kevin Liu, Rania Khalaf, Dieter König, Mike Marin, Vinkesh Mehta, Satish Thatte, Danny van der Rijn, Prasad Yendluri, and Alex Yiu. 2007. *Web Services Business Process Execution Language Version 2.0.* OASIS Standard. https://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html.

[4] Anthropic. 2025. Model Context Protocol Specification, Version 2025-06-18. MCP Specification. https://modelcontextprotocol.io/specification/2025-06-18

[5] Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The Semantic Web: A New Form of Web Content that is Meaningful to Computers will Unleash a Revolution of New Possibilities. *Scientific American* (2001).

[6] Don Box. 2001. A Brief History of SOAP. https://www.xml.com/pub/a/ws/2001/04/04/soap.html.

[7] Douglas Bryan, Vadim Draluk, Dave Ehnebuske, Tom Glover, Andrew Hately, Yin Leng Husband, Alan Karp, Keisuke Kibakura, Chris Kurt, Jeff Lancelle, Sam Lee, Sean MacRoibeaird, Anne Thomas Manes, Barbara McKee, Joel Munter, Tammy Nordan, Chuck Reeves, Dan Rogers, Christine Tomlinson, Cafer Tosun, Claus von Riegen, and Prasad Yendluri. 2002. *UDDI Version 2.04 API Specification.* OASIS Standard. http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.htm.

[8] Roberto Chinnici, Jean-Jacques Moreau, Arthur Ryman, and Sanjiva Weerawarana. 2007. *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language.* W3C Recommendation. https://www.w3.org/TR/wsdl20/.

[9] Jos de Bruijn, Christoph Bussler, John Domingue, Dieter Fensel, Martin Hepp, Uwe Keller, Michael Kifer, Birgitta König-Ries, Jacek Kopecky, Rubén Lara, Holger Lausen, Eyal Oren, Axel Polleres, Dumitru Roman, James Scicluna, and Michael Stollberg. 2005. *Web Service Modeling Ontology (WSMO).* W3C Member Submission. https://www.w3.org/submissions/WSMO/.

[10] Mark Endrei, Jenny Ang, Ali Arsanjani, Sook Chua, Philippe Comte, Pål Krogdahl, Min Luo, and Tony Newling. 2004. *Patterns: service-oriented architecture and web services.* IBM Corporation.

[11] Dieter Fensel, Holger Lausen, Axel Polleres, Jos De Bruijn, Michael Stollberg, Dumitru Roman, and John Domingue. 2006. *Enabling semantic web services.* Springer Science & Business Media.

[12] Carol Geyer. 2008. History of BPEL. https://bpel.xml.org/history.

[13] Google and A2A Project Contributors. 2025. Agent2Agent (A2A) Protocol Specification. https://a2a-protocol.org/v0.3.0/specification/. Version 0.3.0.

[14] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, Anish Karmarkar, and Yves Lafon. 2007. *SOAP Version 1.2 Part 1: Messaging Framework (Second Edition).* W3C Recommendation. https://www.w3.org/TR/soap12/.

[15] Armin Haller, Emilia Cimpian, Adrian Mocan, Eyal Oren, and Christoph Bussler. 2005. WSMX - a semantic service-oriented architecture. In *Proceedings of the 2nd IEEE International Conference on Web Services (ICWS).* 321–328.

[16] Xinyi Hou, Yanjie Zhao, and Haoyu Wang. 2025. On the (In)Security of LLM App Stores. In *Proceedings of the 46th IEEE Symposium on Security and Privacy (SP).* 317–335. https://doi.org/10.1109/SP61157.2025.00117

[17] Dave Ings, Luc Clément, Dieter König, Vinkesh Mehta, Ralf Mueller, Ravi Rangaswamy, Michael Rowley, and Ivana Trickovic. 2010. *Web Services – Human Task (WS-HumanTask) Specification Version 1.1.* OASIS Committee Specification. https://docs.oasis-open.org/bpel4people/ws-humantask-1.1-spec-cs-01.html.

[18] Dave Ings, Luc Clément, Dieter König, Vinkesh Mehta, Ralf Mueller, Ravi Rangaswamy, Michael Rowley, and Ivana Trickovic. 2010. *WS-BPEL Extension for People (BPEL4People) Specification Version 1.1.* OASIS Committee Specification. https://docs.oasis-open.org/bpel4people/bpel4people-1.1.html.

[19] Nickolas Kavantzas, David Burdett, Gregory Ritzinger, Tony Fletcher, Yves Lafon, and Charlton Barreto. 2005. *Web Services Choreography Description Language Version 1.0.* W3C Candidate Recommendation. https://www.w3.org/TR/ws-cdl-10/.

[20] Matthias Klusch, Patrick Kapahnke, Stefan Schulte, Freddy Lecue, and Abraham Bernstein. 2016. Semantic web service search: a brief survey. *Künstliche Intelligenz* 30, 22 (2016), 139–147. https://doi.org/10.1007/s13218-015-0415-7

[21] Jacek Kopecký, Tomas Vitvar, Carine Bournez, and Joel Farrell. 2007. SAWSDL: Semantic Annotations for WSDL and XML Schema. *IEEE Internet Computing* 11, 6 (2007), 60–67. https://doi.org/10.1109/MIC.2007.134

[22] Kelvin Lawrence, Chris Kaler, Anthony Nadalin, Ronald Monzillo, and Phillip Hallam-Baker. 2006. *Web Services Security: SOAP Message Security 1.1 (WS-Security).* OASIS Standard. https://groups.oasis-open.org/higherlogic/ws/public/download/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf.

[23] David Martin, Mark Burstein, Jerry Hobbs, Ora Lassila, Drew McDermott, Sheila McIlraith, Srini Narayanan, Massimo Paolucci, Bijan Parsia, Terry Payne, Evren Sirin, Naveen Srinivasan, and Katia Sycara. 2004. *OWL-S: Semantic Markup for Web Services.* W3C Member Submission. https://www.w3.org/submissions/OWL-S/.

[24] David Martin, Massimo Paolucci, Sheila McIlraith, Mark Burstein, Drew McDermott, Deborah McGuinness, Bijan Parsia, Terry Payne, Marta Sabou, Monika Solanki, Naveen Srinivasan, and Katia Sycara. 2004. Bringing Semantics to Web Services: The OWL-S Approach. In *Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition (SWSWPC).* 26–42.

[25] Massimo Paolucci, Naveen Srinivasan, and Katia P. Sycara. 2004. Expressing WSMO Mediators in OWL-S. In *Proceedings of the Workshop on Semantic Web Services (SWS) at the 3rd International Semantic Web Conference (ISWC).*

[26] Mike P. Papazoglou and Willem-Jan van den Heuvel. 2007. Service oriented architectures: approaches, technologies and research issues. *VLDB J.* 16, 3 (2007), 389–415. https://doi.org/10.1007/S00778-007-0044-3

[27] Marco Pistore, Fabio Barbon, Piergiorgio Bertoli, Dmitry Shaparau, and Paolo Traverso. 2004. Planning and monitoring web service composition. In *Proceedings of the 11th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA)*. Springer, 106–115.

[28] Dumitru Roman, Uwe Keller, Holger Lausen, Jos De Bruijn, Rubén Lara, Michael Stollberg, Axel Polleres, Cristina Feier, Cristoph Bussler, and Dieter Fensel. 2005. Web service modeling ontology. *Applied ontology* 1, 1 (2005), 77–106.

[29] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS)*.

[30] Asir S Vedamuthu, David Orchard, Frederick Hirsch, Maryann Hondo, Prasad Yendluri, Toufic Boubez, and Ümit Yalçinalp. 2007. *Web Services Policy 1.5 - Framework*. W3C Recommendation. https://www.w3.org/TR/ws-policy.

[31] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2609–2634. https://doi.org/10.18653/v1/2023.acl-long.147

[32] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*.