Towards a Modular Ontology for Autonomous Robotic Orchestration

Michael McCain[1,*], Chris Davis Jaldi[1], Susan Shrestha[1], Shreyas Casturi[1] and Cogan Shimizu[1]

[1] *Wright State University, 3640 Colonel Glenn Hwy, Dayton, OH, 45435, USA*

Abstract

Domains that rely heavily on robotics have shown increasing interest in collaborative, intelligent, and reasoning-capable autonomous systems. In pursuit of these interests, a standard method for increasing autonomy among robotic agents and enabling collaboration is the use of knowledge representation in the form of ontologies and knowledge graphs. To further enhance these systems, we are creating a modular ontology for autonomous robotic orchestration, using the Modular Ontology Modeling methodology, which focuses on developing ontology patterns to facilitate reuse. This effort aims to achieve a greater goal, an artificially intelligent orchestrator capable of commanding autonomous robots in unpredictable environments, titled Task Adaptation with Shared Knowledge for Multi-Agent Teaming Systems (TASK-MATS). We introduce well-known robotic architectures that incorporate specific ontologies into their frameworks and discuss how those foundational ontologies inspired an expansion of their concepts to fit our use case. To conclude, we briefly discuss the next steps in our ontology modeling efforts.

Keywords

robotics, ontology, knowledge graph, orchestration, multi-agent system

1. Introduction

Domains that rely heavily on robotics have shown an increasing interest in autonomous systems that are collaborative, intelligent, and capable of reasoning. In pursuit of these interests, a standard method for increasing autonomy among robots and enabling collaboration is the use of knowledge representation in the form of ontologies and knowledge graphs (KGs) [1, 2, 3, 4, 5]. Due to the relatively recent focus on achieving these ambitious and complex autonomous systems, there are many competing approaches, especially in the context of task planning [4, 1, 3, 6, 7, 8]. However, these approaches vary in terms of ontological design or methodology, but do use the World Wide Web Consortium (W3C) consortium's Web Ontology Language (OWL)[1] as the primary logic-based language for semantic representation. Unfortunately, as highlighted in [1], many of these ontologies lack reusability and do not meet the need for the confluence of existing ontologies.

Considering these aspects for the current ontological representations of complex autonomous systems, we want to engineer an ontology that could enable these multi-agent systems (MAS) to be governed and commanded by an AI Orchestrator (AIO) capable of the following: understanding its environment, decomposing goals into atomic subtasks, agent assignment based on capabilities, and status verification. This is part of a greater effort to create a KG-powered MAS, titled Task Adaptation with Shared Knowledge for Multi-Agent Teaming Systems (TASK-MATS). More importantly, it would be designed to support reusability and allow for adaptation to the domain of need. This ontology would be modeled using state-of-the-art (SOTA) best practices to ensure modularity and reusability through the use of ontological design patterns (ODPs) [9, 10, 11, 12]. This would fill two needs. The first would be to

RobOntics'25: International Workshop on Ontologies for Autonomous Robotics, December 10, 2025

*Corresponding author.
mccain.32@wright.edu (M. McCain); jaldi.2@wright.edu (C. D. Jaldi); shrestha.167@wright.edu (S. Shrestha); casturi.2@wright.edu (S. Casturi); cogan.shimizu@wright.edu (C. Shimizu)
0009-0008-3664-6248 (M. McCain); 0009-0000-2287-1198 (C. D. Jaldi); 0009-0005-0728-6851 (S. Shrestha); 0009-0007-9111-9830 (S. Casturi); 0000-0003-4283-8701 (C. Shimizu)

[1] www.w3.org/TR/2012/REC-owl2-primer-20121211/

provide an ontology that allows for more complex autonomous MAS. The second is the demand for reusable ODPs that facilitates the merging of existing ontologies across domains that utilize these MAS.

The rest of this paper is organized as follows. Section 2 showcases how ontologies and KGs are being utilized to enable autonomous robots. Section 3 showcases the current results from the efforts being made to actualize the proposed ontology. Finally, in section 4, we conclude and discuss the future direction of our work.

2. Background

Bestowing robots with autonomous capabilities in dynamic unstructured environments has proven to be a momentous challenge. However, significant headway has been made toward achieving this through the utilization of robotic frameworks that employ ontologies to aid in building an autonomous agent's worldview.

2.1. Ontology for Collaborative Robotics and Adaptation (OCRA)

Olivares-Alarcos et al. [4] introduce a novel framework that focuses on task planning, adaptation, and collaborative behavior between humans and robots in real time, in unstructured environments. A primary motivation for this research was to ensure safe, dependable completion of tasks when human and robotic agents work together in the same space. OCRA was designed with the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [13] + DnS Ultralite (DUL) [14] as its foundational ontology. As noted in [4], this was due to the common usage of a similar framework, KnowRob [15, 16]. Their formalization approach initially used first-order logic (FOL) [17], followed by a formalization in the OWL Description Logic (DL) [18] to enable a more efficient runtime computation at the expense of knowledge representation. The most important takeaway from their research is that, through the use of an ontology, the authors successfully enabled a robotic arm to safely perform the same task alongside a human and adapt to the dynamic environment as needed.

2.2. KnowRob Ontologies

KnowRob 2.0[2] is arguably one of the more successful implementations of a framework for enhancing robotic agent capabilities in unstructured environments[3] [16]. It is a modified version of the initial framework showcased in 2013 [15]. In this newer version, their ontology is the focal point around which the framework is built. In the original version, the authors expanded on the OpenCyc [19] ontology, and they formalized their knowledge using OWL DL. In contrast, the KnowRob 2.0 ontology uses DOLCE + DUL (specified in FOL) as its foundational ontology instead. In addition to this, the authors use patterns from the Socio-physical Model of Activities (SOMA) for Autonomous Robotic Agents[4] [20]. A primary reason for this extreme shift in ontology choice between versions was due to the fact that ODPs have been proven more effective for ontology modeling[5] [13], as highlighted in [9] and [10].

KnowRob 2.0 is significantly more complex than other existing frameworks, as it includes several unique architectural components, such as virtual simulations, episodic memory, and the ability to learn from their outputs. However, to enable any of these to work, an ontology is required to represent the semantic meaning of the data in the knowledge base. Without their ontology, the robots that employ their software would be unable to infer or reason about their environment [16]. Further justifying the significance of semantic knowledge representation in the form of ontologies, in the context of enabling autonomy in robots.

[2]https://www.knowrob.org/
[3]https://github.com/knowrob/knowrob
[4]https://ease-crc.github.io/soma/
[5]https://knowrob.org/ontologies

2.3. MOMo - Modular Ontology Modeling

The Modular Ontology Modeling (MOMo) methodology is described in detail in [11], but is briefly summarized in Figure 1.[6] MOMo provides a systematic approach for developing robust and reusable ontologies designed to function as a schema for a KG. This approach is designed to address limitations in utilizing monolithic ontologies, as they are often difficult to reuse due to either *strong* ontological commitments that result in overspecification or *weak* commitments that cause ambiguity. We have chosen to use MOMo for these benefits, as well as due to our own significant experience with this particular method.

1. Define the use case
2. Make competency questions
3. Identify key notions
4. Match patterns to key notions
5. Instantiate the patterns
6. Systematic axiomatization
7. Assemble the modules
8. Review final product
9. Produce artifacts

Figure 1: The steps taken in the Modular Ontology Modeling methodology (briefly).

Before we discuss the paper's position, we summarize our implementation of [11]. We cover the initial conceptual phases and the creation of the core visual architecture. Specifically, we followed steps 1-5 in Figure 1 by first defining a clear set of competency questions for the modular ontology of robotic orchestration, which guided the identification of key notions such as Agent, Task, Environment, Goal, and more. Each notion was mapped to either an existing ODP template when adaptable, or was modeled as a new pattern, to follow established modeling principles and best practices. The resulting modular structure is expressed through schema diagrams representing inter-module relationships.

Positionally, MOMo builds upon the same design philosophy that underlies traditional ontology engineering frameworks such as DOLCE + DUL, and SOMA, i.e., emphasizing modularization and compositional reuse and enabling a systematic alignment of multiple ODPs across heterogeneous domains. However, while other frameworks achieve this through curated ontological imports or handcrafted integrations, MOMo distinguishes itself by providing an explicit, prescriptive workflow for implementing these principles that formalizes these best practices into a repeatable and transparent methodology. It also serves as a methodological scaffold for applying SOTA modular engineering practices while ensuring each modeling decision is explicit and traceable. This makes it particularly well-suited for the orchestration of complex, cross-domain and ontology-guided MAS that require explicit interoperability between modules. Hence, we contribute and position a growing shift towards pattern-driven ontology engineering that prioritizes reusability, traceability and empirical grounding. The immediate future work will focus on executing steps 6-9 of Figure 1, which involves formalizing axioms, assembling and validating modules and publishing the final artifacts for broader community reuse.

3. Position & Preliminary Results

Our approach toward creating a modular ontology for robotic orchestration followed the methodology presented and outlined in MOMo. This requires that we begin by identifying a use case, determining useful datasets, and developing competency questions that would allow for future verification of the ontology. We reiterate our use case: to construct an SOTA reusable ontology that allows for complex MAS that incorporates an AIO for task assignment and verification in dynamic and unstructured environments. From this, we identified many applicable datasets that encompass data on objects, robotic manipulation, environments, tasking, and more. We then began identifying key notions, or rather, concepts that overlap within the competency questions and data [11, 21].

The key notions act as the basis for modeling the ontology. First, we examined patterns that already exist within MODL [10], where we found well developed ODPs, such as those involving State, Event,

[6]Further resources can be found in https://github.com/kastle-lab/cs7810-intro-to-ke and https://github.com/kastle-lab/kastle-drawbridge.

and SpatioTemporalExtent. We also identified a pattern from another modular ontology (i.e., OntoPret [22]): the pattern structure for Behavior. For brevity, many other patterns were also identified; they are included in our repository. Unfortunately, many of our key notions still lacked suitable ODPs, and they will require further effort through bespoke modeling. Some notable key notions without suitable patterns that we have begun developing for our use case are described in the following subsections of Section 3.

The deliverables for this ontology and the current collection of identified datasets are available in our GitHub repository[7].

3.1. Goal & Task

Task decomposition is and remains a central theme across many prior works. Several frameworks, such as KnowRob 2.0 [16], ORPP [7] and RTPO [6], have each modeled aspects of task planning, execution, or process sequencing, yet they differ significantly in modularity and reusability. For example, KnowRob 2.0 embedded decomposition semantics into its activity model via SOMA's hierarchy, making the relationships tightly coupled to its cognitive architecture. In the same vein, ORPP introduced a skill-based process planning ontology but centered its design around agile manufacturing workflows rather than generalized orchestration. Similarly, RTPO focused on explicit domain knowledge bases for robot task planning but employed monolithic class structures, thereby restricting extensibility.

Building on these past limitations and lessons observed, we developed a Goal-Task module to formalize how high-level objectives can be decomposed into actionable units of work within our orchestration ontology. Following the principles of MOMo, we treat Goal as the abstract representation of the desired outcome while keeping Task as the minimal operational unit required to achieve that outcome, keeping it atomic and hence not further decomposable.

As portrayed in Figure 2, a Goal instantiated relationships to Task through hasTask property for explicit decomposition of work from strategic intent to an actionable executable item. This modeling approach explores a different angle of decomposition and helps preserve the work decomposition hierarchy from a 'what needs to be achieved and done', rather than generic breakdowns, which focus on the 'how it should be done', to capture the broader semantics of intention behind work planning and completion. This also captures the Spatial and Temporal constraints (hasDeadline, isWithinBounds and hasOperationalExtent), which contextualize the broader tasks and goals of a system within real-world limits, also bounding it to their respective Environment extents. Each Goal is further linked to a success condition via hasSuccessState, allowing the AIO to evaluate completion criteria, especially in dynamic environments.

Similarly, individual Tasks were modeled for sequencing for partial order execution and dependency management (dependsOnTask and hasNextTask), as well as references to required objects (requiresObject) and necessary Archetype (requiresArchetype).

Extending several patterns from MODL, these design primitives, unlike several prior frameworks that embed semantics with

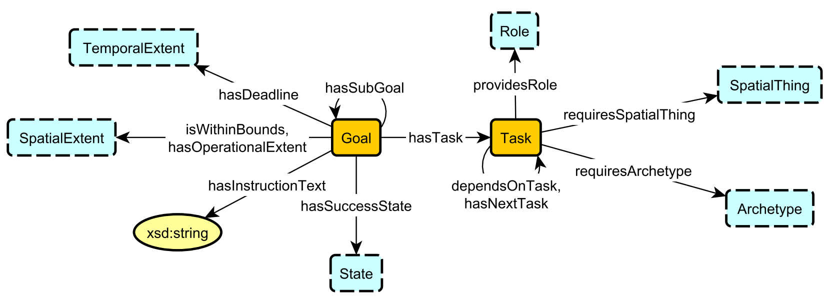
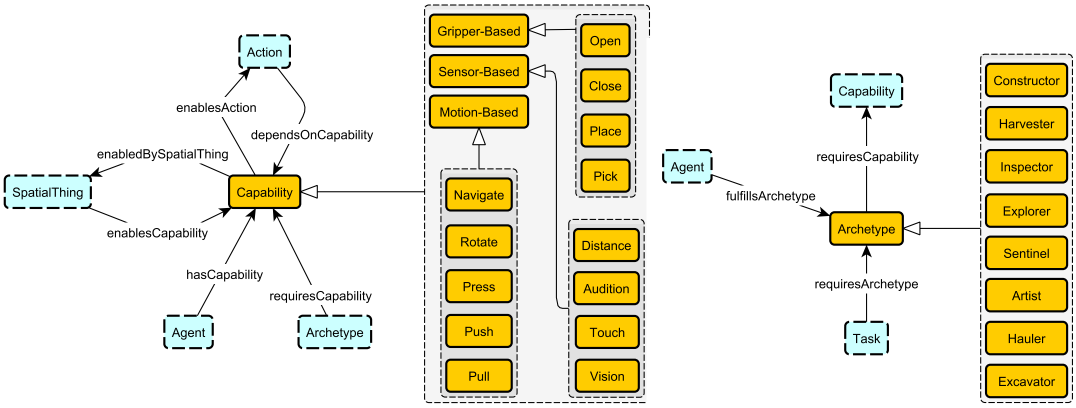


Figure 2: The schema diagram for Goal and Task. The yellow boxes indicate the classes, the blue dashed boxes refer to external schema entities and the yellow ellipses indicate the datatypes tied to those classes. Open arrows indicate SubclassOf relations.

[7]The current work for this ontology https://github.com/kastle-lab/Autonomous-Robotic-Orchestration-Modular-Ontology.



(a) The schema diagram for Capability. (b) The schema diagram for Archetype.

Figure 3: This figure shows the schema diagrams for both Capability (Figure 3a) and Archetype (Figure 3b), as modeled using the standards outlined in MOMo.[8]

static rule sets, ensure compatibility with other modules for ontological consistency across the framework. Modeling was strongly influenced and grounded in task dependencies, temporal ordering and goal hierarchies by datasets such as ALFRED [23] and RH20T-P [24], designed specifically for robot-robot and human-robot collaboration scenarios. In short, this module helps fill the conceptual bridge between intention and execution by connecting goals and tasks, embedding environmental and dependency semantics for AIO to reason over MAS plans, verify desired outcomes and adaptively recompose tasks to maintain executive and structural continuity.

3.2. Archetype & Capability

Both the OCRA and KnowRob frameworks import a Capabilities class as defined by SOMA. However, neither of these ontologies extends or makes use of the class in a meaningful way. Notwithstanding, a well-developed Capability class is required for our use case. In SOMA, the Capability class is entirely abstract, but further extending it would be more appropriate for a MAS with an AIO for tracking the set of capabilities of each robotic agent under its influence. The purpose is to provide the AIO with the capacity to reason about the available agents and their capabilities under its authority. To then appropriate the necessary agents to achieve an overall goal and complete atomic-level tasks, based on the required skills. The current schema for the capability pattern is shown in Figure 3a.

Furthermore, the modeling of the capability pattern was heavily influenced by existing available data. One useful dataset of note is RH20T-P, provided by an internationally collaborative research team from China and Australia. In their dataset, they model what they define as Primitive Skills under two categories: Gripper-Based and Motion-Based [24]. These are useful distinctions to make in the Capabilities class, as the capacity of robotic agents is typically constrained by their hardware and software. The composition of such capabilities results in the next class of discussion, Archetype.

Analysis of Figure 3a reveals a connected class, Archetype, through the relationship requiresCapability, as exhibited in Figure 3b. The Archetype class is a thematic representation of an agent's capability. For example, a robotic Agent of the Explorer Archetype would have specific sensor and motion-based capabilities. We chose to model an Archetype class rather than use the nomenclature Role, because there are important distinctions between the two. While Role is typically used to classify an agent's immediate characteristics, the Archetype class is instead more representative of the culmination of all the characteristics an agent has based on their capabilities (i.e., a set of skills). The specific set of those capabilities gives an agent an archetype, allowing it to fill many roles until it is assigned to an immediate

[8]Figure 2 contains the legend for MOMo schema diagrams.

task, where it will fulfill a specific role. As these patterns become more concrete, the archetypes would be enforced through the constraints imposed by axioms.

4. Conclusions

We are engineering a modular ontology, using SOTA best practices, to enable complex autonomous MAS to include an AIO capable of managing other robotic agents. Currently, there are two well-known exemplary frameworks, OCRA and KnowRob, that showcase the significant role that ontologies play in the pursuit of such complex systems. These frameworks use the same foundational ontologies — DOLCE+DUL and SOMA — and extend them as needed. In our examination of these, we found it necessary to extend and add to some of the concepts presented in the aforementioned ontologies, not necessarily an extension of the ontologies themselves, by creating the ODPs we identified as a requirement for our use case. This is part of a greater effort to create a KG-powered MAS, which we are coining as TASK-MATS.

Future Work We have identified several directions for future work. Of course, we must finalize our own ontology efforts. As a part of this, however, an exhaustive collection of specification data, especially related to capability, must be executed. This ensures that not just an ontological notion is well-modeled, but also that it is empirically validated using real-world scenarios. The TASK-MATS project supports this initiative, but annexing or otherwise integrating into the wider community is a must. Finally, we must look forward into how this ontology can be used in broader neurosymbolic frameworks, incorporating planning and multi-modality. Though the work is mostly preliminary and much future work remains, we conclude that, at the very least, the current state of this work is a proper step in the direction toward achieving our endeavors.

Acknowledgments

Michael McCain and Cogan Shimizu acknowledge support from DAGSI RX26-24: "TASK-MATS: Task Adaptation with Shared Knowledge for Multi-Agent Teaming Systems".

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly to perform a grammar and spelling check. Generative AI was used to identify academic resources during the literature search phase of this research, but these sources were manually reviewed.

References

[1] E. Aguado, V. Gomez, M. Hernando, C. Rossi, R. Sanz, A survey of ontology-enabled processes for dependable robot autonomy, Frontiers in Robotics and AI 11 (2024). doi:10.3389/frobt.2024.1377897.

[2] H. Du, S. Thudumu, R. Vasa, K. Mouzakis, A survey on context-aware multi-agent systems: Techniques, challenges and future directions, 2024. URL: https://arxiv.org/abs/2402.01968v2.

[3] H. Li, P. Goncalves, V. Ragavan, A. Olivares-Alarcos, E. M. Barreto, D. Beler, J. Bermejo, S. Borgo, A. Chibani, J. Carbonera, M. Diab, S. Fiorini, A. Gyrard, M. Habib, A. Khamis, K. Moulouel, H. Nakawala, B. Nguyen, C. Nowak, J. Olszewska, E. Pignaton, E. Prestes, J. Quintas, S. Redfield, R. Sanz, C. Schlenoff, E. Tosello, Ieee standard for autonomous robotics (aur) ontology (2021). URL: https://ieeexplore.ieee.org/document/9774339?utm_source=chatgpt.com. doi:10.1109/ieeestd.2022.9774339.

[4] A. Olivares-Alarcos, S. Foix, S. Borgo, N. G. Alenyà, Ocra – an ontology for collaborative robotics and adaptation, Computers in Industry 138 (2022) 103627–103627. doi:10.1016/j.compind.2022.103627.
[5] A. Ricci, M. Piunti, M. Viroli, Environment programming in multi-agent systems: an artifact-based perspective, Autonomous Agents and Multi-Agent Systems 23 (2010) 158–192. doi:10.1007/s10458-010-9140-7.
[6] X. Sun, Y. Zhang, J. M. Chen, Rtpo: A domain knowledge base for robot task planning 8 (2019) 1105–1105. doi:10.3390/electronics8101105.
[7] C. Z. Sprenger, J. Antonio, N. U. Baier, Orpp—an ontology for skill-based robotic process planning in agile manufacturing, Electronics 13 (2024) 3666–3666. URL: https://www.mdpi.com/2079-9292/13/18/3666. doi:10.3390/electronics13183666.
[8] L. Kinder, T. Käfer, Static and adaptive planning with wot td by generating python objects as intermediary representations using large language models, Lecture notes in computer science (2025) 256–273. doi:10.1007/978-3-031-78955-7_21.
[9] A. Gangemi, Ontology design patterns for semantic web content, Lecture Notes in Computer Science (2005) 262–276. doi:https://doi.org/10.1007/11574620_21.
[10] C. Shimizu, Q. Hirt, P. Hitzler, Modl: A modular ontology design library, 2019. URL: https://arxiv.org/abs/1904.05405.
[11] C. Shimizu, K. Hammar, P. Hitzler, Modular ontology modeling, Semantic Web (2022) 1–31. doi:https://doi.org/10.3233/sw-222886.
[12] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Computing Surveys 54 (2022) 1–37. doi:10.1145/3447772.
[13] S. Borgo, R. Ferrario, A. Gangemi, N. Guarino, C. Masolo, D. Porello, E. M. Sanfilippo, L. Vieu, Dolce: A descriptive ontology for linguistic and cognitive engineering, Applied Ontology Preprint (2021) 1–25. URL: https://content.iospress.com/articles/applied-ontology/ao210259. doi:https://doi.org/10.3233/AO-210259.
[14] A. Scherp, T. Franz, C. Saathoff, S. Staab, F – a model of events based on the foundational ontology dolce+dns ultralite, 2024. URL: https://arxiv.org/abs/2411.16609. arXiv:2411.16609.
[15] M. Tenorth, M. Beetz, Knowrob: A knowledge processing infrastructure for cognition-enabled robots, The International Journal of Robotics Research 32 (2013) 566–590. doi:10.1177/0278364913481635.
[16] M. Beetz, D. Bessler, A. Haidu, M. Pomarlan, A. K. Bozcuoglu, G. Bartels, Know rob 2.0 — a 2nd generation knowledge processing framework for cognition-enabled robotic agents (2018). doi:10.1109/icra.2018.8460964.
[17] R. M. Smullyan, First-Order Logic. Preliminaries, Springer Berlin Heidelberg, Berlin, Heidelberg, 1968, pp. 43–52. URL: https://doi.org/10.1007/978-3-642-86718-7_4. doi:10.1007/978-3-642-86718-7_4.
[18] G. K. Saha, Web ontology language (owl) and semantic web, Ubiquity 2007 (2007) 1–1. doi:https://doi.org/10.1145/1295289.1295290.
[19] D. B. Lenat, Cyc: a large-scale investment in knowledge infrastructure, Commun. ACM 38 (1995) 33–38. URL: https://doi.org/10.1145/219717.219745. doi:10.1145/219717.219745.
[20] D. Beßler, R. Porzel, M. Pomarlan, A. Vyas, S. Höffner, M. Beetz, R. Malaka, J. Bateman, Foundations of the socio-physical model of activities (soma) for autonomous robotic agents, 2020. URL: https://arxiv.org/abs/2011.11972. arXiv:2011.11972.
[21] C. Shimizu, P. Hitzler, A. Krisnadhi, Modular ontology modeling: A tutorial, Applications and practices in ontology design, extraction, and reasoning (2020) 3–20.
[22] A. Ellis, S. Severyn, F. Novakazi, H. Banaee, C. Shimizu, Ontopret: An ontology for the interpretation of human behavior, 2025. URL: https://arxiv.org/abs/2510.23553. arXiv:2510.23553.
[23] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, D. Fox, ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks, in: The IEEE Conference

on Computer Vision and Pattern Recognition (CVPR), 2020. URL: https://arxiv.org/abs/1912.01734.
[24] Z. Chen, Z. Shi, X. Lu, L. He, S. Qian, H. S. Fang, Z. Yin, W. Ouyang, J. Shao, Y. Qiao, C. Lu, L. Sheng, Rh20t-p: A primitive-level robotic dataset towards composable generalization agents, arXiv preprint arXiv: 2403.19622 (2024).