Composable Semantic Frames for Grounding Language in Robot Control Primitives

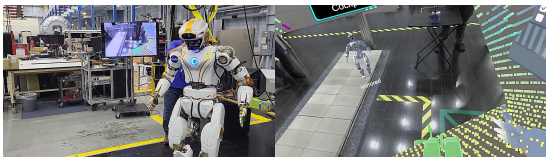
Emily Sheetz[1,2], Matthew Shannon[1], Adam Ingerman[1], Cameron Kisailus[1], Shaun Azimi[2], and Odest Chadwicke Jenkins[1]

***Abstract*— As robots become increasingly capable of taking on challenging tasks, we want robots to be commanded in intuitive ways. Non-expert users in particular should be able to communicate with robots about task goals. As a result, modes of interaction such as language have gained interest for commanding robots. We present *composable semantic frames*, which ground commands in robot control primitives by *composing* high-level commands from lower-level commands. We demonstrate that *composable semantic frames* allow robots to understand and execute a variety of challenging commands, such as those involving multiple verb meanings, command variations, and compound nouns. The robot quickly processes *composable semantic frames* and accurately grounds and executes the commanded tasks, demonstrating the power of *composable semantic frames* for allowing users to intuitively interact with robots.**

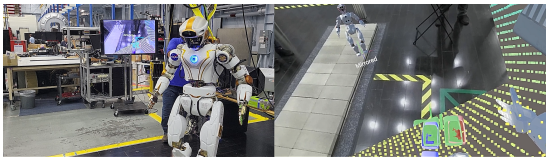
I. INTRODUCTION

Robots are becoming increasingly capable of performing complex tasks, and people are looking to apply robot capabilities to a wider variety of domains. However, these additional complexities make it difficult for non-experts to intuitively interact with robots. Many researchers investigate intuitive ways to interact with and program robots, such as gestures, facial expressions [5], [44], eye-tracking [3], [30], and learning from demonstration [2], [21], [28], [29]. Language in particular has gained much interest as an intuitive interface that can provide a wide variety of rich input signals for commanding robots [42]. RoboFrameNet [43] demonstrates the power of using semantic frames [45] to bridge the gap between language and goal-directed robot actions. Language-based interactions will allow more non-expert users to intuitively interact with robots.

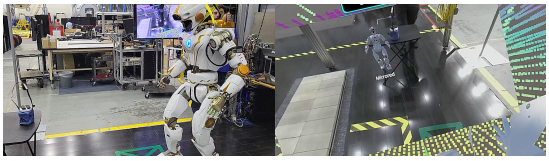
Scaling robot systems to understand all possible mappings of natural language commands to actions is an open question. Systems such as Amazon's Alexa and Apple's Siri [36], [9], [22] indicate that restricted language could be a more scalable solution to providing an intuitive interface for commanding robots. However, limitations in existing semantic frame implementations limit recognizable commands. For example, RoboFrameNet [43] suffers from not being able to differentiate verbs with multiple meanings ("set the table" or "set down the object"), ground command variations (such as "go to Alice's desk" or "go to Charlie's desk"), or recognize compound nouns (such as "Alice's desk" or "left hand"). We must overcome the limitations of existing semantic frame implementations to address the challenge of creating a scalable restricted language for commanding robots.



(a) Operator: "Plan and execute to waypoint."



(b) Robot: "Footstep plan received. Proceed with task?"



(c) Operator: "Yes." Robot: "Proceeding with task."

Fig. 1: Human and robot communicating about a navigation task through a virtual reality (VR) headset. The robot (left) is being commanded by an operator in VR (right).

We take insight from hierarchical robot control schemes and address the challenges of creating a scalable restricted language for robot commands. In robot control, notions of hierarchy and composition allow simple primitives to be combined to create more complex behaviors. Similarly, by composing high-level commands from lower-level semantic frames that are grounded in action, robots can recognize and execute complex commands in a scalable way.

In this paper, we propose *composable semantic frames* (CSFs) as a method for creating a scalable restricted language for commanding robots to perform tasks. CSFs provide more complex command understanding and ground frame actions in robot control primitives. We test our approach on two robot platforms and multiple tasks. Our proposed *composable semantic frames* allow robots to recognize and execute complex commands, thereby providing a scalable approach to using restricted language for commanding robots.

The authors are with the [1]University of Michigan and [2]NASA Johnson Space Center (NASA).

Disclaimer: Trade names and trademarks are used in this report for identification only. Their usage does not constitute an official endorsement, either expressed or implied, by the National Aeronautics and Space Administration.

II. RELATED WORK

A. Language as a Robot Percept

Researchers have long investigated different ways for users to intuitively interact with robots, due to the rich input signals different modalities can provide. In particular, researchers explore how to use language to intuitively communicate task goals to robots. In the 1970s, the system SHRDLU [47], [48] was developed, which carried out natural language commands in a virtual environment. Since then, researchers have aimed to expand the use of natural language to command intelligent agents and robots [42].

Many works demonstrate the power of using natural language to command robots in specific domains. Dzifcak et al. [10] explore how to translate natural language instructions into descriptions of task goals and actions. Chernova et al. [7] use data-mining for robots to ground action-oriented natural language. Matuszek et al. [26] investigate how robots can learn what objects are being referred to in deictic gestures and language (meaning gestures and language that draw attention to objects without naming them directly). Many works explore understanding natural language in route navigation tasks [23], [20], [24], [25]. While these works demonstrate the widespread interest in using language to command robots, scaling to new commands and new domains remains an open question.

Due to the challenges of scaling to truly natural language, commanding robots using restricted language is a useful approach. Voice interfaces–such as Amazon's Alexa, Apple's Siri, Google's Assistant, and Microsoft's Cortana [36], [9], [22]–are part of everyday life. These systems demonstrate the power of restricted language for commanding intelligent agents. Some research indicates that restricted language allows users to achieve similar or better task performance than natural (unrestricted) language without detracting from overall user experience [27]. These works demonstrate the power of using restricted language to communicate intuitively with robots about task goals.

B. Semantic Frames

Semantic frames have been used in the field of natural language processing (NLP) to represent a scene being acted out [40], [45], [46], [16]. FrameNet [40] emphasizes that a verb alone is not sufficient to describe a scene or action, and *frame elements* are necessary to describe agents and direct and indirect objects involved in the action. For example, the verb "give" cannot be acted out until we know what object is being given and to whom. FrameNet also uses *lexical units* to map language into the appropriate semantic frame. Lexical units are hand-annotated to express how frame elements are related to a command.

RoboFrameNet [43] uses semantic frames to serve as a middle-ground between a spoken command and robot action. RoboFrameNet interprets spoken commands as text, then parses the text to instantiate a semantic frame. Representations of object affordances for robotics generally do not explicitly note the direct and indirect objects being acted on, which limits the complexity of robot action that can be performed [49]. For this reason, semantic frames are a particularly useful middle-ground between a spoken command and robot action because it augments the robots understanding of the action being performed.

RoboFrameNet demonstrates the power of semantic frames in allowing robots to comprehend spoken commands. We extend RoboFrameNet by advancing the capabilities and scalability of semantic frames. Rather than focusing on semantic frame instantiation, we place greater emphasis on the execution of the actions represented by semantic frames.

C. Hierarchical Robot Control

Many works control robots using hierarchical control [1] or subsumption architectures [15]. Robots can execute object affordances [11] using a *control basis* of object-centric [4] controllers. A control basis builds up complex actions from simple behavioral building blocks such as grasping [34], [33], [35] or conditioning behaviors [13] such as avoiding joint limits and singularities. Executing complex tasks requires *composition* of the low-level building blocks [39] and sequencing these behaviors [6] to achieve a task goal.

We take inspiration from the *compositions* seen in robot control to ground our *composable semantic frames* in robot action. Similar to how complex robot actions are composed from simple, low-level controllers, our *composable semantic frames* are composed from simple, low-level grounded commands. Composition allows our pipeline to ground high-level commands in a scalable way.

III. METHODS

A. Composable Semantic Frames

For robots to understand a command, we need to specify a *lexical unit* that defines a command. Lexical units contain important information such as synonymous verbs and grammatical dependencies that may be used in the command involving that verb. *Semantic frames* ground the information contained within the corresponding lexical unit by mapping grammatical dependencies to words in the verbal command. Semantic frames can also have *children semantic frames*, which are more specific versions of a command. For example, a `turn` semantic frame may have more specific children frames `turn_left` and `turn_right`.

Our proposed *composable semantic frames* (CSFs) offer several improvements upon previous implementations of semantic frames, specifically RoboFrameNet [43]. The following sections detail the improvements in our CSFs.

1) Optional Frame Elements: Lexical units and semantic frames include *frame elements* to describe grammatical dependencies involved in a verbal command. These frame elements may or may not be *core* to the command, but each frame element is required to understand the command. CSFs differentiate between *required frame elements* and *optional frame elements*. This allows more variation in commands, since some command variations may not use all frame elements.

For example, the commands "give me the block" and "give me the red block" both use the `give` lexical unit and semantic frame. Defining an optional adjective modifier allows both commands to be recognized by telling the CSF to not always expect a modifier. For example, *red* provides optional information to differentiate one block from another, but is not required to understand the command. CSFs can be instantiated as long as the required elements are identified during parsing; the optional elements provide helpful information, but do not prevent the CSF from being instantiated.

2) Head Dependency Relations: The frame elements in each lexical unit and semantic frame have a corresponding grammatical relation. For example, the command "pick up the blue block" includes an adjectival modifier "blue" and a direct object "block". CSFs provide additional information about frame relations by specifying a *head dependency relation*, effectively tying a frame element to other elements it depends on. For example, the head dependency relation for adjectival modifier "blue" would be the direct object "block" since the adjectival modifier describes the direct object.

RoboFrameNet [43] and our CSF pipeline use the Stanford parser [19], which parses the head dependencies of each word in a command. However, whereas RoboFrameNet did not make use of these head dependency relations, our CSF pipeline does. Specifying the head dependency relations allows CSFs to differentiate between frame elements with the same grammatical relation type, and therefore recognize more complex commands. For example, consider the command "stack the red block on the blue block." When parsed, this command involves a direct object (the block being stacked) and an indirect object (the block being stacked on top of). Each of these objects have adjectival modifiers to differentiate the two objects. Previous implementations of semantic frames would not be able to differentiate between the two blocks or determine which block to act on. In contrast, the CSF for `stack` uses head dependency relations to differentiate between the adjectival modifiers; the head dependency relation for "red" is the direct object while the head dependency relation for "blue" is the indirect object. Head dependency relations allow CSFs to make use of additional information in commands and understand more complex commands.

3) Frame Actions: The most important feature of CSFs is that they are grounded in robot action. Each CSF contains a sequence of *actions* required to carry out the commanded task. For CSFs to be grounded in robot action, the actions listed within a CSF must be either: (a) a robot motion control primitive (described further in Section III-B), or (b) a CSF command. By allowing actions within a CSF to be other CSFs, we can create semantic frames that are ***composed*** from other semantic frames. The composable nature of CSFs results in a recursive expansion of actions to obtain a complete sequence of grounded actions for a command.

Composition makes our CSF pipeline modular and scalable. For example, a general purpose CSF `wait_for_confirmation` tells the robot to wait for input from the operator before continuing with a task. Such frames can be reused within other frames; in fact, the `wait_for_confirmation` CSF was used within every experiment in Section IV. Composition of CSFs also means that high-level commands can be created from lower-level commands. With a basis of CSFs grounded in robot control primitives, we guarantee that high-level commands composed of these basis elements can also be grounded in action. Users can communicate high-level task goals to robots without thinking about the low-level action execution. The composability of CSFs provides significant features such as reuse, scalability, high-level task goals, and abstraction of low-level execution.

4) Argument Substitution: Because CSFs use more complex, possibly optional, dependency relations, they also need to (optionally) use these arguments within the frame actions. If a frame element is successfully parsed, it can be substituted in to any of the frame actions within the CSF. This allows the robot to carry out actions involving these frame elements.

Argument substitution in CSFs can be used to differentiate between and act on corresponding objects. For example, the command "stack the red block on the blue block" can be enacted by substituting in the appropriate frame elements to allow the robot to understand it should first pick up the "red block" and then move towards the "blue block." Argument substitution is also useful for command variations. For example, the commands "go to office 103" and "go to office 105" can be recognized using a single CSF `go_to_office`. By substituting in the appropriate office number, the robot can scalably carry out all variations. Argument substitution in CSFs is a powerful tool for execute task variations.

B. Robot Control Primitives

The most important feature of CSFs is that each command is grounded in robot action. To ground the command and enact each CSF, we require that actions within the frames are either other CSF commands or *robot control primitives*. Depending on the robot, the robot control primitives can take a variety of forms. Our implementation of CSFs does not depend on the form of the robot control primitives, just that some basis of primitives exists. For example, robot control primitives can be *affordance templates* [14], *affordance primitives* [31], [32], or a *control basis* [34]. One work defines an example control basis–a set of controller building blocks –as 6D pose, 3D position, alignment (relative rotation), and screw controllers [41].

In our experiments, we use the following robot control primitives: open/close hand, move end-effector to target 6D pose, plan footstep trajectory, and execute footstep trajectory. For the robots we tested on (the Fetch robot and NASA Johnson Space Center's Valkyrie robot [38], [17]), we determined that these robot control primitives were sufficient for a wide variety of tasks. These control primitives correspond directly to operations within the motion control libraries running on these robots (MoveIt [8] for Fetch and IHMC Open Robotics Software [37] for Valkyrie).

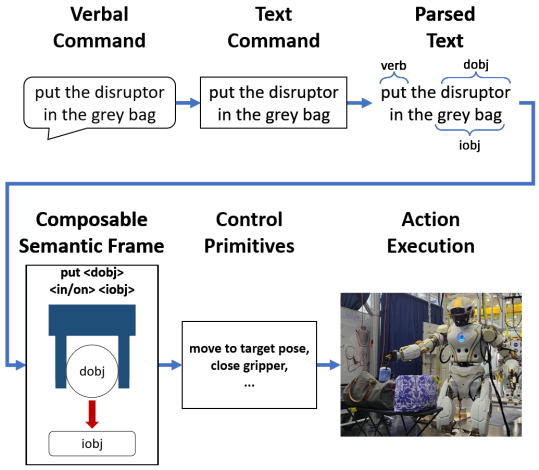


Fig. 2: Our CSF pipeline. The verbal command is parsed and matched with a CSF, which contains the sequence of actions needed to execute the command.

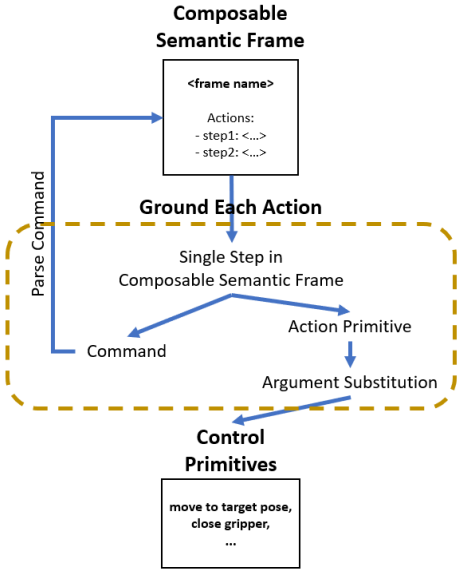


Fig. 3: Our pipeline for converting CSFs to the corresponding sequence of grounded robot control primitives. Each action in a CSF is either a grounded control primitive or a command that can be recursively grounded into another CSF.

IV. EXPERIMENTS AND RESULTS

Figure 2 describes the pipeline for using *composable semantic frames* to ground verbal commands and execute the commanded actions. Figure 3 further details the grounding of verbal commands into sequences of robot control primitives. We assume human-in-the-loop perception through interactive object registration [12]. For the robot to execute a command, it needs to know where the required objects are in the scene.

To demonstrate the capabilities of CSFs, we performed several experiments on the University of Michigan Laboratory for Progress Fetch robot and NASA Johnson Space Center's Valkyrie robot [38], [17], [18].

"Give" Lexical Unit and Composable Semantic Frame

```
# Lexical Unit
# give_lu.yaml
- frame_name: give
  verb: [give]
  description: ""
  frame_element_grammatical_relations:
    - name: give_object
      relation: obj
      head_relation: root
    - name: give_indirect_object
      relation: iobj
      head_relation: root
  optional_frame_element_relations:
    - name: give_det
      relation: det
      head_relation: obj
    - name: give_modifier
      relation: amod
      head_relation: obj
```

```
# Composable Semantic Frame
# give_sf.yaml
name: give
frame_elements:
  - name: give_object
    is_core: True
  - name: give_indirect_object
    is_core: True
  - name: give_det
    is_core: False
  - name: give_modifier
    is_core: False
children:
  - name: give_high_five
    frame_element_relations:
      - parent_name: give_object
        child_value: five
      - parent_name: give_indirect_object
        child_value: ___
      - parent_name: give_det
        child_value: a
      - parent_name: give_modifier
        child_value: high
  - name: give_object_to
    frame_element_relations:
      - parent_name: give_object
        child_value: ___
      - parent_name: give_indirect_object
        child_value: ___
      - parent_name: give_det
        child_value: the
```

Fig. 4: Lexical unit and CSF for `give`. The lexical unit (left) defines optional frame elements and head dependency relations for each frame element, which gives CSFs the flexibility to understand related commands. The CSF (right) contains two related children frames, one of which (blue box) can be seen in Figure 5.

"Give High Five" Composable Semantic Frame

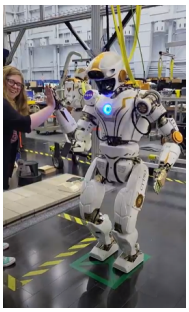
```
# Composable Semantic Frame
# give_high_five_sf.yaml
name: give_high_five
frame_elements:
  - name: give_object
    is_core: True
  - name: give_indirect_object
    is_core: True
  - name: give_det
    is_core: False
  - name: give_modifier
    is_core: False
actions:
  - step1: raise right hand
  - step2*: verify {give_indirect_object} ready to end high five
  - step3: wait for confirmation
  - step4: home all
```

Fig. 5: CSF for command `give_high_five`. The CSF actions use argument substitution (orange box) and composition of other recognizable commands such as "raise right hand," "wait for confirmation," and "home all" (blue box).

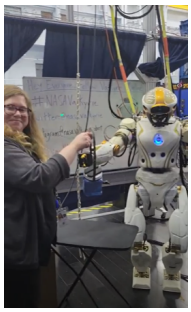
A. Verbs with Multiple Meanings

A single verb can have many different meanings depending on the context. CSFs are able to unambiguously determine what frame corresponds to the command by using optional frame elements and argument substitution. This allows the CSF pipeline to determine which verb meaning is being commanded based on the direct and indirect objects in the command. For example, Figure 4 shows the lexical unit for "give" with several optional frame elements. The CSF for "give" has two children, `give_high_five` (Figure 5) and `give_object_to`. Each child requires substitution of different optional frame elements. Based on the parsing of the command, the CSF pipeline identifies the corresponding child command.

Using the CSF pipeline, Valkyrie is able to understand and

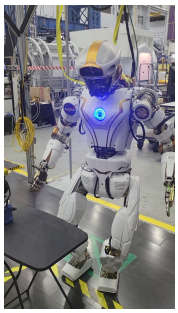


(a) "Give me a high five."

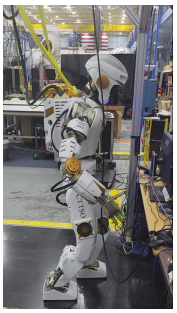


(b) "Give Emily the disruptor."

Fig. 6: CSFs allow robots to understand that the same verb (in this case, "give") involves different actions depending on the objects being acted on.



(a) "Go to Emily's desk."



(b) "Go to Steven's desk."

Fig. 7: CSFs allow robots to scalably understand command variations . A single CSF (`go_to_desk`) represents all desk destinations by using argument substitution.

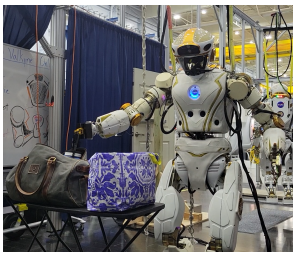
execute the commands "give me a high five" and "give Emily the disruptor," as seen in Figure 6. Though both commands use the same verb "give," the robot understands that the affordances required to execute these commands depend on the direct objects ("high five" and "disruptor" respectively). We see that the CSFs allow the robot to accurately comprehend the multiple meanings associated with these verbs and execute the commanded tasks.

B. Command Variations

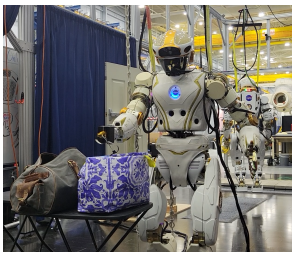
CSFs can scalably instantiate command variations due to argument substitution, as seen in Figure 7. Commands "go to Emily's desk" and "go to Steven's desk" only differ in terms of the final destination. Previous implementations of semantic frames would require separate frames for each office the robot would need to navigate to. Because of argument substitution, CSFs can represent all variations using a single frame. Both commands use a single CSF `go_to_desk` and use argument substitution to handle variations in desk destination. CSFs allow the robot to comprehend and execute command variations in a scalable way.

C. Compound Nouns

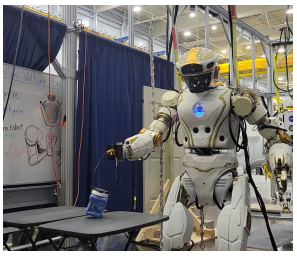
Previous implementations of semantic frames cannot effectively instantiate frames involving compound nouns. This means that any distinctions between nouns such as "grey bag" or "white bag" cannot be understood or acted on appropriately. Due to optional frame elements, head dependency relations, and argument substitution, CSFs can comprehend compound nouns and act on these objects accordingly, as seen in Figure 8. Because the robot comprehends compound nouns, it correctly differentiates between similar object types and executes the appropriate affordances with respect to those objects. Furthermore, these experiments demonstrate recognition of command variations, as all of these experiments use the same `put_object_in_on` CSF.



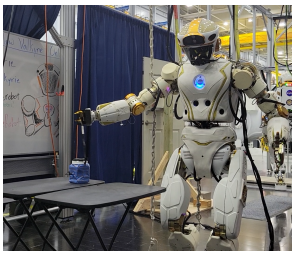
(a) "Put the disruptor in the grey bag."



(b) "Put the disruptor in the white bag."



(c) "Put the disruptor on the left table."



(d) "Put the disruptor on the right table."

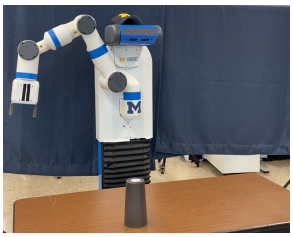
Fig. 8: CSFs allow robots to understand compound nouns and differentiate between multiple similar objects (such as grey bag, white bag, and bag).

D. Command Ambiguity

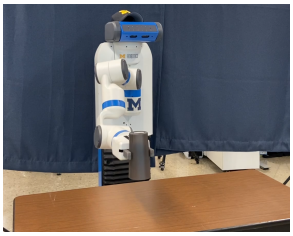
Our CSF pipeline works on multiple robots that execute the grounded control primitives in different ways. Figure 9 shows the Fetch robot executing the command "move that to the left." Since CSFs do not restrict the form of robot control primitives, commands can be executed on multiple robots. The CSF pipeline can also make sense of command ambiguity, as demonstrated by the robot's ability to act on "*that*" object in the scene. Since the only object present in front of the robot is the cup, the robot understands that the only possible grounding for "that" is the cup. CSFs can effectively handle some ambiguity in language and can be executed my multiple robots.

E. Multiple Modes of Interaction

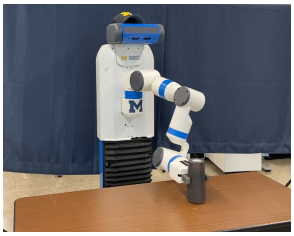
Finally, we wanted to explore how grounded commands could improve ease of interaction with robots through different interaction platforms. NASA's Johnson Space Center has developed a virtual reality (VR) interface for commanding the Valkyrie robot [18] using a VR headset and controllers.



(a) Operator: "Move that to the left."



(b) Fetch picks up the cup.



(c) Fetch places the cup to its left.

Fig. 9: CSFs allow multiple robots to ground commands in action. The robot can also understand some ambiguity in the command, and understands that "move *that* to the left" can only refer to the cup.

Fig. 10: CSF pipeline commanding Valkyrie in virtual reality. Valkyrie executed commands such as "report listening status," "set waypoint," "plan and execute to waypoint," and "give Lewis a high five," pictured above.

The VR interface makes use of voice commands since language is a natural part of human communication. The CSF pipeline significantly expands the voice commands Valkyrie can recognize and is a natural extension to the interactions already supported in VR. Figure 10 shows the robot executing the task "give Lewis a high five." Because the VR headset can provide spoken feedback to the operator, we found that the CSFs provided much more intuitive and conversational interactions with the robot. Operators could command the robot, the robot would execute and wait for feedback, and the operators could prompt the robot to continue or abort the task. Our experiments in different interaction platforms–specifically virtual reality–demonstrate the power of CSFs to improve ease of interaction between robots and humans.

F. Command Processing

To evaluate the safety and responsiveness of our CSF pipeline, we recorded the processing time for each CSF. Due to the added capabilities of CSFs–specifically argument substitution and recursive definitions of frame actions–we need to ensure that commands can be processed quickly, especially if command execution needs to be interrupted for safety purposes.

Task Type	Total Commands	Mean Time (s)	SD Time (s)
Multiple Verb Meanings	24	0.309	0.189
Command Variations	5	0.365	0.179
Compound Nouns	76	0.268	0.201
VR Commands	32	0.324	0.163
All Tasks	**137**	**0.292**	**0.192**

TABLE I: Mean and standard deviation (SD) of processing times for CSF commands. Processing times are reported for each task type as well as aggregate data for all CSF commands.

Table I shows the mean and standard deviation of processing times for each task type as well as aggregate data across all experiments. Note that several trials were performed for each task to verify that commands were being grounded accurately. Some trials were repeated to improve the human-in-the-loop object registration required for robot execution of object. Overall, we see that CSFs can be processed quickly, ensuring responsiveness and safety of robots listening for CSF commands.

V. Discussion and Conclusion

The robot's ability to successfully perform a variety of tasks from verbal commands demonstrates the power of our proposed *composable semantic frames* as a middle-ground between language and robot action. The CSFs greatly improve upon the capabilities of past semantic frame implementations by allowing robots to recognize a wider variety of commands in a scalable manner. Since CSFs are grounded in robot control primitives, we demonstrate that the robot can not only understand spoken commands, but physically execute these commands. By composing semantic frames together to create high-level commands, we ensure that robots can execute tasks from commands that abstract individual actions from the user.

Future work includes expanding the capabilities of semantic frames to incorporate more ambiguous language. Our experiments require that the robot interacts with known, labelled, and registered objects. Future work would also involve testing the effectiveness of CSFs with autonomous segmentation and registration. Overall, our work in *composable semantic frames* demonstrates the power of commanding robots through actions grounded in robot control primitives. Our *composable semantic frames* allow users to intuitively interact with robots in a variety of tasks.

Acknowledgments

This work was supported in part by NASA Space Technology Graduate Research Opportunity (NSTGRO) grant 80NSSC20K1200. The authors would like to thank the members of the NASA Dexterous Robotics Team. Special thanks to Steven Jens Jorgensen, Misha Savchenko, Mark Paterson, Ian Chase, Lewis Hill, and Mina Kian for their mentorship, input, and robot ops support.

REFERENCES

[1] J. S. Albus, A. J. Barbera, and R. N. Nagel, "Theory and Practice of Hierarchical Control," National Bureau of Standards, 1980.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A Survey of Robot Learning from Demonstration," *Robotics and Autonomous Systems*, 2009.

[3] R. Atienza and A. Zelinsky, "Active Gaze Tracking for Human-Robot Interaction," *IEEE International Conference on Multimodal Interfaces*, 2002.

[4] D. H. Ballard, "Task Frames in Robot Manipulation," pp. 16-22, 1984.

[5] V. Bruce, "What the Human Face Tells the Human Mind: Some Challenges for the Robot-Human Interface," *Advanced Robotics*, 1993.

[6] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek, "Sequential Composition of Dynamically Dexterous Robot Behaviors," *International Journal of Robotics Research*, 1999.

[7] S. Chernova, J. Orkin, and C. Breazeal, "Crowdsourcing HRI through Online Multiplayer Games," *AAAI Fall Symposium Series*, 2010.

[8] S. Chitta, I. Sucan, and S. Cousins, "MoveIt! ROS Topics," *IEEE Robotics and Automation Magazine*, 2012.

[9] B. R. Cowan, N. Pantidi, D. Coyle, K. Morrissey, P. Clarke, S. Al-Shehri, D. Earley, and N. Bandeira, ""What Can I Help You With?": Infrequent Users' Experiences of Intelligent Personal Assistants," *International Conference on Human-Computer Interaction with Mobile Devices and Services*, 2017.

[10] J. Dzifcak, M. Scheutz, C. Baral, and P. Schermerhorn, "What to Do and How to Do It: Translating Natural Language Directives Into Temporal and Dynamic Logic Representation for Goal Management and Action Execution," *IEEE International Conference on Robotics and Automation*, 2009.

[11] J. J. Gibson, "The Theory of Affordances," pp. 67-82, 1977.

[12] M. Hagenow, M. Zinn, T. Fong, E. Laske, and K. Hambuchen, "Affordance Template Registration via Human-in-the-Loop Corrections," *arXiv prepring arXiv:2109.13649*, 2021.

[13] S. Hart and R. Grupen, "Natural Task Decomposition with Intrinsic Potential Fields," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 2507–2512, 2007.

[14] S. Hart, S. Dinh, and K. Hambuchen, "The Affordance Template ROS Package for Robot Task Programming," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6227-6234, 2015.

[15] R. Hartley and F. Pipitone, "Experiments with the Subsumption Architecture," *IEEE International Conference on Robotics and Automation (ICRA)*, 1991.

[16] K. M. Hermann, D. Das, J. Weston, and K. Ganchev, "Semantic Frame identification with Distributed Word Representations," *Association for Computational Linguistics*, 2014.

[17] S. J. Jorgensen, M. W. Lanighan, S. S. Bertrand, A. Watson, J. S. Altemus, R. S. Askew, L. Bridgwater, B. Domingue, C. Kendrick, J. Lee, M. Paterson, J. Sanchez, P. Beeson, S. Gee, S. Hart, A. H. Quispe, R. Griffin, I. Lee, S. McCrory, L. Sentis, J. Pratt, and J. S. Mehling, "Deploying the NASA Valkyrie Humanoid for IED Response: An Initial Approach and Evaluation Summary," *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2019.

[18] S. J. Jorgensen, M. Wonsick, M. Paterson, A. Watson, I. Chase, and J. S. Mehling, "Cockpit Interface for Locomotion and Manipulation Control of the NASA Valkyrie Humanoid in Virtual Reality (VR)," NASA Technical Reports Server, 2022. [Online]. Available: `https://ntrs.nasa.gov/citations/20220007587`

[19] D. Klein and C. D. Manning, "Accurate Unlexicalized Parsing," *Association for Computational Linguistics*, 2003.

[20] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward Understanding Natural Language Directions," *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010.

[21] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot Learning from Demonstration by Constructing Skill Trees," *International Journal of Robotics Research*, 2012.

[22] E. Luger and A. Sellen, ""Like Having a Really Bad PA": The Gulf between User Expectation and Experience of Conversational Agents," *CHI Conference on Human Factors in Computing Systems*, 2016.

[23] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the Talk: Connecting Language, Knowledge, and Action in Route Instructions," *AAAI*, 2006.

[24] C. Matuszek, D. Fox, and K. Koscher, "Following Directions Using Statistical Machine Translation," *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010.

[25] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to Parse Natural Language Commands to a Robot Control System," *Experimental Robotics*, 2013.

[26] C. Matuszek, L. Bo, L. Zettlemoyer, and D. Fox, "Learning from Unscripted Deictic Gesture and Language for Human-Robot Interactions," *AAAI Conference on Artificial Intelligence*, 2014.

[27] J. Mu and A. Sarkar, "Do We Need Natural Language? Exploring Restricted Language Interfaces for Complex Domains," *CHI Conference on Human Factors in Computing Systems*, 2019.

[28] S. Niekum, S. Osentoski, G. Konidaris, and A. G. Barto, "Learning and Generalization of Complex Tasks from Unstructured Demonstrations," *IEEE International Conference on Intelligent Robots and Systems*, 2012.

[29] S. Niekum, S. Chitta, A. G. Barto, B. Narthi, and S. Osentoski, "Incremental Semantically Grounded Learning from Demonstration," *Robotics: Science and Systems (RSS)*, 2013.

[30] O. Palinko, F. Rea, G. Sandini, and A. Sciutti, "Eye Gaze Tracking for a Humanoid Robot," *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2015.

[31] A. Pettinger, C. Elliot, P. Fan, and M. Pryor, "Reducing the Teleoperator's Cognitive Burden for Complex Contact Tasks using Affordance Primitives," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[32] A. Pettinger, F. Alambeigi, and M. Pryor, "A Versatile Affordance Modeling Framework using Screw Primitives to Increase Autonomy During Manipulation Contact Tasks," *IEEE Robotics and Automation Letters*, 2022.

[33] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Nullspace Composition of Control Laws for Grasping," *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[34] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Manipulation Gaits: Sequences of Grasp Control Tasks," *IEEE International Conference on Robotics and Automation (ICRA)* vol. 1, pp. 801–806, 2004.

[35] R. Platt Jr, A. H. Fagg, and R. A. Grupen, "Null-Space Grasp Control: Theory and Experiments," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 282–295, 2010.

[36] M. Porcheron, J. E. Fischer, S. Reeves, and S. Sharples, "Voice Interfaces in Everyday Life," *CHI Conference on Human Factors in Computing Systems*, 2018.

[37] J. Pratt, P. Neuhaus, D. Stephen, S. Bertrand, D. Calvert, S. McCrory, G. Robert, G. Wiedebach, I. Lee, D. Duran, and J. Carff, "IHMC Open Robotics Software," Institute for Human and Machine Cognition (IHMC), 2021. [Online]. Available: `https://github.com/ihmcrobotics/ihmc-open-robotics-software`

[38] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, and R. Berka, "Valkyrie: NASA's First Bipedal Humanoid Robot," *Journal of Field Robotics*, 2015.

[39] K. Rohanimanesh, R. Platt, S. Mahadevan, and R. Grupen, "Coarticulation in Markov Decision Processes," *Advances in Neural Information Processing Systems*, 2004.

[40] J. Ruppenhofer, M. Ellsworth, M. Schwarzer-Petruck, C. R. Johnson, and J. Scheffczyk, "FrameNet II: Extended Theory and Practice," *International Computer Science Institute*, 2016.

[41] E. Sheetz, X. Chen, Z. Zeng, K. Zheng, Q. Shi, and O. C. Jenkins, "Composable Causality in Semantic Robot Programming," *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.

[42] S. Tellex, N. Gopalan, H. Kress-Gazit, C. Matuszek, "Robots that Use Language," *Annual Review of Control, Robotics, and Autonomous Systems*, 2020.

[43] B. J. Thomas and O. C. Jenkins, "RoboFrameNet: Verb-Centric Semantics for Actions in Robot Middleware," *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[44] T. Tojo, Y. Matsusaka, T. Ishii, and T. Kobayashi, "A Conversational Robot Utilizing Facial and Body Expressions," *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2000.

[45] D. Trandabat and D. Cristea, "Natural Language Processing Using Semantic Frames," Ph.D. dissertation, University "Alexandru Ioan Cuza" of Iaşi, Romania, 2010.

[46] Y. Y. Wang, L. Deng, and A. Acero, "Semantic Frame-Based Spoken Language Understanding," *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, 2011.

[47] T. Winograd, "Procedures as a Representation for Data in a Computer Program for Understanding Natural Language," MIT Technical Report, 1971.

[48] T. Winograd, "SHRDLU: A System for Dialog," 1972.
[49] P. Zech, S. Haller, S. R. Lakani, B. Ridge, E. Ugur, and J. Piater, "Computational Models of Affordance in Robotics: A Taxonomy and Systematic Classification," *Adaptive Behavior*, 2017.