Cooking Preparation Knowledge using the Functional Object-Oriented Network

David A. Paulius[*] and **Yu Sun**
University of South Florida, Tampa, Florida, USA
{davidpaulius, yusun}@mail.usf.edu

Abstract

We developed the *functional object-oriented network* (FOON) as a graphical knowledge representation for manipulations that can be performed by domestic robots. This bipartite representation focuses on household manipulation activities; for now, our focus is on the domain of cooking preparation activities. A robot can use FOON for solving manipulation problems through a knowledge retrieval procedure. This retrieval procedure aims to determine the necessary steps (as a task tree) to solve a given problem, i.e. to prepare a specific dish or food item within a specific state, given a list of ingredients or utensils that are available for the robot to use. In our most recent work, we modified FOON to account for weights that reflect the difficulty or likelihood of a robot successfully performing the action(s) in a universal FOON. However, certain manipulations may be too difficult for it to perform on its own based on its own physical limitations. To make it easier for the robot, a human can assist to the minimal extent needed to perform the activity to completion by identifying those actions with low success rates for the human to do. In our experiments, it is shown that tasks can be executed successfully with the aid of the assistant.

1 Introduction

In the ideal world, we want to build robots that are capable of performing all tasks for those individuals who are unable to complete the task themselves due to physical limitations. To efficiently program robots that can perform such tasks, a knowledge representation can be created to capture several modalities of information for task planning, specifically: 1) knowing what actions produce specific effects on objects and 2) understanding what objects and states are necessary for producing other objects. Previously in [1; 2], we introduced the functional object-oriented network (FOON), which is a graphical knowledge representation for domestic robots. Although this representation can be generalized to other domains, we specifically focus on cooking activities. A FOON typically combines knowledge from human demonstrations. In [1], we demonstrated how a robot can use a FOON for performing manipulation tasks through task tree retrieval. In [2], our objective was to investigate how we can generalize knowledge in FOON for unseen cases of objects, since a robot's operation is limited by the knowledge contained with a FOON. That is to say: if we do not see a specific use of an object in demonstrations, a FOON will not have that information, and the robot will not be able to perform those actions as a result. To manage this limitation, we explored two ways of expanding knowledge in FOON to other objects similar to what is present in FOON. Similarity of objects considers the object's functionality and meaning.

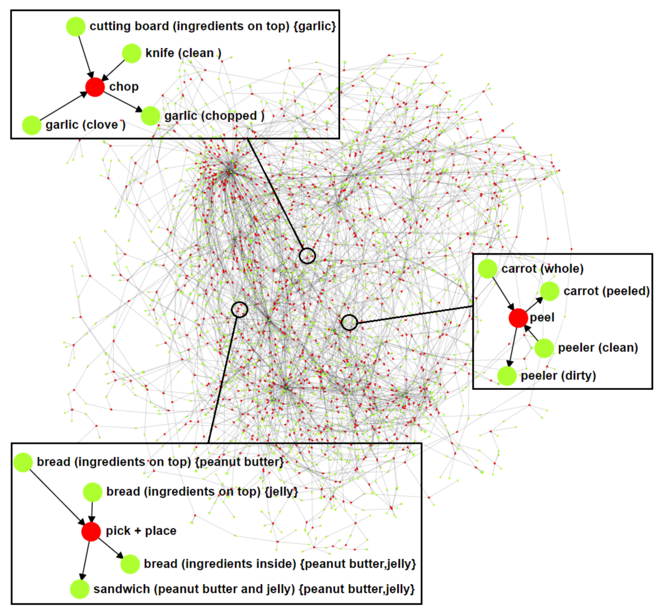

Figure 1: Illustration of a universal FOON combining knowledge from 65 instructional videos. Three examples of functional units are shown, each describing an atomic manipulation.

However, prior to these works, we did not evaluate the performance of FOON using real robotic systems, as to perfectly design such a robot is an exceptionally daunting task. For one, the variability of the environment in which robots work is very dynamic and is likely to feature objects of different

[*]Contact Author

shapes and sizes, while also varying in the position of objects. Secondly, robot motions are not guaranteed to be 100% reliable and can fail occasionally. A robot's capability to perform human-like manipulations heavily depends on how it is made; features such as the type of end-effector it has (e.g. what type of gripper it uses, how many fingers it has, etc.), the number of degrees of freedom and joints it has for its appendages, and the freedom (or lack of) to navigate the environment in search for the items it requires for problem solving. Through the addition of weights, we are better able to capture uncertainty of performing such actions. Therefore, we now introduce success rates as weights in FOON to identify a task sequence that is best suited to the current situation. Furthermore, weights would also be set for robots with different architectures to reflect their ability to perform certain manipulations.

We can leverage the available resources or capabilities of the robot by introducing collaboration with a human assistant. Human-robot collaboration (HRC) is an ongoing research area that focuses on robot and human interaction [3; 4; 5] to solve a common goal and has been extensively studied for areas such as social interaction [6; 7; 8], coordinated tasks [9; 10] rehabilitation [11; 12], and care for the elderly or disabled [13; 14]. The human acts as an assistant to the robot, who has the knowledge needed to perform the tasks as FOON; given a goal, the robot determines the best course of action through task tree retrieval and works with the human to solve the posed problem. This not only makes things easier for the human in reducing the complexity of solving the task (when compared to doing it on his/her own – especially if impaired), but it also improves the robot's chances of succeeding in task tree execution.

2 Functional Object-Oriented Network

The *functional object-oriented network* (FOON) represents manipulations as seen in cooking activities (with possible extension to other domains) by capturing the objects and the activity's motions within a graphical structure. This representation is motivated by the theory of affordance [15], wherein it describes the underlying uses and/or effects of objects afforded to the robot, which are innately depicted though edges connecting objects to actions. To suitably represent activities, a FOON contains two types of nodes: object nodes and motion nodes. Object nodes symbolize any object that is manipulated passively or actively within activities in FOON, while motion nodes symbolize the type of manipulation that connected object nodes participate in for a given action. These motion nodes can be actions commonly performed in cooking such as pouring, cutting, or stirring. In Figure 2 describing the task of stirring a cup of tea using a spoon, the active object would be a *spoon* object that acts upon a passive object *tea cup*, which contains the ingredients *tea* and *sugar*. The stirring manipulation is represented by a motion node of label *stir*. This action results in a change in the contents' state within the *tea cup*, producing *tea*. The composition of object nodes in this manner create a *functional unit*. A functional unit describes the change in the states of objects used in a manipulation action before and after execution; it is important to consider object states to identify when an action has

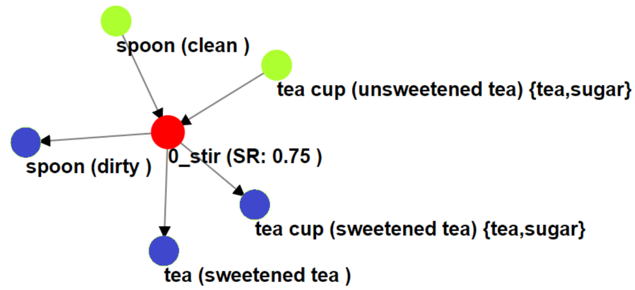

Figure 2: A basic functional unit with two input nodes (in green) and three output nodes (in indigo) connected by an intermediary single motion node (in red) describing the action of stirring tea with sugar to sweeten it. A certain robot has a 75% chance of success in performing this action as indicated by the success rate.

been completed [16]. Each functional unit contains a single motion node describing the action. Typically, an activity is represented by a series of functional units that are connected through common object nodes. *Input* object nodes describe the required state(s) of objects needed to perform the task, and *output* object nodes describe the outcome of performing the action on those input object nodes. Some actions do not necessarily cause a change in all input objects' states, and so there may be instances where there are fewer output object nodes than inputs.

3 Adding Weights to FOON

Up to this point, we have yet to evaluate the innate capability of a robot in task planning with FOON. All motions were previously considered to have equal weights in FOON, implying that all motions can be executed by any robot without difficulty. In other words, the assumption was that any robot should be able to perform the manipulations as well as any other robot or even humans. However, this does not match reality since robots come in different shapes and sizes, meaning that they may not all precisely perform the same manipulations equally. As much as we would like any robot to perform any motion, it is difficult to achieve human-like dexterity as observed in demonstrations. For these reasons, we introduce weights into FOON to reflect how challenging a manipulation is to perform for a given robot. The weights reflect the robot's *success rate* of performing actions. Success rate weights (as percentages) are assigned to each functional unit's motion node and are based not only on the manipulation type, but also on the objects contained within the functional unit. To guarantee that a robot can perform such motions, weights can be used as heuristics for knowledge retrieval; even though several robots will be equipped with the same universal FOON (meaning they will all have knowledge of the same sequence of actions for all activities), different weights will be assigned to them based on: 1) physical capabilities of the robot, 2) past experiences and ability in performing the action, and 3) the tools or objects that the robot needs to manipulate. This can ultimately result in potentially very different task trees. Hence, it is important to note that these weights must first be defined for each type of robot.

4 Task Tree Retrieval

Given a problem defined as a goal, a robot can perform knowledge retrieval to obtain a subgraph that contains functional units outlining the steps it needs to follow to solve it. The searching procedure is driven by a list of items available to the robot in its environment (i.e. the kitchen), which is used to determine the functional units that can be executed in the given scenario due to the availability of inputs to these units. This algorithm is motivated by typical graph-based depth-first search (DFS) and breadth-first search (BFS): starting from the goal node, we search for candidate functional units in a depth-wise manner, while for each candidate, we search among its input nodes in a breadth-wise manner to determine whether or not they are available in our kitchen. A subgraph that is obtained from knowledge retrieval is referred to as a *task tree*. A task tree differs from a regular subgraph, as it will not necessarily reflect the complete procedure from a single human demonstration. Rather, it will leverage the knowledge obtained from multiple sources to produce a novel task sequence. With weights, we can derive an optimal task tree with the highest possible overall rate of success.

5 Human-Assisted Manipulations

With a revised retrieval algorithm using weights, we can obtain optimal, novel task trees from FOON. However, certain trees must be eliminated due to the robot's inability to accomplish the required manipulations described in those task trees; even the execution of the best task tree can still result in failure. To remedy this, we can involve a human assistant in manipulation problems. The human assistant can identify the number of steps out of the total number of steps (as functional units) in a task tree that he/she is able to perform with the robot to cooperatively solve the problem. As input to the task tree retrieval, the assistant can indicate the number of steps as a value M, which cannot exceed the length of the task tree N minus 1 step (as an involvement where N is equal to M means that the human will perform the entire task with no robot assistance in its manipulations). If M is 0, there will be no human involvement in achieving his/her desired goal but at the chance of not being able to perform the entirety of the activity's manipulations. The output of the algorithm can be modified to produce the best task tree based on different values of M, as certain trees may be better to execute due to a higher likelihood of success (assuming that the human assistant can perform the manipulation flawlessly).

In these human-assisted steps, the success rate changes to 100% by default, unless the human assistant's ability to perform the action is impaired in any way (based on the person's health/condition, mood, age, and other factors). Once the human identifies M, the algorithm is run to find the suitable task tree for the given amount of participation. If the human user does not provide a value for M, the optimal value of M can also be determined by the robot; this is done by finding the tree whose success rate at some value of M does not significantly improve over the prior value $M-1$. The robot may still fail its manipulations, but it will not have to worry about performing those that it does not have programmed in its primitives. The M steps would then be modified to indicate that a human assistant should execute those steps when the robot executes the task tree. In the task tree execution phase, the robot will perform its delegated actions, and the remaining M steps are given as instructions to the assistant on how to perform actions on the robot's behalf.

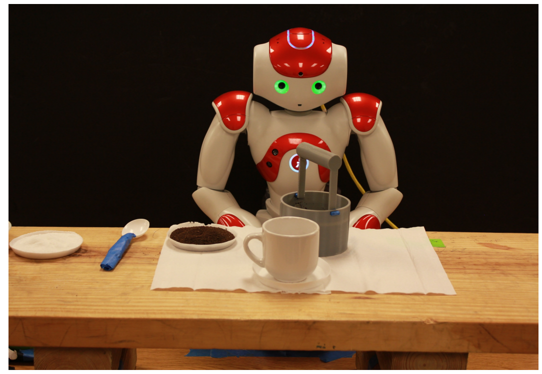

Figure 3: Our experimental setup for demonstrating the use of a weighted FOON with the NAO robot. NAO is performing the tea-making task. Its motor primitives are taught by demonstration.

6 Experimental Results

In our experiments, we show that we can significantly improve a robot's performance with FOON through HRC within the planning and execution phases. We demonstrate that a robot can acquire the ideal task tree for execution, delegate commands to the human assistant, and successfully obtain the goal product for varying levels of involvement. We use Aldebaran's NAO robot to execute manipulations needed to complete the tasks of making tea, mashed potatoes, and ramen noodles. Different variations of preparing each dish were merged together into a single, universal FOON, which was then provided to the algorithm to identify candidate trees for preparing these items. Because the NAO robot itself is very small, its physical capabilities are limited to using smaller versions of items, and furthermore, certain manipulations are very difficult to replicate. Under these circumstances, the robot greatly benefits from human participation in task tree execution. Certain parts of the tasks, such as heating containers to obtain hot water, cannot be left to the robot to perform; for such motions, their nodes were assigned a success rate of 1% to reflect how impossible they are for the robot to do on its own. However, for those motions executable by the robot, we assign higher rates based on our confidence in the robot performing the programmed motion primitives. The task trees obtained through the weighted retrieval approach, along with demonstrations of the robot performing each of these trees, can be viewed within the supplementary material provided here[1]. Without human involvement, the NAO robot attempts to execute the task tree but ends up failing once it encounters the motion it does not know how to perform; however, with human involvement, the robot can finish all of the tasks and produce the final product. In some cases, we did observe that

[1]Video demonstrations can be found at the following link: http://www.foonets.com/human-robot.html

the motion primitives of the robot can fail, rendering the entire sequence as a failure. As future work, we would like to include sensors or behaviour that allow the robot to determine when it has failed a particular action and to determine what it needs to do to recover from the failed action. Even without its own notion of failure, the robot can supplement this through human interaction by communicating with the assistant to determine whether it should perform the action again.

7 Conclusion and Future Work

To summarize, we introduce human-robot collaborative task planning using the graphical knowledge representation known as the functional object-oriented network (FOON). Previously, we have shown that a FOON can be used for obtaining the steps needed to achieve a given goal through task tree retrieval, and that these task trees can be novel and flexible to the given scenario. We introduced a modified retrieval procedure that takes the robot's physical capabilities into account for task planning through the integration of robot success rates. These success rates determine whether the robot can successfully execute the task tree on its own or whether it needs some assistance. To improve the performance of the robot in execution, a human assistant can perform the difficult motions for the robot. We discussed the modified task tree retrieval to acquire the ideal task tree based on the amount of involvement that can be given by the human assistant, and in our experiments, We show that we can obtain suitable task trees that leverage both the robot's and human's capabilities without requiring too much effort from the human assistant. In the future, we would like to explore task tree execution for manipulations done by multiple robots, thereby creating a multi-robot collaborative effort to solving problems. This would require identifying difficulties in performing various types of manipulations so that an optimal task tree can be produced that maximizes the performance of the participating robots. We will demonstrate the interaction between two or more robots, even of different types, to illustrate that FOON can be used for task tree retrieval and execution for any given robot and that plans can be made to synchronize efforts made by the robots to solve the given problem. In addition, we would like to focus more on the robot's recovery from failure to perform a specific action in a FOON task tree since this is also important to successfully execute its given task.

References

[1] D. Paulius, Y. Huang, R. Milton, W. D. Buchanan, J. Sam, and Y. Sun. Functional Object-oriented network for manipulation learning. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2655–2662. IEEE, 2016.

[2] David Paulius, Ahmad B Jelodar, and Yu Sun. Functional Object-Oriented Network: Construction & Expansion. pages 5935–5941, 2018.

[3] Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and autonomous systems*, 42(3-4):143–166, 2003.

[4] Michael A Goodrich, Alan C Schultz, et al. Human–robot interaction: a survey. *Foundations and Trends® in Human–Computer Interaction*, 1(3):203–275, 2008.

[5] Balasubramaniyan Chandrasekaran and James M Conrad. Human-robot collaboration: A survey. In *SoutheastCon 2015*, pages 1–8. IEEE, 2015.

[6] Michael Zinn, Oussama Khatib, Bernard Roth, and J Kenneth Salisbury. Playing it safe [human-friendly robots]. *IEEE Robotics & Automation Magazine*, 11(2):12–21, 2004.

[7] A. Edsinger and C. C. Kemp. Human-robot interaction for cooperative manipulation: Handing objects to one another. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pages 1167–1172. IEEE, 2007.

[8] Paul E Rybski, Jeremy Stolarz, Kevin Yoon, and Manuela Veloso. Using dialog and human observations to dictate tasks to a learning robot assistant. *Intelligent Service Robotics*, 1(2):159–167, 2008.

[9] Bilge Mutlu and Jodi Forlizzi. Robots in organizations: the role of workflow, social, and environmental factors in human-robot interaction. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 287–294. ACM, 2008.

[10] Matthew C Gombolay, Cindy Huang, and Julie A Shah. Coordination of human-robot teaming with human task preferences. In *AAAI Fall Symposium Series on AI-HRI*, volume 11, page 2015, 2015.

[11] Ben Robins, Paul Dickerson, Penny Stribling, and Kerstin Dautenhahn. Robot-mediated joint attention in children with autism: A case study in robot-human interaction. *Interaction studies*, 5(2):161–198, 2004.

[12] Aude Billard, Ben Robins, Jacqueline Nadel, and Kerstin Dautenhahn. Building robota, a mini-humanoid robot for the rehabilitation of children with autism. *Assistive Technology*, 19(1):37–49, 2007.

[13] Henry Kautz, Larry Arnstein, Gaetano Borriello, Oren Etzioni, and Dieter Fox. An overview of the assisted cognition project. In *AAAI-2002 Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*, number 2002, page 6065, 2002.

[14] Kerstin Dautenhahn, Michael Walters, Sarah Woods, Kheng Lee Koay, Chrystopher L Nehaniv, A Sisbot, Rachid Alami, and Thierry Siméon. How may i serve you?: a robot companion approaching a seated person in a helping context. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 172–179. ACM, 2006.

[15] J.J. Gibson. The theory of affordances. In R. Shaw and J. Bransford, editors, *Perceiving, Acting and Knowing*. Hillsdale, NJ: Erlbaum, 1977.

[16] A. B. Jelodar, M. S. Salekin, and Y. Sun. Identifying object states in cooking-related images. *arXiv preprint arXiv:1805.06956*, May 2018.