CoSTAR in Surgery: A Cross-platform User Interface for Surgical Robot Task Specification

Baichuan Jiang, Chris Paxton, Peter Kazanzides, and Gregory D. Hager

***Abstract*— Human-Robot Collaboration (HRC) in surgery has been an emerging field of study in recent years, which aims at incorporating the advantages of both surgeon and machine to improve safety, accuracy and speed. In this work, we propose a cross-platform, open-source framework that would facilitate surgical HRC research. The work is based on CoSTAR, a Behavior-Tree based framework for end-user instruction of industrial robots [1]. To demonstrate its feasibility as a platform for collaborative surgical robot research, we generalized the original system and implemented it on the da Vinci Research Kit (dVRK), while maintaining its full functionality on other robot platforms such as UR5 and KUKA LBR iiwa.**

I. Introduction

Surgical robots have been increasingly adopted in clinical procedures to support surgeons with various tasks, although most currently available systems are under full control of the surgeon. Recent studies have suggested that incorporating further intelligence into the surgical robot will free surgeons from repetitive tasks, reduce large movements of the master manipulator to avoid frequent clutching and readjusting hand position, and achieve better overall precision and accuracy, e.g. [2], [3], [4], [5].

Many of these proposed collaborative schemes aim to take advantage of the reliability that robots offer when performing less critical tasks, while taking advantage of the surgeons' domain knowledge to perform fine manipulation actions. Padoy *et al.* [2] adopted a Hidden Markov Model (HMM) as a way of modeling a surgical procedure in order to automatically alternate between manual and automated subtasks. The Hubot [3] system runs continuously in a loop alternating between fully manual, manual with haptic guidance, and fully automated modes. This kind of "human decide, robot do" scheme was also used in semi-autonomous brain tumor ablation [5].

CoSTAR [1] aims to do something similar for collaborative robots but provides further extensibility than purely ontology-based systems. CoSTAR's core features are a modular architecture that combines planning, perception, and logic via a Behavior Tree-based task specification UI, previously shown on the KUKA LBR iiwa and the Universal Robots UR5. Here, we propose to adapt and expand the system to make it suitable for surgical robotics, where an expert user can construct a Behavior Tree to automate parts of the procedure and adapt it to their specific scenarios. The system setup is shown in Fig. 1.

Authors are with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA

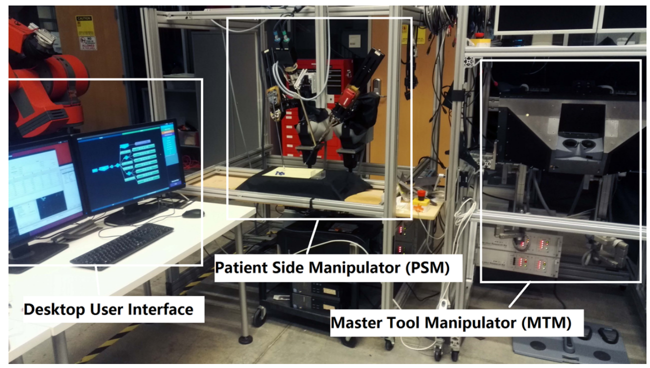

Fig. 1. CoSTAR implementation on dVRK. Users design the task plan using the user interface on the screen to the left, and then use the master console on the right to perform manual portions of the procedure.

II. System Architecture

A modular framework is one key for cross-platform implementation. CoSTAR is composed of a set of modular *Components*, such as `GRIPPER` and `ARM`. Each component C is associated with a set (I, O, s, p, u):

- *Input I and Output O* are represented by ROS topics and they are never explicitly exposed to the end-user.
- *Symbols s* are populated from input as a function of the raw state of the world, which represent objects and positions, such as `LEFT OF` or `PRESSED`.
- *Predicates p* describe the quality of existing symbols or their relationship, which is more formally expressed as a function mapping $p(I, s_0, ..., s_n) \rightarrow [\mathtt{TRUE}, \mathtt{FALSE}]$ that can be used to control workflow.
- *Operations u* are the specific actions that would change the value of stored symbols or the state of the robot.

Another key for modularity is the inheritance within each component. For example, components `PSM` and `DVRK_GRIPPER` for the patient side manipulators are instantiations spawned from the abstract component `ARM` and `GRIPPER` by implementing the required functionalities such as `MOVE` and `CLOSE_GRIPPER`. More detailed description of components can be found in [1].

A. Behavior tree instructor interface

A behavior tree is expressive to represent a great variety of tasks accommodating different scenarios for human-robot collaborative surgery. This is achieved through the linking of internal nodes:

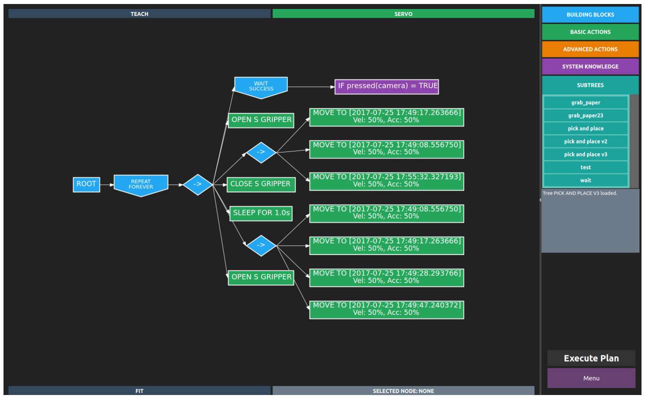

Fig. 2. CoSTAR `INSTRUCTOR` user interface: tabs on the right have different functionalities to build a behavior tree and are able to do subtree manipulation.

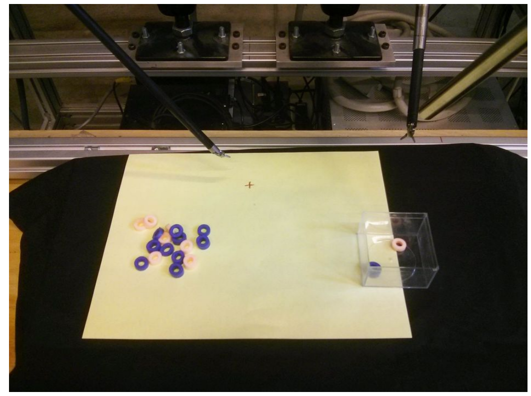

Fig. 3. Debridement experiment setup. PSM1 will pick one debris from the left pile and place it on the red cross, then PSM2 will pick the piece from the red cross and transport it to the tray.

- `SEQUENCE` node: tick each child in order, until each one reports `SUCCESS`. One child's fail will result in the sequence's fail.
- `SELECTION` node: tick each child in order until one returns success.
- `REPEAT N` node: tick children until N successes or failures are reported.
- `RESET N` node: returns the value of child, but resets the child up to N times.

We can combine these structural and operation nodes to create a behavior tree that encapsulates very complex behavior. The process of creation can be greatly simplified using the `INSTRUCTOR` user interface as shown in Fig. 2.

III. Case Study

A simple surgical scenario where the framework will be helpful and easy to use is *debridement*, which involves removing the dead tissue fragments during surgical treatment. We set up the experiment [6] as in Fig. 1 and Fig. 3.

The target in this experiment is to transfer the pile of debridements on the left to the tray on the right. Because it is safer to set the motion scale at a low level during teleoperated robotic surgery, the workspace of the dVRK Patient Side Manipulator (PSM) is limited and requires frequent clutching of the Master Tool Manipulator (MTM) to reorient the slave workspace. Therefore, we propose a two-step procedure: first use PSM1 (left) to manually pick one fragment to a fixed position, then let PSM2 (right) to automatically pick it up and place it in the tray. The second step can be achieved through a learn-from-demonstration (LfD) scheme: first the surgeon demonstrates the pick-and-place procedure while recording waypoints during manipulation. The saved waypoints that follow a safe path can be used to automate the robot during the second step.

The above procedure can be represented by a behavior tree as shown in Fig. 2. The execution order for our user interface is from left to right and from top to bottom. Following the `ROOT` node, a `REPEAT` node is set to repeat the per-fragment action. Then a `SEQUENCE` node follows, representing the automation sequence. The `PREDICATOR` node for pedal press signal is set after the `WAITFORSUCCESS` node, so that the workflow proceeds whenever the pedal is pressed. Finally, operation nodes such as `OPENGRIPPER` and `MOVETOWAYPOINTS` complete the full procedure.

By automating the PSM2 pick-and-place, the surgeon can focus on using PSM1 to place the debris at the fixed location under a safer motion scale. Whenever triggered by a pedal press, PSM2 will repeat the action so that two arms can be operated at the same time, thus greatly increasing the speed.

IV. Conclusion

We described a modified version of the open-source CoSTAR system that allows surgeons to quickly author partially autonomous task plans for the DVRK. This will be of great benefit when multiple robot arms are present, such as a full da Vinci Surgical System, or in a Hand-Assisted Laparoscopic Surgery (HALS) procedure [4] where the surgeon only has one free hand to operate multiple tools..

The modular framework and abstract representation of workflow and environment state (by predicates and symbols) make the proposed system an ideal surgical HRC platform. In the future we will incorporate machine learning-based action detection or vision based surgical robot automation and perception.

References

[1] C. Paxton, A. Hundt, F. Jonathan, K. Guerin, and G. D. Hager, "Costar: Instructing collaborative robots with behavior trees and vision," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, 2017, available as arXiv preprint arXiv:1611.06145.

[2] N. Padoy and G. D. Hager, "Human-machine collaborative surgery using learned models," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5285–5292.

[3] P. Berthet-Rayne, M. Power, H. King, and G.-Z. Yang, "Hubot: A three state human-robot collaborative framework for bimanual surgical tasks based on learned models," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 715–722.

[4] E. Bauzano, B. Estebanez, I. Garcia-Morales, and V. F. Muñoz, "Collaborative human–robot system for hals suture procedures," *IEEE Systems Journal*, vol. 10, no. 3, pp. 957–966, 2016.

[5] D. Hu, Y. Gong, B. Hannaford, and E. J. Seibel, "Semi-autonomous simulated brain tumor ablation with ravenii surgical robot using behavior tree," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3868–3875.

[6] Video available at: https://youtu.be/RqQNLZuuRUE.