A knowledge-based framework for task automation in surgery

Michele Ginesi*, Daniele Meli*, Hirenkumar Nakawala*, Andrea Roberti* and Paolo Fiorini

***Abstract*— Robotic surgery has significantly improved the quality of surgical procedures. In the past, researches have been focused on automating simple surgical actions, however there exists no scalable framework for automation in surgery. In this paper, we present a knowledge-based modular framework for the automation of articulated surgical tasks, for example, with multiple coordinated actions. The framework is consisted of ontology, providing entities for surgical automation and rules for task planning, and "dynamic movement primitives" as adaptive motion planner as to replicate the dexterity of surgeons. To validate our framework, we chose a paradigmatic scenario of a peg-and-ring task, a standard training exercise for novice surgeons which presents many challenges of real surgery, e.g. grasping and transferring. Experiments show the validity of the framework and its adaptability to faulty events. The modular architecture is expected to generalize to different tasks and platforms.**

I. INTRODUCTION

Autonomous surgery involves standard challenges of complex robotic tasks, like the execution of safe and precise human-like movements, real-time understanding and execution of the surgical workflow, reactive decision-making especially during unexpected and faulty events. In the past, automation of complex tasks by an intelligent robotic system has been widely investigated. In the "KnowRob" system [1], a knowledge model consisting of an ontology knowledge base of the task and fine-grained action representation for reasoning on sensory information has been used to drive manipulation and exploration tasks in uncertain environment. While the "Knowrob" relies mostly on the prior knowledge, other researches investigated learning of semantically relevant action sequences for manipulation tasks, e.g. [2]. An alternative approach to full task automation starts from no prior knowledge and tries to infer a task model only from observations with the machine learning techniques. Relevant examples are [3] for autonomous driving using Markov decision process and [4] for end-to-end deep learning of manipulation skills. Learning has the advantage to enrich the prior knowledge of the task, which may not cover all possible situations. [5] gives a perspective on the application of machine learning techniques on surgical data. However, the robustness of the learned model strongly depends on the quality and size of the learning dataset, especially for dynamic and uncertain scenarios. In surgery, a proper dataset cannot be generated, due to the complexity and safety issues of the scenario. For this reason, in our work, we chose a knowledge-based model. A knowledge model encodes expertise of the surgeon, thus it is expected to enhance predictability and interpretability, by providing access to the inferred knowledge, for surgical safety. So far, only elementary surgical tasks, made of just one action (e.g., needle insertion [6]) have been automated in a laboratory based set-up. Recently, the automation of suturing has been implemented [7], however it lacks adaptation and re-planning to failure and unexpected events. This paper provides two main contributions:

- A *modular framework with a hierarchical reasoning* for the automation of a structured surgical task, where multiple actions must be coordinated. At the task level, an ontology encodes the prior knowledge as rules and verifies the pre-conditions for the execution of the main actions. At the lower motion level planning, motion planning is implemented with the framework of dynamic movement primitives (DMPs).
- Replanning to *failure events* in the surgical scenario, both at the task level (triggering conditions in the ontology) and at the motion level planning (exploiting fast re-planning of DMPs with environmental changes).

A two-layer framework with sensory information (vision and kinematics) is used to verify pre-conditions and effects in the ontology rules, as well as to drive the online DMP execution. Ontologies have already been successfully used for monitoring and abstract modeling of the surgical workflow [8], but to our knowledge they have never been applied in full autonomous surgical tasks and systems. DMPs have been chosen because they can replicate the human movement from just one single demonstration, which is essential to encode the dexterity of the surgeon in the autonomous system. Another important feature of our framework is the modularity, that is, the components (ontology, DMPs) are separated from each other. This modular implementation offers more versatility to different hardware architectures and tasks.

The rest of the paper is structured as follows: In section II, we present the planning units of the full framework and show how they co-operate for real-time execution; section III describes the case of study of the peg-ring task, pointing out the relevance of it to real surgery, and how the framework is adapted to automate it; section IV represents the experimental protocol used to validate the framework; in section V, we summarize our contributions and suggest future extensions of this work.

II. MATERIALS AND METHODS

In this section, we outline the two main components of our planning framework: an ontology-based knowledge

Authors are with Department of Computer Science, University of Verona, Via S. Francesco, 22, 37129, Verona, Italy. paolo.fiorini@univr.it

*Authors contributed equally and in the alphabetical order

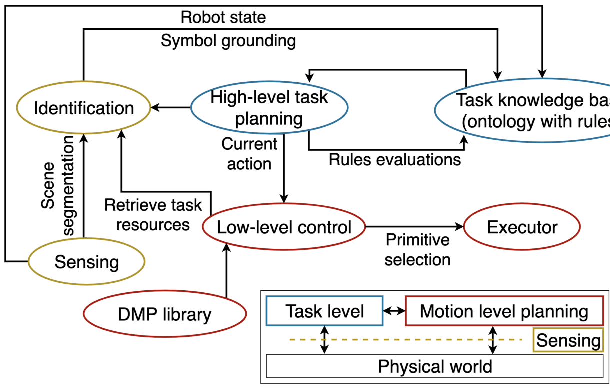

Fig. 1. Proposed framework for autonomous surgery. Low-level control with DMPs (red) and high-level reasoning with ontology (blue) interact to execute the task. The components exchange information through perception modules (yellow).

model which encodes procedural description, as actions and transition conditions, and drives the task-level execution, and DMPs as a motion planner. A block diagram explaining the framework is shown in Fig. 1, highlighting the co-ordination of the components through the perceptual information.

A. Ontology for automation in surgery

An ontology provides a knowledge representation of key concepts related to the domain of discourse with properties, relationships, and constraints [9]. Moving from a description of the surgical task as a set of actions, i.e. elementary operations, and transition rules between them, the ontology specifies the properties of the task, robot capabilities e.g. actions, resources e.g. objects, and rules deriving instance-level axioms for task planning. The ontology offers the opportunity to easily determine the task description by the users, which enhances the interpretability and reliability of the reasoner.

The ontology is developed using a top-down methodology. Considering extendability with Core Ontology for Robotics and Automation (CORA) [10] for complex robotic tasks in surgery, we separated object and process branches based on Suggested Upper Merged Ontology (SUMO) [11] than Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [12] or Basic Formal Ontology [13]. The generic ontology classes, i.e. *State*, *Action*, *Object*, *MathematicalOrcomputationalThing*, *Exit conditions* and *Spatial information*, etc. are shown in Fig. 2. Object branch, i.e. *Object*, is used to present task resources, e.g., a set of needed physical objects, while process branch is used to specify entities representing surgical actions. The logical propositions are specifically defined for the procedural entities, *Action* and *State*. Temporal relations are specified between the entities representing *Action*. Spatial information is represented with the abstract concepts, e.g. *position*, *orientation*, etc., which are defined for *object*. *Action* defines the *Goal*, an externally defined behaviour or state of the system, that robot is trying to accomplish. If the overall *Goal* is to move an object to a particular location, the robot may be told to have a *goal* of reaching a specific interim pose which is derived implicitly by interpreting the sensor data, e.g. vision. *State* represents whether the workflow described by the prior model is in the initial, transition, or final state. The surgical actions and constraints could be easily extended by importing upper ontologies, e.g. Ontology for Surgical Process Modeling (OntoSPM) [8]. However, ontology mapping will be needed in order to import several upper ontologies. To encode workflow description, we use SWRL [14] instead of temporal planning languages like PDDL [15], to do the Ontology Web Language-Description Logic (OWL-DL) based deductive reasoning. As a reasoning engine, we used Pellet [16].

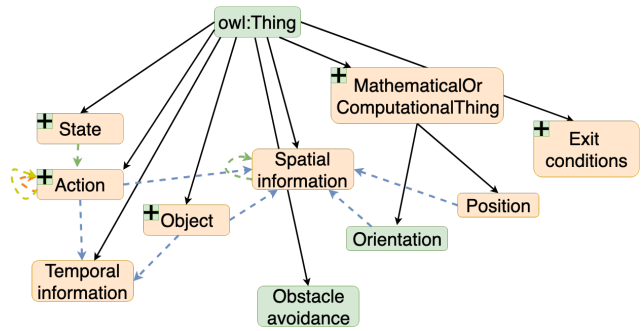

Fig. 2. High-level conceptual structure representing entities of the ontology. The rectangles represent the classes of ontology. While the black arrows show the class hierarchies, other coloured arrows represent relations between the classes. The orange colored rectangles/classes and its subclasses are used in this work.

B. Dynamic Movement Primitives

Each action at the task level is associated with a specific Cartesian trajectory with a semantic meaning. The trajectory is expressed as a Dynamic Movement Primitives (DMPs) [17]. In DMPs, a point-to-point trajectory is generated by solving the following second order Ordinary Differential Equation (ODE) of spring-mass-damp type:

$$\begin{cases} \tau\dot{\mathbf{v}} = \mathbf{K}(\mathbf{g}-\mathbf{x}) - \mathbf{D}\mathbf{v} - \mathbf{K}(\mathbf{g}-\mathbf{x}_0)s + \mathbf{K}\mathbf{f}(s) & \text{(1a)} \\ \tau\dot{\mathbf{x}} = \mathbf{v} & \text{(1b)} \end{cases}$$

in which $\mathbf{x}$ and $\mathbf{v}$ are, respectively, the trajectory and its velocity, $\mathbf{x}_0$ and $\mathbf{g}$ are the starting and goal position, $\mathbf{K}$ and $\mathbf{D}$ are the elastic and damping matrix, and $\mathbf{f}$ is a *forcing* term, which depends on a re-parametrization of the time governed by the so-called *canonical system*

$$\tau\dot{s} = -\alpha s, \qquad \alpha > 0. \tag{2}$$

The matrices $\mathbf{K}$ and $\mathbf{D}$ in (1a) are chosen to be diagonal, in order to separate the evolution of the solution along each dimension, and such that the system is critically damped ($\mathbf{K}_{i\,i} > 0$, $\mathbf{D}_{i\,i} = 2\sqrt{\mathbf{K}_{i\,i}}$). In DMPs, a desired trajectory $(\hat{\mathbf{x}}(t), \hat{\mathbf{v}}(t))$, $t \in [t_0, t_1]$ is used in (1) to compute the desired forcing term $\hat{\mathbf{f}}(s)$, where s is obtained by integrating (2). Then, $\hat{\mathbf{f}}$ is approximated using Gaussian Radial Basis Functions $\{\psi_i(s)\}_{i=0,1,\ldots,N}$:

$$\hat{\mathbf{f}}(s) \approx \frac{\sum_{i=0}^{N} \mathbf{w}_i \psi_i(s)}{\sum_{i=0}^{N} \psi_i(s)}\, s.$$

Once the weights have been computed via least-squares method, equation (1) can be integrated to generate a trajectory which is of similar shape to the learned one, but can have different initial and final positions, as well as different time duration. Moreover, DMPs can handle obstacle avoidance. Indeed, an additional term $\varphi(\mathbf{x})$ can be added to equation (1a) to move the trajectory away from the obstacle. To model solid obstacles, *superquadric potential functions* $U(\mathbf{x})$ [18] are used.

We used DMPs due to the complexity of modeling surgical gestures, which are optimal according to a meaningful cost function. While for the classical robotic tasks, minimum-energy or minimum-time trajectories can be generated using splines, sampling-based or geometric planners for instance [19], optimal surgical gestures rely on the dexterity of the surgeon's hand, which cannot be directly encoded as a cost function. For this reason, a "Learning from Demonstration" approach is preferred.

III. EXPERIMENTAL SCENARIO

A. Task description

The peg-and-ring task is consisted of placing 3 colored rings on the respective pegs as shown in Fig. 3. A finite state machine (FSM) describing the task is derived from [20] and adapted to single manipulator case. Each state of the FSM is associated with one action in the nomenclature of the surgical workflow [21]. An action is a single gesture with a specific semantic meaning. In the peg-and-ring task, three main actions are performed:

- *"Move to start"*: the robot initially moves to a standard configuration at the beginning of the task;
- *"Move and Grasp"*: the robot goes to the next ring in the sequence, then grasps it;
- *"Carry and Leave"*: the robot carries the grasped ring to the corresponding peg, then releases the gripper;

Additionally, a failure condition is raised when the ring falls during the *"Carry and Leave"* state. In this case, a *"Drop"* state commands the opening of the gripper. Afterwards, the *"Move and Grasp"* state is invoked again and the framework re-plans the task from the last ring. Although failure can occur also in *"Move and Grasp"* state (e.g., the robot is not able to grasp the ring), we decided not to add another recovery state, since such an unwanted situation is immediately recognized when the robot starts *"Carry and Leave"*, and re-planning can be triggered.

We have chosen this task because it is a standard training task for novice surgeons [22], and it presents several common issues of real surgery, e.g. object identification, grasping and transfer, re-planning in case of failure or dynamic scene, coordination of simple action sequences in arbitrary order, depending on the initial conditions (initial sequence of rings). Moreover, to the best of our knowledge, this task has only been partially automated in the context of surgery so far [23].

B. Framework implementation

In this section, we represent how a hierarchical reasoning with DMPs and ontologies is implemented for the peg-and-ring task. As illustrated in Fig. 4, the two modules are coordinated through sensors, guaranteeing continuous online adaptation.

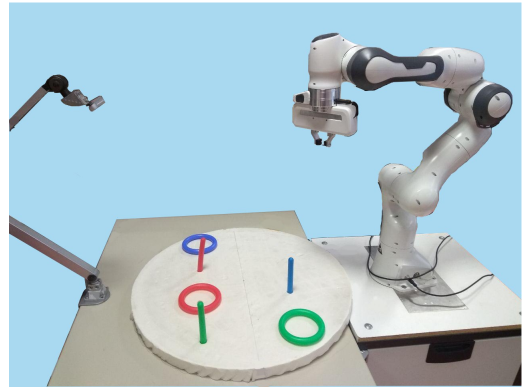

Fig. 3. The setup with the camera and the robot.

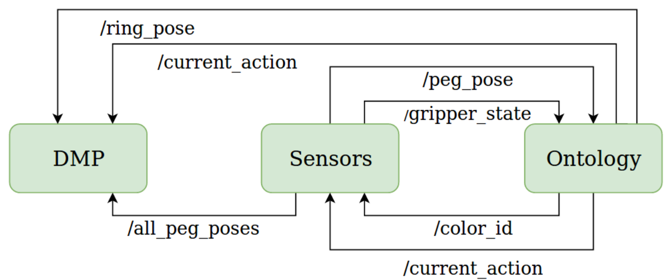

Fig. 4. Structure of the general framework for the peg-and-ring task. The arrows show the ROS topics used for communication between the main modules.

1) High-level reasoning: To effectively handle planning process at the task level, we exploit spatio-temporal reasoning with the ontology. Spatial reasoning is achieved by classifying objects in the environment. Classes and their instances (*pegs* and *rings*) have geometric properties (*spatial pose*) and *colorID* semantic property, which identifies the specific color. The pose of the pegs is computed at the beginning of the task and stored, since they are fixed. A *gripper* class instance is also generated to reason on the status of the robot's end-effector. We follow Allen's temporal interval algebra to do qualitative temporal planning between two actions, e.g. *Move and Grasp* takes place before *Carry and Leave*. To implement logical task planning, the preconditions and effects are implemented in a form of production rules.

The preconditions and effects are:

1) *Rule-1 to start classification in ontology*:
   Preconditions: End-effector at initial configuration and gripper is open
   Effects: Classification of objects, decomposition of state and action sequences, assignment of colorIDs to rings
2) *Rule-2 for "Move and Grasp" action*:
   Preconditions: Gripper is open, end-effector at standard location and colorID
   Effects: The ring is grasped.

3) <u>*Rule-3 for "Carry and Leave" action*</u>:
Preconditions: Status of execution is "Move and Grasp" and gripper is closed
Effects: the ring is on the peg.
4) <u>*Rule-4 for end of task*</u>:
Preconditions: Status of execution at "Carry and Leave", gripper is open and ColorID is the last one
Effects: End of task
5) <u>*Rule-5 for re-planning at failure*</u>:
Preconditions: Status of execution at "Carry and Leave", gripper is open and tracking status is false
Effects: the ring is dropped, execute "Move and Grasp" action

The inferred action at each time step is communicated to the low-level reasoning module, while the colorID information is used to query the vision system.

2) Motion level planning: At the motion planning level, reasoning is needed to efficiently avoid obstacles and successfully reach the target points defined by the high-level reasoning engine and the vision system. DMPs are able to avoid obstacles in real-time, adding repulsive superquadric potential functions surrounding the obstacles, i.e. pegs. However, we cannot set the same radius for the potential in all the states of the task. In fact, during "Carry and Leave", the robot is holding the ring, so the potential must be enlarged to take into account its additional encumbrance. On the other hand, during "Move and Grasp" state, the potential is shrunk to take into account only the minimum size of the end-effector for safety purpose, thus increasing the available workspace of the robot. Moreover, DMPs only generate a trajectory in the Cartesian space for the end-effector's origin frame, considering the robot as a single point. Since this assumption does not hold in our case, we have to take into account the encumbrance of the manipulator.A fast random sampling of possible joint configurations is implemented which determines the given Cartesian pose while avoiding obstacles.The robot's built-in standard tree-based motion planner then interpolates between consecutive configurations. To guarantee singularity-free continuous motion, the time step in the integration of eq. (1a) must be sufficiently small.

C. Perception

For the peg-and-ring task, we used an RGB-D sensor to observe and co-ordinate the execution of the entire task. The "Sensors" component takes point clouds and the rectified color images as inputs. Point clouds are then subsampled and filtered to reduce the processing demand. We use RANSAC [24] to extract the parameters of the plane to segment the point clouds with the objects, i.e. pegs and rings. Using an Euclidean clustering, rings and pegs in the scene are separated in a different set of point clouds. A grasping point is calculated on the clusters of the rings using the model of the ring and considering the minimum distance to the x-axis from the robot base frame. The cluster of pegs, which are pre-calibrated in the robot frame, are used for acquiring the goal location while carrying rings. We use RGB images to track the ring when it is attached to the gripper. The rings are tracked by comparing the distance between pose of the end-effector, which is projected in the camera frame, and the centroid of the ring tracked within a threshold of 70 pixels around the centroid. For segmentation and tracking, we use PCL and OpenCV.

IV. EXPERIMENTAL RESULTS

A. Experimental protocol

We performed several experiments to test our framework:

1) **Execution of standard peg-ring task**, accomplishing final goal of moving all rings into pegs, in different orders and positions.
2) **Handling failure conditions**, at task level (re-planning when drop occurs) and at motion level (dynamic environment).

Fig. 3 shows the real setup used to test the components of our framework, with a 7-DOF industrial manipulator "Panda" from Franka Emika (Germany), a desktop computer with a 3.5 GHz Xeon E5-1620 v3 CPU (Inter, Corp.) running on a real time linux kernel (Kernel 4.16.18-rt12) and a Realsense RSD415 camera (Intel, Corp.).

The ontology for task-level reasoning has 113 logical axioms, 40 classes, 14 object properties and 17 data properties. We initialized 12 classes, 4 object properties and 6 data properties for the peg-and-ring task. Ontology was processed in 258 ms by Pellet, guaranteeing real-time performances.

At motion planning level, the hyperparameters of DMPs are set to $\alpha = 3$ and $\mathbf{K} = \mathrm{diag}(1050)$. Moreover, a tolerance of 5% on the distance from the goal point is set to solve numerical issues. The integration time step is fixed to 10 ms to guarantee smooth continuous motion. For our implementation we learned two DMPs from a dataset of 15 task manual executions: one for the *Move and Grasp* gesture, and one for the *Carry and Leave*. In Fig. 6, we represent an example on DMPs. Two different manual *Carry and Leave* Cartesian trajectories are shown, and the general DMP is plotted for one of them. It is evident how the primitive encodes the shape of the gesture, which is common to the two manual trajectories, and is able to preserve it when changing the start and goal points. This is relevant in surgery, where the shape of the movement is a signature of its semantic meaning (e.g., circular motion and wrist orientation for suturing), and allows a connection to the higher-level knowledge base for enhanced interpretability. For obstacle avoidance, the radius of the cylindric repulsive potential surrounding the pegs is 3 cm (minimum size of the end-effector) in the *"Move and Grasp"* state, while it is 8 cm (radius of the ring) during *"Carry and Leave"*. We segmented objects in the scene comparing the Euclidean distances between the clouds obtained using the manual segmentation, which was done by extracting the points that fall inside of a 2D polyline once it is projected on screen, and by the visual inspection in Rviz.

B. Results

1) Full task execution: In this experiment, the robot started from a pre-defined initial configuration to initiate reasoning. We designed the initial positions of the rings

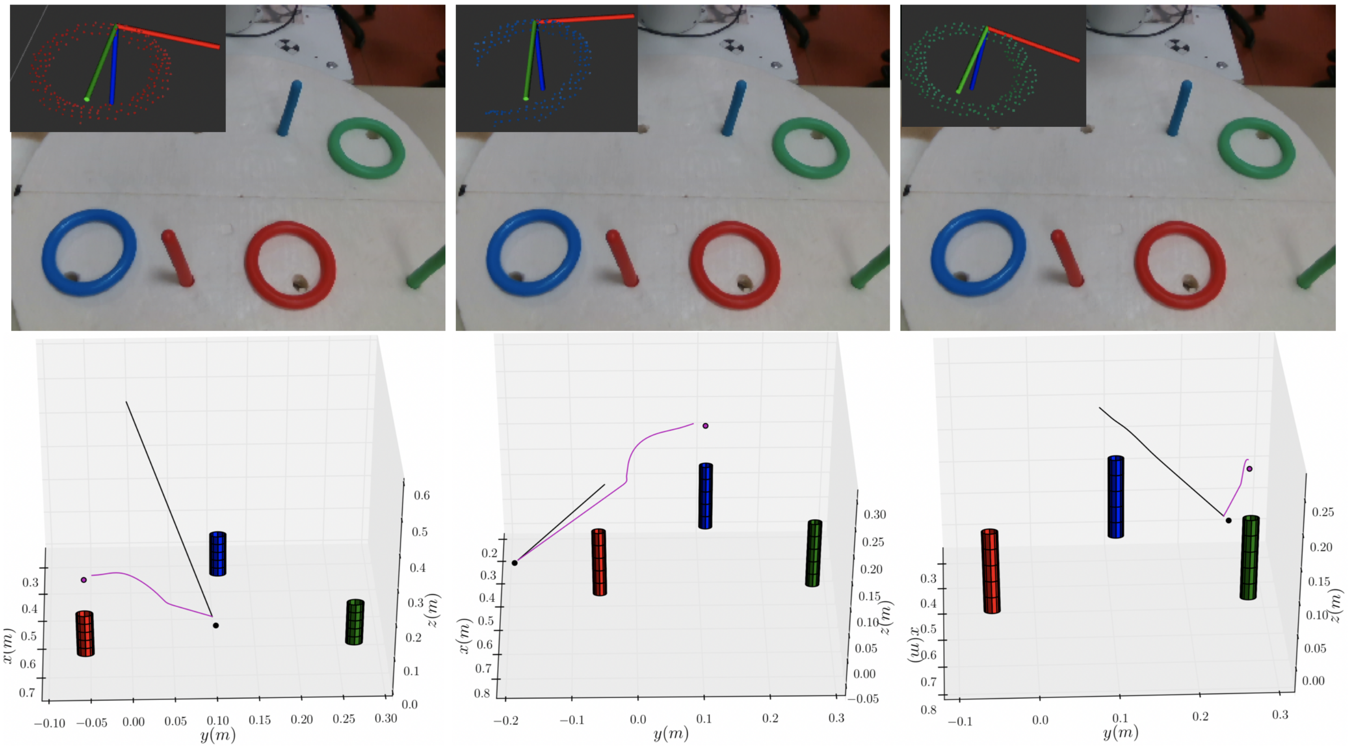

Fig. 5. Results for one execution of the full task. The upper row shows the segmentation of the rings as seen from the camera and the target grasping pose. The lower row shows the trajectories and goal points for *Move and Grasp* (black) and *Carry and Leave* (magenta) actions for the three colors (red, blue, green from left to right) and the real pegs.

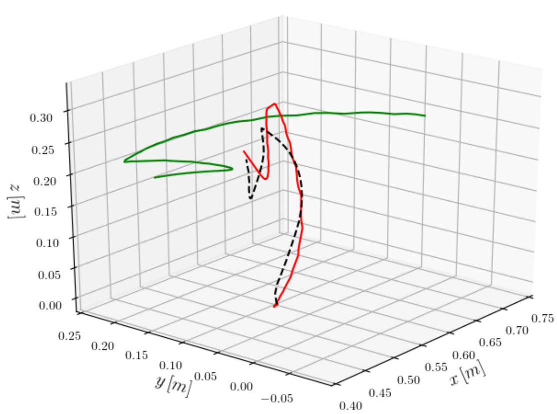

Fig. 6. Example of spatial adaptation of DMPs. The green and red trajectories are human demonstrations of the *Carry and Leave* gesture (the first for the green ring, the second for the red one). The black dashed line shows the DMP learned from the *Carry and Leave* dataset and adapted to the start and goal position of the red one.

in order to execute three different kinds of motion during *"Carry and Leave"*: peg on the side of the ring (red), peg in a far diagonal position (blue) and peg in front of the ring (green). In order to prove the repeatability of our framework the full task was executed 10 times with different orderings of the rings, and rings were always placed successfully. The attached video clarifies the experiments. As an example, in Fig. 5, the upper row shows the scene as seen from the camera, with the segmented point cloud and the target grasping pose computed by the camera for each ring in the upper left corner of the base, retrieved from RViz simulator. The lower row shows the executed trajectories for *"Move and Grasp"* and *"Carry and Leave"* for each ring. The trajectories deviate from the original DMPs because of the presence of the obstacles, especially in *Carry and Leave* for the enlargement of the obstacles. The offset between the end point of the trajectories and the actual goal is due to the tolerance in the DMP.

2) Failure recovery and dynamic adaptation: In this scenario, at first the ring was purposefully removed from the gripper of the robot while carrying. The system was able to detect the fall of the ring and trigger *Drop*, and *Move and Grasp* again, proving the task-level adaptability to failures. Then, we moved the ring away from its original position while the robot was approaching it, to show the adaptive re-planning of DMPs at the motion level. We tested both scenarios, that is ontology replanning and DMPs replanning, to prove repeatability 5 times, and re-grasping was always successful. The attached video shows the real executions with the robot.

V. DISCUSSION AND CONCLUSIONS

In this work, a knowledge-based modular framework for automation of surgical procedures involving many actions has been presented, integrating task-level ontology reasoning with adaptive motion planning with DMPs. The ontology encodes prior task knowledge, providing interpretability and reliability of the autonomous execution, which are essential for the safety requirement in surgery. Continuously querying the sensors, failures at the task level and changes in the environment are for the first time handled in a surgical framework, exploiting adaptability of DMPs. DMPs also can easily learn human movements, which is advantageous to replicate typical surgeons gestures, e.g., knot-tying. The benefits of the framework were tested on a surgical training peg-and-ring task, with an abstract representation, that is bigger objects and so on, of a real task.

At the task level, logical planners, e.g., a discrete planner with Answer Set Programming or PDDL [25] could be compared against ontologies, in terms of flexibility and scalability to more complex surgical scenarios. At the motion level, the online computational cost could be significantly improved extending obstacle avoidance to quaternion DMPs [26]. In this way, the time of computing collision-free inverse kinematics would be saved and the wrist dexterity of the surgeon would also be encoded in the motion primitives. We will also extend DMPs using the Associative Skill Memories [27] framework to learn the gestures that require certain type of interactions with the environment, e.g. "cutting a tissue", which is more apparent in realistic scenarios. We will test this framework by adding more variability in the scenario (e.g., rings with different shapes and mixed colors). We will show the easy transfer of the architecture to the surgical da Vinci robot, exploiting the modularity of the framework and experimenting on a realistic task.

ACKNOWLEDGMENT

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, ARS (Autonomous Robotic Surgery) project, grant agreement No. 742671.

REFERENCES

[1] M. Tenorth and M. Beetz, "Knowrob – a knowledge processing infrastructure for cognition-enabled robots," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 5, pp. 566–590, 2013.

[2] M. Wächter, S. Schulz, T. Asfour, E. Aksoy, F. Wörgötter, and R. Dillmann, "Action sequence reproduction based on automatic segmentation and object-action complexes," in *13th IEEE-RAS International Conference on Humanoid Robots*, 2013, pp. 189–195.

[3] E. Galceran, A. Cunningham, R. M. Eustice, and E. Olson, "Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment," *Autonomous Robots*, vol. 41, no. 6, pp. 1367–1382, 2017.

[4] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[5] Y. Kassahun, B. Yu, A. T. Tibebu, D. Stoyanov, S. Giannarou, J. H. Metzen, and E. Vander Poorten, "Surgical robotics beyond enhanced dexterity instrumentation: a survey of machine learning techniques and their role in intelligent and autonomous surgical actions," *International journal of computer assisted radiology and surgery*, vol. 11, no. 4, pp. 553–568, 2016.

[6] R. Muradore, P. Fiorini, G. Akgun, D. E. Barkana, M. Bonfe, F. Boriero, A. Caprara, G. De Rossi, R. Dodi, O. J. Elle, F. Ferraguti, L. Gasperotti, R. Gassert, K. Mathiassen, D. Handini, O. Lambercy, L. Li, M. Kruusmaa, A. O. Manurung, G. Meruzzi, H. Q. P. Nguyen, N. Preda, G. Riolfo, A. Ristolainen, A. Sanna, C. Secchi, M. Torsello, and A. E. Yantac, "Development of a cognitive robotic system for simple surgical tasks," *Int J of Advanced Robotic Systems*, vol. 12, no. 4, p. 37, 2015.

[7] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4178–4185.

[8] B. Gibaud, G. Forestier, C. Feldmann, G. Ferrigno, P. Gonçalves, T. Haidegger, C. Julliard, D. Katić, H. Kenngott, L. Maier-Hein, K. März, E. De Momi, D. A. Nagy, H. Nakawala, J. Neumann, T. Neumuth, J. R. Balderrama, S. Speidel, M. Wagner, and P. Jannin, "Toward a standard ontology of surgical process models," *International J CARS*, vol. 13, no. 9, pp. 1397–1408, 2018.

[9] T. Gruber, "A translation approach to portable ontologies," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[10] C. Schlenoff, E. Prestes, R. Madhavan, P. Goncalves, H. Li, S. Balakirsky, T. Kramer, and E. Miguelez, "An ieee standard ontology for robotics and automation," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 1337–1342.

[11] A. Pease, I. Niles, and J. Li, "The suggested upper merged ontology: A large ontology for the semantic web and its applications," *AAAI Technical Report*, vol. WS-02-11, 2002.

[12] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, "Sweetening ontologies with dolce," *Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web. EKAW, Lecture Notes in Computer Science*, vol. 2473, 2002.

[13] P. Grenon and B. Smith, "Snap and span: Towards dynamic spatial ontology," *Spatial Cognition and Computation*, vol. 4, no. 1, pp. 69–103, 2004.

[14] I. Horrocks, P. Patel-Scheider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "A semantic web rule langauge combining owl and ruleml. w3c member submission 2004," accessed: 2019-01-12. [Online]. Available: https://www.w3.org/Submission/SWRL

[15] M. Fox and D. Long, "Pddl2.1: An extension to pddl for expressing temporal planning domains," *Journal of Artificial Intelligence Research*, vol. 20, pp. 61–124, 2003.

[16] S. Evren, B. Parsia, B. Cuenca Grau, A. Kalyanpur, and Y. Katz, "Pellet: a practical owl-dl reasoner," *Web Semant.: Sci, Serv Agents World Wide Web*, vol. 5, no. 2, pp. 51–53, 2007.

[17] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal, "Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance," in *Robotics and Automation, IEEE International Conference on*. IEEE, 2009, pp. 2587–2592.

[18] M. Ginesi, D. Meli, A. Calanca, D. Dall'Alba, N. Sansonetto, and P. Fiorini, "Dynamic movement primitives: Volumentric obstacle avoidance," 2019, under revision.

[19] M. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, 2018.

[20] M. Hong and J. Rozenblit, "Modeling of a transfer task in computer assisted surgical training," *MSM16 Proceedings of the Modeling and Simulation in Medicine Symposium*, p. 4, 2016.

[21] F. Lalys and P. Jannin, "Surgical process modeling: a review," *International J CARS*, vol. 9, no. 3, pp. 495–511, 2014.

[22] N. J. Soper and G. M. Fried, "The fundamentals of laparoscopic surgery: its time has come." *Bull Am Coll Surg*, vol. 93, no. 9, pp. 30–32, 2008.

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

[24] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[25] Y. Jiang, S. Zhang, P. Khandelwal, and P. Stone, "Task planning in robotics: an empirical comparison of pddl-based and asp-based systems," *arXiv preprint arXiv:1804.08229*, 2018.

[26] A. Ude, B. Nemec, T. Petri, and J. Morimoto, "Orientation in cartesian space dynamic movement primitives," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2997–3004, 2014.

[27] P. Pastor, M. Kalakrishnan, L. Righetti, and S. Schaal, "Towards associative skill memories," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*. IEEE, 2012, pp. 309–315.