Politecnico Di Milano

Biomedical Engineering
School of Industrial and Information Engineering

Milano Leonardo Campus
20131, Milan Italy

Master Thesis

Surgical Process Modeling of Robotic Partial Nephrectomy with Answer Set Programming

Sara Sabry

Matriculation Number: 897010

19.11.2020

Supervised by
Prof. Dr. Elena De Momi

Thesis advisors
Dr. Hirenkumar Nakawala
Daniele Meli PhD

I would first like to thank Dr. Elena De Momi for giving me the opportunity to work on such an important field of research.

I am extremely grateful to Dr. Hirenkumar Nakwala. The success of this project would not have been possible without his guidance and support.

I would also like to give special thanks to Dr. Aldo Marzullo and Daniele Meli PhD for all their help. Their assistance was invaluable in the completion of this thesis. Furthermore, I would like to thank Alice Segato PhD for her guidance in writing the thesis.

Finally I would like to give special acknowledgement to Plamen Pasliev, without whom I would not have been able to do this.

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Milan, 19.11.2020

Sara Sabry

Abstract

Having full autonomy in robotic surgery would revolutionize the quality of healthcare. However, one of the biggest challenges faced in automating a surgical procedure is the lack of data that can be used in statistical or Machine Learning approaches. Another challenge stems from the high variability between patients, and the differences between surgical techniques used for the same procedure. Current approaches require large amounts of data to provide accurate predictions, which is seldom available for surgery.

The aim of this study is to create an AI system that is able to generate a surgical process. A Robotic Partial Nephrectomy is used as a prototypical implementation. First, Surgical Procedural Knowledge is extracted from articles written by experts on the techniques of the procedure, and then formalized using Surgical Process Modeling strategies. Using this formalized knowledge, Answer Set Program (ASP) rules and constraints were created. ASP uses external atoms, and the knowledge it is given a priori, to reason on the surgical process.

Real surgical video annotations were used for evaluation. Anatomies were extracted from these surgeries, and given as input to the ASP algorithm, to generate the surgical process. Actual Steps and Actions were compared to those predicted by the ASP system. The overall accuracy of the model in predicting Steps and Actions was found to be 95.3%. Actions were predicted with a mean precision of 97% with a standard deviation of 6.5%, a mean recall of 89% with a standard deviation of 14.3%, and a mean F1-score of 92% with a standard deviation of 8.1%, computed over all the annotations. Additionally, expert urologists validated the correctness of the surgical workflow provided by the ASP system.

ASP allows for a robust, and flexible surgical process generation. This system can be integrated with a situation awareness module, as well as a surgical robot, to increase the level of autonomy in surgery.

Sommario

Avere piena autonomia nella chirurgia robotica rivoluzionerebbe la qualità dell' assistenza sanitaria. Tuttavia, una delle maggiori sfide affrontate nell'automazione di una procedura chirurgica è la mancanza di dati che possano essere utilizzati in approcci statistici o di Machine Learning. Un'altra sfida deriva dall'elevata variabilità tra i pazienti e dalle differenze tra le tecniche chirurgiche utilizzate per la stessa procedura. Gli approcci attuali richiedono grandi quantità di dati per fornire previsioni accurate, che raramente sono disponibili per la chirurgia.

Lo scopo di questo studio è quello di creare un sistema di IA in grado di generare un processo chirurgico. Una Nefrectomia Parziale Robotica viene utilizzata come implementazione prototipale. In primo luogo, la conoscenza della procedura chirurgica viene estratta da articoli scritti da esperti sulle tecniche della procedura, e poi formalizzata utilizzando strategie di Modellazione del Processo Chirurgico. Utilizzando questa conoscenza formalizzata, sono state create le regole e i vincoli del Answer Set Program (ASP). L'ASP utilizza atomi esterni, e la conoscenza che gli viene data a priori, per ragionare sul processo chirurgico.

Per la valutazione sono state utilizzate vere e proprie annotazioni video chirurgiche. Le anatomie sono state estratte da questi interventi, e date come input all'algoritmo ASP, per generare il processo chirurgico. I passi e le azioni reali sono stati confrontati con quelli previsti dal sistema ASP. L'accuratezza complessiva del modello nella previsione di Passi e Azioni è risultata essere del 95,3%. Le azioni sono state previste con una precisione media di 97% con una deviazione standard di 6.5%, un richiamo medio di 89% con una deviazione standard di 14.3%, e un F1-score medio di 92% con una deviazione standard di 8.1%, calcolato su tutte le annotazioni. Inoltre, urologi esperti hanno convalidato la correttezza del flusso di lavoro chirurgico fornito dal sistema ASP.

L'ASP consente di generare un processo chirurgico robusto e flessibile. Questo sistema può essere integrato con un modulo di consapevolezza della situazione, così come un robot chirurgico, per aumentare il livello di autonomia in chirurgia.

Contents

1 Introduction

1.1 Motivation

Over the last few decades, surgical autonomy has been extensively studied with minimally invasive surgery (MIS). MIS provides tremendous benefits over open surgeries in the reduction of infections [14], as well as a general improvement of cosmetic outcomes, and a reduction in recovery time. With its increase in popularity, and with the rise of surgical robots, autonomy in surgery became an important field of research. Increasing the level of autonomy [1] in surgery can improve the quality of healthcare. As described in [2], this autonomy can increase the level of safety and reduce the patient's recovery time. It can also reduce the chances of mistakes that usually occur because of fatigued surgeons. Having full autonomy in surgery can also help human kind in its quest for space exploration. It is expected that in the early 2030s, longer space flights to the moon and mars will probably begin. The mars exploration is planned to last about two and a half years. However, medical crew members might not have the proper training to treat all the possible pathologies they may face [3]. For that, a fully autonomous surgical approach might be useful, or even life saving. Semi-autonomous system can also be used if fully autonomous ones are not yet available. Crew members without full medical training can perform the incisions and closure parts of the surgical procedure while having the robot do all the rest [4].

Artificial Intelligence (AI) research for autonomous robotic surgery includes all aspects of a surgical procedure. This involves research that ranges from situation awareness and scene understanding, safe and explainable surgical plan generation, to dexterous trajectory generation. A lot of research has been successful in interpreting data from sensors to guide the execution of simple tasks such as knot-tying [5] or drilling [6]. Others have been successful in proving greater dexterity by a surgical robot in executing some simple surgical actions compared to humans. Some examples include the studies [7], [8] and [9] further explained in Chapter 2. Current AI such as machine learning (ML) raises many problems in applications such as surgery. For one, ML algorithms are often treated as a black box, where the predictions generated are not explainable. Methods such as Convoluted Neural Networks (CNN) have yielded exceptional results for image recognition. Other AI, including Ontologies, Knowledge Representation, and probabilistic models have also been shown to be useful for detection and diagnostics. However, a large amount of data is needed to get accurate results. In surgery, this data is scarcely available, and the intra-operable variations between surgeries increase this difficulty even more.

While there are significant advances in robotic surgery in terms of mechanical abilities such as scene understanding with sensors, and having robotic arms that provide greater dexterity, there is still opportunity for improvement to reach full automation in surgery. AI systems are still limited in terms of their ability to judge and plan surgical procedures. Being able to reason on information not known in advance is a human cognition ability that AI is yet to perfect [1].

One of the greatest challenges in automating surgical planning is the lack of well-defined rules on how to specifically perform a procedure [1]. Expert surgeons usually do not all perform an operation in the exact same manner. There are too many variables between patients and preferences in techniques, to have a consensus on how to exactly perform a surgical procedure. This thesis attempts to use AI for real-time surgical process generation. Our focus is to only deal with the decision making portion of the surgery, using a knowledge representation language: Answer Set Programming (ASP). ASP is a declarative rule based logic programming language that uses knowledge on the domain to reason on the surgical process. ASP can reason in real-time on information about the environment from sensors, by using the knowledge it is given a priori by experts.

1.2 Objective

This thesis attempts at providing a declarative reasoning approach to modeling a surgical procedure. This AI system will also be able to adapt to environmental changes by quickly re-planning the Surgical Process (SP). For that, we chose a logic programming language: Answer Set Programming (ASP). ASP is a powerful problem solving tool, often used for planning problems and robotics. The aim of this work is to apply this AI method on a Robotic Partial Nephrectomy (RPN). Surgical Procedural Knowledge is extracted from articles written by experts, on the principles and techniques of this procedure. The RPN process is first formalized, and then rules within the ASP program are created using common sense, from surgeon's description of the task. The ASP rules are high level knowledge extracted from descriptions of the surgical technique. When combined with constraints, a safe, explainable and reliable system is guaranteed.

1.3 Scope

One of the integral parts in surgical automation is the generation of safe and explainable plans. In this thesis, the rules and relations that govern the ASP system are based on the descriptions of the robotic partial nephrectomy technique. In these two articles [10, 11], small differences in the order of steps can be seen. However, the sequential understanding of the specifications of the procedure are very similar. For this reason, the ASP algorithm should provide the conditions necessary for a successful surgical process, while also leaving room for flexibility.

The main contributions of this work are described as follows:

- A robotic partial nephrectomy will be modeled using Surgical Process Modeling strategies. This model will be generic enough to be expanded to other robotic laparoscopic procedures.
- The ASP algorithm will give a justifiable surgical process that gives information on the different granularity levels: Phase, Step, Instrument, Action and Anatomy.
- Input for the ASP system will be assumed to come from sensors, acting as external atoms, on the anatomies identified. Two types of outputs can be achieved with this system: a complete surgical workflow generation, or a real time surgical procedure that reflects a more realistic scenario.
- For this work, scene understanding and the actual executions of these actions are not dealt with, but merely the decision making and reasoning part.

Finally, this work will be evaluated both quantitatively and qualitatively. Real surgical video annotations will be used for the comparison between actual surgical steps and actions, and those predicted by the ASP system. In addition, four urologists will evaluate the provided workflow and give their expert opinion on how the system as a whole can be improved and be applicable in real surgical scenarios.

1.4 Outline

This thesis is divided into 7 chapters.

Chapter 2 'Related Work' describes some relevant research that has been done to achieve autonomy in surgery. It takes the reader through some advances in minimally invasive surgery and robot-assisted surgery, to different levels of autonomy in surgery.

Chapter 3 'Fundamentals' explains some important concepts that need to be understood for Surgical Process Modeling and Knowledge Representation in the field of Surgery.

Chapter 4 'Problem Description' clarifies the difference between Machine Learning and Machine Reasoning, and how a combination of the two would could achieve an ideal scenario.

Chapter 5 'Methodology' describes the methods used in the thesis to achieve the goal proposed in order to address some of the problems mentioned. The logic behind the program is described as well as the exemplary surgical procedure being modeled.

Chapter 6 'Experimental Results and Validation' portrays the effectiveness of the program proposed, and evaluates it by comparing it to real surgical video annotations. a qualitative analysis was also conducted with the help of four urologists.

Chapter 7 'Conclusion' briefly summarizes this thesis, while recounting the problems faced and the reasons behind the results achieved. This will be followed by a short description of potential future work.

2 Related Work

2.1 Minimally Invasive Surgery, Automation and the use of Artificial Intelligence

Let us first begin by giving some background in the field of robotic surgery and autonomy.

2.1.1 Minimally Invasive Surgery

Minimally invasive surgery (MIS) has been a hot topic for decades. Researchers have been trying to find ways to reduce the risk of surgical complications, postoperative pain, and recovery time [12]. This approach has also allowed for a reduction in hospital stays which led to a decrease in cost for healthcare organizations [12].

For these reasons, along with the improved cosmetic outcomes, minimally invasive techniques have become widely used around the world in many fields of surgery [12].
Surgical site infections (SSIs) are very common in patients who have just undergone surgery, according to the Centers for Disease Control and Prevention [13], [14]. In [15], the relation between SSIs, morbidity and rates of readmission after being discharged from the hospital is presented. Furthermore, the effect of minimally invasive surgery on the risk of SSIs has been studied in [13] and [16]. Gandaglia et al. [16] evaluated the role of minimally invasive techniques on the risk of SSIs after surgery. After some logistical regression analyses, they were able to show that MIS significantly reduced the likelihood of SSIs. Another study by [17] evaluated SSIs in minimally invasive urological surgery, and compared this approach with a traditional open one. Their results showed that the risk of SSI for MIS was again reduced, and that MIS yielded better perioperative outcomes as well as lower complication rates [17]. A plethora of research initiatives has been done showing this significant improvement in surgical outcomes when using MIS compared with laparotomy, as seen in [18] and [19].

2.1.2 Robot-Assisted Surgery (RAS)

Minimally invasive surgery has incentivised the use of robotics in surgery. To perform MIS, surgeons used robots to improve precision during surgery, which also lead to a reduction in recovery time for patients [20, 21, 22]. Robot-assisted surgeries have proven to be more advantageous over conventional ones [23, 24].

One of the most crucial roles that a robot may play in semi- or fully automated surgery is the reduction of medical errors. Medical errors have been shown to be the third leading cause of death in the United States of America in 2016, after cancer and heart diseases [25, 26].

The da Vinci control system [27] was a revolutionary step towards achieving this goal. Nowadays, it's being used in many different fields including General surgery, Urology, Cardio-Thoracic, Gynecology, Otolaryngology and Pediatrics [28]. Since its inauguration, many different versions have been introduced [29] which resulted in more sophisticated surgery. Over the years, the three-dimensional visual acuity was improved, along with an increase in degrees of freedom. This was a big milestone when it came to accuracy and accessibility in surgery [29].

Surgical Robots have also been able to help surgeons by taking over some tasks, allowing them to focus on the more critical issues at hand. This lead to a reduction in their fatigue during operation, and thus reducing the likelihood of them making mistakes [26, 1].

For more complex tasks, surgical robots can improve the quality of surgical operation performed by giving surgeons greater dexterity and visualization, which ultimately leads to less errors [29, 24].

Advances in the surgical robot sensing capabilities improved the skills of surgeons operating in confined or intricate environments [23]. Technologies for sensing and actuation have seen a lot of progress in the past few decades, which has led to much more effective and precise surgery [7].

In one recent study [30], Hwang et al. were the first to ever manipulate da Vinci robotic arms with depth sensing. They used a Zivid One Plus RGB+depth (RGBD) camera. This camera provided images with 1920x1200 pixels at 13 frames per second and a depth resolution of 0.5mm [30]. They used the example of the peg transfer task commonly used for surgical training [31]. Surgeons were able to perform this task with eminent dexterity [32]. Their attempt at using this new technology with the goal of achieving higher standards of surgery showed promising results.

2.1.3 Autonomy in robotic surgery

The idea of automating surgery has been around for decades, and there have been many attempts at using some kind of artificial intelligence to further advance the field of surgery.

An example can be seen in the study led by Shademan et.al [8]. Robot-assisted surgeries are still dependent on the skills of the surgeon handling the arms. However, this study of a supervised autonomous robotic surgery on soft tissue demonstrated how this system outperformed a task such as suturing. A three-dimensional and near-infrared fluorescent (NIRF) light-field camera directed by an autonomous suturing algorithm was used in an open surgery setting. The latter named STAR, Smart Tissue Autonomous Robot, seen in Figure 2.1, is comprised of a robotic arm made of lightweight materials, and has a laparoscopic suturing tool attached to its end [8]. Thanks to its eight-degrees of freedom (DOF), it is able to perform tasks with great dexterity and accuracy. The combination of this NIRF techonology with the plenoptic imaging system allows it to recognize NIRF markers on tissue targets [8].

The study compared some of the ex-vivo sutures performed by STAR to those performed by highly skilled surgeons with at least 7 years of experience. For a first evaluation, STAR was found to perform better than hand-sewn suturing (OPEN), laparoscopy (LAP) and current RAS with da Vinci Surgical System techniques. STAR showed more consistent spacing between sutures leading to higher leak pressure resistance. An in-vivo supervised surgery was also performed in pig intestines through a laparotomy using STAR, and was compared with an OPEN control. Anatomy evaluations showed similar leak pressures to those of the OPEN control surgery.

The results of this study of supervised autonomous surgery have shown that STAR is safer than other types of surgeries, and would give access to better surgical techniques with higher efficacy, all while removing the human factor of surgeon experience [8].

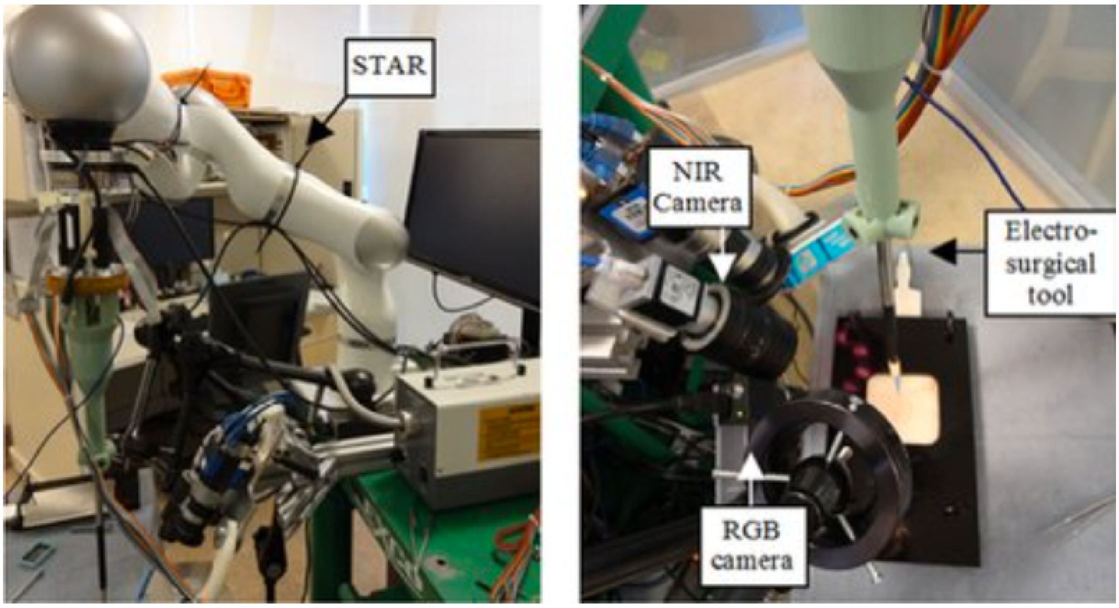

Figure 2.1: Smart Tissue Autonomous Robot (STAR) [33]

Another experiment was done with the STAR system evaluating its performance in an electrosurgery for tumor resection [33]. Their experiments showed that, despite taking more time than surgeons, STAR achieved more consistent results with less deviation than any human modality [33]. They demonstrated how this robotic system was able to outperform highly experienced surgeons in accurately completing the cutting and charring tasks. STAR was also able to successfully perform a pseudo-tumor resection using visual servoiung. Again, a semi-autonomous robotic system was demonstrated to be feasible and very promising [33].

Automating surgical tasks can provide important benefits for surgeons and clinicians. In this study, [9], researchers attempted to use a novel sensing modality at the catheter tip for autonomous navigation inside one of the most complex human organs, the heart. This haptic vision used machine learning and image processing algorithms combined with intracardiac endoscopy to form a hybrid touching and imaging sensor. The latter can identify what it is touching as well as how much pressure is being applied [9]. Their in vivo experiments with this autonomous navigation system were evaluated and compared with operator-controlled robot motions that had manual navigation. Their results showed successful completion of the intracardiac catheter navigation task, with similar efficacy and procedure time when compared with expert manual navigation [9]. This work successfully exhibited an alternative approach to robotic surgery.

One of the aspects of surgical autonomy that we have yet to discuss is the development of a cognitive robotic system that is able to reason and to make decisions intra-operatively, depending on real-time information acquired by sensors. This has been one of the long-term goals in research relating to autonomy in surgical robotics [34, 35]. In the following section, we discuss some of the most recent work in autonomous surgical task planning.

2.2 State of the Art application of ASP

Answer Set Programming (ASP) can make explainable decisions by taking in information from sensors as well as prior knowledge given by experts on the task at hand.

In a recent study done by Ginesi et. al., they used a da Vinci Research Kit (DVRK) with ASP for task planning while integrating it with Dynamic Movement Primitives (DMPs) for real-time obstacle avoidance and trajectory generation [36]. DMPs learn from a small dataset of gestures of surgeons to mimic their dexterity.

In their study, they focused their efforts on validating this framework using a peg and ring task, since surgeons often use this example for training in surgery. In this peg and ring example much like in a real surgery, robot arms must move while avoiding obstacles. They are also required to grasp and precisely place small objects with great efficacy. Another similarity to a real surgery is having a dynamic environment, meaning the information acquired from sensors can change in real time. For this reason, the AI must be able to quickly re-plan in response to those changes. It must also be ale to deal with failures in an explainable way. The AI takes in these new conditions, and gives a new model in order to achieve the final goal [36].

The main limitation of this work [36] is that the system assumed that the knowledge the AI had of the situation and task was comprehensive. Surgical scenarios are so complex that it is quite hard to have enough labeled training examples or to encode a full picture of the domain knowledge. There are so many variations in patient's anatomies that are simply not known in advance. The conditions that were thought to be known before surgery can also change over time. This makes having a comprehensive domain knowledge unrealistic in the field of robotic surgery [36].

For this reason, the team of Meli et. al [37] performed another experiment as a continuation of this previous one. Their main focus was to have a formalism that is able to learn the initial incomplete knowledge from a small number of human executions of this task. For that, they used Inductive Logic Programming (ILP) where examples of task executions are labeled in order to learn the axioms that govern the dynamics of the domain. They added that to the underlying ASP system that is able to reason with incomplete domain knowledge. Their system was in fact able to learn from this limited number of examples about the domain knowledge.

Their experimental results showed to be a successful step towards learning and automation in surgical robotics when compared with probabilistic approaches or machine learning, where these black-box tools cannot perform well with a limited number of examples to learn from.

In our study, we will use ASP on a more complex surgical task, that of a robotic partial nephrectomy.

With all these efforts to automate robotic surgery, advances in the field would be difficult without having re-usable knowledge (data) that can be easily shared and understood [38]. For that, an agreement on the language and definitions of the terms are needed. This is where ontology languages come into play. For successful knowledge representation, the data needs to be shareable and reusable among different researchers [39]. This is further explained in Chapter 3.

3 Fundamentals

3.1 Surgical Process Modeling Strategies

Current advances in the field of surgical autonomy lean more towards machine learning techniques. Some of these examples can be seen in [40] and in [41] using Hidden Markov models. Others use Dynamic Time Warping techniques as seen in [42]. Moreover, some research has used statistical analyses [43] and Random Forests [44] as well. Combinations of some of these techniques can also be seen in [45] and [46]. However, these methods don't use standardized medical background knowledge in a recoverable way. Conversely, some approaches can formalize this medical knowledge generically. Examples of such formal methods include Unified Modeling Language (UML) [47] or Description Logics [48] and ontological representations [49].

Hence, Surgical Process Modeling (SPM) techniques need to be based on ontology [50]. That way, different approaches can be coherent, interoperable and comparable, which in turn increases the value of data. A four level translational approach to SPM is described in [50]:

- Natural Language Level
  The first level is the natural language level related to the users. It allows them to include their knowledge and experience into the model and acts as an interface for result communication and analysis.

- Conceptual / Ontological Level
  The second level is the conceptual or ontological level and allows for domain knowledge analysis with regards to ontology.

- Formal / Mathematical Level
  The third is the formal level, and allows one to formalize domain knowledge into mathematical formulations in order to determine its purpose.

- Implementation Level
  The final level is that of implementation and deals with the actual realization of the formalization done in the mathematical level, in a machine-processable language.

A surgical procedure was first described in [51] as a sequence of steps. The latter can also be referred to as a workflow. For decades, researchers have been trying to model surgical procedures using different modeling strategies. Each modeling strategy is characterized by its own granularity levels, the methods with which they acquire data, and modeling approaches [52]. However, in this thesis, we will be focused on the modeling part of SPM as seen in figure 3.1.

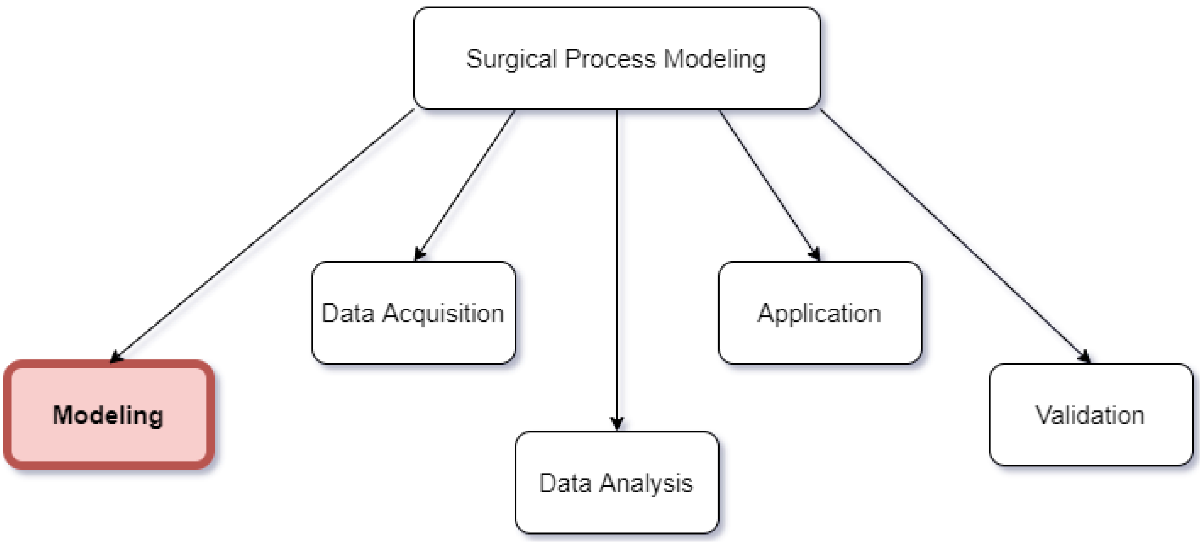

Figure 3.1: General overview of Surgical Process Modeling [53]

The first step in surgical process modeling is defining the the modeling approach. After that, we must illustrate the different granularity levels, and how the information used for modeling is acquired.

3.1.1 Modeling Approach

Modeling a surgical procedure can be done in two different ways: top-down and bottom-up [54]. When using a top-down approach, the procedure is described from highest abstraction levels to lowest, meaning, from lowest to highest granularity. Contrarily, a bottom-up model goes from the highest level of granularity to the lowest [52].

With the top-down modeling approach, the whole procedure is first described with high abstractions before getting into more details, which makes it less likely to make mistakes in identifying the lower abstraction activities [52]. In this thesis, we will be using a top-down approach for Knowledge Representation, where information about the procedural technique is extracted from literature.

When using a bottom-up approach, low-level activities are first modelled, which makes it more precise. The latter comes at the cost of having a complex process with results that might be different from reality. An example of such an approach is when speaking of Machine Learning (ML) methods, which will be further explained in section 3.2.

Before we can explain how these two modeling approaches are used in surgery, we must understand how a procedure is described according to granularity levels.

3.1.2 Granularity Levels

Surgical process modeling is done by taking into account the concept of granularity level. *"A granularity level is defined as the level of abstraction at which the surgical procedure is described"* [53]. On a granularity axis, the lowest level is the procedure itself, with the highest abstractions. The higher the granularity level, the more detailed it is, and the lower the abstractions. The procedure itself, which is the lowest granularity level, is then decomposed into Phases, Steps, Activities, and Motions.

A surgery can be divided into main events called phases. These Phases are composed of Steps. Each Step is a sequence of Activities. An Activity is a physical task and consists of a list of Motions. Motions can be described as a surgical task that involves the movement of only one hand [52, 53]. In this thesis, we focus on the Phases, Steps and Activities. An Activity will be described by an Action, done on an Anatomy, using an Instrument.

After understanding the different granularity levels, the next step would be to define the different methods of acquiring information in surgery before beginning to model the procedure.

3.1.3 Information Acquisition

Information is acquired differently depending on the modeling approach. In this thesis, a top-down approach for modeling is used for Knowledge Representation. Correspondingly, the information about the procedural technique is extracted from literature.

For bottom-up approaches such as ML methods, data is acquired according to the methods described herein. In general, data acquisition can either be computer-based, or be done manually. Having a complete and solid data set is essential for surgical process modeling since it would affect the whole workflow study.

Manual data acquisition can be done online or offline. For online observation of the surgical process workflow, data is recorded by the observer present in the OR. In their papers, [51, 55, 56] used manual data acquisition methods. Some of the advantages of this method include the fact that the observer can see and interact with clinical team members.

However, human error is very high when trying to record large amounts of data in the OR. Having offline data acquisition has been shown to overcome these limitations, as done by [51, 56, 57], but the observer could no longer interact with the clinical team since he or she is gathering the information through video recordings. Another problem is that low-level data cannot always be provided by observations in the OR.

The emergence of computer-based data collection technologies overcame some of these limitations. Automated data acquisition was able to eliminate human error by using sensors, image recognition and processing techniques [58, 59]. This tracking system can be used in the Operating Room (OR), or on videos of the surgical procedure.

However, the surgical process modeling field is very complex, which renders this data collection task challenging. Some of the limitations of the process include the fact that some specific tasks in the procedure cannot be properly identified, i.e., the signal might not be able to show what the purpose of a certain instrument is. It might not be able to detect the small differences between using an electrical surgical knife to dissect fat, or cut lesions.

Developing reliable sensors and optical tracking systems (OTS) is a challenge. Both the optical and electromagnetic tracking systems (ETS) [60, 61] have drawbacks. For the OTS, the tracking markers should always be visible to the tracking systems, and they need to be attached to rigid materials. Therefore, the latter cannot track flexible instruments or soft tissues. These problems can be eliminated using an ETS, but unfortunately, when having metal objects nearby, the performance of the ETS worsens.

With this data acquired, one must learn to represent the information in a comprehensible way.

3.1.4 Model Representation

The results of a surgical process model need to be represented in a way that is easily understood before they can be interpreted. This can be done in two ways:

- Descriptive Representations
  Where the description is done using text [41, 62].

- Numerical Representations
  Where representation of the system is done using programming languages or mathematical relations [63].

A lot of well known formal and semi-formal languages are numeric representations. Some examples include UML [64], BNPN [65], workflow diagrams [56, 66], and YAWL, a workflow modeling language [63].

Descriptive representations are simple and easy to understand, but are not detailed enough to have a proper analysis of the model since not all the relations between the entities are represented. They usually need to be combined with a numerical approach that provides these missing details. We are able to make qualitative analyses as well as simulations thanks to numerical representations. The downside to this method is that it is not easy to achieve, and that once it's done, it leaves no room for flexible changes [52].

Finally, after acquiring the information with different granularity levels and representing it, we can discuss the two main modeling approaches used in surgery, and how Knowledge Representation exhibited a clear advantage.

3.2 Machine Learning Vs. Machine Reasoning

To achieve Autonomy in Robotic Surgery, there are five main parts to deal with: Object Recognition, Navigation, Human-Robot interaction, Manipulation and Decision Making. In this thesis we will be dealing with the decision making portion of surgery, as seen in figure 3.2. For that, we are using Answer Set Programming for Knowledge Representation and task reasoning in robotic surgery, while assuming a sensing module gives the AI system information on the environment.

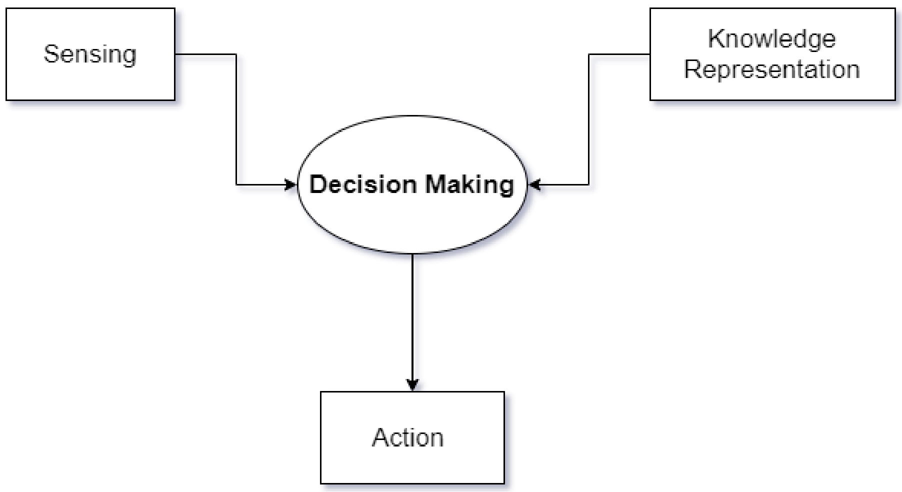

Figure 3.2: Simplified Decision Making Process with an AI method

In many of the aforementioned research in Chapter 2 and in [67], [68] and [30], the AI relies on tasks described using finite state machines (FSM). In these works, the environment is assumed to be static. In such cases, the AI is not aware and cannot react to anomalous events such in dynamic environments. Machine learning systems (ML) are bottom-up approaches to modeling. They are able to recognize patterns from large datasets and make predictions for similar scenarios. ML methods cannot predict on a new situation or solve a problem that it has never seen before.

In [69] and [70], **statistical (hidden Markov Model) and data-driven (neural networks) models** are used in robotic surgery to overcome some of these limitations and to improve situation awareness. However, to be able to achieve accurate results with these methods, a great deal of data is required. Unfortunately in surgery, this data is usually unavailable, which prohibits us from getting accurate results with these machine learning systems.

Another problem that arises when speaking about machine learning (ML) is the fact that they are black-box tools, as seen in figure 3.3. This means that the predictions or plans that are generated cannot be explained or monitored. In [71], they were able to prove how some explanations can be manipulated, which furthermore impacts the appropriateness of this autonomous system in applications where safety is prioritized, such as surgery. AI systems need to be explainable to be trusted, and as mentioned in [72], Neural Networks in combination with a Knowledge-based approach might be the answer. However, in this thesis, we will attempt the latter.

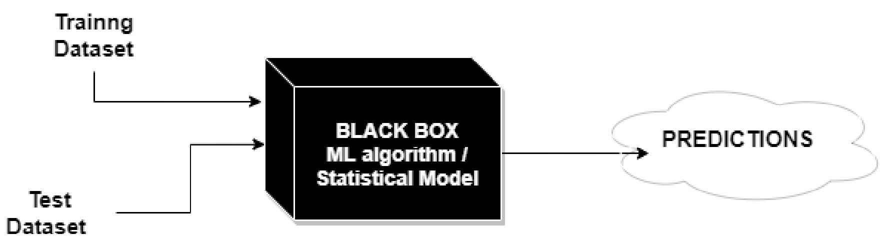

Figure 3.3: Black Box in Machine Learning

For the reasons mentioned above, researchers have switched their focus to Knowledge-Based reasoning systems [73, 74], where in the context of surgical robotics, information collected from surgeons' prior expertise on surgical tasks are encoded. This allows for a better understanding of the execution workflow [36].

After understanding the different Modeling Approaches that can be used in surgery, one must recognize how to represent the knowledge to be used in the modeling of a procedure. The next section will cover some of the main types of Knowledge Representation (KR) techniques, while focusing on Logical Representation and Answer Set Programming.

4 Knowledge Representation and Answer Set Programming

In this chapter, we will describe the Knowledge Representation techniques, the method used to achieve the objective of this research, and how ASP addresses some of the limitations of current approaches.

4.1 Knowledge Representation and Logic Programming

The goal of Knowledge Representation (KR) is to: *"Develop formalism for providing high-level descriptions of the world that can be effectively used to build intelligent applications."*[75]. Some keywords in this definition can be broken down and further described:

- "Formalism:"
  Having formal and unambiguous semantics where, if the word or phrase is taken out of context, it has only one interpretation [76].
- "High-level descriptions:"
  Representing information in an explainable and human readable way.
- "Intelligent applications:"
  Where the AI is able to deduce new knowledge from what is given [77].
- "Effectively used:"
  The reasoning approach should lead to usable execution.

There are four main types of KR techniques as seen in figure 4.1. In this thesis, we chose to use a logical representation approach, with Answer Set Programming. In the following sections, we will briefly describe those different techniques in order to elucidate the reason for this choice.

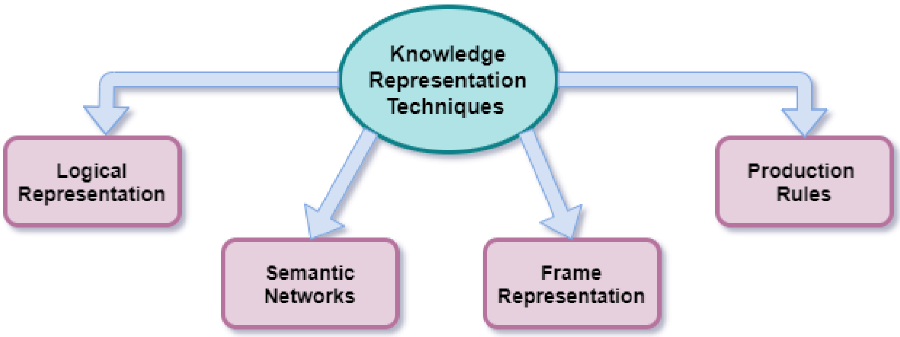

Figure 4.1: Four main types of Knowledge Representation Techniques

4.1.1 Knowledge Representation Techniques

Before we talk about the four main types of knowledge representation techniques, some important concepts need to be touched upon:

Ontologies
Ontology groups and makes relations between different entities [77]. It also organizes them by categorizing the information according to their differences and similarities. In this way, the knowledge stored becomes more easily accessible and understood by machines. Ontology is most helpful when the knowledge being represented is domain specific, such as in medicine [77]. It is useful for integrating information, as well as retrieving it.

Description Logic
Description logics are often used in ontological modeling [76]. The most important part of a Description Logic is the Concept Language. There are three main categories: concept names, role names, and constructors [75, 78].

In this example:
$Mammal \cap \exists\, has - mother.Dog \cap \forall\, drinks.Milk$

"Mammal", "Dog" and "Milk" are concept names that assign a name to groups of objects. "has-mother" and "drinks" are role names that define relationships between objects. $\cap$, $\exists$ and $\forall$ are constructors that can relate concept names to role names [78].

Concept definitions would be in this form:

$Puppy \equiv Mammal \cap \exists\ has - mother.Dog\ \cap \forall\ drinks.Milk$

To better explain some of the constructors used in these examples, table 4.1 shows the main ones and their corresponding definitions. There are two atomic types: "C" and "D" are concept names, while "r" is a role name [78].

Bayesian Network for distributed representation
On a different note, when knowledge is uncertain, a Bayesian network uses conditional probability to reason and make prediction. This network is composed of nodes representing variables connected to one another by links. These links have different strengths of associations and/or dependencies, characterized by each conditional probability [77].

Concept constructor	Definition
$\neg C$	Negation
$C \cap D$	Conjunction
$C \cup D$	Disjunction
$\exists r.C$	Existential restriction
$\forall r.C$	Value restriction
$\top$	Top concept
$\bot$	Bottom concept

Table 4.1: Concept Constructors and their definitions [78]

After briefly defining these important concepts, we will explain the different ways to represent knowledge. KR can be done using Semantic Networks, Production Rules, Frames, or Logic. In this thesis, knowledge will be represented using a logic-based approach for reasoning. However, we must first establish the differences between these techniques.

1. **Semantic Network**
   Some early KR systems used Semantic Networks to represent declarative knowledge. The formalism is a graph that models entities and their respective relations. Entities are represented using nodes, and the relations are

described using arcs or links that connect these nodes [79]. Semantic networks are generally simple and easy to understand and represent knowledge in the natural language. However, since they depend on the creator of the system and are not intelligent, some problems can arise from unclear semantics when applied in automation. In figure 4.2, the links "is-a" and "has" have more than one meaning, which is one of the drawbacks of these networks. Using logical formalism can overcome this issue.

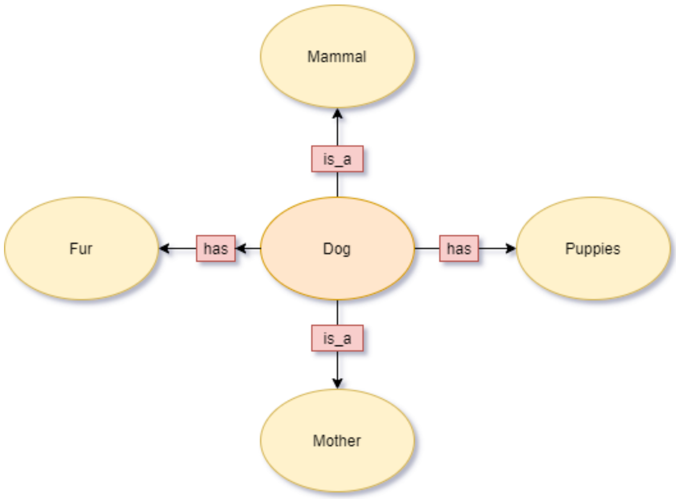

Figure 4.2: Semantic Network example

2. **Production Rules**
   This rule based system is also expressed in the natural language. When conditions are met, the production rule is activated and leads to the execution of an action. This system is mainly used for procedural knowledge comprised of a library of rules [79].
   An example of such a system is this if and $then$ statement:

   $If\ the\ dog\ has\ puppies,\ then\ she\ must\ feed\ them.$

   This rule has $has\ puppies$ as a condition, and if the former is met, then she, (the dog) $must\ feed\ them.$

3. **Frame Representation**
   A frame representation can be seen as a mix between declarative and procedural knowledge. It is mainly used for stereotypical scenarios, which emulates human memory. Frames are used in many applications in AI where the data (entities) is grouped according to attributes [79].

4. **Logical Representation**
   As seen in figure 4.2, some ambiguities exist in the natural language. This can be overcome using logical representations (LR). With LR, conclusions are drawn using rules and conditions [79]. This logic-based system consists of syntax and semantics that assure sound inference. Any sentence in the natural language can be translated into logics using syntax and semantics. Syntax determines what symbols can be used for knowledge representation and how to write them, while semantics describe the rules that allow this knowledge to be interpreted while assigning them meaning. This representation is what helps us perform logical reasoning [79].

For more details, please check the papers [76] and [79].

Our main focus in this chapter is to understand why we chose to use Answer Set Programming, how it works, and what makes it is a powerful problem solving, non-monotonic logical reasoning tool for knowledge representation.

4.1.2 Logical reasoning with Answer Set Programming (ASP)

The framework proposed by [36] is based on ontology and is aimed at automating a peg-and-ring task. Some limitations in their work were evidently due to ontologies, since they are only able to reason with the prior representations of the situations they had. Therefor, ontologies have been shown to be most useful in just aiding the understanding of humans about the situation, as in [80].

On a different note, reasoning on knowledge that is changing or incomplete in a dynamic system, such as one relying on the input from sensors, can offer better flexibility for planning. This can be done with non-monotonic programming [81] and has been shown to be useful in scenarios where safety is critical and where the task itself is difficult, such as surgery. *"Nonmonotonic reasoning was originally motivated by the need to capture in a formal logical system aspects of human* *commonsense reasoning that enable us to withdraw previous conclusions when new information becomes available"* [81]. This is one of the reasons why we chose to use logic programming.

Answer set programming is a logical knowledge representation and reasoning approach. Knowledge is expressively represented as facts, rules and constraints within a code, and the reasoning is executed by answer set solvers (ex: Clingo, Clasp, DLV). ASP allows for commonsense reasoning [82], where failure to declare a knowledge assumes negation, which is one the ways ASP accommodates for non-monotonic reasoning. This form of reasoning has been shown to be useful is safety critic situations such as aerospace [83], autonomous driving [84] and cognitive robotics [85]. Similarly, ASP can be used in another scenario where safety is of utmost importance: robotic surgery.

ASP has been successfully applied in robotic planning where it was able to reasoning on complete as well as incomplete knowledge [86]. This is particularly important for applications where information cannot be fully given in advance, such as in surgery. With incomplete knowledge of the domain, the planning problem becomes more difficult [86]. A solution to that is to have conditional planning, where ASP is required to give a model of how a goal state can be reached, knowing the initial state. It also does that in the presence of sensing actions.

4.1.3 Syntax and Semantics of Answer Set Programming

The rules and facts in ASP have predicates that represent static knowledge. Predicates do not change within the program. Each predicate has an arity (>=0) representing the number of atoms contained within that predicate [87].

An example can be explained simply with the predicate mammal. In $mammal(cat)$ and $mammal(dog)$, where "cat" and "dog" represent the atoms within this predicate. The fact that a cat is a mammal and a dog is a mammal doesn't change and is merely descriptive. Variables inside predicates always start with a capital letter.

Rules In every ASP rule, there exists two parts: a head and a body. A head of a rule is indicated by what is on the left of the "if" statement represented by the symbol $:-$. The body of the rule is represented by the statements on the right of this symbol.

For example:
$move(A, B) :- free(B),\ not\ attached(A),\ A \,!= B.$

For an action *move* to be executed, all the preconditions stated in the body after $:-$ must be true in that given state of knowledge. In this example, we assume that it is sufficient to not know that A is attached and that B is free to be able to move A on B. Here, $!=$ stands for inequality and it is a built-in predicate.

Negation There are two types of negations in ASP: Default (weak) negation and True (strong) negation. Default (or weak) negation is used when dealing with knowledge that is incomplete. Strong negation is used when the statement is known to be false. Weak negation in programs is written as *not* and strong negation is written as $-$.

When there is strong negation in a body of a rule, it means that the head of the rule cannot be executed unless that negated predicate is known to hold.
not a implies that a is not believed to be true, whereas $-a$ means that a is believed to be false.

Facts If the body of a rule is empty,

$move(A, B).$

the action stated always qualifies for execution. An empty body means that body of the rule can be replaced with the "true" statement.

Integrity Constraints If the body head of a rule is empty,

$:-move(A),\ attached(A).$

this means that the conjugation of the literals in the statement cannot be simultaneously true. In the same state, it can't happen that A is moved and attached.

External Atoms
Some atoms may be external, which means they can be defined by sensors, or by manual input.

Rules, Facts and Integrity Constraints are the main constituents in an ASP algorithm. Any knowledge can be represented using these semantics described. The succeeding section will briefly describe ASP solving.

4.1.4 Answer Set Solving

In this thesis, we use Clingo for grounding and solving. Clingo computes answer sets by first grounding the external atoms. This is done by giving true values to the variables. A variable-free atom is also said to be ground. This is then followed by the having the solver check the preconditions within the rules, and see which ones hold. If all the preconditions within the body of a rule hold, then the head of the rule is said to be true. The time step is incremented until the goal is reached. In this thesis, the goal specifies to display just the predicates that are true per time step. In this way, the ASP system gives the Action(s) that should be executed, per time step.

In this thesis, ASP is used for surgical process modeling of a robotic partial nephrectomy. These previous two Chapters 2 and 3 provided the background and fundamentals required to understand how this modeling was done. Using the different granularity levels that can be seen in a surgical procedure, an ASP system is meant to reason on knowledge given by expert surgeons on the surgical technique. The predicates and terminologies within the algorithm must be unambiguous to allow for the proper reasoning and solving by the ASP solver. However, before giving a detailed account of the ASP algorithm created for the surgical process, we must first describe the problems being addressed in this thesis, and why a knowledge-based reasoning approach is used for modeling a surgical procedure.

4.2 Problem Description

In this thesis we describe an application of a logic-based non-monotonic programming language (ASP), in the context of surgical robotics. For that, as an example, we will attempt to model a Robotic Partial Nephrectomy (RPN). The ASP model is generalized, and can be extended to different laparoscopic surgeries.

This laparoscopic surgery consists of executing a certain number of actions, using different instruments held by two robotic arms: PSM1 and PSM2. An Anatomy will be given in real-time to the ASP system by a sensing module, while the instruments to be used in the procedure are assumed to be known. This sensing module is out of the scope of this thesis, and so anatomies will be taken from [11, 10]. ASP will then reason on the next action to be done on the given anatomy, while also providing the phase and step of this action to be performed. A visualization of the "big picture" can be seen in figure 4.3. Surgical procedure knowledge is extracted from [11] and [10] to create the rules and constraints that govern the ASP algorithm. The anatomy is assumed to come from a sensing module, and given as input for the ASP model to general the Surgical Process. In a future work, Deep-Onto neural networks [88] can potentially be combined with this ASP approach for a better surgical workflow recognition system, with real-time task planning. Deep-Onto combines deep learning networks (CNN + LSTM) with ontologies to make high-level predictions on a surgical workflow.

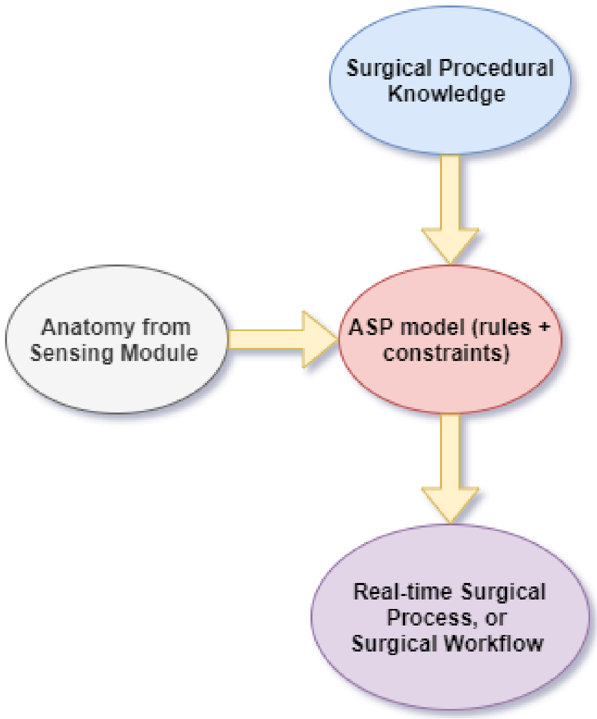

Figure 4.3: Surgical Process Generation by ASP given Anatomy and Knowledge from experts (the sensing module is outside the scope of this thesis).

5 Methodology

We begin with a high-level explanation of the ASP approach to Surgical Process Modeling. This is followed by an illustration of the programming tools used in this thesis. We then describe the surgical procedure that is meant to be modeled. Next, we recount the logic behind the ASP encoding. Lastly, we illustrate how we got a surgical process in real-time, and the feedback loop assuring a dynamic system.

5.1 Overview

The goal of this thesis is to create a generic model of a specific laparoscopic robotic surgery. To model a surgical process, we need to define the type of data available, the knowledge that we assume to be known, and some relationships between entities.

First, we create the ASP logic program using common sense reasoning. Knowledge is represented using a logical programming language (ASP) whereas the reasoning is done using an answer set solver (Clingo). We used Jupyter Notebooks running on a Python kernel for development. We also used popular python libraries such as numpy and pandas. In a surgical context, variability between procedures is very high, so the ASP program is required to reason with incomplete knowledge of the domain to make decisions.

As an input for the logic program, we will use anatomies that are assumed to come from sensors. This input will be incrementally added to the ASP code to simulate a dynamic and real-time surgical procedure. Since performing an action at one time step affects the decision about what can be done at the next time step, the output from the ASP model will be sent back as an additional input for the following action to be executed. Figure 5.1 illustrates the general high-level approach to surgical process modeling using ASP. First the SPM problem is identified, and ASP is used for the logic program. Anatomy is given as input to the

ASP program, and Clingo solves the problem at hand.

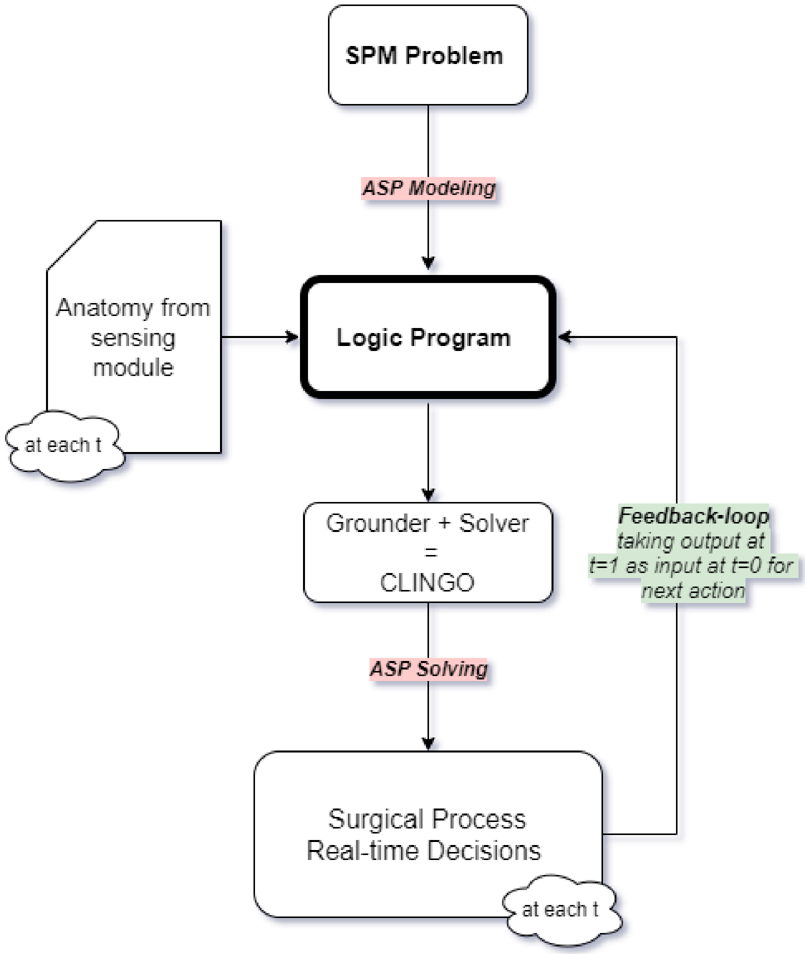

Figure 5.1: High-level architecture of the ASP modeling approach

5.2 The surgical procedure: Robotic Partial Nephrectomy (RPN)

The required surgical task to be performed by ASP is that of a RPN using the da Vinci surgical robot as in [11]. The objective is to perform an **Action** on each

Anatomy given as input from the previous sensing step in order to complete the goal of the surgery: removal of the kidney tumor. For this task to be completed, a multitude of **Surgical Steps** needs to be performed to first reach the kidney and the tumor inside the bowel. There are two da Vinci surgical robotic arms **PSM1** and **PSM2**, replacing the right and left arms of the surgeon respectively. The relationships between **Instruments** and the actors using them are assumed to be known.

The procedure starts after small incisions are made in the patient's abdomen for port placements as seen in figure 5.2.

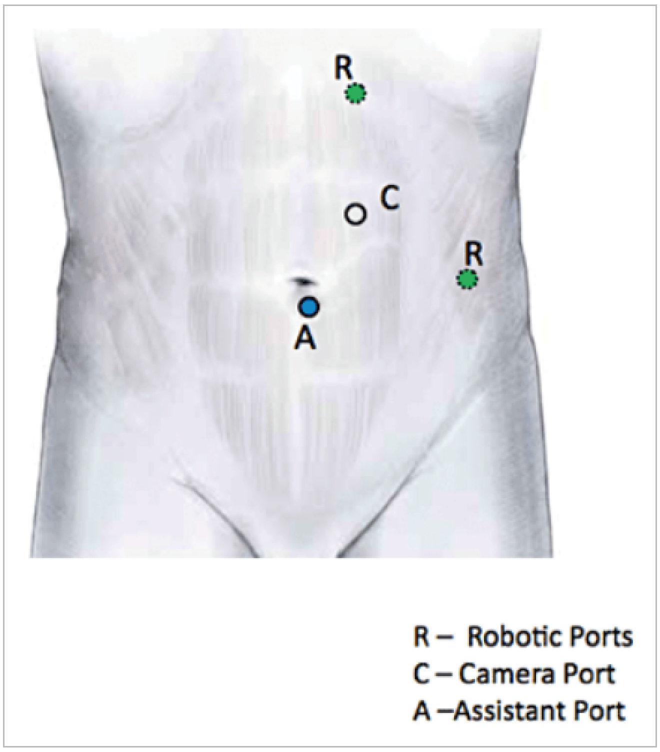

Figure 5.2: Port Placement for Robot-Assisted Partial Nephrectomy [10]

The operative technique for this RPN is modeled using the ones described in [10] and [11]. As per the granularity levels described in 3.1.2, figure 5.3 shows the phases at the lowest granularity level required for the completion of this surgical procedure. We first began by formalizing the procedure into Phases, Steps, Actions and Instruments, using the description of the procedural technique seen in [11].

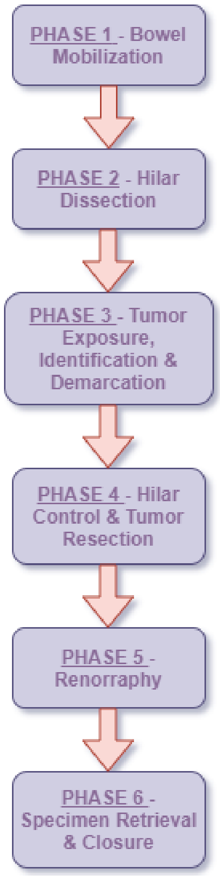

Figure 5.3: RPN Sequential Phases

In figure 5.3, we can see a sequential order of phases, starting from "Bowel Mobilization" to "Closure". The steps within each phase are not necessarily sequential, and different surgeons perform the surgical tasks a little bit differently. This is taken into account within the ASP system by having the AI algorithm reason on what steps should be done given the anatomy and the previous action, rather than merely having a strict sequential workflow.

Not mentioned in figure 5.3 but portrayed within the ASP algorithm are the formalized **Steps, Actors, Actions, Anatomies and Instruments.** In each Surgical Step within a Phase, there is at least one Action done by an Actor using an Instrument on an Anatomy. Each Action performed in a time step affects the choice of the next one. At each time step, we consider some knowledge to be true in order to perform the Action. Such knowledge of the domain that we assume to be known a priori are the relationships between each instrument and respective actor, as well as which instruments are needed for the action to be performed. This knowledge is easily acquired from studying the surgical procedure and understanding the techniques, much like the surgeon's knowledge about the procedure before performing it.

Hilar Dissection cannot be performed unless the bowel is mobilized to expose and reach the kidney. Once the hilar (fascia and surrounding tissue) is dissected, the tumor is exposed, and the margins, size and location of the tumor are identified and demarcated. Once that is done, the hilar must be controlled by clamping the artery and vessels coming and going from the key organ at play (where the tumor is attached) to control bleeding. This is followed by the resection of the tumor along the previously demarcated line. After separation of the tumor from the kidney, renorraphy starts by suturing the open kidney as well as open vessels. After renal reconstruction, the hilar clamps are removed. The specimen is then retrieved in an entrapment sac placed inside the patient by the assistant. Finally a drain is placed and the incisions are closed.

5.3 ASP Encoding

ASP is a non-monotonic reasoning tool in logic programming. This means that the addition of more knowledge to our ASP domain in the form of statements or relations can affect the outcome. ASP solvers can reason and plan the problem at hand with either a large set of knowledge, or an incomplete one [37]. In surgery, we assume that this domain knowledge is incomplete since it is very difficult to predict every single variant inside the patient.

The ASP solving Algorithm 1 describes the general ASP program solving logic,

Algorithm 1 ASP Incremental Solving Algorithm [36]

```
Input: ASP model with rules, constraints and external atoms (1 anatomy from sensors)
Output: Surgical Action, Phase, Step
Ground external atoms
Plan = [], t = 1, Action = null
while not goal do
    if Action ! = null then
        Ground effects of Action
    Check pre-conditions for actions at t
    if action is possible then
        Select Actions possible at t
        Plan.append(Action(t))
    else
        return Unsatisfiable
return Phase, Step, Action
```

similar to the one done by Ginesi et al. in [36]. However, their goal was to generate the plan with the shortest number of time steps possible. In this thesis, the goal is different.

The goal is defined using a simple integrity constraint. It states that the ASP program cannot generate more than one answer set per time step. This is done by limiting the query on the time step t.

In this thesis, the goal defined within the program is to execute one action per time step. This means that, every time the ASP gets a new external atom as input, in this case, anatomy, the ASP provides the solution to what action needs to be done on that anatomy. External atoms are received by the situation awareness module [36], when there is change in the environment. For this surgery, we assume to have sensors that are able to accurately detect an anatomy, and to give this information to the ASP system.

To describe the ASP domain, we need to define the different types of knowledge in the domain. *Statics* are domain attributes that are always true. *Fluents*, on the other hand, are domain attributes that can change over time. *Actions* depend on the relations and rules described within the code. *Terms* can either be variables or object constants. Terms are ground if they have no variables [37]. A predicate is a description or a relation between terms. Terms inside predicates called atoms are *literals*, and the number of atoms within a predicate defines its arity.

Statics include:

- actors: $robotArm(psm1)$, $robotArm(psm2)$ and the assistant
- objects: $object(endobag)$ and $object(drain)$
- instruments: (monopolarCurvedScissors, prograspForceps, roboticNeedleDriver, and carterThomasonDevice)
- tools: (lockingAllisClamp, ultraSoundProbe, bulldogClamp, vicrylSutureSH1, vicrylSutureCT1 and hemolockClip).

Fluents are the descriptive predicates:

- $with(Actor, Instrument)$ relates the instrument used by the actor.
- $usingTool(Actor, Tool)$ relating an actor with the tool being used.

Actions describe the activities performed in each Step, and are characterized by preconditions and effects. The action predicate relates the action done on the anatomy using the specified instrument. Those performed by robotArm(psm1) using $monopolarCurvedScissors$:

- $incise(Anatomy, monopolarCurvedScissors, t)$,
- $dissect(Anatomy, monopolarCurvedScissors, t)$,
- $mark\_margins(Anatomy, monopolarCurvedScissors, t)$,
- $resect(Anatomy, monopolarCurvedScissors, t)$.

Those performed by robotArm(psm1) using $roboticNeedleDriver$ for renorraphy. To better demonstrate this renorraphy phase, figure 5.4 clearly shows the layers of the kidney to be sutured. The action predicate relates the action done on the anatomy using the specified tool:

- $deepLayerClosure(Anatomy, vicrylSutureSH1, t)$,
- $outerLayerClosure(Anatomy, vicrylSutureCT1, t)$.

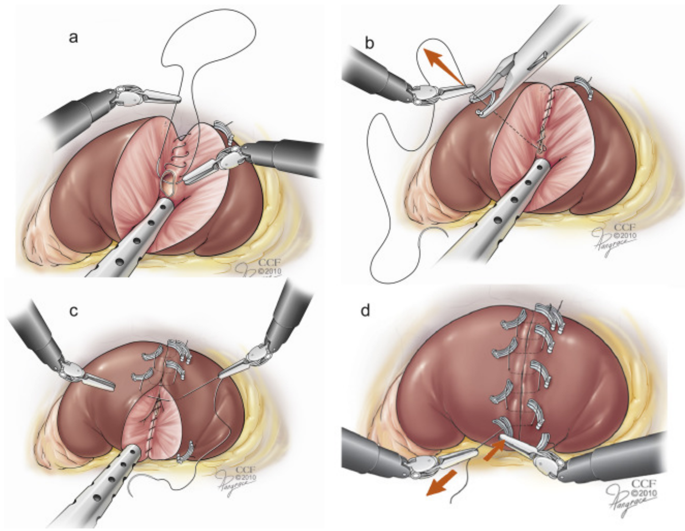

Figure 5.4: Deep and outer layers of the kidney to be sutured in the Renorraphy phase [11]

Those performed by robotArm(psm1) using $carterThomasonDevice$. The action predicate relates the action done on the anatomy using the specified instrument:

- $close(Anatomy, carterThomasonDevice, t)$.

Those performed by robotArm(psm2) using $proGraspForceps$. Similar to the ones mentioned above, these action predicates relate the action done on the anatomy and the instrument:

- $hold(Anatomy, proGraspForceps, t)$,
- $put\_in(Anatomy, Object, proGraspForceps, t)$,
- $unclamp(Anatomy, proGraspForceps, t)$.

Those performed by robotArm(psm2) using $proGraspForceps$. These action predicates relate the anatomy and the tool used to execute the action:

- $retract(Anatomy, lockingAllisClamp, t)$,
- $clamp(Anatomy, bulldogClamp, t)$,
- $detect(Anatomy, ultraSoundProbe, t)$,
- $secure\_suture(Anatomy, hemolockClip, t)$.

Those performed by the assistant. These action predicate are a little different than the ones mentioned above. Not all the actions done by the assistant are modeled in this thesis. However, the following ones are crucial for the logic of the program. These decribe the insertion or placement of objects inside the patient, through the assistant port:

- $insert(Object, inPatient, assistantPort, t)$,
- $place(Object, inPatient, assistantPort, t)$.

Now, in order to understand the preconditions for these action predicates, the anatomies in the abdomen need to be categorized into different types: solid organ, tissue or fascia, organ with attached tumor (key organ), arteries and veins, and tumor. In this way, the ASP model can reason on what type of actions can be executed on the anatomies according to their types. For example, the ASP model knows that one should not clamp the tissue or fascia, but only arteries or veins.
For that, we have specific predicates indicating what type of action is allowed to be executed on an anatomy:

- $canBeDissected(Anatomy)$
- $canbeRetracted(Anatomy)$
- $canBeIncised(Anatomy)$
- $canBeClamped(Anatomy)$
- $canBeResected(Anatomy)$
- $canBeRetrieved(Anatomy)$

Other predicates decribe the different layers of the anatomy that is expected to be dealt with in the procedure:

- $deepLayer(Anatomy)$

- $outLayer(Anatomy)$

This knowledge stems from common sense reasoning and gives the ASP its reasoning power, and hence, intelligence. This also allows the ASP program to be expanded onto other laparoscopic surgeries since the rules are general enough to be applied for different procedures.

In order to understand what ASP uses for reasoning, an example of an action predicate along with its preconditions and effects will be examined.

To execute the following action predicate:
$mark\_margins(Anatomy, monopolarCurvedScissors, t)$,
the preconditions within the body of the rule must be true first:
$detect(Anatomy, ultraSoundProbe, t-1)$,
$with(psm1, monopolarCurvedScissors)$.

This means that to mark the margins on the tumor (given here as a variable) using the monopolar curved scissors, the instrument needs to be with the robot arm PSM1, and this anatomy needs to have been detected by an ultrasound probe in the previous time step. However, not all surgeons use an ultrasound probe before marking the tumor margins. For this reason, the ASP system should allow for such flexibility in the surgical procedure. This can be seen in a different rule, where marking the margins on the tumor is allowed even if it has not been detected with an ultrasound probe. This rule will have different preconditions than the ones stated in this example.

The action predicate *detect* has its own set of preconditions that must be true before it can be executed. In this thesis, the use of instruments and tools is assumed to be known. In a later stage of the work, the system will be programmed to give information on when the instruments or tools need to be changed. For now, since it assumed to be always true, $usingTool(psm2, ultraSoundProbe)$ is true if stated at the beginning of the program in the *base* section of the ASP program. If it is not stated, then the ASP model will generate a process without this ultrasound detection step.

Furthermore, the anatomy must be reached before it can be detected. In this ASP model, it is required that the renal capsule be dissected for the tumor to be reached. It is also possible to dissect the gerota's fascia covering the organ to reach the tumor.

To execute the following action predicate:
$reached(Anatomy)$,
the precondition
$dissect(renalCapsule, monopolarCurvedScissors, t-1)$
or $dissect(gerotasFascia, monopolarCurvedScissors, t-1)$
needs to have been executed at the previous time step. Once the renal capsule or gerota's fascia is dissected, the anatomy is reached and can be detected, and then the margins can me marked.

Flexibility in executing the steps need to be demonstrated since different surgeons can choose to perform the procedure differently. Restrictions on what cannot be allowed are stated, but the ASP must also allow for more than one workflow possibility. As we will see in the succeeding section, there are two possible way to visualize the output from the ASP model. We can either have real time task generation, or an entire surgical workflow.

5.4 Surgical Workflow and Real-time Output

Before describing the types of surgical plans the ASP can generate, we must be briefly reminded of the knowledge it understands. For an Action to be performed at each time step (t), ASP takes in the Anatomy from the sensors as input in the form of an external atom, along with information on the previous action performed, and the instruments and tools being used in the whole procedure.

The output from the ASP model gives each **Action** to be executed on the given **Anatomy**, along with the current **Phase** and **Step**. Another program takes this output from ASP, and sets t to 0. In this way, every output from the model shows the previous action at $t = 0$ and the next one at $t = 1$. In some cases, there can be more than one action happening at the same time step. However, this is only the case for different actors. Every actor (robot arm) can execute one single action at every time step t. For example, while PSM1 is incising, it can happen that PSM2 is holding that same anatomy. However, PSM1 cannot be incising and dissecting an anatomy at the same time step t.

As portrayed in Algorithm 2, the system incrementally adds an anatomy as input for the ASP program. The user can also manually add an anatomy and the previous action at t=0, to get the next action with phase and step. This is useful for real-time task generation of the surgical procedure. This is shown in 6.1.1.

Algorithm 2 ASP solving with incremental addition of previous action to input

```
Input: A list of anatomies
Returns: Full surgical workflow
function GET_SURGICAL_WORKFLOW(list_of_anatomies)
    previous_actions = ”, workflow = []
    for anatomy in list_of_anatomies do
        output = get_asp_output(anatomy, previous_actions)
                ▷ Output contains phase, step, actions, instruments and anatomy.
        workflow = update_workflow(workflow, output)
        previous_actions = output[actions]
    return workflow
```

To generate a full surgical workflow given a list of anatomies, the system uses common sense rules from surgeon's description of the tasks. Another program takes the ASP output containing information on the phase, step, action, and instruments on that given anatomy, and stores it. To get the next output, the latter extracts the information on the previous action executed, along with the new anatomy from the list. This is repeated until it reaches the end on the anatomy list. In this way, the full surgical workflow can be achieved before beginning the surgical procedure.

For real time surgical reasoning in a scenario where a sensor is directly communicating with the ASP system, the information on the anatomy it detects affects the output. Ideally, the ASP takes this anatomy, and decides what can be done on it, depending on what the previous action was. To do that, the systems takes in this information on the current anatomy and previous action, and generates the action that should be executed on this given anatomy, as well as the information on the phase, step, and instruments.

The next chapter will give examples of such outputs. The results of the ASP system will be evaluated quantitatively. For that, a comparison between actual steps from real surgeries will be compared to those predicted by the ASP system. In addition, a qualitative analysis will be done with the help of experts in the field or urology and robotic surgery.

6 Results and Validation

In a first part of this chapter, the implementation of this logic program in a surgical setting is described, as well as the generated surgical process. Next, two types of validations are done for this work. First, a quantitative analysis using real surgical video annotations is performed. And second, experts in the field or urology and minimally invasive robotic urological surgery test out the program to give qualitative feedback.

6.1 Surgical Process Modeling with ASP

In order to design the ASP code that can model a surgical process in real time, a study on the surgical process was done. Since this thesis only deals with the ASP reasoning part of surgical autonomy, we acquired the possible anatomies to be expected from articles and textbooks on the surgical technique [11, 10].

In this thesis, we assume to have a list of possible anatomies, and an idea of what the sensing module will see. The order of the Anatomies is not completely random, since for instance, the kidney is not reached unless the organs and tissues that surround it are retracted and incised, respectively. From the formalized procedural technique explained in these papers [11, 10], the order of Phases are the same, but the order of Steps, and hence Actions within these Steps, can slightly differ. Just like in a real surgery, different surgeons can perform the same procedure a little differently, but main events such as Phases, are inevitably sequential since they give meaning to the procedure itself.

The methodology reported in this thesis can be used twofold:

- First, one anatomy at a time is given as input to the ASP system in real time. This is assumed to be coming from a sensing module. The program then gives an output for the action to be executed at t=1, on that given anatomy. The output generated gives information on the Phase and Step, as well as the Action performed on the Anatomy and the Instrument used.

This application can be useful for monitoring purposes, but it will not be evaluated within this research.

- The second method is to give the ASP system a list of anatomies. The order of the anatomy list is to be expected in real surgery, and it is taken from [11, 10]. This is more useful for planning, and it is a more general case of the first method described. This will be validated in section 6.2.

6.1.1 Real-time Surgical Process Reasoning

As described, a first report of the results of the model uses one anatomy at a time, similar to what we can expect in a surgery in real time. A sensor would distinguish an anatomy, and send it to the model as input. When that is done, the model generates the action to be executed, and saves it as *previous_action*. The ASP solver Clingo takes from 0.100 to 0.150 seconds to generate each answer set. This action is then sent back as an input for the next action to be executed. For every output, the model displays the previous action (at t=0) and the next one (at t=1) to be executed on the current anatomy. Some actions have a prerequisite action as one of its pre-conditions, which means that a certain action needs to be executed first before it can perform the next.
For example, to resect the tumor, a series of actions need to have been already done before jumping onto this essential one. For more details on the logic behind the program, please refer to section 5.3.

For every output, the model displays the previous Phase, Step, Action, Anatomy and Instrument at t=0, and the current ones at t=1. Some examples of outputs will be presented in this section.

In the bowel mobilization phase, some connective tissues need to be incised before we can reach the key organ with a tumor, in this case, the kidney. The order of these incisions are not discrete, and depending on the anatomy seen by the sensor, the robot arm should perform the action on what it is currently seeing. For example, as seen in Figures 6.1, 6.2, and 6.3, *previous_action* is empty because the user does not need to input a previous action in order for the next one to be executed. The logic behind this stems from the fact that if this anatomy is seen by the sensor, there are no preconditions of previous action that needs to be activated before this incision action becomes true. The robot arm can either start by incising $gerotasFascia$, or $lineofToldt$, or $posteriorMesocolon$, depending on what the system receives from the sensor. The following figures bellow illustrate the real time surgical process generation by the ASP system. The model takes the anatomy as input, assumed to come from sensors, and generates the action to be executed on it.

```
ANATOMY FROM SENSORS:
currentAnatomy(gerotasFascia).
Previous action:
NEXT STEP:
phase(bowelMobilization,1).
step(medialMobilization,1).
hold(gerotasFascia,proGraspForceps,1).
incise(gerotasFascia,monopolarCurvedScissors,1).
```

Figure 6.1: Incising gerota's fascia does not depend on a previous action

```
ANATOMY FROM SENSORS:
currentAnatomy(lineofToldt).
Previous action:
NEXT STEP:
phase(bowelMobilization,1).
step(medialMobilization,1).
hold(lineofToldt,proGraspForceps,1).
incise(lineofToldt,monopolarCurvedScissors,1).
```

Figure 6.2: Incising line of Toldt does not depend on a previous action

```
ANATOMY FROM SENSORS:
currentAnatomy(posteriorMesocolon).
Previous action:
NEXT STEP:
phase(bowelMobilization,1).
step(medialMobilization,1).
hold(posteriorMesocolon,proGraspForceps,1).
incise(posteriorMesocolon,monopolarCurvedScissors,1).
```

Figure 6.3: Incising posterior mesocolon does not depend on a previous action

However, this is not the case for most of the action predicates in the Answer Set Program. In figure 6.4, the system allows for hilar dissection without having an action predicate in *previous_action*. This case is different than the one mentioned above for incisions because the system assumes that the hilar vessels are reached since they are given as input from the sensor. Once the sensor sees the hilar vessels, it is presumed that the bowel mobilization phase is over since the sensors would not have been able to see this anatomy if that wasn't the case.

```
ANATOMY FROM SENSORS:
currentAnatomy(hilarVessels).
Previous action:
NEXT STEP:
phase(hilarDissection,1).
step(hilarVesselsDissection,1).
dissect(hilarVessels,monopolarCurvedScissors,1).
```

Figure 6.4: Anatomy is presumed to be reached

A third type of action predicate reasoning is when it is necessary that a previous action be executed for the next one to be true. Figures 6.5 and 6.6 are some examples of such a case. This means that, an action predicate has as one of its preconditions, *previous_action*, that is, another action predicate at t=0. When the sensor reports *currentAnatomy*(*tumor*), the system cannot reason on what needs to be done next without being sure that the anatomy *renalCapsule* has been dissected beforehand. However, as mentioned before, this is just an example of this action predicate, and resection of the tumor can be executed given different preconditions than the ones mentioned here.

```
ANATOMY FROM SENSORS:
currentAnatomy(tumor).
Previous action: dissect(renalCapsule,monopolarCurvedScissors,0 ).
NEXT STEP:
phase(tumorExposure_Demarcation,1).
step(tumorIdentification,1).
detect(tumor,ultraSoundProbe,1).
```

Figure 6.5: Actions that depend on the the previous action: Detecting the tumor after dissection of renal capsule

In Figure 6.5 , $dissect(renalCapsule, monopolarCurvedScissors, 0)$ has to be true before $detect(tumor, ultraSoundProbe, 1)$ can be executed. After detecting the tumor at t=0, the sensor still sees the tumor, and sends this information as input for the next action to be executed. The next predicate action then suggests the next action to be $mark\_margins(tumor, monopolarCuvedScissors, 1)$ since the latter has the action predicate $detect(tumor, ultraSoundProbe, 0)$ as a precondition.

```
ANATOMY FROM SENSORS:
currentAnatomy(tumor).
Previous action: clamp(renalVein,bulldogClamp,0).
NEXT STEP:
phase(hilarControl_tumorResection,1).
step(tumorResection,1).
resect(tumor,monopolarCurvedScissors,1).
```

Figure 6.6: Actions that depend on the the previous action: Resecting tumor after clamping renal vein.

Similarly, the system will not allow for the tumor to be resected (at t=1), unless the renal vein has been clamped in the previous step at t=0, as seen in figure 6.6. In this figure, it can also be understood that clamping the artery and vein is essential before resecting the tumor. In [10], it is specified that the artery must be clamped before the vein. For this reason, the action predicate $clamp(renalArtery, bulldogClamp, 0)$ is a precondition for clamping the vein when it is given as input for the program by the sensors.

Now that we have presented one of the ways to use the ASP system, we will discuss another surgical process generation method in the succeeding section.

6.1.2 Surgical Workflow generation

An alternative way to show the results is by giving the model a list of anatomies. The order of anatomies is one that would be expected in a real procedure. An example of such a list is depicted in table 6.1. For the model to generate a plan for the whole procedure, an additional program takes the information of the executed action, and then stores this information to be used for the next action. In this way, the model does an incremental solving to execute the next action. The model automatically gets all the information about the previous action, including phase and step, as well as the current anatomy given from the list.

Anatomies to be expected on the Right	Anatomies to be expected on the Left
attachment to liver	attachment to spleen
liver	spleen
line of Toldt	line of Toldt
gerota's fascia	gerota's fascia
posterior mesocolon	posterior mesocolon
psoas muscle	psoas muscle
hilar vessels	hilar vessels
gerota's fascia	gerota's fascia
renal capsule	renal capsule
tumor	tumor
tumor	tumor
renal artery	renal artery
renal vein	renal vein
tumor	tumor
kidney	kidney
kidney	kidney
renal artery	renal artery
renal vein	renal vein
tumor	tumor
gerota's fascia	gerota's fascia
kidney	kidney

Table 6.1: List of anatomies to be expected as input from sensors

For every anatomy in the list, the model generates the surgical Phase, Step, as well as which Instrument is used to execute the specific Action on the given Anatomy. Figure 6.7 is an example of the full workflow that can be generated by the model. This table shows a sequential workflow that is dependent on the anatomies given. This model represents a potential workflow for a partial nephrectomy done on the right side of the patient.

	Phase	Step	Instruments	Action	Anatomy
0	bowelMobilization	liverRetraction	[monopolarCurvedScissors]	[dissect]	[attachmentToLiver]
1	bowelMobilization	liverRetraction	[lockingAllisClamp]	[retract]	[liver]
2	bowelMobilization	medialMobilization	[proGraspForceps, monopolarCurvedScissors]	[hold, incise]	[lineOfToldt, lineOfToldt]
3	bowelMobilization	medialMobilization	[proGraspForceps, monopolarCurvedScissors]	[hold, incise]	[gerotasFascia, gerotasFascia]
4	bowelMobilization	medialMobilization	[proGraspForceps, monopolarCurvedScissors]	[hold, incise]	[posteriorMesocolon, posteriorMesocolon]
5	bowelMobilization	medialMobilization	[proGraspForceps, monopolarCurvedScissors]	[hold, incise]	[psoasMuscle, psoasMuscle]
6	hilarDissection	hilarVesselsDissection	[monopolarCurvedScissors]	[dissect]	[hilarVessels]
7	tumorExposure_Demarcation	tumorExposure	[monopolarCurvedScissors]	[dissect]	[gerotasFascia]
8	tumorExposure_Demarcation	tumorExposure	[monopolarCurvedScissors]	[dissect]	[renalCapsule]
9	tumorExposure_Demarcation	tumorIdentification	[ultraSoundProbe]	[detect]	[tumor]
10	tumorExposure_Demarcation	markingTumorMargins	[monopolarCurvedScissors]	[mark_margins]	[tumor]
11	hilarControl_tumorResection	hilarClamping	[bulldogClamp]	[clamp]	[renalArtery]
12	hilarControl_tumorResection	hilarClamping	[bulldogClamp]	[clamp]	[renalVein]
13	hilarControl_tumorResection	tumorResection	[monopolarCurvedScissors]	[resect]	[tumor]
14	renorraphy	suturingExcisionBed	[vicrylSutureSH1, hemolockClip]	[deepLayerClosure, secure_suture]	[kidney, kidney]
15	renorraphy	suturingRenalCapsule	[vicrylSutureCT1, hemolockClip]	[outerLayerClosure, secure_suture]	[kidney, kidney]
16	renorraphy	hilarUnclamping	[proGraspForceps]	[unclamp]	[renalVein]
17	renorraphy	hilarUnclamping	[proGraspForceps]	[unclamp]	[renalArtery]
18	renorraphy	inspectingRenorraphy	[1)]	[inspectSutures]	[kidney]
19	retrieval_closure	retrievingTumor	[endobag]	[put_in]	[tumor]
20	retrieval_closure	reconstruction	[roboticNeedleDriver]	[suture]	[gerotasFascia]
21	retrieval_closure	drainage	[inPatient]	[place]	[drain]

Figure 6.7: Surgical workflow generation with ASP (right side of patient)

	Phase	Step	Instruments	Action	Anatomy
0	bowelMobilization	medialMobilization	[monopolarCurvedScissors]	[dissect]	[attachmentToSpleen]
1	bowelMobilization	medialMobilization	[proGraspForceps, monopolarCurvedScissors]	[hold, incise]	[lineOfToldt, lineOfToldt]
2	bowelMobilization	medialMobilization	[proGraspForceps, monopolarCurvedScissors]	[hold, incise]	[gerotasFascia, gerotasFascia]
3	bowelMobilization	medialMobilization	[proGraspForceps, monopolarCurvedScissors]	[hold, incise]	[posteriorMesocolon, posteriorMesocolon]
4	bowelMobilization	medialMobilization	[proGraspForceps, monopolarCurvedScissors]	[hold, incise]	[psoasMuscle, psoasMuscle]
5	hilarDissection	hilarVesselsDissection	[monopolarCurvedScissors]	[dissect]	[hilarVessels]
6	tumorExposure_Demarcation	tumorExposure	[monopolarCurvedScissors]	[dissect]	[gerotasFascia]
7	tumorExposure_Demarcation	tumorExposure	[monopolarCurvedScissors]	[dissect]	[renalCapsule]
8	tumorExposure_Demarcation	tumorIdentification	[ultraSoundProbe]	[detect]	[tumor]
9	tumorExposure_Demarcation	markingTumorMargins	[monopolarCurvedScissors]	[mark_margins]	[tumor]
10	hilarControl_tumorResection	hilarClamping	[bulldogClamp]	[clamp]	[renalArtery]
11	hilarControl_tumorResection	hilarClamping	[bulldogClamp]	[clamp]	[renalVein]
12	hilarControl_tumorResection	tumorResection	[monopolarCurvedScissors]	[resect]	[tumor]
13	renorraphy	suturingExcisionBed	[vicrylSutureSH1, hemolockClip]	[deepLayerClosure, secure_suture]	[kidney, kidney]
14	renorraphy	suturingRenalCapsule	[vicrylSutureCT1, hemolockClip]	[outerLayerClosure, secure_suture]	[kidney, kidney]
15	renorraphy	hilarUnclamping	[proGraspForceps]	[unclamp]	[renalVein]
16	renorraphy	hilarUnclamping	[proGraspForceps]	[unclamp]	[renalArtery]
17	renorraphy	inspectingRenorraphy	[1)]	[inspectSutures]	[kidney]
18	retrieval_closure	retrievingTumor	[endobag]	[put_in]	[tumor]
19	retrieval_closure	reconstruction	[roboticNeedleDriver]	[suture]	[gerotasFascia]
20	retrieval_closure	drainage	[inPatient]	[place]	[drain]

Figure 6.8: Surgical workflow generation with ASP (left side of patient)

However, the model can also generate the surgical procedure done on the left side of the patient. Since the output depends on the anatomies it receives from the sensors, ie the list it is given as input, it will not execute the actions on the anatomies that are exclusively on the right side, such as the liver. This can be seen in figure 6.8.

The ASP model can also understand how to perform the procedure, when given a different list of anatomies. In figure 6.9, the tumor is retrieved earlier than is seen in the previous two workflows. This choice is possible when the endobag is inserted, and the tumor is identified by the sensing module. Additionally, if the ultrasound probe is not used, the ASP system allows for the margins to be marked without the tumor identification step.

	Phase	Step	Instruments	Action	Anatomy
0	bowelMobilization	medialMobilization	[proGraspForceps, monopolarCurvedScissors]	[hold, incise]	[lineOfToldt, lineOfToldt]
1	bowelMobilization	medialMobilization	[proGraspForceps, monopolarCurvedScissors]	[hold, incise]	[gerotasFascia, gerotasFascia]
2	hilarDissection	hilarVesselsDissection	[monopolarCurvedScissors]	[dissect]	[hilarVessels]
3	tumorExposure_Demarcation	tumorExposure	[monopolarCurvedScissors]	[dissect]	[gerotasFascia]
4	tumorExposure_Demarcation	markingTumorMargins	[monopolarCurvedScissors]	[mark_margins]	[tumor]
5	hilarControl_tumorResection	hilarClamping	[bulldogClamp]	[clamp]	[renalArtery]
6	hilarControl_tumorResection	hilarClamping	[bulldogClamp]	[clamp]	[renalVein]
7	hilarControl_tumorResection	tumorResection	[monopolarCurvedScissors]	[resect]	[tumor]
8	retrieval_closure	retrievingTumor	[endobag]	[put_in]	[tumor]
9	renorraphy	suturingExcisionBed	[vicrylSutureSH1, hemolockClip]	[deepLayerClosure, secure_suture]	[kidney, kidney]
10	renorraphy	suturingRenalCapsule	[vicrylSutureCT1, hemolockClip]	[outerLayerClosure, secure_suture]	[kidney, kidney]
11	renorraphy	hilarUnclamping	[proGraspForceps]	[unclamp]	[renalVein]
12	renorraphy	hilarUnclamping	[proGraspForceps]	[unclamp]	[renalArtery]
13	renorraphy	inspectingRenorraphy	[1)]	[inspectSutures]	[kidney]
14	retrieval_closure	reconstruction	[roboticNeedleDriver]	[suture]	[gerotasFascia]
15	retrieval_closure	drainage	[inPatient]	[place]	[drain]

Figure 6.9: Surgical workflow generation with ASP without ultrasound detection, and with an earlier tumor retrieval

In these examples seen in figures 6.7, 6.8, and 6.9, every row indicates a time step. For each anatomy given by the sensing module, Action, Instrument, Step and Phase are generated. Before moving on to the evaluation of the model, let us first discuss some of the challenges faced when modeling such a surgical procedure.

6.1.3 Technical Challenges

Modeling a surgical procedure comes with many challenges. First, current methods find difficulty in modeling both right and left sided procedures simultaneously. With the ASP system, external atoms are provided by the sensing module, and so the algorithm will not predict an action to be executed on an anatomy that is exclusively on one side of the patient, such as the liver or the spleen.

Before modeling such a complex procedure, care must be taken when making technical choices. The technique of the surgery depends on how to get the best exposure of the kidney to reach the tumor. This will affect how or in what way the bowel is mobilized at the beginning of the procedure. This depends on the size, location, and complexity of the tumor. Consequently, there are many ways to perform a surgery. This makes the SPM process for a robotic partial nephrectomy more difficult. For this reason, the ASP algorithm allows for some flexibility in performing the procedure. This is achieved by having the possibility of generating the surgical procedure non-sequentially. However, the model is generalized enough to only include the information that is to be expected in a standardized approach. In a later stage of this work, the Renal score as well as the location and size of the tumor will be taken into account to allow for a more detailed and realistic surgical plan.

Moreover, in some cases, adhesions can be found after port insertion. This can affect where the ports are placed. Adhesions must be released to allow for greater freedom of the robotic arms. For this purpose, the model of the procedure assumes to begin after port incisions are made and robotic docking. In this way, the robotic arms are considered to be at the correct location and are ready to start the dissection.

Another aspect that must be considered is the choice of hilar control. Some surgeons prefer to clamp both the renal artery and the renal vein, while others only clamp the artery. In a standardized approach, as seen in [11] and [10], they are both clamped. Some studies revealed that there are no differences in surgical outcomes between only clamping the artery and clamping both artery and vein [89, 90]. In this model, we are choosing to clamp both the artery and vein, which provides a more complete and generalized approach.

As previously mentioned, the Renal score affects the surgical technique. If the tumor is endophytic, an enucleation would be required. However, in this generalized model of the procedure, only resection is suggested, which simplifies the technique. In a later stage of this work, the Renal score, hence the complexity of the tumor, will be taken into account, and it will be able to suggest an alternative approach to dissecting the tumor.

In this thesis, many aspects of the procedure were simplified allowing for a more generalized and high level approach of a robotic partial nephrectomy.

In the next section, we will evaluate the results of the ASP model both quantitatively and qualitatively.

6.2 Quantitative Analysis with real Surgical Procedures

For a quantitative evaluation of the results achieved by the ASP model, real surgical video annotations are used. A comparison between actual and predicted surgical steps is made, as well as actual and predicted actions. First, the surgical data was manipulated in order to be used in this validation.

6.2.1 Surgical Video Annotations

In order to evaluate the results obtained by the ASP system in generating a surgical plan, real surgical procedures were used. Videos of robotic partial nephrectomies were manually annotated by a urologist and compiled by Dr. Hirenkumar Nakawala. In these files, information was given at every time step, with the duration of each Phase, Step and Actions done on each Anatomy seen. This data was organized according to different granularity levels, providing information on phases, steps, actions, anatomies, as well as all instruments used. Therein, different instruments used by the surgeon's right hand, left hand, and assistant were distinguished.

For this data to be useful, the annotations had to be cleaned up to be made comparable to the type of outputs we achieved with the ASP system. Consequently, all instruments used by different actors in these annotations are merged into one "Instrument" attribute. Repetitions of actions and instruments within a step are also eliminated to clarify the output. Furthermore, in these video annotations, certain terminologies are different from the ones the ASP uses or generates. For instance, as seen earlier in this chapter, the renal artery must be clamped before the tumor can be resected. In the video annotations, a distinction between renal artery and renal vein is not made. In the annotations, it is said that the renal artery is clamped, but there is no mention of it later being unclamped in the procedure. Similarly, hilar vessels are said to be unclamped when there was no previous mention of them being clamped in any of the previous steps. The surgeon who annotated these surgeries used vague terminologies that the ASP system might find too ambiguous. For this reason, terminologies that are different from the ones within the ASP program are modified, providing that they convey the same meaning. Terminologies for the Phases, Steps, Anatomies and Actions that were different from the ones used in [11, 10], follow these mapping tables 6.2, 6.3, 6.4, 6.5.

Terminology from Annotations	Terminology in ASP from [11, 10]
$hilumDissection$	$hilarDissection$
$tumorExposure$	$tumorExposure\_Demarcation$
$tumorResection$	$hilarControl\_tumorResection$
$closure$	$retrieval\_closure$

Table 6.2: Phase Mapping

Terminology from Annotations	Terminology in ASP from [11, 10]
$mobilization$	$medialMobilization$
$dissection$	$hilarVesselsDissection$
$identification$	$tumorExposure$
$ultrasound$	$tumorIdentification$
$marking$	$markingTumorMargins$
$clamping$	$hilarClamping$
$resection$	$tumorResection$
$removal$	$retrievingTumor$
$midollarSuturing$	$suturingExcisionBed$
$corticalSuturing$	$suturingRenalCapsule$
$unclamping$	$hilarUnclamping$

Table 6.3: Step Mapping

Terminology from Annotations	Terminology in ASP from [11, 10]
incise, *cut*, *dissect*	*incise*, *dissect*
mark	*mark_margins*
put	*put_in*
suture	*deepLayerClosure*
suture	*outerLayerClosure*
suture	*secure_suture*
suck	*suction*

Table 6.4: Action Mapping

Terminology from Annotations	Terminology in ASP from [11, 10]
ligamentBetweenSpleenAndKidney	*attachmentToSpleen*
ligamentBetweenLiverAndKidney	*attachmentToLiver*
Mesocolon	*posteriorMesocolon*
kidneyCapsule	*renalCapsule*
renalArtery	*renalArtery*
hilarVessels	*renalArtery*
hilarVessels	*renalArtery*

Table 6.5: Anatomy Mapping

After cleaning the annotations and adjusting the terminologies, only three surgeries were complete and usable for this validation. In the next section, we will explain how this validation was done, as well as the accuracy calculated for predicting steps and actions by the ASP system. A confusion matrix for the predicted actions is presented, as well as the variability in precision, recall and F1-score for each action, across the three different surgeries used for this evaluation.

6.2.2 Predicting Surgical Steps

To begin evaluating the efficacy of the ASP system in generating a surgical process, an algorithm takes the anatomy from the annotations and inputs it to the ASP system. The ASP algorithm then uses this anatomy to generate a predicted step. Once that is done, the action executed is stored, and later used with the next anatomy from the annotations list to predict the next step. This is repeated until the algorithm goes through the entire list of anatomies from the annotations. To better illustrate this validation, let us refer to the results seen in the figures bellow. In these figures, the numbering column on the left side reflects the time step t. At each t, an Anatomy is extracted from the annotations, and given to the ASP system as input, along with the previous action executed at t-1. The column "ActualStep" are the Steps seen in the real surgery, while the column "PredictedStep" represent the Steps predicted by the ASP system. For every "Anatomy" at each time step, an "ActualStep" is compared with a "PredictedStep".

	Anatomy	ActualStep	PredictedStep
0	['gerotasFascia', 'lineOfToldt', 'bowel']	medialMobilization	[medialMobilization, medialMobilization, medialMobilization]
1	['hilarVessels']	hilarVesselsDissection	[hilarVesselsDissection]
2	['gerotasFascia']	tumorExposure	[tumorExposure]
3	['tumor']	tumorIdentification	[tumorIdentification]
4	['tumor']	markingTumorMargins	[markingTumorMargins]
5	['renalArtery', 'renalVein']	hilarClamping	[hilarClamping, hilarClamping]
6	['tumor']	tumorResection	[tumorResection]
7	['tumor']	retrievingTumor	[retrievingTumor]
8	['kidney']	suturingExcisionBed	[suturingExcisionBed]
10	['kidney']	suturingRenalCapsule	[suturingRenalCapsule]
11	['renalArtery', 'renalVein']	hilarUnclamping	[hilarUnclamping, hilarUnclamping]
12	['kidney']	inspectingRenorraphy	[inspectingRenorraphy]
13	['gerotasFascia']	reconstruction	[reconstruction]
14	['kidney']	drainage	[drainage]

Figure 6.10: Surgery 1 - Actual vs Predicted Steps

In figures 6.10 and 6.11, we can observe that the predicted steps of Surgery 1 and Surgery 2, perfectly match the actual steps in the annotations. The fact that some "PredictedStep" is mentioned more than once is due to the number anatomies it is given. In these figures, at some time steps, more than one anatomy can be seen, when only one was needed to make the prediction. Since the ASP system takes in one anatomy per time step, it predicts the Step for each Anatomy. That is why, in some cases, the "PredictedStep" is mentioned three times, if three anatomies were seen in the annotations, as we can see at time step 0 in figure 6.10. This however does not affect the comparison to be made with the "ActualStep". In these two examples seen in figures 6.10 and 6.11, the order of anatomies are different which can be expected in different surgeries. And yet, the ASP system was still able to correctly predicted every step in the surgery. This demonstrates the flexibility and robustness of the system in predicting surgical steps.

	Anatomy	ActualStep	PredictedStep
0	[lineOfToldt, gerotasFascia, posteriorMesocolon, attachmentToSpleen]	medialMobilization	[medialMobilization, medialMobilization, medialMobilization, medialMobilization]
1	[hilarVessels]	hilarVesselsDissection	[hilarVesselsDissection]
2	[gerotasFascia]	tumorExposure	[tumorExposure]
3	['tumor']	tumorIdentification	[tumorIdentification]
4	['tumor']	markingTumorMargins	[markingTumorMargins]
5	[renalArtery, renalVein]	hilarClamping	[hilarClamping, hilarClamping]
6	['tumor']	tumorResection	[tumorResection]
7	['kidney']	suturingExcisionBed	[suturingExcisionBed]
8	['kidney']	suturingRenalCapsule	[suturingRenalCapsule]
9	[renalArtery, renalVein]	hilarUnclamping	[hilarUnclamping, hilarUnclamping]
10	['kidney']	inspectingRenorraphy	[inspectingRenorraphy]
11	['tumor']	retrievingTumor	[retrievingTumor]
12	[gerotasFascia]	reconstruction	[reconstruction]
13	['kidney']	drainage	[drainage]

Figure 6.11: Surgery 2 - Actual vs Predicted Steps

In figure 6.12, the tumor is not detected with an ultrasound since this step is not essential for the correct completion of the procedure. ASP was able to skip the detection since the use of the ultrasound instrument was deactivated at the beginning of the model. When the anatomy given is the *tumor*, it correctly predicted to mark the margins right away. This shows the flexibility of ASP's in executing the procedure.

After calculating the individual accuracy for each of the three surgeries, the overall accuracy was found to be 100%. The ASP algorithm achieved high accuracy in predicting surgical steps. This is due to the fact that it uses logic reasoning to make decisions. When the ASP system receives a new anatomy as input, and with

	Anatomy	ActualStep	PredictedStep
0	['gerotasFascia']	medialMobilization	[medialMobilization]
1	['hilarVessels']	hilarVesselsDissection	[hilarVesselsDissection]
2	[gerotasFascia]	tumorExposure	[tumorExposure]
3	[tumor]	markingTumorMargins	[markingTumorMargins]
4	[renalArtery, renalVein]	hilarClamping	[hilarClamping, hilarClamping]
5	[tumor]	tumorResection	[tumorResection]
6	['kidney']	suturingExcisionBed	[suturingExcisionBed]
7	['kidney']	suturingRenalCapsule	[suturingRenalCapsule]
8	[renalArtery, renalVein]	hilarUnclamping	[hilarUnclamping, hilarUnclamping]
9	['kidney']	inspectingRenorraphy	[inspectingRenorraphy]

Figure 6.12: Surgery 3 - Actual vs Predicted Steps

its knowledge on the previous action executed in the previous step, it uses the rules and constraints within its algorithm to decide what action should be done next. The logic behind its reasoning capabilities allows for a more flexible process generation by the AI. ASP integrates the understanding that there is more than one way to execute a surgical procedure, and it is able to generate a surgical process given anatomies. In the following section, predicted actions will be evaluated in the same manner.

6.2.3 Predicting Actions

Similar to the experiment described in the previous section, actions predicted by the ASP model are compared with actual ones. The model extracts the anatomy from the annotations, and uses it to decide which action to be executed. It later saves this action to be used in the reasoning of the next one. The same three surgical procedures are used. An important matter to consider is that fact that in the annotations, only the actions done by the dominant hand are displayed. However, the ASP model gives information on both robot arms, and for this reason, some differences can be seen.

	Anatomy	ActualAction	PredictedAction
0	['gerotasFascia']	['incise']	[hold, incise]
1	['hilarVessels']	['dissect']	[dissect]
2	[gerotasFascia]	['dissect']	[dissect]
3	[tumor]	['mark_margins']	[mark_margins]
4	[renalArtery, renalVein]	['clamp']	[clamp, clamp]
5	[tumor]	['resect']	[resect]
6	['kidney']	['deepLayerClosure']	[deepLayerClosure, secure_suture]
7	['kidney']	['outerLayerClosure']	[outerLayerClosure, secure_suture]
8	[renalArtery, renalVein]	['unclamp']	[unclamp, unclamp]
9	['kidney']	['inspectSutures']	[inspectSutures]

Figure 6.13: Comparison between Actual and Predicted Actions for Surgery 3

Figure 6.13 shows a case where the ASP model was able to perfectly predict the actions to be executed, and yielded an accuracy of 100%. This is due to the flexibility of the ASP model in understanding what actions should be executed on the given anatomy, knowing what has been done in the previous step. In this example, detection with an ultrasound is not done, and the ASP system was able to correctly go from the dissection to marking the margins on the tumor when it is received as input. However, as can be seen in some times steps, the ASP model predicted two actions to be done, compared to only one seen in the annotations. This is due to the fact that the ASP model takes into account the actions done by both robotic arms, whereas the annotations are only displaying what is being done by the surgeon's dominant hand. For this reason, when the PredictedAction is $hold, incise$ and the ActualAction is $incise$, it is still considered correct. This applies the rest of the actions predicted by the model.

Figure 6.14 shows the results achieved by the ASP model in predicting actions of Surgery 1. Anatomies were given to the model in the order seen in the video annotations. The ASP was able to correctly predict 93.75% of the actions to be executed in the procedure. The error was due to the fact that the model did not anticipate inspecting the gerota's fascia after suturing it, and before placing the drain, as can be seen in time step 14. This is one of the limitations with the ASP algorithm. Since it used the knowledge extracted from techniques that always

	Anatomy	ActualAction	PredictedAction
0	['lineOfToldt']	incise	[hold, incise]
1	['bowel']	move	[move]
2	['gerotasFascia']	incise	[hold, incise]
3	['hilarVessels']	dissect	[dissect]
4	['gerotasFascia']	dissect	[dissect]
5	['tumor']	detect	[detect]
6	['tumor']	mark_margins	[mark_margins]
7	['renalArtery', 'renalVein']	clamp	[clamp, clamp]
8	['tumor']	resect	[resect]
9	['tumor']	put_in	[put_in]
10	['kidney']	deepLayerClosure	[deepLayerClosure, secure_suture]
11	['kidney']	outerLayerClosure	[outerLayerClosure, secure_suture]
12	['renalArtery', 'renalVein']	unclamp	[unclamp, unclamp]
13	['gerotasFascia']	suture	[suture]
14	['gerotasFascia']	inspect	[inspect, suture, place]
15	['kidney']	place	[place]

Figure 6.14: Comparison between Actual and Predicted Actions for Surgery 1 shows some of the limitations

presumed to inspect the sutures on the kidney, when it came to inspecting the gerota's fasica, it could not make a correct prediction, as seen at time step 14 in figure 6.14. It attempted to inspect the anatomy given, but also to suture it as well as to place it, which is incorrect.

Figures 6.15 and 6.16 show one of the important factors that affect the process generation by the ASP system. These two experiments have only one difference in the order of anatomies. In 6.15, the anatomies seen at time step 11, after deep layer closure of the kidney, are the renal artery and vein. However, since the ASP model understands that after deep layer closure, the outer layer of the kidney must also be sewn, it does not allow for the completion of the procedure, although it correctly predicted all the previous actions. The accuracy of the ASP system in predicting actions when the order of anatomies are not to be expected is of 68.75%.

	Anatomy	ActualAction	PredictedAction
0	['lineOfToldt']	incise	[hold, incise]
1	['posteriorMesocolon']	incise	[hold, incise]
2	['attachmentToSpleen']	dissect	[dissect]
3	['gerotasFascia']	dissect	[hold, incise]
4	['hilarVessels']	dissect	[dissect]
5	['gerotasFascia']	dissect	[dissect]
6	['tumor']	detect	[detect]
7	['renalArtery', 'renalVein']	mark_margins	[mark_margins, clamp]
8	['renalArtery']	clamp	[clamp]
9	['tumor']	resect	[resect]
10	['kidney']	deepLayerClosure	[deepLayerClosure, secure_suture]
11	['renalArtery', 'renalVein']	unclamp	[]
12	['kidney']	outerLayerClosure	[]
13	['tumor']	put_in	[]
14	['gerotasFascia']	suture	[hold, incise]
15	['kidney']	place	[]

Figure 6.15: Comparison between Actual and Predicted Actions for Surgery 2 shows that the order of the anatomies matter for the flow of actions executed

	Anatomy	ActualAction	PredictedAction
0	[lineOfToldt, gerotasFascia, posteriorMesocolon, attachmentToSpleen]	['incise', 'incise', 'incise', 'dissect']	[hold, incise, hold, incise, hold, incise, dissect]
1	[hilarVessels]	['dissect']	[dissect]
2	[gerotasFascia]	['dissect']	[dissect]
3	['tumor']	['detect']	[detect]
4	['tumor']	['mark_margins']	[mark_margins]
5	[renalArtery, renalVein]	['clamp']	[clamp, clamp]
6	['tumor']	['resect']	[resect]
7	['kidney']	['deepLayerClosure']	[deepLayerClosure, secure_suture]
8	['kidney']	['outerLayerClosure']	[outerLayerClosure, secure_suture]
9	[renalArtery, renalVein]	['unclamp']	[unclamp, unclamp]
10	['kidney']	['inspectSutures']	[inspectSutures]
11	['tumor']	['put_in']	[put_in]
12	[gerotasFascia]	['suture']	[suture]
13	['kidney']	['place']	[place]

Figure 6.16: Comparison between Actual and Predicted Actions for Surgery 2

In 6.16, if the order is adjusted, the model allows for the outer layer closure of the kidney, and then proceeds by correctly predicting all succeeding actions, with an accuracy of 100%.

The results of this experiment showed the robustness of the ASP system, and how it would not allow for some safety critical actions to be executed, if the previous one needed to be done first. It allows for some flexibility in executing the steps and actions for non essential activities. However, some actions are highly dependent on the previous ones. An example is shown in 6.15 where the order of anatomies taken as input is paramount. This makes sense in the flow of the procedure. The ASP model was right in stopping the completion of the surgery since the outer layer of the kidney must be sewn before unclamping the vein and artery. This assures the correctness of the surgical technique, as well as the safety of the patient.

6.2.4 Results of Quantitative Validation

To better understand the results achieved by the ASP model in predicting actions, accuracy, precision, recall and F1-score were calculated, and can be seen in table 6.6. This is done by looking at the differences between Actual and Predicted Actions, and noting the true positives, true negatives, false positives and false negatives. These were calculated from figures 6.13, 6.14, 6.15 and 6.16.

Precision, also know as positive predictive value, is defined as true positives divided by the sum of true positives and false positives. It answers the question of how many of the predicted items are relevant. Recall is defined as true positives divided by the sum of true positives and false negatives. It shows how many of the relevant items are selected. Recall expresses how much of the true positives were correctly identified. F1-score is a combination of the two metrics. For example, we predicted dissect correctly 10 out of 11 times. Hence, all 10 predictions are relevant and the precision is equal to 1, but not all relevant instances were selected, hence the recall is 0.91.

The overall accuracy of the system in predicting actions was found to be 90.6%. This can be defined as the degree of correctness of the ASP system in predicting actions when given anatomies. This is mostly due to the dependence of the ASP model on the order of anatomies. Some actions cannot be executed unless others were shown to be done at a previous step. This assures safety for the patient, as well as making sure the procedure makes sense.

Action	**Precision**	**Recall**	**F1-score**
dissect	1	0.91	0.953
incise	0.83	1	0.907
mark_margins	1	1	1
clamp	0.8	1	0.889
resect	1	1	1
deepLayerClosure	1	1	1
outerLayerClosure	1	0.75	0.857
unclamp	1	0.75	0.857
inspectSutures	1	1	1
place	1	0.6667	0.8
put_in	1	0.6667	0.8
suture	1	0.6667	0.8
move	1	1	1
detect	1	1	1

Table 6.6: Precision, recall and f1-score for each action.

Finally, when looking at both surgical steps and actions, 95.3% of the predictions were correct. While considering only predicted actions, the model had a mean precision of 0.97 with a standard deviation of 0.065, a mean recall of 0.89 with a standard deviation of 0.143, and a mean F1-score of 0.92 with a standard deviation of 0.081, computed over all the annotations. This means that 97% of the predicted actions were relevant, while 89% of these relevant predictions were correct.

This validates the model, and shows the feasibility in using ASP for such an application.

To better visualize the performance of our algorithm, a confusion matrix is presented in figure 6.17. This shows the errors that were made by the system in predicting actions.

predictedActions

actualAction	incise	dissect	move	detect	mark_margins	clamp	resect	deepLayerClosure	outerLayerClosure	unclamp	inspectSutures	put_in	place	suture	emtpy
incise	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
dissect	1	10	0	0	0	0	0	0	0	0	0	0	0	0	0
move	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
detect	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
mark_margins	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
clamp	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
resect	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0
deepLayerClosure	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
outerLayerClosure	0	0	0	0	0	0	0	0	3	0	0	0	0	0	1
unclamp	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1
inspectSutures	0	0	0	0	0	0	0	0	0	0	3	0	1	1	1
place	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1
put_in	0	0	0	0	0	0	0	0	0	0	0		2	0	1
suture	1	0	0	0	0	0	0	0	0	0	0	0	0	2	0
empty	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6.17: Confusion Matrix

In the subsequent section, a qualitative analysis will be conducted. This will be done with the help of four urologists who have previously performed this procedure, and who have experience with minimally invasive robotic surgery.

6.3 Qualitative Analysis - Expert Review

In this section, four urologist who are experts in robotic surgery were interviewed. Since they have all previously performed robotic partial nephrectomies, their professional inputs one the procedure was of crucial importance in the improvement of the system. The video recordings of each talk can be presented with this thesis upon request.

6.3.1 Dr.med. Alexander Heinze

Dr. Alexander Heinze is a staff urologist at the Marienkrankenhaus, one of the largest denominational hospitals in northern Germany. After completing his general surgery residency at Centro Medico ABC in Mexico, he moved to Belgium where he received a fellowship in Urological Robotic Surgery. He then received his promotion to Dr.Med. in Hamburg Germany at Martini Klinik. Dr. Heinze specializes in minimally invasive treatments as well as robot-assisted surgeries, specifically for the treatment of oncological diseases.

Being interested in innovations in Medical technologies, Dr. Heinze was kind enough to help conduct a qualitative analysis of this project.

After briefly describing the logic behind the ASP algorithm and what type of knowledge the systems uses to reason, the model generated by the ASP was presented. First, a full workflow of the surgery was described. The results were shown to him in a clear table format as seen in figures 6.7 and 6.8. A second method of using this project, the real-time process generation, was then presented.

Dr. Heinze first started by studying the workflow of the generated model. After analysing it, he reported that the procedure in fact "follows the steps of a real surgery, and it is correctly predicting what needs to be done at each step, with the correct instruments". He also added that he believes that "this project could have real applications, and that it has a lot of potential." In addition, he believes it to be particularly useful for educational purposes, such as for "young surgeons who are not yet experts in performing this procedure." The model could act as a virtual checklist for the best way to achieve the greatest outcome. Another way that this project could be used in practice is with the rise of new surgical robots that will be using AI for decision making. He mentioned that he is currently working with companies in Germany that are working on introducing these types of innovations, and that when they come to the market next year, they could potentially find this project very useful.

However, being an expert in his field, he had some ideas for future work and some features that can be added to make this model more complete for practical purposes. For instance, he believes it would be useful to have the system indicate when an instrument should be changed. This is one of the limitations of this work, and we will work on adding this feature at a later stage. On another note, he mentioned the importance of accounting for intraoperative complications in the proposed surgical system. In real surgery, "one needs to be prepared to deal with complications, and it would be useful if the system can make decisions in case of high stress or bleeding." For instance, the model should react to bleeding by postponing the next dissection scheduled, and performing stitches to stop the bleeding. Another potential case would be if we reach the end of suturing the kidney and bleeding is still detected, the system should be able to suggest performing more stitches, or putting some material on the renal capsule that helps with healing, such as hemostatics.

These suggestions were very useful in understanding how to improve this system, and they will be taken into account at a later stage of this work. Dr. Heinze is involved in innovation technologies at a university in Mexico, and he proposed to add this project to his next talk because he believes it can help inspire others.

6.3.2 Dr.med. Hector Sandoval Barba

Dr.Hector Sandoval Barba is a urologist trained in Mexico and the UK. He is currently a consultant in Mexico, performing open, laparoscopic and robotic surgeries. Dr.Barba has performed robotic partial nephrectomies numerous times, but is currently specialized in oncology. His input was of extreme importance for the future progress of this project. His suggestions will be taken into account in a later work to improve this algorithm of the current system, and to make it more applicable in real surgery.

After understanding what type of information the ASP algorithm is working with, Dr.Barba suggested that it might be too simple to be applied in real surgery. There are numerous aspects that need to be considered before choosing how to approach the problem at hand. First, before docking the surgical robot, adhesion inside the patient might have to be released. The current ASP model assumes to begin the procedure after the incisions are made and the robotic arms are in place. He mentioned that before this step, port insertions can depend on these scar tissues inside the patient. Since the start of the surgery will depend on the adhesions found in the patient, Adhesiolysis should be considered before robotic docking.

Next, he mentioned that the model should consider the different techniques that can be used to release the ligaments that hold the kidney in place. This is one of the technical challenges faced in this project. For this reason, the ASP model simply used a standard technique that was found to be most effective. However, in a later phase of the study, the location of the tumor will have to be taken into account. This will affect how the dissection will begin in the procedure. The technique would be different if the tumor was located in the posterior or anterior part of the kidney. Dr.Barba added this statement: "for instance, if we are dealing with a posterior tumor, the ligaments would be released laterally. However, if it was an anterior tumor, they would have to be released medially." For this reason, it would very important to consider the location of the tumor for modeling this procedure since it would affect how the kidney will be dissected in order to reach the tumor.

Additionally, the techniques to partial nephrectomy mainly depend on the surgeon's criteria. Some choose to clamp the artery, others prefer increasing the pressure on the peritoneum. Alternately, some surgeons selectively clamp only the artery that is feeding the tumor. Since the ASP model is attempting to replace the surgeon down the line, it will need to take all of these into account.

On a different note, dissimilar to what has been done in this thesis so far, he believes that the "renal vein should not be clamped since it will create more pressure in the kidney, which will cause more bleeding, and hence less visibility". However, in [90] and [89], it was was shown that there is actually no difference in the outcome of the procedure if the renal artery was clamped alone, or if both artery and vein were clamped. This emphasizes on how surgeons have different preferences in how to perform the procedure, which is one of the main challenges of this thesis.

Another aspect of the procedure that needs to be considered is whether to perform tumor resection or enucleation. This choice depends on the complexity of the tumor. If the tumor is endophytic, which means it grew inwardly, it would not be easy to delineate its margins and one would have to completely remove the mass without cutting into it with the help of an ultrasound probe. For this reason, classifying the complexity of the tumor is of crucial importance since it would change the way the procedure is performed or planned. In a later stage of this work, the complexity of the tumor will be assessed using renal score [91].

Finally, Dr. Barba concluded our talk by saying that "this is a very ambitious project, but obviously one has to build a stone upon another stone". Even though there are many improvements that need to be done for this to be applicable in real operations , it is probably "the future of certain surgeries". He continued by adding that he would like to continue helping this project move forward.

6.3.3 Dr.med. Alejandro Cumming

Dr. Alejandro Cumming works at one of the only private hospitals in Mexico that have the da Vinci Surgical Robot. With his experience in robotic surgery, he found that having a system that is able to generate the surgical process autonomously would be useful.

Similar to Dr. Barba, he also finds that it is very important to consider the Renal score when attempting to model the procedure. It is an essential instrument that helps categorize the tumor, and to see how complex the procedure could get.

After inspecting one of the workflows generated by the ASP system, he was pleased with the model being able to distinguish between being on the left or the right side of the patient. The model simply reasons on the anatomies it is given as input, therefor it would not try to execute an action that is observed exclusively on one side. However, this is one of the limitations of this project. The ASP system assumes the anatomy it is given as input is seen one at a time, and that each anatomy is correctly identified.

Finally, Dr. Cumming believes that this system would be most useful for complex tumors, and with its high accuracy, and given that robotic arms provide great dexterity, the procedure can assure the highest preservation of nephrons in the kidney when removing the tumor.

6.3.4 Dr.med. Karen Mendoza

Dr. Karen Mendoza is a urologist trained in robotic and minimally invasive surgery in Mexico. After evaluating the workflow generated by the ASP system, she approved of its completeness. Dr.Karen declared that it correctly illustrates the sequence of events "to be expected in a real procedure."

However, she was concerned about whether the size and location of the tumor were taken into account by the ASP model since that would affect the procedural technique. She also suggested that it would be useful to consider that patient's risk of complications. If the patient has had a previous surgery, or has been treated with radiotherapy, this might affect the performance of the surgery. She believes it would be beneficial to know in advance whether complications are to be expected or not. Moreover, as previously mentioned, the ASP system in a later stage should be able to account for complications during surgery in real time, which can alleviate her concerns.

Finally, Dr. Karen was very impressed with the work involved in this project, and added that "it is a very innovated way to see the procedure." She is looking forward to seeing what the project leads to, and how it will benefit the decision making process in surgery.

The next chapter will conclude this thesis, and give an outlook about the future work.

7 Conclusion

For a fully autonomous surgical robot, three main aspects of a procedure need to be addressed: scene understanding and situation awareness, explainable and safe generation of the surgical plan, and the execution of the surgical task by dexterous robotic arms. In this thesis, we will only limit ourselves to the portion of the procedure that requires reasoning, specifically the surgical process. Just as in real surgery, environmental changes can be expected. For that, the system that is generating the surgical process must be able to adapt to these dynamic conditions. This was achieved using a declarative rule-based non-monotonic programming language, ASP.

The surgical plan generated by the ASP system remains rather high level. The procedure is assumed to begin after the incisions are made and surgical robotic docking. The ASP algorithm uses knowledge given by experts about the procedure, as well as information on the environment. This AI algorithm can generate two types of surgical outputs: a full workflow of the procedure, or real-time reasoning about the surgical process. To get a full workflow, a list of anatomies can be given to the ASP system. It then uses logic, with the knowledge it is given a priori, to decide on what action should be executed on that given anatomy. The generated procedure is explainable, and can be easily modified.

This thesis was successful in generating accurate surgical processes for a robotic partial nephrectomy. When given information on different anatomies, as well as the previous action, the ASP algorithm correctly generated the Action that should be executed on that given Anatomy, as well as providing the Phase, Step, and Instrument used. Real surgical video annotations of RPN were used to evaluate the ASP model. It successfully predicted the steps and actions in the procedure, with an accuracy of 95.3%. Actions were predicted with a mean precision of 97%, a mean recall of 89%, and a mean F1-score of 92%.

At present, some of the knowledge that ASP assumes to be always true are the instruments used by the robot arms. However, during the procedure, the robot arms change instruments. This is currently not modeled by the ASP system. In a later phase of this work, the system will be able to indicate when the instruments need to be changed before executing the next action. This will be done by adding actions that indicate the insertion of the new instrument to be used for the next step. This change in instruments mainly occurs before the renorraphy phase, so it can be integrated within the algorithm using this information.

Another limitation is when faced with complications. In real surgery, a surgeon must always be ready to deal with possible difficulties. This will be added to the system in future work. For instance, if the sensors detect bleeding, the system should indicate to stop the dissection and to do some stitches where the bleeding is coming from. Only once the bleeding has stopped can the dissection be continued. Another potential occurrence could happen after reaching the end of the kidney suturing step. If bleeding is detected, the system needs to be able to suggest adding more stitches, or to apply some material on the organ that supports healing. This will be done by adding a supplementary action that has *bleeding* within its preconditions. This *bleeding* literal will act as an external atom activated by the sensing module.

After speaking with Dr.Hector Barba who has performed this procedure numerous times, several matters were brought to light. Firstly, the technique and extent of mobilizing the bowel depends on the location of the tumor within the kidney. Before beginning the surgery, a surgeon must know the RENAL Score [91] in order to decide how to perform the procedure. This will help in deciding how much medial mobilization should be carried out, and choosing which parts of the tissues to incise first. Before being able to apply such a technology in real surgery, information about the renal mass must be considered and integrated within the ASP system. The algorithm will have to integrate this information and allow for different possible plan generation depending on the RENAL score.

This project was found to be useful and promising by Dr.Alexander Heinze. He believes that it can be used by young surgeons or trainees for the time being. They can use the full surgical workflow as a virtual checklist when performing the surgery, or to help them make decision during the surgery when uncertain about what should be done next. In the future, this work can be combined with Deep-Onto [88] for a more complete and accurate surgical workflow generation. It can also be integrated with the da Vinci surgical robot with the goal of increasing the level of autonomy in robotic surgery.

List of Acronyms

CNN	Convoluted Neural Networks
LSTM	Long Short-Term Memory
SSI	Surgical Site Infection
SPM	Surgical Process Modeling
RPN	Robot Partial Nephrectomy
ML	Machine Learning
ASP	Answer Set Programming
NIRF	Near-Infrared Fluorescent
ILP	Inductive Logic Programming
UML	Unified Modeling Language
OR	Operating Room
OTS	Optical Tracking System
ETS	Electromargnetic Tracking System
KR	Knowledge Representation
FSM	Finite State Machines

Bibliography

[1] M. Yip and N. Das, "Robot autonomy for surgery- the encyclopedia of medical robotics," *Minimally invasive surgical robotics*, 2017.

[2] G.-Z. Yang, J. Cambias, K. Cleary, E. Daimler, J. Drake, P. E. Dupont, N. Hata, P. Kazanzides, S. Martel, and R. V. Patel, "Medical robotics regulatory, ethical, and legal considerations for increasing levels of autonomy," *Science Robotics*, 2017.

[3] S. S. Panesar and K. Ashkan, "Surgery in space," *British Journal of Surgery*, 2018.

[4] S. Panesar, Y. Cagle, D. Chander, J. Morey, J. Fernandez-Miranda, and M. Kliot, "Artificial intelligence and the future of surgical robotics," *Annals of Surgery*, 2019.

[5] D.-L. Chow, R. C. Jackson, M. C. Cavusoglu, and W. Newman, "A novel vision guided knot-tying method for autonomous robotic surgery," *IEEE International Conference on Automation Science and Engineering*, 2014.

[6] C. Coulson, R. Taylor, M. G. A. Reid, D. Proops, and P. Brett, "An autonomous surgical robot for drilling a cochleostomy: preliminary porcine trial," *Clinical Otolaryngology*, 2008.

[7] M. I. Tiwana, S. J. Redmond, and N. H. Lovell, "A review of tactile sensing technologies with applications in biomedical engineering," *Sensors and Actuators A: Physical*, 2012.

[8] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. W. Kim, "Supervised autonomous robotic soft tissue surgery," *Science Translational Medicine*, vol. 8, no. 337, pp. 337ra64–337ra64, 2016.

[9] G. Fagogenis, M. Mencattelli, Z. Machaidze, B. Rosa, K. Price, F. Wu, V. Weixler, M. Saeed, J. E. Mayer, and P. E. Dupont, "Autonomous robotic

intracardiac catheter navigation using haptic vision," *Science Robotics*, vol. 4, no. 29, 2019.

[10] S. Sukumar and C. G. Rogers, "Robotic partial nephrectomy: surgical technique," *BJU International*, 2011.

[11] J. H. Kaouk, A. Khalifeh, S. Hillyer, G.-P. Haber, R. J. Stein, and R. Autorino, "Robot-assisted laparoscopic partial nephrectomy: Step-by-step contemporary technique and surgical outcomes at a single high-volume institution," *European Association of Urology*, 2012.

[12] S. Darzi and Y. Munz, "The impact of minimally invasive surgical techniques," *Annu Rev Med*, 2004.

[13] K. E. Sands, D. S. Yokoe, D. C. Hooper, J. L. Tully, T. C. Horan, R. P. Gaynes, S. L. Solomon, and R. Platt, "Detection of postoperative surgical-site infections: comparison of health plan–based surveillance with hospital-based programs.," *Infect Control Hosp Epidemiol.*, 2003.

[14] "National nosocomial infections surveillance (nnis) report, data summary from october 1986-april 1996," *Am J Infect Control*, 1996.

[15] R. D. Daneman N, Lu H, "Discharge after discharge: predicting surgical site infections after patients leave hospital.," *J Hosp Infect.*, 2010.

[16] G. Gandaglia, K. Ghani, A. Sood, J. Meyers, J. Sammon, M. Schmid, B. Varda, A. Briganti, F. Montorsi, M. Sun, M. Menon, and Q. T. A.S. Kibel, "Effect of minimally invasive surgery on the risk for surgical site infections results from the national surgical quality improvement program (nsqip) database," *JAMA Surg.*, 2014.

[17] J. A. R. de Vermandois, G. Cochetti, M. D. Zingaro, A. Santoro, M. Panciarola, A. Boni, M. Marsico, G. Gaudio, A. Paladini, P. Guiggi, R. Cirocchi, and E. Mearini, "Evaluation of surgical site infection in minimally invasive urological surgery," *Open medicine (Warsaw, Poland)*, 2019.

[18] R. Aimaq, G. Akopian, and H. Kaufman, "Surgical site infection rates in laparoscopic versus open colorectal surgery.," *Am Surg*, 2011.

[19] J. Scalici, B. Laughlin, M. Finan, B. Wang, and R. Rocconi, "The trend towards minimally invasive surgery (mis) for endometrial cancer: an acs-nsqip evaluation of surgical outcomes.," *Gynecol Oncol*, 2015.

[20] F. Corcione, C. Esposito, D. Cuccurullo, A. Settembre, N. Miranda, F. Amato, F. Pirozzi, and P. Caiazzo, “Advantages and limits of robot-assisted laparoscopic surgery: preliminary experience.,” *Surgical Endoscopy and Other Interventional Techniques*, 2005.

[21] C. Loukas and E. Georgiou, “Surgical workflow analysis with gaussian mixture multivariate autoregressive (gmmar) models: a simulation study.,” *Computer Aided Surgery*, 2013.

[22] T. Vidovszky, W. Smith, J. Ghosh, and M. Ali, “Robotic cholecystectomy: learning curve, advantages, and limitations.,” *Journal of Surgical Research*, 2006.

[23] M. T. Thai, P. T. Phan, T. T. Hoang, S. Wong, N. H. Lovell, and T. N. Do, “Advanced intelligent systems for surgical robotics,” *Advanced Intelligent Systems*, 2020.

[24] B. Chughtai, D. Scherr, J. D. Pizzo, M. Herman, C. Barbieri, J. Mao, A. Isaacs, R. Lee, A. E. Te, S. A. Kaplan, P. Schlegel, and A. Sedrakyan, “National trends and cost of minimally invasive surgery in urology,” *Urology Practice*, 2015.

[25] M. Makary and M. Daniel., “Medical error: The third leading cause of death in the u.s.,” *BMJ*, 2016.

[26] J. Anderson and K. Abrahamson, “Your health care may kill you: Medical errors.,” *Stud Health Technol Inform.*, 2017.

[27] P. Kazanzides, Z. Chen, A. Deguet, G. Fischer, R. Taylor, and S. DiMaio, “An open-source research kit for the da vinci surgical system,,” *IEEE International Conference on Robotics and Automation*, 2014.

[28] J. H. Palep, “Robotic assisted minimally invasive surgery,” *Journal of minimal access surgery*, vol. 5, 2009.

[29] K. D. Chang, A. A. Raheem, and K. H. Rha, “Novel robotic systems and future directions,” *Indian Journal of Urology*, 2018.

[30] M. Hwang, D. Seita, B. Thananjeyan, J. Ichnowski, S. Paradis, D. Fer, T. Low, and K. Goldberg, “Applying depth-sensing to automated surgical manipulation with a da vinci robot,” 2020.

[31] A. M. Derossis, G. M. Fried, M. Abrahamowicz, H. H. Sigman, J. S. Barkun, and J. L. Meakins, "Development of a model for training and evaluation of laparoscopic skills," *The American Journal of Surgery*, 1998.

[32] G. M. Fried, L. S. Feldman, M. C. Vassiliou, S. A. Fraser, D. Stanbridge, G. Ghitulescu, and C. G. Andrew, "Proving the value of simulation in laparoscopic surgery," *Annals of Surgery*, 2004.

[33] J. D. Opfermann, S. Leonard, R. S. Decker, N. A. Uebele, C. E. Bayne, A. S. Joshi, and A. Krieger, "Semi-autonomous electrosurgery for tumor resection using a multi-degree of freedom electrosurgical tool and visual servoing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3653–3660, 2017.

[34] G. P. Moustris, S. C. Hiridis, K. Deliparaschos, and K. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature," *The international journal of medical robotics and computer assisted surgery*, 2011.

[35] D. B. Camarillo, T. M. Krummel, and J. K. S. Jr, "Robotic technology in surgery: past, present, and future," *The American Journal of Surgery*, 2004.

[36] M. Ginesi, D. Meli, and A. Roberti, "Autonomous task planning and situation awareness in robotic surgery april 2020," *Autonomous Robotic Surgery*, 2020.

[37] D. Meli, P. Fiorini, and M. Sridharan, "Towards inductive learning of surgical task knowledge: a preliminary case study of the peg transfer task," *Procedia Computer Science*, 2020.

[38] A. Olivares-Alarcos, D. Beßler, A. Khamis, P. Gonçalves, M. Habib, J. Bermejo, M. Barreto, M. Diab, J. Rosell, J. Quintas, J. Olszewska, H. Nakawala, E. Pignaton de Freitas, A. Gyrard, S. Borgo, G. Alenyà, M. Beetz, and H. Li, "A review and comparison of ontology-based approaches to robot autonomy," *The Knowledge Engineering Review*, vol. 34, 12 2019.

[39] D. Katic, C. Julliard, A.-L. Wekerle, H. Kenngott, B. P. Müller-Stich, R. Dillmann, S. Speidel, P. Jannin, and B. Gibaud, "Lapontospm: an ontology for laparoscopic surgeries and its application to surgical phase recognition," *International Journal of Computer Assisted Radiology and Surgery volume*, 2015.

[40] T. Blum, N. Padoy, H. Feussner, and N. Navab, "Workflow mining for visualization and analysis of surgeries," *Int J Comput Assist Radiol Surg*, 2008.

[41] L. Bouarfa, P. Jonker, and J. Dankelman, "Discovery of high-level tasks in the operating room," *J Biomed Inform*, 2011.

[42] A. Ahmadi, T. Sielhorst, R. Stauder, M. Horn, H. Feussner, and N. Navab, "Recovery of surgical workflow without explicit models," *Med Image Comput Comput Assist Interv*, 2007.

[43] C. Reiley, H. Lin, B. Varadarajan, B. Vagolgyi, S. Khudanpur, D. Yuh, and G. Hager, "Automatic recognition of surgical motions using statistical modeling for capturing variability.," *Stud Health Technol Inform*, 2008.

[44] R. Stauder, A. Okur, L. Peter, A. Schneider, M. Kranzfelder, and N. N. H. Feussner, "Random forests for phase detection in surgical workflow analysis," *The 5th international conference on information processing in computer-assisted interventions (IPCAI)*, 2014.

[45] F. Lalys, L. Riffaud, X. Morandi, and P. Jannin, "Surgical phases detection from microscope videos by combining svm and hmm.," *Med Comput Vis Recognit Tech Appl Med Imaging*, 2011.

[46] N. Padoy, T. Blum, S. Ahmadi, H. Feussner, M. Berger, and N. Navab, "Statistical modeling and recognition of surgical workflow.," *Med Image Anal*, 2010.

[47] P. Jannin and X. Morandi, "Surgical models for computer-assisted neurosurgery.," *Neuroimage*, 2007.

[48] O. Burgert, T. Neumuth, F. Lempp, R. Mudunuri, J. Meixensberger, s. A. D. G. Strauss, P. Jannin, and H. Lemke, "Linking top-level ontologies and surgical workflows," *Int J Comput Assist Radiol Surg*, 2006.

[49] T. Neumuth, B. Kaschek, D. Neumuth, M. Ceschia, J. Meixensberger, G. Strauss, and O. Burgert, "An observation support system with an adaptive ontology-driven user interface for the modeling of complex behaviors during surgical interventions.," *Behav Res Methods*, 2010.

[50] D. Neumutha, F. Loebe, H. Herre, and T. Neumuth, "Modeling surgical processes: A four-level translational approach," *Artificial Intelligence in Medicine*, 2010.

[51] L. Mackenzie, J. Ibbotson, C. Cao, and A. Lomax, "Hierarchical decomposition of laparoscopic surgery: a human factors approach to investigating the operating room environment,"

[52] M. Gholinejad, A. J. Loeve, and J. Dankelman, "Surgical process modelling strategies: which method to choose for determining workflow?," *Minimally-Invasive Therapy Allied Technologies*, 2019.

[53] F. Lalys and P. Jannin, "Surgical process modelling: a review," *International Journal of Computer Assisted Radiology and Surgery*, 2014.

[54] T. Neumuth, *Surgical process modeling- Theory, Methods, and Applications*. 2011. ISBN 978-3-00-038630-5.

[55] L. Riffaud, T. Neumuth, X. Morandi, C. Trantakis, J. Meixensberger, O. Burgert, B. Trelhu, and P. Jannin, "Recording of surgical processes: a study comparing senior and junior neurosurgeons during lumbar disc herniation surgery," *Neurosurgery*, 2010.

[56] A. Loeve, J. Al-Issawi, F. Fernandez-Gutierrez, T. Lango, J. Strehlow, S. Haase, M. Matzko, A. Napoli, A. Melzer, and J. Dankelman, "Workflow and intervention times of mr-guided focused ultrasound - predicting the impact of new techniques.," *Journal of biomedical informatics*, 2016.

[57] R. Malik, P. White, and C. Macewen, "Using human reliability analysis to detect surgical error in endoscopic dcr surgery," *Clin Otolaryngol*, 2003.

[58] T. Neumuth, B. Kaschek, D. Neumuth, M. Ceschia, J. Meixensberger, G. Strauss, and O. Burgert, "An observation support system with an adaptive ontology-driven user interface for the modeling of complex behaviors during surgical interventions," *Behav. Res Methods*, 2010.

[59] Rockstroh, "Video-based detection of device interaction in the operating room.,"

[60] Y. S. Chae, S. H. Lee, H. K. Lee, and M. Y. Kim, "Optical coordinate tracking system using afocal optics for image-guided surgery," *International Symposium of Optomechatronics Technology*, 2015.

[61] H. Zhang, F. Banovac, R. Lin, N. Glossop, B. J. Wood, D. Lindisch, E. Levy, and K. Cleary, "Electromagnetic tracking for abdominal interventions in computer aided surgery," *Computer Aided Surgery*, 2006.

[62] J.-P. Ashis and B.-S. Petra., "Workflow integration matrix: a framework to support the development of surgical information systems," *Design Studies*, 2008.

[63] M. Bandt, R. Kuhn, S. Schick, and H. Meyer, "Beyond flexibility- workflows in the perioperative sector of the healthcare domain," *Electronic Communications of the EASST*, 2011.

[64] T. Neumuth, A. Pretschner, C. Trantakis, M. Fischer, H. Lemke, and O. Burgert, "An approach to xml-based description of intraoperative surgical workflows," *R. Eckstein, R. Tolksdorf, (Eds.), Humboldt-Univ, Berlin*, 2005.

[65] S. Barbagallo, L. Corradi, J. de Ville de Goyet, M. Iannucci, I. Porro, N. Rosso, E. Tanfani, and A. Testi, "Optimization and planning of operating theatre activities: an original definition of pathways and process modeling," *BMC Med Inform Decis Mak.*, 2015.

[66] J. Qi, Z. Jiang, G. Zhang, R. Miao, and Q. Su, "A surgical management information system driven by workflow," *IEEE International Conference on Service Operations and Logistics, and Informatics*, 2006.

[67] S. Sen, A. Garg, D. Gealy, S. McKinley, Y. Jen, and K. Goldberg, "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization," *IEEE International Conference on Robotics and Automation*, 2016.

[68] T. D. Nagy and T. Haidegger, "A dvrk-based framework for surgical subtask automation," *Acta Polytechnica Hungarica*, 2019.

[69] R. Muradore, P. Fiorini, and G. Akgun, "Development of a cognitive robotic system for simple surgical tasks," *Int J of Advanced Robotic Systems*, 2015.

[70] Y. Kassahun, B. Yu, A. T. Tibebu, D. Stoyanov, S. Giannarou, J. H. Metzen, and E. V. 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*, 2016.

[71] C. J. Anders, P. Pasliev, A.-K. Dombrowski, K.-R. Müller, and P. Kessel, "Fairwashing explanations with off-manifold detergent," 2020.

[72] F. Calimeri, F. Cauteruccio, L. Cinelli, A. Marzullo, C. Stamile, G. Terracina, F. Durand-Dubief, and D. Sappey-Marinier, "A logic-based framework leveraging neural networks for studying the evolution of neurological disorders," *Under consideration for publication in Theory and Practice of Logic Programming*, 2019.

[73] A. Loutfi, S. Coradeschi, M. Daoutis, and J. Melchert, "Using knowledge representation for perceptual anchoring in a robotic system," *International Journal on Artificial Intelligence Tools*, 2008.

[74] M. Tenorth and M. Beetz, "Knowrob – a knowledge processing infrastructure for cognition-enabled robots," *International Journal of Robotics Research (IJRR)*, 2013.

[75] T. Springer and A.-Y. Turhan, "Employing description logics in ambient intelligence for modeling and reasoning about complex situations," *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 3, pp. 235–259, 2009.

[76] M. Krotzsch, F. Simancik, and I. Horrocks, "A description logic primer," *Department of Computer Science, University of Oxford, UK.*

[77] P. Tanwar, T. V. Prasad, and K. Dutt, "A tour towards the various knowledge representation techniques for cognitive hybrid sentence modeling and analyzer," *International Journal of Informatics and Communication Technology*, 2018.

[78] F. Baader and W. Nutt, "Basic description logics," *The description logic handbook: theory, implementation, and applications*, 2003.

[79] C. Ramirez and B. Valdes, "A general knowledge representation model for the acquisition of skills and concepts," *Int. J. Softw. Sci. Comput. Intell.*, vol. 4, p. 1–25, Oct. 2012.

[80] S. Puls, J. Graf, H. Worn, and I. Maurtua, "Cognitive robotics ¨in industrial environments," *Human Machine Interaction-Getting Closer InTech*, 2012.

[81] Y. Dimopoulos, B. Nebel, and J. Koehler, "Encoding planning problems in nonmonotonic logic programs," *European Conference on Planning*, 1997.

[82] T. Mueller, “Commonsense reasoning using answer set programming,” *IBM Watson Group and IBM Research*, 2014.

[83] M. Balduccini, M. Gelfond, R. Watson, and M. Nogueira, “The usa-advisor: A case study in answer set planning,” *International Conference on Logic Programming and Nonmonotonic Reasoning*, 2001.

[84] M. Gebser, P. Obermeier, T. Schaub, M. Ratsch-Heitmann, and M. Runge, “Routing driverless transport vehicles in car assembly with answer set programming,” *Theory and Practice of Logic Programming*, 2018.

[85] H. Levesque and G. Lakemeyer, “Cognitive robotics.,” *In: van Harmelen F, Lifschitz V, Porter B (eds) Handbook of Knowledge representation. Elsevier*, 2007.

[86] E. Erdem and V. Patoglu, “Applications of asp in robotics,” *KI - Künstliche Intelligenz*, 2002.

[87] T. Eiter, W. Faber, N. Leone, G. Pfeifer, and A. Polleres, “A logic programming approach to knowledge-state planning, ii: The system,” *Artificial Intelligence*, vol. 144, p. 157–211, Mar 2003.

[88] H. Nakawala, R. Bianchi, L. Pescatori, O. D. Cobelli, G. Ferrigno, and E. D. Momi., “Deep-onto network for surgical workflow and context recognition,” *International Journal for Computer Assisted Radiology Surgery.*, 2019.

[89] K. A. Blum, D. J. Paulucci, R. Abaza, D. D. Eun, A. Bhandari, J. C. Delto, L. S. Krane, A. K. Hemal, and K. K. Badani, “Main renal artery clamping with or without renal vein clamping during robotic partial nephrectomy for clinical t1 renal masses: Perioperative and long-term functional outcomes,” *Urology*, 2016.

[90] W. Liu, F. Wang, X. L. Hui Ding, J. Tian, Y. Gong1, D. Li, H. Wang, D. Kaushik, R. Rodriguez, and Z. Wang, “Comparison of the peri- and postoperative parameters between artery-only and artery-vein clamping techniques in partial nephrectomy: a systematic review and meta-analysis,” *International Journal of Clinical and Experimental Medicine*, 2017.

[91] D. Cantera, A. Kutikova, B. Manleya, B. Eglestonb, J. Simhana, M. Smaldonea, E. Tepera, R. Viterboa, D. Y. Chena, R. E. Greenberga, and R. G.

Uzzoa, "Utility of the r.e.n.a.l.-nephrometry scoring system in objectifying treatment decision-making of the enhancing renal mass," *Urology*, 2011.