Classification of Modeling for Versatile Simulation Goals in Robotic Surgery

Stefan Jörg, Rainer Konietschke, and Julian Klodmann

Abstract Simulation is common practice for surgeon training in particular for robotic surgery. This paper introduces further relevant applications of simulation that improve patient safety. Therefore, the design of a modular simulator for minimally invasive robotic surgery is presented. The authors introduce a classification of hierarchical levels of modeling details for the three aspects Application, System, and Patient. Furthermore, the principal use case classes Training, Workflow Validation, Workflow Design, Monitoring, and Robot Design of simulation for robotic surgery are introduced. For each class standard simulator setups are presented. The use of the classification is exemplified for Training and Robot Design use cases.

Key words: surgical robotics, simulation, patient safety

1 Introduction

It is common practice, to use the method of simulation to train surgeons with the aim to improve patient safety [5] [11] [2]. Training has been recognized as an even more important issue when surgical robotic systems are involved because their use requires new skills [6].

Other than surgical training, there are more useful applications of simulation that increase patient safety in the context of robotic surgery, e.g. preoperative planning or workflow optimization.

The intention of this paper is to introduce the relevant applications of simulation for robotic surgery and to describe their relation to patient safety. In particular, the relevant properties of simulation in robotic surgery are

---

Robotics and Mechatronics Center, German Aerospace Center (DLR), Wessling, Germany, e-mail: `stefan.joerg@dlr.de`

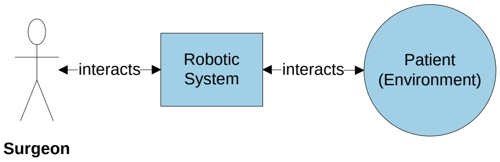

Fig. 1 The Robotic System Simplifies the Interface Between Surgeon and Environment. The Surgeon Only Interacts with the Robotic System.

identified and then used to classify the simulation types as use cases which are defined intended applications.

As robotic surgery involves complex technical systems the authors view the simulation thereof from a systems engineering point of view. The System Modeling Language SysML is an industry standard for the specification, analysis, design, verification, and validation of such systems [10]. Following SysML, we define a *use case* as the description of the intended *goal* of a simulation.

To establish a common ground, the definition of a Simulator for Robotic Surgery is given in the following section. Based on this, the main components of a simulator for robotic surgery are introduced. Sec. 3 introduces the concept for approximation of the real world with levels of details with the aim to optimize the modeling and simulation effort for a given use case. Sec. 4 introduces the principal use case classes of simulation. Sec. 5 exemplifies the setup of the simulator.

2 A Simulator For Robotic Surgery

In this section the definition of a simulator for robotic surgery is presented. This definition will be later used for the discussion of the relevant use cases and the simulator setups related to that.

There are various definitions of simulation in the literature that reflect the broad application domains of this method [4] [12]. We use the more formal definition of *simulation* as observation or execution of a model over time. A *model* is an approximate representation of a real-world system. A *simulator* implements and executes a model to perform a simulation. *Analysis* is the process to verify and validate the simulation result. Thus, the method of simulation involves the tasks to model, simulate, and analyze.

A simulator interacts with an *entity-under-test*. If the entity is human, the simulation is called *interactive* (human-in-the-loop). Surgeon training is a well-known example of interactive simulation.

Simulation is *real-time* when the simulation time advances with the physical time of the real-world. If the simulator interacts with a real-world system

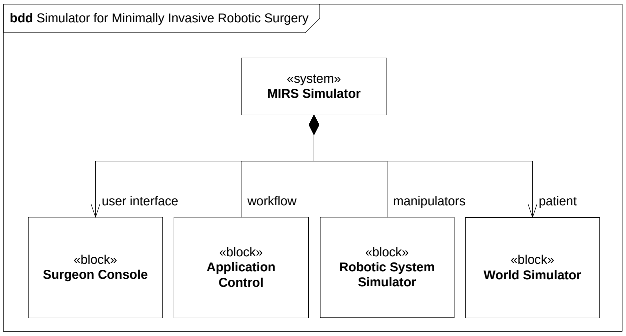

Fig. 2 A Modular Simulator for Minimally Invasive Robotic Surgery (MIRS)

the simulation has to be real-time. Hence, interactive simulation is always real-time.

A model is *discrete-event* or *continuous*. A discrete-event model updates its state upon the occurrence of an event. In general, events occur aperiodically. In contrast to that, a continuous model defines its state as a continuous function of time, e.g. as differential equations. Advanced simulations of real-world systems use a combination of both [4].

Practically, the *implementation* of a simulation approximates the continuous states at periodic steps in simulation time. Similar to that robotic systems are implemented as discrete signal systems. Discrete signal computation means that at certain points in time sensor values are acquired and a control law is executed that generates new motor commands. This similarity can be leveraged to use the same implementation of control laws for the simulation and the real system.

In robotic surgery, the robotic system is usually the main way a surgeon interacts with the patient. Therefore, the focus of this paper is on models where only the robotic system is the intermediate between surgeon and environment (see Fig. 1). This simplifies the user interface. Instead of various surgeon tools only the user console of the robotic system interfaces directly with the surgeon. Especially, for the tele-robotic setups of minimally invasive surgery it is natural to use the same interface for simulation and real-world scenarios [1].

Fig. 2 depicts the main components of a simulator for minimally invasive robotic surgery. The four components Surgeon Console, Application Control, Robotic System Simulator and World Simulator partition the model of the surgical robotic system into corresponding classes: User interface, workflow, manipulators, and patient.

This modularity of the simulator can be used to adapt the simulator to various use cases. The simulator's use case determines the details of each model (its relevant aspects) and the interface between each component (exchange of data between models). Level of detail introduces a classification of aspects of a model's approximation of the real world. The following section discusses the main aspects and introduces corresponding hierarchies of abstraction levels.

3 Level Of Details

A model is an approximation of the real world. Levels of modeling details introduce a classification of this approximation. The main purpose of the introduction of levels of modeling details is the scaling of modeling and computation effort according to the simulation's use case. Hence, a proper classification has to grasp the relevant issues of the simulator's application domain. We refer to these relevant issues as the aspects of the model. For robotic surgery, we identified three relevant aspects of the model:

- The Application Model, which captures the surgical procedure *(Application-Centric)*
- The System Model, which captures the implementation of the robotic system *(System-Centric)*
- The Patient Model, which captures the environment with the focus on the patient *(Patient-Centric)*

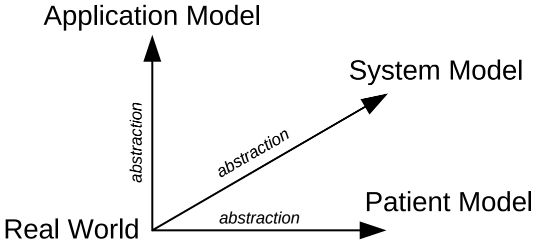

Fig. 3 Three Distinct Aspects of Simulation for Robotic Surgery are Application, System, and Patient.

Together, they span the modeling space for robotic surgery (see Fig. 3). Depending on the simulation's use case, each aspect is of more or less interest, i.e. the approximation needs to be more or less realistic. For each aspect, a hierarchy of abstraction levels further classifies the modeling detail. Starting at the most abstract level each successive level adds a new detail to the model. Finally, Level 0 represents the real-world. These hierarchies for each aspect are the method to easily adapt a simulator setup to the requirements of a distinct use case. In the following, the hierarchies for the three aspects are introduced.

3.1 Application-Centric Modeling Hierarchy

The aspect of application has the focus on how detailed the surgical procedure is modeled.

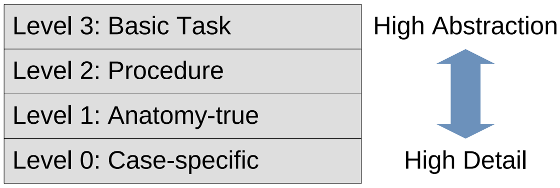

Fig. 4 Application-Centric Levels of Modeling Details

Level 3 Basic Task: On this level, the application model captures only basic tasks, such as suturing or cutting. There is no workflow and no surgical goal.

Level 2 Procedure: The procedure level adds the workflow to the basic tasks. This includes a surgical goal.

Level 1 Anatomy-true: This level adds the details of human anatomy, such as organs and vessels.

Level 0 Case-specific: This level adds the specific data and anatomy parameters of a certain case (patient).

3.2 System-Centric Modeling Hierarchy

The system model has the focus on the modeling details of the implementation of the surgical robotic system. This refers to the main components of a surgical simulator as depicted in Fig. 2. Therefore, for each level the impact on each component is discussed, except the Application Control component which is mainly affected by the Application-Centric aspect.

Level 3 Kinematics: Only the kinematical aspects are modeled on this level, dynamic effects are not considered (e.g. masses, inertia, and forces).

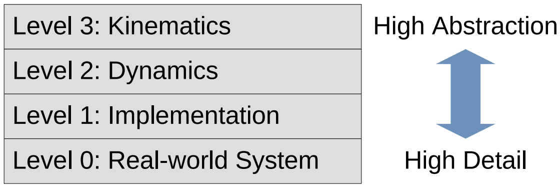

Fig. 5 System-Centric Levels of Modeling Details

Surgeon Console: User input is simulated and limited to motion, e.g. playback of pre-recorded user input, dedicated test patterns. The **robotic system** is modeled as an ideal positioning device: It follows its desired trajectory instantly. **World:** Manipulation of objects in the environment is limited to motion.

Level 2 Dynamics: This level introduces dynamics, i.e. the effects of masses and forces and torques w.r.t. motion. Relevant dynamic effects of implementation properties are also modeled on this level, e.g. torque ripple, latency, friction. **Surgeon Console:** Simulated user input is augmented by the dynamic properties of input devices. Thus, the effect of input devices on the system, e.g. its control laws, etc., can be investigated. **Robotic System:** The movement of the robot is constrained by its dynamics (e.g. inertia, maximum motor torque, friction, etc.) and by external forces of the environment. An accurate dynamics model of the robotic system is necessary for this [8]. **World:** Interaction of objects (including the robot) in the environment is modeled with forces and torques.

Level 1 Implementation: This level adds the cycle-true computation of discrete signal systems. The cyclic execution at discrete points in time is only an approximation of the continuous physical world. Level 1 models this cyclic execution of the system. Thus, the effects on the system's performance due to this discrete approximation can be incorporated to the simulation. Level 1 is useful to simulate the behavior of implementations of complex systems. Details of the mechatronic implementation are added to the simulation. These include communication errors and delays, computation time, quantization of sensor values. **Surgeon Console:** Mechatronic implementation of input and feedback devices. **Robotic System:** Mechatronic implementation of the robotic system. **World:** There are no implementation details of the environment.

Level 0 Real System: This level represents the real-world system, e.g. the robot and its tools, input devices, GUIs, 3D-Displays, the patient, animal models or phantoms.

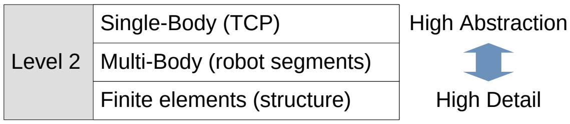

Fig. 6 Reasonable Levels of Modeling Details within Level 2 (Dynamics)

The three levels of kinematics, dynamics and implementation cover the most relevant details of simulation in descending order. It is reasonable to refine this main classification within each level. Especially Level 2, the dynamics, has varying aspects of interest depending on the simulation's use case. Fig. 6 depicts a possible hierarchy within Level 2.

3.3 Patient-Centric Modeling Hierarchy

The patient model regards the details of the environment with the focus on the patient. Basdogan et al. [5] state that versatility in patient modeling, finding the right realism and integrating material properties of organs are all important issues for surgical simulation. The following hierarchy defines three major abstraction steps of the environment model.

Fig. 7 Patient-Centric Levels of Modeling Details

Level 3 Rigid Body: On this level, the environment is modeled as rigid body geometry with stiff object contacts.

Level 2 Soft Tissue: This level adds the detail of material properties to the modeled objects (e.g. soft tissue for organs).

Level 1 Functional: This level adds functions to the tissue, i.e. blood flow, objects are cuttable, tear-able.

Level 0 Real-World: The patient, a phantom or an animal.

4 A Classification Of Principal Use Cases

Starting the classification from a systems engineering point of view the principal use case classes of simulation for robotic surgery are discussed in this section. As robotic surgery involves sophisticated mechatronic systems it is reasonable to derive the categories of application from that domain. Including training from the domain of surgery, the four main categories of goals for simulation in the context of robotic surgery are as follows:

Training - goal of training is to use the method of simulation to improve the skills of involved humans.

Design - goal is to use simulation for the design process of a system. This includes the robotic system, the design of a surgical procedure, etc.

Validation - goal is to identify the proper system or task specifications.

Verification - goal is to use simulation to determine whether a system or procedure works as specified.

In the following, the principal use case classes for simulation of robotic surgery are discussed and related to the four categories. Figures 8-13 show the configuration of the modular simulator, introduced in Section II, for the principal use case classes.

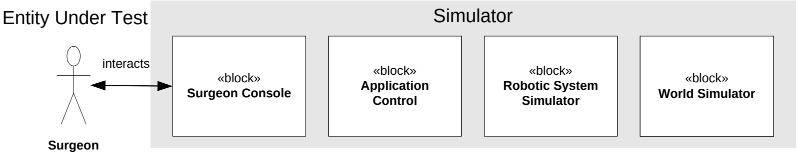

Fig. 8 Simulator Setup for the Class Surgeon Skill Training

Use Case Class Surgeon Skill Training (Category: Training) It is common practice to use simulation to train surgeon skill either on a certain surgical procedure or to operate the robotic system. **Goal:** Improve skills of surgeons. **Typical Setup:** Requires all four components of the simulator (see Fig. 8). The surgeon is the entity-under-test. Hence, the simulation is always interactive. **Patient safety** is immediately affected by surgeon skills. However, a training curriculum is required that proves to enhance surgeon skills [3].

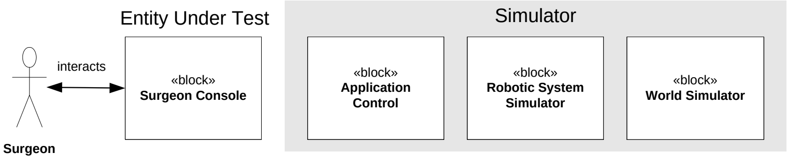

Fig. 9 Simulator Setup for the Class User Interface Designs

Use Case Class User Interface Design (Category: Design, Validation) **Goal:** Improve and validate the design of the surgeon console. **Typical Setup:** The simulator (see Fig. 8) consists of Application Control, Robotic System and World simulators. The Surgeon Console is the entity-under-test. **Patient Safety:** A validated user interface reduces the risk of mal-operation. An improved user interface puts the surgeon's focus on the surgical procedure. It provides the surgeon with relevant information only and alerts him of impending risks.

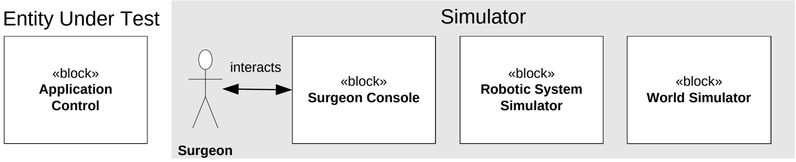

Fig. 10 Simulator Setup for the Class Workflow Validation

Use Case Class Workflow Validation (Category: Validation). Workflow falls into two categories: Surgical Procedure and Technical Workflow. The former has the focus on the surgical goal, the latter on the operation of the robotic system. **Goal:** Validate a given workflow with human-centered (usability) methods. **Typical Setup:** The simulator setup (see Fig. 10) is similar to the class User Interface Design, except that Application Control is the entity-under-test. **Patient safety:** A workflow design that has been validated with the surgeon-in-the-loop reduces the risk for the patient because the workflow has been proven to be more likely performed well by surgeons.

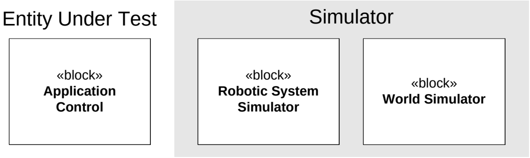

Fig. 11 Simulator Setup for the Class Workflow Design

Use Case Class Workflow Design (Category: Design). **Goal:** Improve the design of a surgical or technical workflow. Pre-operative planning falls into this class. **Typical Setup:** Requires both the robotic system and the world simulators. Application Control is the entity-under-test (see Fig. 11). **Patient safety:** Improved workflow for a certain procedure (e.g. shorter operation time, simplified tasks) reduces the risk for the patient.

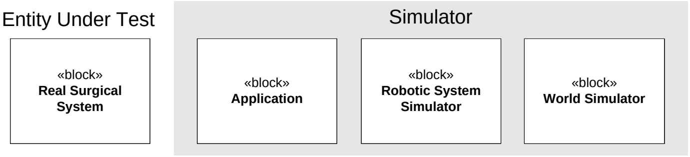

Fig. 12 Simulator Setup for the Class Monitoring

Use Case Class Monitoring (Category: Verification). During surgery, the simulator runs in parallel to the real system. A monitor compares the simulation state with the real state. It estimates whether the real system works as intended. A variant of monitoring is predictive simulation, where the future state of the real system is estimated from the current simulation state [7]. **Goal:** Continuously verify real system during operation. **Typical Setup:** The simulator comprises of Application Control, Robotic System Simulator, and World Simulator (see Fig. 12) . The real system is the entity-under-test. **Patient safety:** Monitoring increases patient safety by reducing the risk to deviate from a planned procedure. The comparison of simulated and real state enables to detect system failures, deviations from the planned workflow, and planning errors.

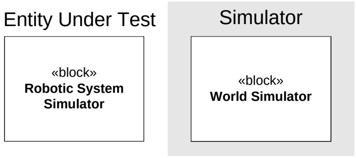

Fig. 13 Simulator Setup for the Class Robot Design

Use Case Class Robot Design (Category: Design, Verification, Validation) **Goal:** Improve the design of the robotic system, e.g. optimize the kinematics for a certain workspace. **Typical Setup:** Requires only the World Simulator (see Fig. 13). The Robotics System Simulator is the entity-under-test. **Patient safety:** A better robotic system that is more suitable for the intended procedure reduces the overall risk for the patient.

5 Simulator Configurations For Use Cases

This Section exemplifies how to setup the simulator regarding its modeling aspects and details for the two principal use case classes Surgeon Skill Training and Robot Design.

5.1 Surgeon Skill Training

The *Surgeon Skill Training* class can be divided into several training goals: (1) **Technical:** training the use of the technical system, i.e. the surgical robotics system, (2) **Procedure:** training a certain surgical procedure and (3) **Patient-specific:** training a patient-specific surgical intervention.

For the **Technical** goal, the mere *basic task* may be sufficient in terms of the *Application*. In case the methods to step through the workspace are part of the training, also the *procedure* needs to be modeled. As concerns the *System*, a simple *kinematics* modeling can usually show most of the relevant system behavior. However, if e.g. the robotic system offers force feedback, also some *dynamics* should be modeled to allow the user to understand this functionality. In the dimension of *Patient*, a *rigid-body* environment is often used; see first training steps in [9]. The force feedback functionality however could be better introduced by using *soft tissue* modeling. If the robotic system uses special instruments (e.g. cautery devices), also a *functional* modeling may be required.

The second goal, **Procedure**, usually involves an *anatomy-true* modeling in *Application*. To allow for interaction between surgical tools and patient, *dynamics* need to be modeled in *System*, and the *Patient* should be modeled either as *soft tissue* or even *functional*.

For the third goal, **Patient-specific**, the *Application* needs to be modeled *case specific*, otherwise it is similar to the second goal Procedure.

5.2 Robot Design

Simulation methods are widely used to optimize a *Robot Design* also in non-medical areas. Usual goals in this class are to find (1) **Kinematics:** the kinematics structure of the robot (e.g. number and sequence of joints, link lengths), (2) **Dynamics:** the dynamics parameters (e.g. masses of the segments, stiffnesses, maximum accelerations, applicable forces and torques), or (3) **Implementation:** the appropriate implementation (e.g. sensor resolution, motor characteristics).

In terms of *Application*, the task needs to be *anatomy-true* in most cases to allow for good comparability with reality. The classification w.r.t. the *System* is straight-forward regarding the goal names. As concerns the *Patient*, a *rigid body* or *soft tissue* modeling can be chosen.

6 Conclusions

The hierarchical levels of modeling details of the three aspects Application, System, and Patient enable the setup of a simulator for versatile goals in robotic surgery. The presented use case classes are only the basis from that the actual use cases are derived further. How this works, is illustrated in the previous section. Moreover, metrics of the impact on patient safety should be developed for each use case. The presented classification is currently used for the implementation of a Robotic System Simulator within the project SAFROS (Patient Safety in Robotic Surgery).

The developed concepts are also applicable to tele-robotics in general, e.g. the operation of robots in hazardous environments such as space or disaster zones, etc. Herein the aspect of Environment of Fig. 3 moves from Patient to the new domain. The levels of modeling details only differ slightly from the proposed ones. Furthermore the described methods enable to combine different simulations (e.g. workflow, monitoring and simulations to design robots), often used separately in various fields of robotics, within one modular simulator. This can be done by merging the modular simulations and their classified use cases with compatible interfaces.

Acknowledgements The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 248960. http://www.safros.eu

References

1. da Vinci skills simulator. URL `http://www.intuitivesurgical.com/products/skills_simulator/`. Last visited 30. Jan 2012
2. Agency for Healthcare Research and Quality: Improving patient saftey through simulation research (2008). URL `http://www.ahrq.gov/qual/simulproj.htm`
3. Alimisis, D., Vicentini, M., Fiorini, P.: Towards a problem-based training curriculum for surgical robotics: the SAFROS project. In: T. Bastiaens, M. Ebner (eds.) Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications, pp. 297–302. Chesapeake, VA (2011)
4. Banks, J. (ed.): Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice. Wiley & Sons, New York (1998)
5. Basdogan, C., Sedef, M., Harders, M., Wesarg, S.: VR-based simulators for training in minimally invasive surgery. Computer Graphics and Applications **27**(2), 54–66 (2007). DOI 10.1109/MCG.2007.51
6. Herron, D., Marohn, M.: A consensus document on robotic surgery. Surgical Endoscopy **22**, 313–325 (2008). DOI 10.1007/s00464-007-9727-5
7. Hirzinger, G., Brunner, B., Dietrich., J., J.Heindl: Sensor-based space robotics-ROTEX and its telerobotic features. IEEE Transactions on Robotics and automation **9**(5) (1993)
8. Klodmann, J., Konietschke, R., Albu-Schaffer, A., Hirzinger, G.: Static calibration of the DLR medical robot MIRO, a flexible lightweight robot with integrated torque sensors. In: Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pp. 3708–3715. IEEE, San Francisco,CA, USA (2011). DOI 10.1109/IROS.2011.6095097
9. Lerner, M., Ayalew, M., Peine, W., Sundaram, C.: Does training on a virtual reality robotic simulator improve performance on the da vinci surgical system? Journal of Endourology **24**(3) (2010)
10. Object Managment Group, http://www.omg.org/spec/SysML/1.2: OMG Systems Modeling Language (OMG SysML), Version 1.2 (2006). Document formal/2010-06-01
11. Satava, R.: Historical review of surgical simulation - a personal perspective. World Journal of Surgery **32**(2), 141–148 (2008). DOI 10.1007/s00268-007-9374-y
12. Sokolowski, J., Banks, C. (eds.): Principles of Modeling and Simulation: A Multidisciplinary Approach. Wiley & Sons, New York (2009)