ENGINEERING MATHEMATICS
AND COMPUTING LAB

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Towards an Intelligent Framework for Personalized Simulation-enhanced Surgery Assistance:

Linking a Simulation Ontology to a Reinforcement Learning Algorithm for Calibration of Numerical Simulations

Nicolai Schoch, Vincent Heuveline

Preprint No. 2017-05

Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)

www.emcl.iwr.uni-heidelberg.de

Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)
ISSN 2191–0693
Preprint No. 2017-05

Affiliation of the Authors

Nicolai Schoch[a,1], Vincent Heuveline[a]

[a] *Engineering Mathematics and Computing Lab (EMCL),*
Interdisciplinary Center for Scientific Computing (IWR),
Heidelberg University, Germany

[1] *Main Author: Nicolai Schoch, OrcID 0000-0002-7806-6435,*
mailto:nicolai.schoch@iwr.uni-heidelberg.de

Impressum

Heidelberg University
Interdisciplinary Center for Scientific Computing (IWR)
Engineering Mathematics and Computing Lab (EMCL)

Im Neuenheimer Feld 205,
69120 Heidelberg
Germany

Towards an Intelligent Framework for Personalized Simulation-enhanced Surgery Assistance: Linking a Simulation Ontology to a Reinforcement Learning Algorithm for Calibration of Numerical Simulations

Nicolai Schoch, Vincent Heuveline

September 28, 2017

Abstract

Evolving our previous research results in the context of cognition-guidance and patient-specifity for simulation-enhanced cardiac surgery assistance, in this work we further investigate on (1) a machine learning framework which allows to patient-individually calibrate soft tissue material parameters for subsequent simulation, and (2) a profound knowledge management framework which may enhance the ontology-driven overall setup of the cognition-guided surgery simulation in a clinic environment.

Rather than being a closed research work with an in-depth theory backup and a complete evaluation, we here present a technical report and some interesting experimental works that are to serve for further research and development.

1 Introduction

When dealing with complex surgery, it is highly important to ensure a holistic, knowledge- and experience-based, patient-specific surgical treatment planning. The surgeons and staff in the operation room (OR) need to account for the entirety of available medical patient data, know how to handle an abundance of technical developments, and stay on top of current surgical expert knowledge in order to define the possibly best suitable surgical treatment strategy. There is hence a huge potential for computer assistance and IT support, also and in particular regarding surgery simulation, which enables surgeons not only to plan but to simulate, too, several steps of an intervention and to thus forecast relevant surgical situations.

In our previous works, we have focussed on supporting minimally-invasive mitral valve reconstruction (MVR) surgery. MVR is a complex operation that is to re-establish the functionality of an incompetent mitral valve (MV) through implantation of an artificial annuloplasty ring that reshapes the morphology of the valvular apparatus and thus allows for proper valve closure again [2]. We aimed at supporting MVR surgery by providing the surgeon with biomechanical FEM-based MVR surgery simulations [15,17]. These simulations are to enable the surgeon to assess the simulated *post-operational* MV behavior *before* the actual operation. However, in order to be really beneficial to the surgeon, these simulations must fulfill certain criteria, which comprise:

- they must be patient-specific, i.e., the available patient-individual medical data and information must holistically be processed and integrated into the simulation,
- they must be based on surgical expert knowledge and medical evidence,
- the biomechanical and patient model which underlies the simulation must be comprehensive, and

- the setup and execution of the simulation (and of all preceeding simulation preprocessing steps) needs to be fully automated and integrated into the surgical treatment workflow.

Hence, in our former work, research on simulation-enhanced, cognition-guided, patient-specific cardiac surgery assistance has be conducted. The overall work is thoroughly described and documented in the PhD Thesis by Nicolai Schoch [15], and several dedicated publications focus on the miscellaneous research fields and component parts of the overall prototypic system. In summery, first a biomechanical MV/MVR model has been described and an FEM-based MVR surgery simulation has been developed [16] using the FEM software toolkit HiFlow^3 [1]. Following, the Medical Simulation Markup Language (MSML) was introduced, and it was described how – through its features and functionalities – it simplifies the biomechanical modeling workflow [22]. It was then detailed, how, by means of the MSML and a set of dedicated MVR simulation preprocessing operators, patient-individual medical data is comprehensively analyzed and processed in order for the fully automated setup of HiFlow^3-based MVR simulation scenarios [18]. Finally, the entire work was integrated into the cognitive system architecture of the joint research project *Cognition-Guided Surgery* (SFB TRR 125). Emphasis was put on its semantic knowledge and data infrastructure [3] as well as on the setup of its cognitive software components [19], which eventually facilitate – at least to a certain extend – cognition-guidance and patient-specifity for the overall simulation-enhanced MVR assistance pipeline [17].

See Figure 1 for an illustration of the overall cardiac surgery assistance workflow and system setup. All data and information is gathered, structured and semantically annotated in the common Knowledge Base [3]. Two separate chains then further process both image and non-image as well as parameter data [19]. Merging these two information processing chains, the Simulation Preprocessing component, which is built onto the functionalities of the MSML [22], takes care of the proper biomechanical model and simulation scenario setup [18]. Finally, the Numerical Simulation Application executes the afore set-up MVR surgery simulation scenario [17], and hands the results over to the postprocessing and visualization component.

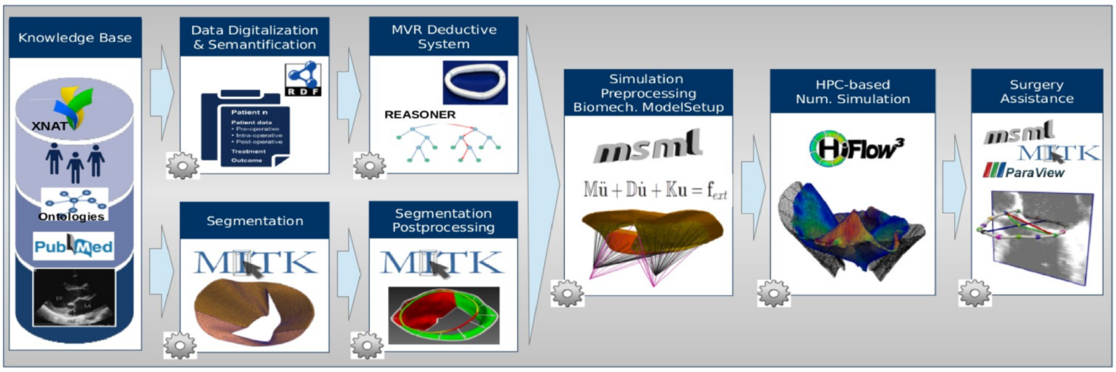


Figure 1: Workflow diagram and overall setup of the cognition-guided, patient-individual, simulation-enhanced MVR surgery assistance system prototype, as described in [15].

In these previous works, we have proposed and implemented, for the first time, a prototypic system for simulation-enhanced, cognition-guided, patient-specific cardiac surgery assistance, and its functionality and performance were successfully evaluated. We have shown that – through its cognitive, data-driven pipeline setup – medical patient data and surgical information can be analyzed and processed comprehensively and fully automatically, which hence presents an important step towards a simulation-enhanced, cognition-guided, patient-specific cardiac surgery assistance system.

Nevertheless, there is still a long way to go for simulation-enhanced surgery assistance in order to be really applicable and beneficial to surgeons in the operation room. Among others, this is due to two major issues which have so far been treated only in second place:

On the one hand, this is the ***patient-specific, fine-grained calibration*** of the biomechanical model and simulation, e.g., with respect to the *patient-individual tissue material parameters*, which cannot be properly measured, and which are hence taken from anatomy lexica, specific domain literature or from simply averaging over patient collectives.

On the other hand, this is the systems' ***missing medical and technical evidence***. Surgery simulation is rarely established in the OR due to the sheer complexity of an adequately and comprehensively set-up surgery simulation scenario. Setting up an adequate surgery simulation not only requires an in-depth knowledge and understanding of the patient's anatomy, its functionality and possible diseases, as well as profound experience in the surgical context. In order to optimally set up the simulation scenario and to properly interpret its results, this also demands proficient skills in the context of Finite Elements, simulation development and simulation environments. It is hence urgently required to semantically represent and standardize this knowledge and experience in order for a further automation of the entire biomechanical modeling and simulation workflow, and for evidence-based, better-accepted surgery simulation results [9].

Hence, having conducted research on the basics of cognition-guidance for simulation-enhanced cardiac surgery assistance, i.e., building on the previously above described achievements that result from our recent works, we now aim at focussing on the following two fields in particular:

On the one hand, we look at the ***patient-individual calibration*** of soft tissue material parameters by means of ***machine learning*** methods. And on the other hand, we work further towards a profound ***knowledge management*** and an ***ontology-driven setup*** for cognition-guided, simulation-enhanced surgery assistance.

Please note, rather than being a closed research work with an in-depth theory backup and a complete evaluation, this paper is more of a description of some interesting experimental works that followed the work in the context of the collaborative research project *Cognition-Guided Surgery* (SFB TRR 125). Resulting from this work, one can recognize that further research is required in the future, and this paper may give a first hint at how promising this research may be.

2 Methods and Evaluation

In this section, first, we investigate on the patient-individual calibration of simulated soft tissue deformations by means of machine learning (ML) algorithms. We will develop a reinforcement learning (RL) algorithm which allows for identification for soft tissue material parameters using real image and deformation data.

Second, we further develop the surgery simulation ontology proposed in our previous work [20], in order to setup a profound knowledge management framework and to allow for an ontology-driven overall setup for our cognition-guided, patient-individual, simulation-enhanced surgery assistance prototype.

Both sections contain their respectively own evaluation, and each a short discussion, too.

2.1 Machine Learning-based Simulation Calibration

In this subsection, we first present our RL test scenario, then outline the idea of reinforcement learning and motivate why it fits to simulation calibration tasks. Following, we describe our RL implementation and the simulation interface, as well as conduct an evaluation and a short discussion of the results.

The test scenario for ML-based parameter identification and simulation calibration. We simulate the bending of a beam which is fixed on its left side and which is subject to pressure and gravity on the right, see Figure 2. The beam is constituted of an unknown homogenous soft tissue material, and its deformation is simulated using the Saint-Venant Kirchhoff model of linear elasticity in order to describe the relationship between stress and strain [16]. The therein contained two Lame constants λ and μ, as well as the beam's material density ρ are unknown.

By means of an available simulation result (in terms of *real* image and deformation data for the same beam object), we now try to find out the *actual* material properties.

This situation can be seen *in analogy* to a typical situation in the operation room: The surgeon shall be provided with a simulation of the behavior of a patient's organ, and the organ has previously been subjected to some imaging technique, so the respective image data and deformation measurements are already available. However, eventhough the morphology of the organ can be obtained from the image data through segmentation, there is no means yet to specifically measure the in vivo material properties in order for utilizing these in a subsequent biomechanical simulation.

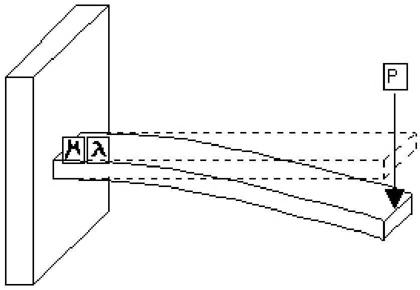


Figure 2: Draft of the test scenario for parameter identification.

Hence, in whichever way this simulation is set-up, it will not be patient-specific with respect to the material parameters, *if* there is no preceeding parameter calibration [17]. The latter, i.e., the parameter identification and calibration, is exactly, what this work is investigating on in this section, so the goal is to calibrate the simulation material parameters in order to fit them to the available image or deformation data.

Back to the test scenario, we intend to calibrate the material parameters through fitting them to image and deformation data which has artificially been produced for testing purposes. Our *fitting measure* follows the idea of the DICE coefficient and computes the *Root Mean Square Error* (RMSE) of the beam, which is obtained from the (not yet entirely calibrated) deformation simulation on the one hand, and the artificially produced deformed beam data on the other hand. On a side note: we record the material parameters that we use for producing the artificial beam deformation data, in order to later evaluate the quality of the parameter calibration.

The idea of reinforcement learning and its application to our test scenario. When looking at the different classes of machine learning (ML) algorithms, it becomes obvious that those that belong to the *reinforcement learning* (RL) type may likely be most applicable for the above described simulation calibration problem. On the one hand, we have our (not yet calibrated) simulation setup and simulation results, and we have the respective image and deformation data, which we try to get as close to with our simulation results as possible. And on the other hand, we have a quality or fitting measure, the RMSE value, that tells us how good we are with the calibration.

In this context, RL can be seen as the *interaction* of some *intelligent agent or system* with a given *environment*, namely the above describe quantities and observations [21]. See Figure 3 for an illustration of this concept.

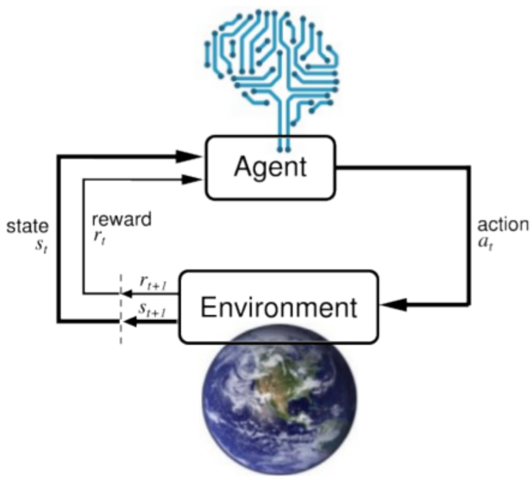


Figure 3: Schematic draft of the concept of reinforcement learning: an agent interacts with an environment, makes observations, gets rewards and performs further actions.

RL is learning what to do and learning how to map situations/observations to actions through defining a (numerical) reward for the respective actions. The learner is not told which action to take, but instead must discover which action will yield the maximum reward.

As an example, one may think of a pet that you want to train to sit down. A common technique to train animals is to let them know that some treat is just about to get served when they fulfill a respective task. This is essentially *reinforcing* your pet to practice the intended behavior. You tell them to sit down, make them sit down and follow up with a treat. And as time passes, the pet gets used to this causality series, and whenever it hears the 'sit down' command, it responds by sitting down in anticipation of a treat. In this example, the artificial agent is represented by the pet, the reward function is the treat, and the resulting action is the sit down behavior.

Transferring this to our simulation calibration problem, the artificial agent is represented by the computer which is aware of the current guess of the material parameter combination along with the resulting 3D morphology representation of the deformed beam. The reward function is a function of the RMSE value, and the smaller the RMSE value, the higher the reward for the agent. Lastly, the action is an adaption of the current parameter set, aiming for an increase of the reward in the next step. We will see below, how such an adaptation of the parameter set may look like.

More formally, the mathematical framework for defining a solution in an RL scenario is a *Markov Decision Process* [21], where we have

- a set S of states s_t,
- a set A of actions a_t,
- a reward function r_t, which depends on s_t and a_t, and
- a policy Π, as well as a value V.

In order to get from the initial state s_0 to the final state s_N, an action a_t or possibly several actions have to be taken, and for each action that is taken, a reward r_t which can be both positive or negative is given. The set of performed actions defines the policy Π, or vice versa, the policy determines the actions that are performed and why. Lastly, the returned rewards sum up in the value V, hence, the goal is to maximize the rewards by choosing the optimal policy. Mathematically speaking, this means we have to maximize the expected value $E(r_t|\Pi, s_t)$ for all possible values of S for a time step t.

Coming back to our test scenario, given some initial material parameter combination, say $\{\lambda_i, \mu_i\}$, the set of possible *actions* now consists of the following possibilities:

- Action 1: do nothing, i.e., no *real* parameter manipulation, which yields $\{\lambda_i, \mu_i\}$,
- Action 2: increase parameter λ by $\Delta\lambda$, which yields $\{\lambda_i + \Delta\lambda, \mu_i\}$,
- Action 3: decrease parameter λ by $\Delta\lambda$, which yields $\{\lambda_i - \Delta\lambda, \mu_i\}$,
- Action 4: increase parameter μ by $\Delta\mu$, which yields $\{\lambda_i, \mu_i + \Delta\mu\}$,
- Action 5: decrease parameter μ by $\Delta\mu$, which yields $\{\lambda_i, \mu_i - \Delta\mu\}$.

With this set, an *action*, which should yield an increase of the reward, would be an *adaptation* of the whatsoever current material parameter combination to a slightly better parameter combination, i.e., to a combination which is a little closer to the real material parameters.

With respect to the policy, the most straight-forward approach is a greedy approach, which is literally known as the *epsilon greedy* approach. This approach consists in looking at the current state of the agent/environment, and looking at all possible actions and their respective direct rewards, and then selecting and executing the best of them.

Besides this epsilon greedy approach, there are other ways to solve the problem. The major categories are

- policy-based, where the focus is to find the optimal policy,
- value-based, where the focus is put on finding the optimal value, i.e., cumulative reward, and
- action-based, where the focus is on what the optimal action is at each step.

We refer to Sutton and Barto [21] for more general information on these approaches, and to a paper which conducts a dedicated survey on different RL algorithms [6].

Implementation and evaluation of an RL algorithm for solving our test scenario. In the following, we describe the implementation of our *Q-Learning* RL algorithm. Q-Learning is a policy-based learning algorithm, where the function approximator is usually implemented as a neural network [21]. For the beginning, in our case, it is a straight-forward implementation without a NN-based Q-value approximation. See Figure 4 for a structural draft of the algorithm.

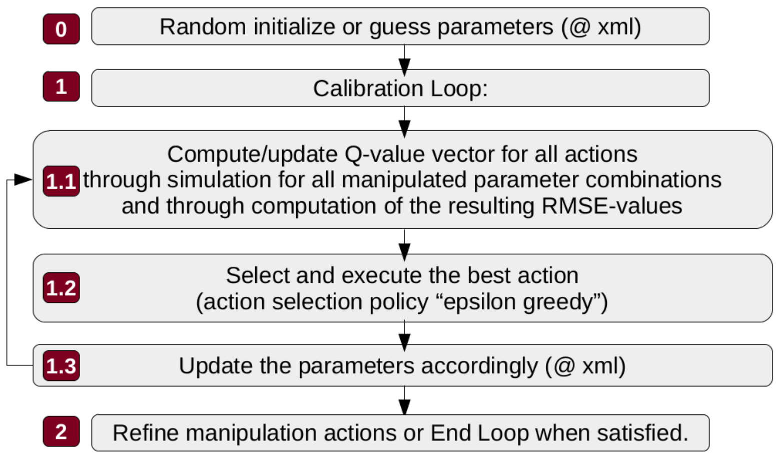


Figure 4: Draft of the basic structure and setup of the implementation of our reinforcement learning algorithm for simulation calibration.

In an initialization step (0), the set of parameters, i.e., the parameter combination, is randomly initialized, or – if possible – guessed, and written into the HiFlow^3 simulation input xml file.

The subsequent calibration loop (1) then comprises the following substeps: (1.1) Initialize (and then respectively update) the *Q-value vector*, the components of which contain the scalar-valued rewards for the respectively possible actions in each step. To compute the rewards, first, the deformation scenarios have to be simulated for all possible parameter manipulations (see the 5 actions above), and second, from the obtained simulation results, the reward (in terms of RMSE-value) has to be computed for each scenario. (1.2) From all 5 available actions, select and then actually execute only the best action, following the *epsilon greedy* action selection policy. (1.3) Update the parameter combination according to the respectively selected action in the simulation input xml file in order for the next step in the loop.

Steps (1.1) to (1.3) are repeated, until (2) the internal state memory re-discovers the same parameter combination in two subsequent steps of the loop. This means that the *no manipulation* Action 1 has been found to be the best, which in turn means that a real manipulation of the parameter combination does not yield any better results.

At this point, the parameter manipulation values $\Delta\lambda$ and $\Delta\mu$ are *refined*, i.e., divided by 2, and the loop (1) starts over again. The refinement is repeated arbitrarily often, until the desired accuracy is obtained. – The thus obtained parameter combination should correspond to the actual (*real*) soft tissue material parameter combination.

The source code of this implementation is available open-source on GitHub:
`https://github.com/NicolaiSchoch/rlalgo-for-hf3-esim-calib`

Executing the Q-Learning algorithm for our test scenario, we may observe what is shown in Figures 5 and 6: Figure 5 (left side) visualizes a successfully obtained gradual calibration of soft tissue material parameters. Figure 6 (left side) shows the respective iterative decrease of the obtained RMSE-values for both parameters with 3 refinements.

However, depending on the initial values of our parameter combination, we may also observe what is visualized on the right side in Figure 5. The corresponding plot of the RMSE-values as shown in Figure 6 (right side) reveals the obvious problem:

Since our epsilon greedy approach basically represents a *gradient descent* approach, we may get trapped in *local minima*, where – with the given manipulation values $\Delta\lambda$ and $\Delta\mu$ – no further action is able to improve the respectively current parameter combination. The global minimum can hence not be reached.

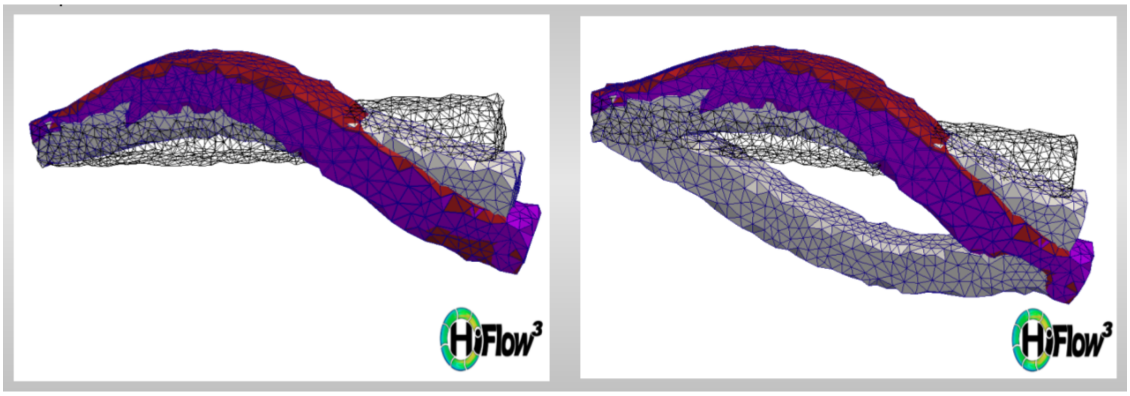


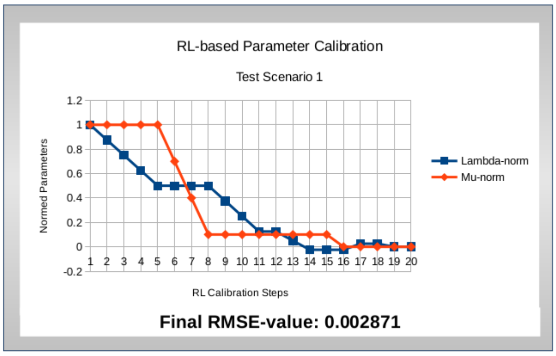
Figure 5: Visualization of the iterative results of the presented calibration algorithm. The wireframe representation shows the initial state of the test object, the violet representation shows the real deformation behavior of the object as extracted from the real test scenario deformation data, and the grey representations show the simulation results after stepwise calibration by means of the suggested RL-algorithm.
(***Left:***) The grey-colored object iteratively approaches the real data through successful gradual material parameter calibration. (***Right:***) The grey-colored object as obtained after a failing material parameter calibration.

Therefore, it is important to combine what is called *exploitation* and *exploration*. Applied to our test scenario, this means that it is necessary to randomly initialize *several* parameter combinations (exploration) and to conduct the calibration for each of them (exploitation). Having obtained the resulting RMSE-values for each of them, one may select the best (optimum), which then represents the global minimum.

The above presented algorithm is not yet too sophisticated, and there are several ways to improve the obtained setup: Obviously, as mentioned earlier, one may deploy a *Neural Network* (NN), in the form of a so-called *Multilayer Perceptron*, in order to learn and approximate the Q-value function. In other words, this means we may train a NN with a large set of random deformation simulation scenarios (i.e. with different material parameter combinations and their associated 3D deformation simulation results) and their corresponding RSME values. A *state* in our test scenario may therefore be represented as a feature vector $\mathbf{x}(s_t)$, containing both the 3D coordinates of the deformed beam object and the associated material parameter combination. And accordingly, the *value function* $V(s_t, \mathbf{w})$ can be computed as $V(s_t, \mathbf{w}) = \mathbf{x}(s_t)^T \cdot \mathbf{w} = \sum_j x_j w_j$. After training, the NN can then be used to *approximate* the Q-value vector, *instead of* computing (and respectively simulating) it as part of step (1.1) in the above Q-Learning algorithm. After the compute-intensive training of the NN, this may have the potential of saving lots of computation time for the calibration, since every execution of step (1.1) previously entailed the expensive execution of 5 simulations (1 for each action). The respective scripts that were used to create the training data sets are available open-source on GitHub, too:
`https://github.com/NicolaiSchoch/rlalgo-for-hf3-esim-calib`

Besides that, we note that there are many more further RL algorithms, some of which may be well suitable for the given problem statement. Therefore, we refer again to the survey in [6]. This being said, we particularly mention two well-established RL libraries which contain a large set of RL algorithms: Keras-RL [13] and VowPalWabbit [7].

Both, the concept of implementing a Multilayer Perceptron in order to learn the Q-value function approximation, and also the idea of using external RL libraries in order for a more efficient RL-based simulation calibration, are subject to ongoing work with our partners at the AIFB at the Karlsruhe Institute for Technology. A dedicated publication is to present further joint results soon.



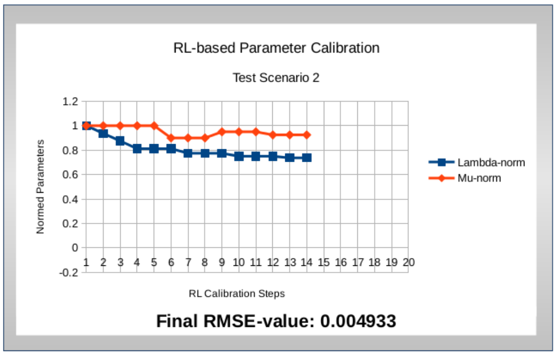


Figure 6: Plot of the iterative calibration-based development of the material parameters $\{\lambda_i, \mu_i\}$. For better understanding and visualization, the parameters have been scaled such that the initial parameter values are at 1.0, and the real parameter values are at 0.0. Hence, optimally, the blue and red lines would go down from 1.0 to somewhere close to 0.0. This is the case in the ***left*** plot, where one can observe that initially, for four iterations through the loop, the best rewards were obtained through manipulation of the λ parameter. Subsequently, actions manipulating the μ parameter were more rewarding up to the point where Action 0 got chosen, which then triggered the first refinement of the parameter manipulation values $\Delta\lambda$ and $\Delta\mu$. Going on as above and refining twice more finally yields a really good approximation of the calibrated parameters to the real parameters, which is indicated through a very low RMSE value of 0.002871. Opposedly, on the ***right*** plot, the two lines do not go down from 1.0 to 0.0, which indicates that the calibration algorithm got trapped in a local minimum, where an RMSE-value of only 0.004933 is obtained.

Short discussion and outlook. To conclude, we want to shortly discuss, what *is* and what *is not* (potentially) possible with this setup? Most obviously, it has to be emphasized here, that the quality assessment of our RL-based material parameter calibration algorithms is conducted on the results of the respectively obtained soft tissue deformations, and here in turn based on the ***RMSE value***, which is a value that is closely related to the so-called DICE coefficient. However, as for the DICE coefficient, it also holds for our RMSE value, that a comparison and quality assessment only works well for rather small deformations with still enough object overlap, such as for instance for a liver or bladder deformation simulation, or for the beam in our test scenario. All of these objects are rather compact. Opposedly, when simulating the deformation behavior of a mitral valve, where large deformations occur to the extremely thin leaflet tissue, the DICE coefficient and the RMSE value quickly rise and get close to 1.0, such that they are not useful anymore. See Figure 7 for illustration.

Besides this, the above suggested test scenario as well as our liver deformation experiments both have rather small ***DoF numbers*** (the 3D objects have less than 12.000 DoFs). For these experiments, our Q-Learning algorithm could achieve full material parameter calibration in less than 45 minutes. However, we expect a notably worse ***performance*** for higher DoF numbers, such that a Q-Learning-based material parameter calibration algorithm such as the above one may not be applicable anymore in reasonable times. Yet, a NN-based Multilayer Perceptron may achieve better results, which is subject to future work.

In this context, one may also need to consider ***Data Assimilation*** techniques [8], which seem promising especially in the realm of calibration of biomechanical models and simulations [12]. They allow, e.g., through Kalman Filters, to specifically factor in the underlying biomechanical model in terms of the PDE, and thus may facilitate data assimilation in a more efficient way based on the integration of this a-priori knowledge.

Lastly, we started to investigate on ways and means to ***integrate expert and domain knowledge*** into the suggested setup. On the one hand, this could be realized through a concept which is commonly known as ***Inverse Reinforcement Learning***. In Inverse RL, a NN is trained by means of data sets

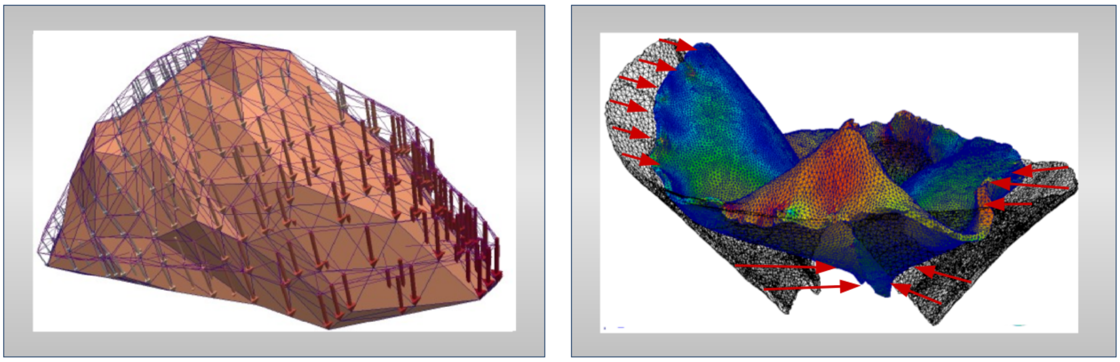


Figure 7: Comparison of two simulated objects subjected to deformation. On the ***left***, a liver, which is a rather compact object, where rather small deformations occur. The resulting RMSE values for different material parameter combinations usually are meaningful and applicable for comparison and fitting measurements, since there is still lots of overlap between the two deformed objects. Opposed to this, on the ***right***, a mitral valve object, which consists of very thin and loose tissue, and rather large deformations occur. Here, small manipulations of the material parameter combinations have a huge impact on the deformation behavior, and quickly affect completely different deformation results which do *not* overlap anymore at all, such that the RMSE values for these respectively different parameter combinations are *not* meaningful for comparison anymore.

that were obtained from observing the course of actions which a simulation expert (e.g., ourselves) takes throughout the entire simulation setup and calibration process. Hence, in the data sets, we find recorded, how initial parameter combinations were guessed by a simulation expert, and also how and on which basis – after each obtained interim simulation result – adaptations to the material parameter combinations were made. Yet, a well-known problem in setting up an Inverse RL framework is the training: Large amounts of training data are needed in order to properly optimize the weights of the NN. The acquisition of these training data sets is however very expensive, as a real human simulation expert has to be observed over a long series of exhaustive, manual simulation calibration processes.

Nevertheless, the concept of Inverse RL is promising and certainly worth implementing and evaluating in a suitable application context, such that we plan to further elaborate this joint idea along with our partners at the AIFB at the Karlsruhe Institute for Technology. The thus obtained results will also be presented in a dedicated publication.

On the other hand, and this will yield the transition to Section 2.2, one may bridge to ***knowledge models***, i.e., ontologies, in order to represent and consider domain and expert knowledge as well as commonly avowed uncertainties.

Knowledge models allow to formally collect, structure and describe (domain) knowledge, e.g., about the considered soft tissue material parameters. As such, knowledge about *quantitative relations* between parameters or about *physical restrictions* of the increase or decrease of parameters may, for instance, be represented in a linked knowledge model, in order to thus guarantee for a better initialization of the parameters in step (0) of the above algorithm. Similarly, this formalized knowledge may restrict unphysical random manipulations such as affected through the above actions 1 to 5, if parameters grow too big or decrease below a certain unphysical threshold. Last but not least, one can represent expert knowledge about typical *uncertainties* with respect to these parameters themselves, their measurement processes, and to their imaging-based acquisition. Like this, both tolerance values and suitable simulation samples can adequately be determined and implemented in the algorithm.

2.2 Knowledge Management and a Surgery Simulation Ontology

In this subsection, we first re-introduce the general idea of knowledge models and ontologies. Thereafter, we review the benefits which our work on cognition-guided, patient-specific, simulation-enhanced cardiac surgery assistance has obtained through being an ontology-driven system. This gives the motivation to further elaborate on the respective surgery simulation knowledge model towards a heavyweight ontology, which is presented subsequently. Lastly, we shortly discuss our work and give a brief outlook.

Knowledge models and ontologies. Knowledge models are to capture, organize, structure and interrelate knowledge and information in a formal way. In this respect, an *ontology* – according to Gruber (1993) – is an explicit specification and formalization of a shared conceptualization [4].

The typical *building blocks* of an ontology are classes, instances/individuals, rules, restrictions, properties, axioms and relations. These allow to abstractize and generalize things or entities or properties, etc., and to group them together, to categorize them, etc.

Previous works seen from the ontology perspective. In our previous works, we have designed and implemented a simulation-enhanced, cognition-guided, patient-specific cardiac surgery assistance system prototype [15]. It is part of the cognitive system architecture of the joint research project *Cognition-Guided Surgery* (SFB TRR 125). Its semantic knowledge and data infrastructure [3] as well as the setup of its cognitive software components [19] facilitate *cognition-guidance* and *patient-specifity* for the overall simulation-enhanced MVR assistance pipeline which was introduced in Section 1, see again Figure 1. By means of the *cognitive, data-driven pipeline*, medical patient data and surgical information is analyzed and processed comprehensively, efficiently and fully automatically, in order to finally set up patient-individual MVR simulation scenarios.

Apart from this work, we have proposed a first concept for *semantic surgery simulation* in [20]. The approach aims at simplifying the usability of surgery simulations via a semantic representation of simulation properties and an underlying numerics properties decision tree. Initially, this approach was designed to be linked and combined with the before-mentioned works, and to thus complement and augment the overall system.

From an ontology point of view, looking at the surgery assistance pipeline in Figure 1, we found that – through the backup of the underlying lightweight ontology – our ***(lightweight) ontology-driven system*** facilitates that:

- our cognitive software components are automatically triggered as part of a data-driven information proceesing pipeline,
- comprehensive patient data and information processing is fully automated,
- the knowledge base can be queried, and reasoning as well as knowledge retrieval is facilitated,
- the general simulation setup and simulation preprocessing is autonomously controlled,
- the processing of data and information is based on medical/surgical evidence.

When adding our Machine Learning component (as presented in Section 2.1) to this lightweight ontology-driven system, the resulting advantages that are obtained through the ontology-layer quickly become apparent. We are able to:

- automate the fitting of real patient data to simulation data, for the calibration of simulation material parameters,
- account for domain and expert knowledge, and
- account for sources of uncertainties, as represented in the ontology.

Towards a heavyweight surgery simulation ontology. Given this long list of ontology-based benefits, in our recent work, we intended to further develop the above-mentioned lightweight surgery simulation ontology [20] towards a more profound knowledge management framework for ontology-driven, cognition-guided, patient-individual, simulation-enhanced surgery assistance. We thus worked towards

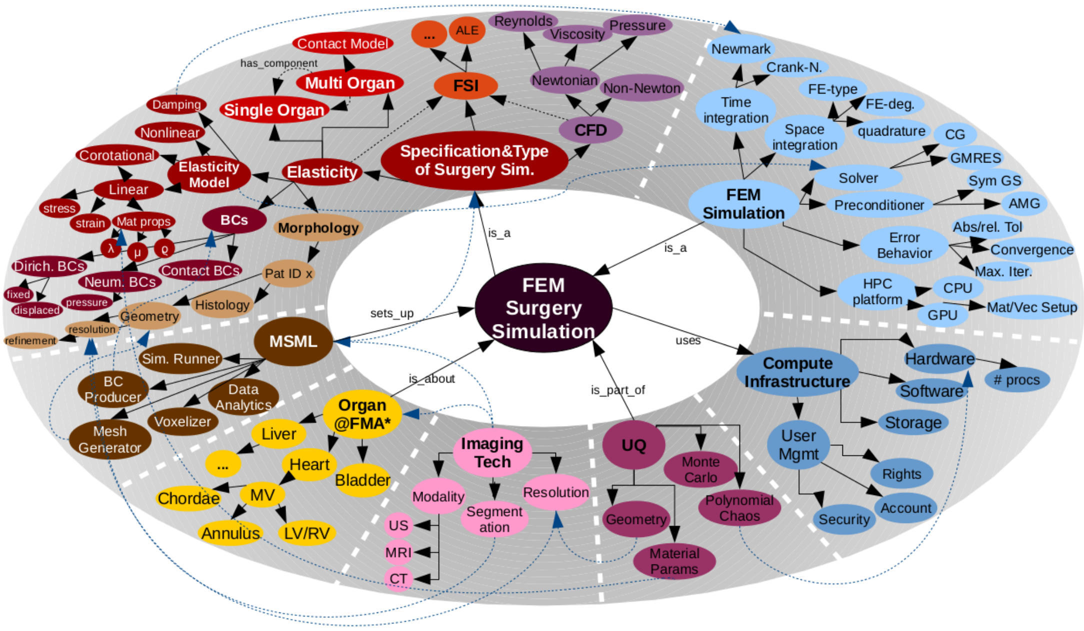


Figure 8: Radial structure diagram visualization of the developed ***FEM surgery simulation ontology***.

a sophisticated, formal, heavyweight ontology, which is mathematically more rigorous, machine readable and still consistent with the before-mentioned developments in the context of the SFB TRR 125 *Cognition-Guided Surgery*.

Therefore, we *captured unified and agreed-upon knowledge*, both for the *surgical and anatomical* as well as for the *biomechanical modeling and simulation* context. The obtained knowledge model thus, e.g., defines meanings, classes, instances, dependencies and relations between concepts and things, and explicitly specifies the respective domain knowledge, such as that an FEM surgery simulation is a FEM simulation which in turn necessarily entails a space integration mechanism and which possibly involves a time integration scheme in case of a dynamic, i.e., instationary, simulation. Similarly, for the surgical context, one may, for instance, formally state that a patient's organ morphology is always obtained from the same patient's organ segmentation, which in turn is always obtained from an imaging technique such as ultrasound which was applied again to the same respective patient.

Using the open-source ontology development environment ***Protégé*** [10], we formalized the concept of our knowledge model in the ***Ontology Web Language*** (OWL) [5]. See Figure 8 for a visualization of the obtained ontology in the form of a radial structure diagram.

In the center, we have the core idea of the ontology, the `FEM Surgery Simulation`, which inherits its general features and functionalities from the class `FEM Simulation` (bright blue) but is closely linked to `surgery` via the concepts of the *Foundational Model of Anatomy* (FMA) [14] (yellow). The class `FEM Simulation`, in turn, comprises all features and properties that make up a general FEM Simulation, starting from time and space integration, via solvers and preconditioners, to the approximation error behavior and the exploited High-Performance Computing platform. An instance of an FEM Surgery Simulation can then be further subdivided into several specifications and types of surgery simulations (red/reddish/violet), such as `CFD` or `elasticity` simulations, or, combining these two, an `FSI` simulation, etc.

When exemplarily looking at an elasticity simulation, it generally specifies an `elasticity model`, `boundary conditions` and the object's `morphology`. All of these are again related and linked to other classes and concepts, via rules, inheritance relations, or other dependencies. For instance, the `boundary conditions` are created through the MSML and its comprehensive set of simulation preprocessing operators (brown), and material properties as well as morphology information is obtained from the FMA (yellow) and from imaging techniques (pink).

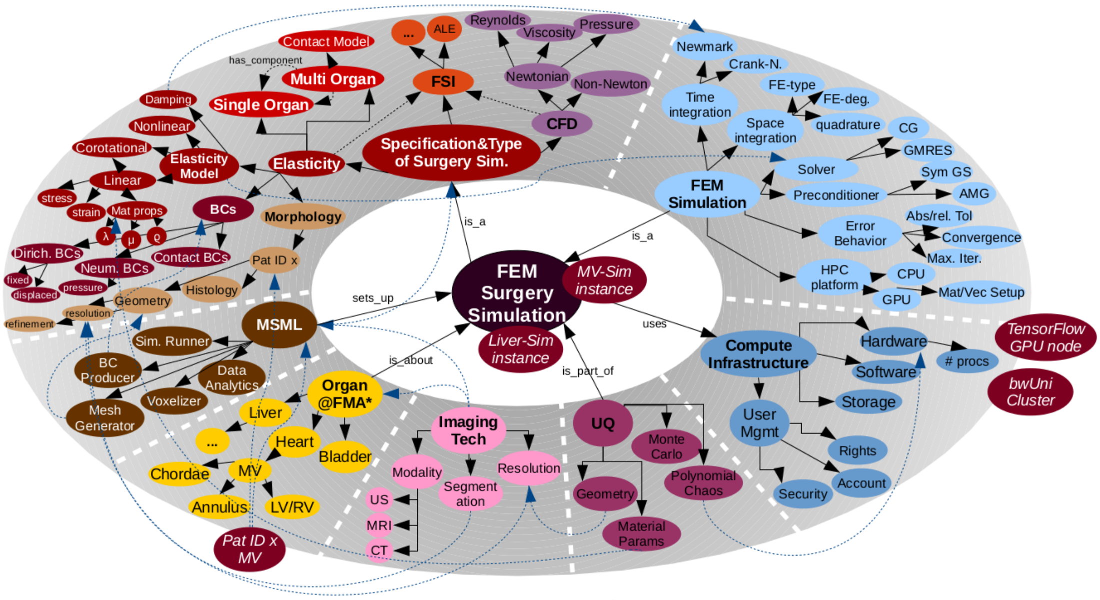


Figure 9: Radial structure diagram visualization of the developed ***FEM surgery simulation ontology***, including a small set of possible instantiations (in dark red).

In particular, we emphasize the concept of `Uncertainty` and of `Uncertainty Quantification` (violet), which, e.g., is connected via geometry and material property information to the respective classes and instances in the field of `elasticity simulations` (red).

So far, this has been *purely abstract knowledge modeling*, and the presented *knowledge model* or *ontology* has not yet been applied. In order to apply it, it is scaled up (i.e., further specified and extended) and deployed in a repository, a *knowledge base*. Then, coming back to the MVR surgery assistance use case and applying our FEM surgery simulation ontology, we can start – as a user of this ontology – to create information, i.e., real instances, and take them to populate the knowledge base. And in the other direction, the user may – by means of a front end application which connects via an interaction layer to the ontology deployment environment – query the knowledge base to retrieve information, e.g., for decision support:

We could, e.g., create an instance `male Patient X`, and let him have a `liver simulation` and a `mitral valve simulation`, see Figure 9. And these simulations may be executed on a specific compute infrastructure instance, such as on a `TensorFlow GPU node` or on the `bwUniCluster` at the KIT, Germany.

With these instances being created, our ***ontology-driven system*** is now able to *autonomously* go on, i.e., infer missing knowledge about the respective instances through querying, reasoning and general knowledge retrieval, and further process all information all through the entire surgery assistance pipeline as depicted in Figure 1. We have motivated the respective steps at the beginning of this section, and their functionalities and results were shown as well as elaborted on in our previous works [3, 19, 20].

Short discussion and outlook. Having developed this first prototype of a heavyweight knowledge model for FEM surgery simulation, our knowledge model has to be evaluated. For the ***evaluation of our ontology***, we plan to stick to the suggested methodology in [11], and want to answer the following questions:

- Are *goal*, *scope* and *range* of our ontology fulfilled? - I.e., is it adequate in the context of our desired surgery simulation application field? Has the domain properly and comprehensively been represented? Have concepts, classes, individuals and relations been properly generalized and linked with each other?

- Can we answer the *competency questions* we asked in the initial phase of the ontology development? - I.e., is it user-oriented, and can the answers be given through querying the ontology and the associated knowledge base?
- Is the ontology *complete*, *consistent* and *logically set-up*? - Is the ontology development environment-based consistency checking successful?
- Is *compatibility* with the employed *reasoning engines* given? - I.e., can new knowledge be inferred through reasoning?
- And finally, can our ontology stand its ground when fully *integrated* into our cognition-guided, patient-specific, simulation-enhanced cardiac surgery assistance system prototype [15]? - Does it improve and enhance the earlier system setup? - Is medical, surgical and technical evidence obvious?

In order to answer the last question, we plan to validate the complete setup by means of surgery application scenarios motivated through our clinical research partners, e.g., in cardiac, vascular and laparoscopic surgery. Moreover, to evaluate the overall ontology-driven data and information processing as well as the compatibility with reasoning engines, the single pipeline components can be evaluated separately, e.g., by means of testing the machine results versus those from a human domain expert.

3 Summary, Conclusion and Outlook

We conclude with a short summary of the presented research and development.

Building on top of our previous works on cognition-guidance and patient-specifity for simulation-enhanced cardiac surgery assistance, we have evolved and further developed the given setup in this work and particularly investigated on

1. a machine learning framework which allows to patient-individually calibrate soft tissue material parameters for subsequent simulation, and
2. a profound knowledge management framework which may enhance the ontology-driven overall setup of the cognition-guided surgery simulation in a clinic environment.

We have presented our ideas, concepts and implementations both in the machine learning context and in the knowledge modeling context. However, rather than being a closed research work with an in-depth theory backup and a complete evaluation, we here presented a technical report and some interesting experimental works that are to serve for further research and development.

Putting all building blocks of our research together, we obtain what is visualized in Figure 10. The three left bottom blocks denote our previous works on the ***biomechanical surgery simulation*** itself [16, 17], on the ***MSML-based simulation preprocessing*** [18, 22], and on the ***semantic data infrastructure*** which our ***cognitive software applications*** work on [3, 19]. Through complete integration of these three blocks, we are able – to some extend – to set up cognition-guided and patient-specific surgery simulation scenarios for cardiac surgery assistance [15].

In addition to this, the right bottom block now represents the here suggested ***Machine Learning framework*** unit, which we designed in order to fine-calibrate the above obtained cognition-guided, patient-specific surgery simulation setup through calibration of the soft tissue material parameters, see Section 2.1. Like this, we intend to achieve (or, at least go towards) a fully personalized surgery simulation for cardiac surgery assistance purposes.

Adding as a top layer onto all these blocks our ***surgery simulation knowledge model / ontology***, as presented in Section 2.2, we finally enable ***fully automated, fully personalized, intelligent surgery simulation assistance***, Figure 10.

In terms of future work, we see the following items:

- evolve the Reinforcement Learning algorithms and simulation calibration setup,
- evaluate the Machine Learning-based simulation calibration approach versus a Data Assimilation-based simulation calibration approach,
- evaluate the proposed surgery simulation ontology, desirably in the context of surgical applications.

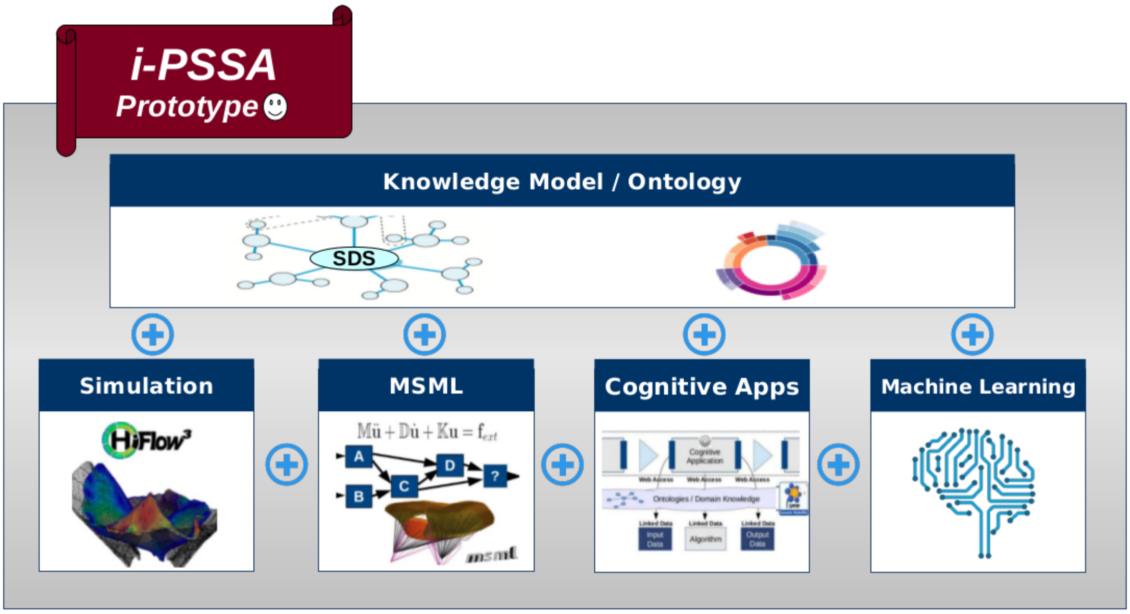


Figure 10: Illustration of the overall setup and the contained closely connected building blocks of our ***intelligent Personalized Simulation-enhanced Surgery Assistance*** system prototype, in short: ***i-PSSA***.

Lastly, we want to emphasize again, that this work is to be seen as a technical report on our recent experimental works. In-depth studies on the underlying theory and an exhaustive evaluation are, however, missing so far, and up to further research.

Acknowledgements:
We gratefully acknowledge Patrick Philipp and York Sure-Vetter from the *Institut für Angewandte Informatik und Formale Beschreibungsverfahren* (AIFB) at the *Karlsruhe Institute of Technology* (KIT), Karlsruhe, Germany. We have conducted intense conversations and enlightening discussions on both Machine Learning and Knowledge Modeling. Building onto the here presented ideas and prototypes, we may assume to obtain promising joint future works in this field.

References

[1] H. Anzt, W. Augustin, M. Baumann, T. Gengenbach, T. Hahn, A. Helfrich-Schkarbanenko, V. Heuveline, E. Ketelaer, D. Lukarski, A. Nestler, S. Ritterbusch, S. Ronnas, M. Schick, M. Schmidtobreick, C. Subramanian, J.-P. Weiss, F. Wilhelm, and M. Wlotzka. HiFlow^3 – a hardware-aware parallel finite element package. *Tools for High Performance Computing 2011*, pages 139–151, 2012.

[2] A. Carpentier, D. Adams, and F. Filsoufi. *Carpentier's Reconstructive Valve Surgery*. Saunders W.B. Saunders/Elsevier, 2010.

[3] A. Fetzer, J. Metzger, D. Katic, K. März, M. Wagner, P. Philipp, S. Engelhardt, T. Weller, S. Zelzer, A.M. Franz, N. Schoch, V. Heuveline, M. Maleshkova, A. Rettinger, S. Speidel, I. Wolf, H. Kenngott, A. Mehrabi, B. Müller, L. Maier-Hein, H.-P. Meinzer, and M. Nolden. Towards an open-source semantic data infrastructure for integrating clinical and scientific data in cognition-guided surgery. *Proc. SPIE. Medical Imaging 2016*, 9789:978900–978908, 2016.

[4] T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[5] P. Hitzler, M. Kroetzsch, B. Parsia, P.F. Patel-Schneider, and S. Rudolph. Owl 2 – web ontology language: Primer (second edition), 2012.

[6] L.P. Kaelbling, M.L. Littman, and A.W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research 4*, pages 237–285, 1996.

[7] John Langford. Vowpal wabbit. `https://github.com/JohnLangford/vowpal_wabbit`, 2016.

[8] K.J.H. Law, A.M. Stuart, and K.C. Zygalakis. *Data Assimilation: A Mathematical Introduction.* Springer, 2015.

[9] L. Maier-Hein, S. Vedula, S. Speidel, N. Navab, R. Kikinis, A. Park, M. Eisenmann, H. Feussner, G. Forestier, S. Giannarou, M. Hashizume, D. Katic, H. Kenngott, M. Kranzfelder, A. Malpani, K. Maerz, T. Neumuth, N. Padoy, C. Pugh, N. Schoch, D. Stoyanov, R. Taylor, M. Wagner, G.D. Hager, and P. Jannin. Surgical data science: Enabling next-generation surgery. *Nature Biomedical Engineering*, 2017.

[10] N.F. Noy and D.L. McGuinness. Ontology development 101: A guide to creating your first ontology, 2000.

[11] L. Obrst, B. Ashpole, W. Ceusters, I. Mani, S. Ray, and B. Smith. The evaluation of ontologies: Toward improved semantic interoperability. *Revolutionizing Knowledge Discovery in the Life Sciences*, pages 139–158, 2007.

[12] I. Peterlik and A. Klima. Towards an efficient data assimilation in physically-based medical simulations. *IEEE International Conference on Bioinformatics and Biomedicine (BTBM) 2015*, 2015.

[13] Matthias Plappert. keras-rl. `https://github.com/matthiasplappert/keras-rl`, 2016.

[14] C. Rosse and J.L.V. Mejino. The foundational model of anatomy ontology. *Anatomy Ontologies for Bioinformatics: Principles and Practice*, 6:59–117, 2007.

[15] N. Schoch. Towards cognition-guided patient-specific numerical simulation for cardiac surgery assistance (phd thesis). *Heidelberg Document Server HeiDOK*, 2017.

[16] N. Schoch, S. Engelhardt, R. De Simone, I. Wolf, and V. Heuveline. High performance computing for cognition-guided cardiac surgery: Soft tissue simulation for mitral valve reconstruction in knowledge-based surgery assistance. *Modeling, Simulation and Optimization of Complex Processes, In: Proc. High Performance Scientific Computing (HPSC) 2015*, 2015.

[17] N. Schoch and V. Heuveline. Towards cognition-guided patient-specific fem-based cardiac surgery simulation. *Springer's LNCS Lecture Notes in Computer Science, Proceedings of FIMH 2017*, 10263:115–126, 2017.

[18] N. Schoch, F. Kißler, M. Stoll, S. Engelhardt, R. de Simone, I. Wolf, R. Bendl, and V. Heuveline. Comprehensive patient-specific information preprocessing for cardiac surgery simulations. *Int J CARS*, 11(6):1051–1059, 2016.

[19] N. Schoch, P. Philipp, T. Weller, S. Engelhardt, M. Volovyk, A. Fetzer, M. Nolden, R. de Simone, I. Wolf, M. Maleshkova, A. Rettinger, R. Studer, and V. Heuveline. Cognitive tools pipeline for assistance of mitral valve surgery. *Proc. SPIE. Medical Imaging 2016*, 9786:978603–978603–8, 2016.

[20] N. Schoch, S. Speidel, Y. Sure-Vetter, and V. Heuveline. Towards semantic simulation for patient-specific surgery assistance. *Online Proceedings of the International Workshop on Surgical Data Science 2016*, 2016.

[21] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction.* The MIT Press, Cambridge, Massachusetts, London, England, 2017.

[22] S. Suwelack, M. Stoll, S. Schalck, N. Schoch, R. Dillmann, R. Bendl, V. Heuveline, and S. Speidel. The medical simulation markup language – simplifying the biomechanical modeling workflow. *Journal on Studies in Health Technology and Informatics*, 196:394–400, 2014.

recent issues

No. 2017-04	Martin Wlotzka, Thierry Morel, Andrea Piacentini, Vincent Heuveline: New features for advanced dynamic parallel communication routines in OpenPALM: Algorithms and documentation
No. 2017-03	Martin Wlotzka, Vincent Heuveline: An energy-efficient parallel multigrid method for multi-core CPU platforms and HPC clusters
No. 2017-02	Thomas Loderer, Vincent Heuveline: New sparsing approach for real-time simulations of stiff models on electronic control units
No. 2017-01	Chen Song, Markus Stoll, Kristina Giske, Rolf Bendl, Vincent Heuveline: Sparse Grids for quantifying motion uncertainties in biomechanical models of radiotherapy patients
No. 2016-02	Jonas Kratzke, Vincent Heuveline: An analytically solvable benchmark problem for fluid-structure interaction with uncertain parameters
No. 2016-01	Philipp Gerstner, Michael Schick, Vincent Heuveline, Nico Meyer-Hbner, Michael Suriyah, Thomas Leibfried, Viktor Slednev, Wolf Fichtner, Valentin Bertsch: A Domain Decomposition Approach for Solving Dynamic Optimal Power Flow Problems in Parallel with Application to the German Transmission Grid
No. 2015-04	Philipp Gerstner, Vincent Heuveline, Michael Schick : A Multilevel Domain Decomposition approach for solving time constrained Optimal Power Flow problems
No. 2015-03	Martin Wlotzka, Vincent Heuveline: Block-asynchronous and Jacobi smoothers for a multigrid solver on GPU-accelerated HPC clusters
No. 2015-02	Nicolai Schoch, Fabian Kiler, Markus Stoll, Sandy Engelhardt, Raffaele de Simone, Ivo Wolf, Rolf Bendl, Vincent Heuveline: Comprehensive Pre- & Post-Processing for Numerical Simulations in Cardiac Surgery Assistance
No. 2015-01	Teresa Beck, Martin Baumann, Leonhard Scheck, Vincent Heuveline, Sarah Jones: Comparison of mesh-adaptation criteria for an idealized tropical cyclone problem
No. 2014-02	Christoph Paulus, Stefan Suwelack, Nicolai Schoch, Stefanie Speidel, Rdiger Dillmann, Vincent Heuveline: Simulation of Complex Cuts in Soft Tissue with the Extended Finite Element Method (X-FEM)
No. 2014-01	Martin Wlotzka, Vincent Heuveline: A parallel solution scheme for multiphysics evolution problems using OpenPALM
No. 2013-04	Nicolai Schoch, Stefan Suwelack, Stefanie Speidel, Rediger Dillmann, Vincent Heuveline: Simulation of Surgical Cutting in Soft Tissue using the Extended Finite Element Method (X-FEM)
No. 2013-03	Martin Wlotzka, Edwin Haas, Philipp Kraft, Vincent Heuveline, Steffen Klatt, David Kraus, Klaus Butterbach-Bahl, Lutz Breuer: Dynamic Simulation of Land Management Effects on Soil N2O Emissions using a coupled Hydrology-Ecosystem Model
No. 2013-02	Nicolai Schoch, Stefan Suwelack, Rüediger Dillmann, Vincent Heuveline: Simulation of Surgical Cutting in Soft Tissue using the Extended Finite Element Method (X-FEM)

Preprint Series of the Engineering Mathematics and Computing Lab (EMCL)

www.emcl.iwr.uni-heidelberg.de