Expert-based Probabilistic Modeling of Workflows in Context of Surgical Interventions

Patrick Philipp and Jürgen Beyerer
Vision and Fusion Laboratory IES
Karlsruhe Institute of Technology KIT
Karlsruhe, Germany
Email: p.philipp@kit.edu

Yvonne Fischer and Jürgen Beyerer
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB
Karlsruhe, Germany

***Abstract*—To provide assistance functions in context of surgical interventions, the use of medical workflows plays an important role. Workflow models can be used to assess the progress of an on-going surgery, enabling tailored (i.e., context sensitive) support for the medical practitioner. Subsequently, this provides opportunities to prevent malpractices, to enhance the patient's outcome and to preserve a high level of satisfaction. In this work, we propose a framework which enables a formalization of medical workflows. It is driven by a dialog of medical as well as technical experts and is based on the Unified Modeling Language (UML). An easy comprehensible UML activity serves as a starting point for the automatic generation of more complex models that can be used for the actual estimation of the progress of a surgical intervention. In this work, we present translation rules, which allow to transfer a given UML activity into a Dynamic Bayesian Network (DBN). The methods are presented for the application example of a cholecystectomy (surgical removal of the gallbladder).**

Keywords: Surgical Workflow, Dynamic Bayesian Networks, Unified Modeling Language, Assistance, Planning

I. INTRODUCTION

In contemporary medicine, the use of advanced computer-based assistance (including both software and hardware) has become an evolving area [1]. E.g. the global market for medical robotics and computer-assisted surgical equipment is projected to grow to 4.6 billion dollars by 2019 (using a five-year compound annual growth rate of 7%) [2].

Regarding the example case of surgical interventions, a computer-based assistance can be realized in various forms – therefore we outline three typical forms of assistance and their relation to situation detection in the next subsections.

In robot assisted surgery [3], where medical robots perform specific sub-actions of a surgery, the assistance should ideally seamlessly integrate into the workflow of the human operating team. Basis is the assessment of the progress of the surgery, i.e., the current phase of an underlying medical workflow. Another form of computer-based assistance in medical context are the initiation of administrative processes (e.g. automatic notification of the medical ward that a surgery ends soon), or the support in preparation of a surgical report (e.g. providing start and end time of a surgery) [4]. Both forms strongly depend on the detection of the progress of an on-going surgery as well as the knowledge of a pre-defined surgical workflow.

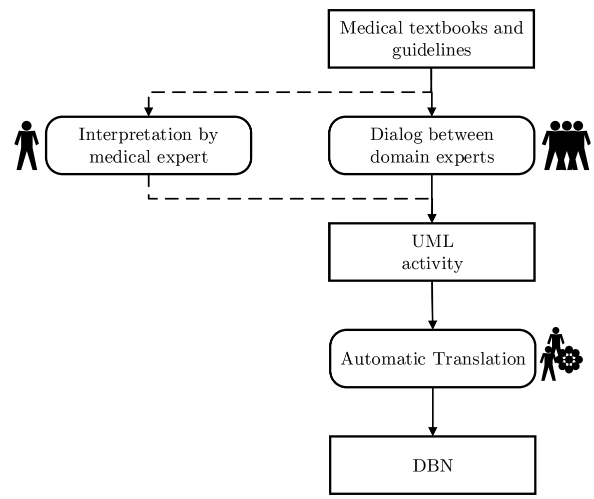

Fig. 1. Medical knowledge can be transformed into a UML activity of the surgical workflow. The process of formalization can be carried out by an expert's dialog or by the medical expert himself. The resulting activity serves as an interface to more complex models used for the actual detection of a surgical step. In this work, we introduce translation rules, transferring a given UML activity into a Dynamic Bayesian Network (DBN).

To tackle the challenge of the huge amount of available data (e.g. patient records or surgical guidelines including specific actions for emergency procedures), an assistance providing tailored (e.g. context sensitive) information during an on-going surgery is of interest [5]. Thereby, the current progress of a surgery and possible derivations of a pre-defined surgical workflow are considered to support the operating team (e.g.) with suitable examination results or options for actions.

The guidance of an operating team opens up a scope for optimization. This includes for example the decision support of the surgeon. Overall, this can help to prevent malpractices, enhance the patient's outcome and preserve a high level of satisfaction for patients as well as employees.

II. Modeling Approach

Consequently, key stones of a tailored assistance, taking the specific conditions of an on-going surgery into account, are: Firstly an appropriate representation of the underlying workflow; secondly, a reliable detection of the current progress by synchronizing this workflow model with sensory data.

Addressing the first challenge of a proper workflow representation, we introduce a framework for an expert-based formalization of medical knowledge. The Unified Modeling Language (UML) is used to provide a graphical representation of the workflow by UML activities, facilitating the dialog of technical and medical experts.

For the second challenge of a reliable progress detection, we elaborate the use of a probabilistic model based on Dynamic Bayesian Networks (DBNs). In order to link these two necessities, we propose a framework that allows for the automatic translation of UML activities into DBNs. To model workflows by a UML activity, medical knowledge has to be condensed and formalized. This formalization process can be realized via a dialog of experts from the medical, and the technical domain, c.f. Fig. 1.

Due to their easy comprehensibility, UML activities also offer the possibility for the medical expert to modify an already formalized workflow by him- or herself. With the activity representation serving as a bypass, we believe that barriers can be reduced: The fact of practitioners being able to understand and modify the formalized representation of the workflow by themselves, reduces their fear of regimentation and allows them, e.g., to easily adapt a workflow to specific boundary conditions. For the actual implementation of an assistance function that supports the practitioner during a surgery, we translate UML activities into more complex models. The structure and parameters of these models are automatically generated.

The idea of combining UML activities with other, more complex models is the subject of various research [6], [7], [8]. E.g. Störrle et al. [9], [10] and Agrawal [11] introduced a translation of UML activities into Petri Nets, and therefore linking the usability of UML modeling with the analytic power of the target model. To our knowledge, none of the present research facilitates UML activities in context of a dialogue of experts from different domains (e.g. technical and medical domain experts). The same applies to the translation of UML activities into DBNs. In [1] we introduced the translation from UML activities of diagnostic processes into Bayesian Networks (BNs). In this paper we show how this approach can be adapted to surgical workflows. I.e., this is done by extending the translation of UML activities to Dynamic Bayesian Networks (DBNs). We evaluate the resulting DBNs with a common use case for surgical process modeling [12]: a cholecystectomy (the surgical removal of the gallbladder). Our approach can easily adapted to other use cases, e.g. a hip replacement [13].

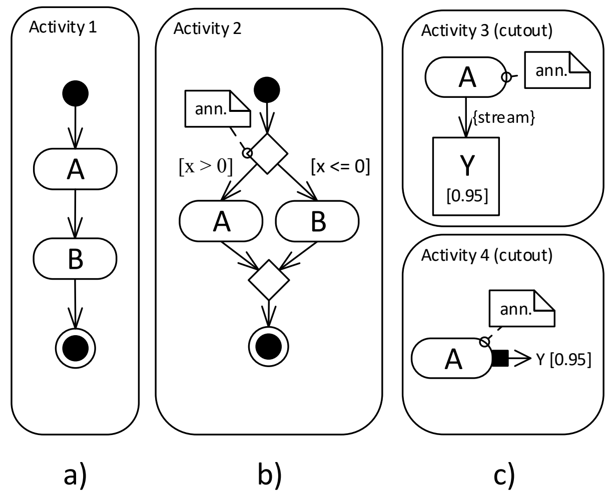

Fig. 2. Two typical routings of workflow models in context of surgical interventions. In a) the actions A and B are performed one after another (sequentially). In b), a decision has to be made in order to perform either A or B. The decision is represented by a guard (expression). If it evaluates to true, the corresponding action is performed. c) shows to equivalent representation of a streamed object. I.e. an object flow can occur during A is performed. Furthermore, both activities contain a textual annotation which can be used to store additional information.

III. UML activities

UML is used and accepted in industry worldwide [14]. UML activities have been chosen as an interface because of their easy comprehensibility for the medical- as well as the technical domain experts [15]. The understanding of the workflow representation is a necessary precondition for the dialog of experts.

A. Fundamentals

UML activities can be used to graphically represent how a particular process or algorithm proceeds [19]. The activity diagrams are constructed from different shapes that are connected with directed edges (arrows) and support the modeling of sequences of actions, as well as of choice, iteration, and concurrency [14].

Fig. 2 shows typical routings which may appear in workflows, such as a cholecystectomy. The start of an activity is represented by a black dot (initial node), the end of an activity is symbolized by a double circle (activity final). Single actions of the workflow are symbolized by rounded rectangles. Arrows represent the flow and thereby the order in which actions can be performed.

While Subfigures a) and b) of Fig. 2 show a two different kind of routings, Subfigure c) depicts two equivalent parts of an activity. The fact that an object flow can occur while action A is performed (and not only after A is finished) is specified by the keyword {stream}. An abbreviated form is a black pin, which is depicted in activity 4. Both activities in Subfigure c) contain a textual annotation associated with action A to store additional information.

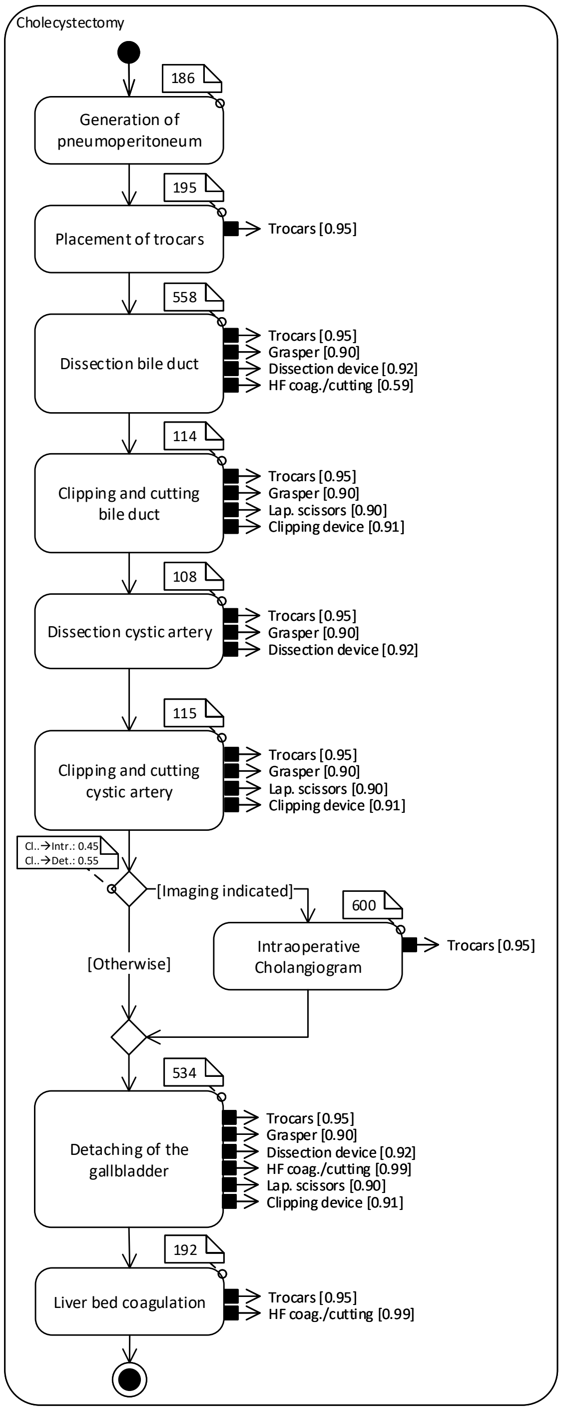

Fig. 3. The figure shows the UML activity of a cholecystectomy. At the beginning there is a sequential order of actions. The action "Intraoperative Cholangiogram" is optionally performed. The activity is derived from a description in ref. [16], [17] and [18].

B. Modeling of a surgical workflow

Fig. 3 depicts the formalized model of a cholecystectomy (surgical removal of the gallbladder). In the upper part of this UML activity, a sequence of actions is shown. The first action

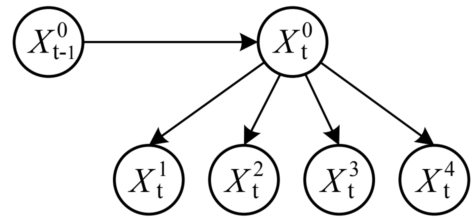

Fig. 4. Simplified graph of a 2-Slice Temporal Bayesian Network (2TBN). In context of a DBN, a 2TBN (or: $B_{\rightarrow}$) is used as template for consecutive time steps t. For simplification, we omitted $X^{1:4}_{t-1}$, since in this example, there is no direct dependency to $X^{0:4}_{t}$.

is the injection of carbonic acid gas to inflate the abdomen (stomach). The mean time of this action is represented by the corresponding annotation. Then the trocars (sharpened tubes) are used to break trough the abdominal wall. E.g., trocars are used to enable the placement of additional medical instruments during the surgery. Black pins represent the observation probability of trocars in this action (cf. Fig. 3).

In subsequent actions, the cystic duct and the artery are exposed, clipped and dissected (divided by cutting). An intraoperative cholangiogram (radiographic imaging of the bile ducts with contrast medium) is optionally performed, therefore a decision has to be made. After merging the two possible flows, the gallbladder is dissected from the liver bed and the gallbladder is extracted.

IV. Dynamic Bayesian Networks

There are many reasons for considering Dynamic Bayesian Networks (DBNs) [20] as a modeling tool for dynamic systems [21]. With respect to the modeling of medical workflows, we opted for DBNs because they combine a reasonable tradeoff between expressiveness and complexity, and include probabilistic models that have proved successful in practice (e.g. Hidden Markov Models) [12]. Additionally, due to their factorized state space, DBN models allow improved modularity and interpretability. In contrast to a Hidden Markov Model, their state space can be expressed in a factored form and not only as a single discrete random variable. Furthermore, concerning Kalman Filter Models, DBNs allow for arbitrary probability distributions (not only for unimodal linear-Gaussians) [18].

Whereas DBNs provide a good tradeoff between expressiveness and complexity to a knowledge engineer, the formal representation of a DBN is less suited for a joint dialog of a medical and a technical domain expert. Moreover, the modification of the model by the medical expert himself seems not to be feasible. Consequently, we introduce translation rules, transferring a comprehensible UML activity into a DBN.

A. Fundamentals

A Bayesian network (BN) is a probabilistic graphical model (PGM), combining graph theoretic approaches with approaches of probability theory [22]. A Dynamic Bayesian

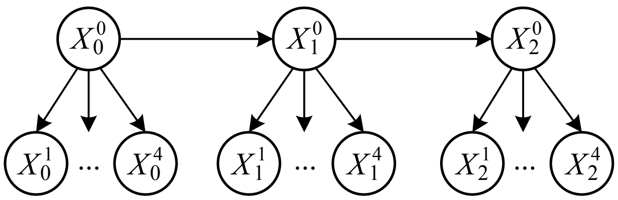

Fig. 5. Example of a DBN unrolled for 3 slices. For simplification some nodes are omitted, which is graphically represented by "...".

network (DBN) is an extension of a BN, also taking the temporal dependencies of variables into account [21]. A DBN is given by a pair

$$DBN = (B_0, B_\rightarrow),$$

where the Bayesian Network B_0 uses $P(X_0^{0:N})$ to specify the a-priori probability distribution over random variables $X^{0:N}$ in a time step with index 0. Furthermore, $B_\rightarrow$ specifies the conditional probability distribution over discrete time steps t by using

$$P(X_t^{0:N} | X_{t-1}^{0:N}) = \prod_{n=0}^{N} P(X_t^n | \mathrm{Pa}(X_t^n)) . \quad (1)$$

In (1), $\mathrm{Pa}(X_t^n)$ is the set of X_t^n's parents in the corresponding graph. The parents can be in the same time slice (e.g. representing instantaneous causation) or the previous one (i.e. we assume the model to be fist-order Markov). In the latter case, arcs point to time slices with ascending index, reflecting the causal flow of time [21]. Compare Fig. 4 for an exemplary graph of $B_\rightarrow$ which is also known as a two-slice temporal Bayesian network (2TBN).

For T time-slices, the joint distribution can be graphically represented by an "unrolling" of the DBN, using B_0 as the initial distribution and $B_\rightarrow$ as template for each following time slice. Refer Fig. 5 for a DBN unrolled for 3 time slices. Similar to HMMs, parameters of such a DBN, having N children, can be grouped as follows (cf. Fig. 5):

$$P(X_0^0 = i) = \boldsymbol{\pi}(i) \quad (2)$$
$$P(X_t^0 = j | X_{t-1}^0 = i) = \boldsymbol{A}(i, j) \quad (3)$$
$$P(X_t^1 = j | X_t^0 = i) = \boldsymbol{B}^{(1)}(i, j) \quad (4)$$
$$\ldots$$
$$P(X_t^N = j | X_t^0 = i) = \boldsymbol{B}^{(N)}(i, j) \quad (5)$$

Thereby (2) shows the inital probability distribution associated with root node X^0 at time step 0 (X_0^0). Please note, the statement $P(X = x)$ is a shorthand for an event $\omega \in \Omega : f_X(\omega) = x$, whereby the set of possible outcomes is denoted by Ω and f_X maps an event ω to a possible value of a random Variable X. We denote possible values x of X by $Val(X)$ [22].

In (3) the probability distributions of state transitions is given. With this, the dependencies over time (and between states) are expressed. Consequently, $\boldsymbol{A}(i, j)$ is a adjacency matrix extended by transition probabilities for entries unequal 0. In (4) and (5) the probability distribution for observations concerning a child is given. Please note, that a HMM can be specified by a single matrix $\boldsymbol{B}(i, j)$, since the corresponding probability distribution can not factorized like in case of a DBN – that means, graphically, the root node would have only one child incorporating the probability distribution.

V. Translation of a UML activity into a DBN

Given a UML activity represented as a graph $\mathcal{U} = (N, F)$, the set N can be further divided into different sets of nodes given by:

- $\mathcal{A}$: Set of actions,
- $\mathcal{I}, \mathcal{E}$: Set of initial node, set of final nodes,
- $\mathcal{B}$: Set of decision and merge nodes (branch nodes),
- $\mathcal{O}$: Set of object nodes.

The set of object nodes is given by the set of data pins. A node that is part of one of the node sets $\mathcal{I}, \mathcal{E}, \mathcal{B}$ is called a control node. Furthermore, the set F is given by

- $\mathcal{KF}$: Control flow, i.e. activity edges connecting actions and control nodes, as well as edges between themselves.
- $\mathcal{DF}$: Object flow, i.e. activity edges connecting actions and object nodes or between control nodes and object nodes.

A. Building the DBN structure

Formally, the translation $[[\mathcal{U}]]$ of a UML activity $\mathcal{U}$ to the network structure of B_0 and $B_\rightarrow$ of the DBN is given by:

$$[[(N, F)]] = (V_{B_0}, E_{B_0}, V_{B_\rightarrow}, E_{B_\rightarrow}),$$

using

$$V_{B_0} = \{v_e \mid e = (e_i, e_j) \in \mathcal{DF}, e_i \in \mathcal{A}, e_j \in \mathcal{O}\} \cup \{v_\mathrm{r}\}, \quad (6)$$
$$E_{B_0} = \{(v_\mathrm{r}, e_j) \mid (e_i, e_j) \in \mathcal{DF}, e_i \in \mathcal{A}, e_j \in \mathcal{O}\} \quad (7)$$
$$V_{B_\rightarrow} = \{v_{e^*}^* \mid e^* = (e_i^*, e_j^*) \in \mathcal{DF}, e_i^* \in \mathcal{A}, e_j^* \in \mathcal{O}\} \quad (8)$$
$$\cup \{v_\mathrm{r}^*\} \cup V_{B_0} , \quad (9)$$
$$E_{B_\rightarrow} = \{(v_\mathrm{r}^*, e_j^*) \mid (e_i^*, e_j^*) \in \mathcal{DF}, e_i^* \in \mathcal{A}, e_j^* \in \mathcal{O}\} \quad (10)$$
$$\cup \{(v_\mathrm{r}, v_\mathrm{r}^*)\} . \quad (11)$$

According to (6), output pins of UML actions are transformed into vertices of the graph of B_0. Additionally, an auxiliary node v_r is added to the network to represent the root of the resulting Bayesian structure.

Please note, that UML nodes with equal names (e.g. as equal output pins of different actions) are indeed translated several times. However, since the destination node has the same name in each case, it is added to the vertex set V_{B_0} of the graph of B_0 only once. That's because adding an element to a set where the element is already part of, does not alter the set.

Furthermore (7), the object flow connecting an action to an object is translated into an directed edge from the root node v_{r} of B_0's graph to the corresponding child.

The vertex set $V_{B_{\rightarrow}}$ of the graph of $B_{\rightarrow}$ is given by (8) and (9). First, output pins of UML actions are transformed into vertices of the graph of $B_{\rightarrow}$. Then the set containing root node v_r^* and the set V_{B_0} are included. An asterisk is used to avoid ambiguities and to mark a vertex belonging to slice t of the graph of $B_{\rightarrow}$ (cf. Fig. 4 for an example of a 2TBN).

The set of edges $E_{B_{\rightarrow}}$ (10) contains edges from the root node v_{r}^* of $B_{\rightarrow}$'s graph to the corresponding children. The edges are derived from the object flow connecting an action to an object. Finally, the root node v_r of slice t-1 and the root node v_r^* of slice t are connected by the directed edge $(v_{\mathrm{r}}, v_{\mathrm{r}}^*)$.

B. Parametrization of the DBN structure

Besides the structure, the parameters of the DBN have to be specified. First, the values of the (discrete) random variables associated with v_r^* and v_r have to be specified. Let X_t^0 be the random variables associated to v_r^* for timesteps t, and X_{t-1}^0 be the random variables associated to v_r. The values of these random variables correspond to a natural number, where each of the different possible phases of the surgery is assigned a unique natural number by a bijective function $f : \mathcal{A} \rightarrow \{1, \ldots, N\}$, using $N = |\mathcal{A}|$. Without loss of generality, we assume that the numbers assigned by f correspond to a lexicographical order. We get:

$$Val(X_t^0) = \{f(v) \mid v \in \mathcal{A}\},$$
$$Val(X_{t-1}^0) = \{f(v) \mid v \in \mathcal{A}\},$$

using $Val(\cdot)$ to denote the set of values an associated random variable can take. For all other random variables a Bernoulli distribution is used. That means, random variables $X^{1:N}$ associated with child nodes of v_r and v_r^* are binary-valued, e.g.

$$Val(X_t^i) = \{0, 1\},$$

using $i = 1, \ldots, N$ with $N = |V_{B_{\rightarrow}} \backslash (\{v_r^*\} \cup V_{B_0})|$. Furthermore, the initial state distribution, and the CPD of observations and of state transitions are specified. Thereby, the initial state distribution π is given by:

$$\pi(f(v)) = \begin{cases} 1, & \text{if } (u, v) \in \mathcal{KF} \wedge u \in \mathcal{I}, \\ 0, & \text{otherwise}, \end{cases} \tag{12}$$

with $f : \mathcal{A} \rightarrow \{1, \ldots, |\mathcal{A}|\}$ being the bijective function assigning a unique natural number to each action $v \in \mathcal{A}$ as defined above. Since there is only one dedicated start state, the probability of its presence in the initial distribution is 1; all other state are assigned with 0 as initial probability (12).

The probability distribution of state transitions are given by (13). Their representation equals to an adjacency matrix having transition probabilities as edge weights (similar to HMMs). The matrix $\boldsymbol{A}$ is given by:

$$\boldsymbol{A}(i, j) = \begin{cases} \frac{u.\text{an.du}-1}{u.\text{an.du}}, & \text{if } u = v, u \subseteq \mathcal{A} \\ 1 - \frac{u.\text{an.du-1}}{u.\text{an.du}}, & \text{if } (u, v) \in \mathcal{KF} \\ & \wedge \{u, v\} \subseteq \mathcal{A} \\ 1, & \text{if } (u, v) \in \mathcal{KF} \\ & \wedge u \in \mathcal{A} \wedge v \in \mathcal{E} \\ b.\text{an.prob}(u, v) & \\ \cdot \left(1 - \left(\frac{u.\text{an.du}-1}{u.\text{an.du}}\right)\right), & \text{if } \exists b \in \mathcal{B} : \{(u, b), (b, v)\} \\ & \subseteq \mathcal{KF} \wedge \\ & \nexists w \in \mathcal{A} : (w, b) \in \mathcal{KF} \\ 0, & \text{otherwise}, \end{cases} \tag{13}$$

where $i = f(u), j = f(v)$, with $f : \mathcal{A} \rightarrow \{1, \ldots, |\mathcal{A}|\}$ again being the bijective function assigning a unique natural number to each action as defined above. The number u.an.du gives the average duration of the phase associated to action $u \in \mathcal{A}$, and b.an.prob(u, v) gives the probability of a transition to phase v, given phase u will be left. Both u.an.du as well as b.an.prob(u, v) are specified in the annotation for the corresponding action $u \in \mathcal{A}$ and branch and merge nodes $b \in \mathcal{B}$, respectively, as explained in Section IV. Please note that a directed edge from a source node u to a target node v is given by the edge $e = (u, v)$, where $\{u, v\}$ denotes the set of a source and a target node.

In the UML activity, several pins denote features, and their corresponding values are observed during surgery. Some of the pins $o \in \mathcal{O}$ are attached to different actions, but represent the same feature, i.e., they have the same name. Let $\mathcal{O}_{\text{name}}$ be the set of unique pin names present in the activity. We define a surjective function $g : \mathcal{O} \rightarrow \mathcal{O}_{\text{name}}$. This function returns the name of a given object node o. We define a second, bijective function $h : \mathcal{O}_{\text{name}} \rightarrow \{1, \ldots, N\}$, with $N = |\mathcal{O}_{\text{name}}|$. This function h lexicographically assigns a unique natural number to each element of $\mathcal{O}_{\text{name}}$.

To specify the underlying CPD for each feature, we define one matrix $\boldsymbol{B}^k$ for each element $o_{\text{name}} \in \mathcal{O}_{\text{name}}$:

$$\exists!\, \boldsymbol{B}^k\ \forall\, o_{\text{name}} \in \mathcal{O}_{\text{name}} : k = h(o_{\text{name}}).$$

The elements $b_{i,j+1}^k$ of each matrix $\boldsymbol{B}^k$ are given by

$$b_{i,j+1}^k = \begin{cases} P(X_t^k = j | X_t^0 = i), & \text{if } j = 1 \\ 1 - P(X_t^k = j | X_t^0 = i), & \text{otherwise}, \end{cases}$$

where $j \in Val(X_t^k) = \{0, 1\}$, and $i \in Val(X_t^0) = \{f(v) \mid v \in \mathcal{A}\}$ as defined above.

$$
\boldsymbol{A} = \begin{pmatrix}
0.9912 & 0 & 0 & 0 & 0.0088 & 0 & 0 & 0 & 0 \\
0 & 0.9913 & 0.0048 & 0 & 0 & 0 & 0.0039 & 0 & 0 \\
0 & 0 & 0.9981 & 0 & 0 & 0 & 0 & 0.0019 & 0 \\
0.008 & 0 & 0 & 0.9982 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.0093 & 0 & 0 & 0.9907 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.9946 & 0 & 0 & 0.0054 \\
0 & 0 & 0.0017 & 0 & 0 & 0 & 0.9983 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 & 0 \\
0 & 0 & 0 & 0.0051 & 0 & 0 & 0 & 0 & 0.9949
\end{pmatrix} \tag{14}
$$

$$
\boldsymbol{B}^2 = \begin{pmatrix}
0.10 & 0.10 & 0.10 & 0.10 & 0.10 & 0.90 & 0.90 & 0.90 & 0.90 \\
0.90 & 0.90 & 0.90 & 0.90 & 0.90 & 0.10 & 0.10 & 0.10 & 0.10
\end{pmatrix}^T \tag{15}
$$

The conditional probabilities $P(X_t^k = j | X_t^0 = i)$ are derived from the state of the objective nodes. Each objective node $o \in \mathcal{O}$ is associated with a probability value o.prob, and

$$
P(X_t^k = 1 | X_t^0 = i) = \begin{cases}
o.\text{prob}, & \text{if } \exists (a, o) \in \mathcal{DF} \\
& \text{with} \\
& a \in \mathcal{A}, o \in \mathcal{O}, \\
& (h \circ g)(o) = k, \\
& f(a) = i \\
\theta, & \text{otherwise.}
\end{cases}
$$

Using $\theta = 0.1$ as default value for false positive predictions of the presence of features. With this, a UML activity can be translated into a DBN, using parameters $\boldsymbol{\pi}$ as initial state distribution, $\boldsymbol{A}$ for state transitions, and matrices $\boldsymbol{B}^k$ for the probability distributions of observations.

VI. Results

For the UML activity of the cholecystectomy, the translation to DBNs' structure results in graphs similiar to the ones depicted in Fig. 4 and Fig. 5. In our case, the structures comprise 6 children, since 6 different features are used. These features are (in lexicographic ordering) the usage of the:

1) clipping device
2) dissection device
3) grasper
4) HF coag./cutting
5) lap. scissors
6) trocars

The values of the (discrete) random variables associated with v_r and v_r^* are as follows:

$$
Val(X_t^0) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\},
$$
$$
Val(X_{t-1}^0) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}.
$$

Thereby an lexicographic ordering of the names of surgical phases is used, to assign the corresponding number. The ordering is given by:

1) Clipping and cutting bile duct
2) Clipping and cutting cystic artery
3) Detaching of the gallbladder
4) Dissection bile duct
5) Dissection cystic artery
6) Generation of pneumoperitoneum
7) Intraoperative Cholangiogram
8) Liver bed coagulation
9) Placement of trocars

Consequently, the initial state distribution is given by:

$$
\boldsymbol{\pi} = (0, 0, 0, 0, 0, 1, 0, 0, 0)^T ,
$$

since the surgical intervention (and the corresponding UML activity) always starts with the generation of a pneumoperitoneum.

Concerning the state transition probabilities, $\boldsymbol{A}$ is given by (14). Regarding the equation in (13), the row entries of the matrix in (14) sum up to 1. Please note, the values are rounded, to allow a compact representation of $\boldsymbol{A}$.

For simplicity, we do not show all observation probabilities represented by $\boldsymbol{B}^k$. In (15), the transpose of matrix B^2 is shown, which encodes the probabilities of the detected usage of the grasper, given a surgical phase.

The first row corresponds to the probability that the grasper is not used in a phase. This is specified by the column of the matrix. I.e. the columns represent the different surgical phases. The second row is the probability of using the grasper, given a phase. I.e. the columns sum up to 1.

Please note, the probabilities incorporate two sources of expert knowledge: Firstly, the probability that a specific instrument is used, given a specific phase (medical knowledge). Secondly, the probability a specific instrument is detected as used given a specific phase (technical knowledge). E.g. even if a medical domain experts estimates a probability of the usage of an instrument in a specific phase by 1, the technical expert has to take the accuracy of the used algorithms into account, e.g. an accuracy of the detection of usage of 90%.

On the one hand, limitation of the proposed process are estimation errors introduced by a limited number of experts providing the corresponding probabilities and the influence of unknown variables not expressed in the UML model. On the other hand, the resulting (expert-based) models can further be refined with sources of knowledge, e.g. with a large amount of observational and/or expert data, as soon as it is available. Furthermore, to deal with unknown variables, the introduction of leak variables in context of constructs like "noisy or" [23] is possible to represent causes not modeled. Performance measurements for the proposed DBN model can be found in [24]. Thereby, we used a DBN to estimate the progress of a synthetic surgery.

VII. Conclusion

In this paper we introduced a framework for modeling surgical interventions. The framework utilizes UML activities as an interface for Dynamic Bayesian Networks (DBNs). These models are used for the actual estimation of the progress of a surgical intervention. As an example application, we presented a cholecystectomy (removal of the gallbladder), a common use case.

Activities are utilized as an interface, because of their comprehensibility for the medical as well as the technical domain expert. This is a necessary precondition for the dialog of experts, and also allows for workflow modifications by the medical practitioner on his own. This can help to lower barriers of workflow modeling – e.g. reduce the fear of regimentation on the side of the medical domain expert. At the same time, the self-determined formalization can help to close the gap between theoretical knowledge and practical solutions in context of a surgical intervention.

By using only one comprehensible activity, several advanced models can be obtained by using translation rules. In the present work we focused on the translation of a UML activity into Dyanamic Bayesian Networks. This model can then be used to provide a basis for advanced assistance functions, taking the progress of a surgery into account.

References

[1] P. Philipp, Y. Fischer, D. Hempel, and J. Beyerer, "Framework for an interactive assistance in diagnostic processes based on probabilistic modeling of clinical practice guidelines," in Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology, Elsevier, pp. 371–390, 2016.

[2] A. McWilliams, "Medical robotics and computer-assisted surgery: The global market," BCC Research, vol. HLC036F, 2014.

[3] R. A. Faust, Robotics in Surgery: History, Current and Future Applications. Nova Publishers, 2007.

[4] N. Padoy, T. Blum, H. Feussner, M.-O. Berger, and N. Navab, "On-line recognition of surgical activity for monitoring in the operating room." AAAI, pp. 1718–1724, 2008.

[5] S. Glöser, "Frage der Woche an Prof. Dr. med. Markus Schwarz," Dtsch Arztebl International, vol. 112, no. 23, p. 4, 2015.

[6] Z. Han, L. Zhang, and J. Ling, "Transformation of uml activity diagram to yawl," Enterprise Interoperability, Springer, pp. 289–299, 2010.

[7] W. van der Aalst and K. M. Van Hee, Workflow management: models, methods, and systems. MIT press, 2000.

[8] A. Hofstede, N. Russell, W. Aalst, and M. Adams, Modern Business Process Automation: YAWL and its Support Environment. Springer, 2010.

[9] H. Störrle, "Semantics of control-flow in UML 2.0 activities," in Visual Languages and Human Centric Computing, 2004 IEEE Symposium on Visual Languages and Human-Centric Computing, 2004, pp. 235–242.

[10] H. Störrle, "Semantics and verification of data flow in UML 2.0 activities," Electronic Notes in Theoretical Computer Science, vol. 127, no. 4, pp. 35–52, 2005.

[11] B. Agarwal, "Transformation of uml activity diagrams into petrie nets for verification purposes," International Journal Of Engineering And Computer Science, vol. 2, no. 3, pp. 798–805, 2013.

[12] F. Lalys and P. Jannin, "Surgical process modelling: a review," International journal of computer assisted radiology and surgery, vol. 9, no. 3, pp. 495–511, 2014.

[13] I. D. Learmonth, C. Young, and C. Rorabeck, "The operation of the century: total hip replacement," The Lancet, vol. 370, no. 9597, pp. 1508 – 1519, 2007.

[14] OMG - Object Management Group. (2014.08.01) Unified Modeling Language Superstructure Version 2.4.1. [Online]. Available: http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF/

[15] P. Philipp, Y. Fischer, D. Hempel, and J. Beyerer, "Modeling of clinical practice guidelines for interactive assistance in diagnostic processes," in HIMS 2015, International Conference on Health Informatics and Medical Systems, july 27-30, Las Vegas, USA, 2015, pp. 3–9.

[16] N. J. Soper, L. L. Swanström, and S. Eubanks, Mastery of endoscopic and laparoscopic surgery. Lippincott Williams & Wilkins, 2008.

[17] G.-Q. Ding, W. Cai, and M.-F. Qin, "Is intraoperative cholangiography necessary during laparoscopic cholecystectomy for cholelithiasis?" World Journal of Gastroenterology: WJG, vol. 21, no. 7, p. 2147, 2015.

[18] N. Padoy, "Workflow and activity modeling for monitoring surgical procedures," Ph.D. dissertation, Université Henri Poincaré-Nancy I, 2010.

[19] C. Rupp, S. Queins, et al., UML 2 glasklar. Carl Hanser Verlag GmbH Co KG, 2012.

[20] T. Dean and K. Kanazawa, "A model for reasoning about persistence and causation," Computational intelligence, vol. 5, no. 2, pp. 142–150, 1989.

[21] K. P. Murphy, "Dynamic bayesian networks: representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, 2002.

[22] D. Koller and N. Friedman, Probabilistic graphical models: principles and techniques. MIT press, 2009.

[23] F. J. Diez, "Parameter adjustment in Bayes networks. The generalized noisy OR-gate," in Proceedings of the Ninth international conference on Uncertainty in artificial intelligence, 1993, pp. 99–105.

[24] P. Philipp, L. Schreiter, J. Giehl, Y. Fischer, J. Raczkowsky, M. Schwarz, H. Woern, and J. Beyerer, "Situation detection for an interactive assistance in surgical interventions based on dynamic bayesian networks," CRAS 2016, 6th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery, 2016.