From Observations to Collaborative Simulation: Application to Surgical Training

Guillaume Claude, Valérie Gouranton, Benoît Caillaud, Bernard Gibaud, Pierre Jannin, Bruno Arnaldi

From Observations to Collaborative Simulation: Application to Surgical Training

Guillaume CLAUDE[1], Valérie GOURANTON[1], Benoit CAILLAUD[2], Bernard GIBAUD[3], Pierre JANNIN[3], and Bruno ARNALDI[1]

[1]INSA de Rennes, IRISA/Inria, F-35042 Rennes cedex, France [†]
[2]Inria/IRISA, Campus de Beaulieu, F-35042 Rennes cedex, France [‡]
[3] INSERM, UMR 1099, Université de Rennes 1, LTSI F-35043 Rennes, France [§]

Figure 1: *We use a virtual replica of an existing operating room of our local university hospital center.*

Abstract

In surgical training, Virtual Reality systems are mainly focused on technical surgical skills, leaving out procedural aspects. Our project aims at providing a novel approach to the use of Virtual Reality addressing this point. In our project, we propose an innovative workflow to integrate a generic model of the procedure, generated from real case surgery observation, as the scenarios model in the virtual reality training system. In this article we present how the generic procedure model is generated and its integration in the virtual environment.

Categories and Subject Descriptors (according to ACM CCS): D.2.2 [Software]: SOFTWARE ENGINEERING/Design Tools and Techniques—Petri nets H.5.1 [Information Systems]: INFORMATION INTERFACES AND PRESENTATION/Multimedia Information Systems—Virtual Reality H.5.2 [Information Systems]: INFORMATION INTERFACES AND PRESENTATION/User Interfaces—Training

1. Context and Objectives

Until recently, surgical education relied on the old see one, do one (and teach one) paradigm. According to [Sat96], surgical education is now expected to rely on computer-based systems with simulation capabilities. Studies on surgical skills, (e.g. [CIRM06], [Spe78] or [McC97]) have shown the importance of both conceptual and procedural knowledge in the decision making process.

Our project aims to develop virtual environments for procedural training based on observations of real case surgeries (Figure 2).

This project is integrates three research topics: (1) Surgical Process Data Organisation and Acquisition, (2) Surgical Process Mining, and (3) Collaborative Virtual Environment for Training. These topics match with the three stages of a specific workflow. First, we acquire data, during real case surgeries of a specific procedure, using the observation module of the "b<>com SurgeTrack" software suite (Figure 3). Observations are formalised and structured as Surgical Process Models (SPM, proposed by [LJ14]). They rely on a formal representation of the surgical domain defined in the open source ontology OntoSPM proposed by [GPP14]. Observations are then saved in XES format. The second stage uses the XES files to generate a generic model of the procedure (gSPM) based on a binary net formalism. Finally, the third stage integrates the procedure model as the specification of all the possible scenarios in a collabo-

† fistname.lastname@irisa.fr
‡ fistname.lastname@inria.fr
§ fistname.lastname@univ-rennes1.fr

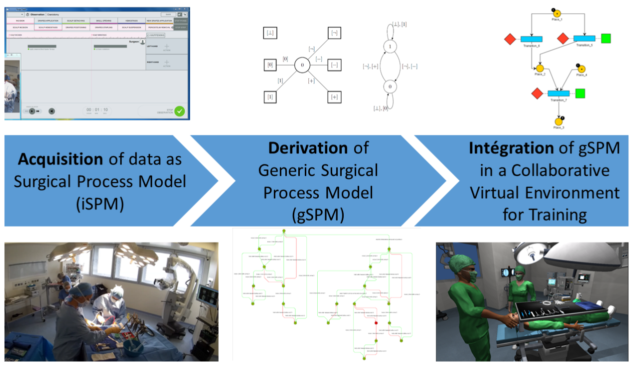

Figure 2: *We provide Collaborative Virtual Environments for Training in Surgical Procedures, based on observations of real case surgeries. Observation data is integrated into a system providing a Generalised Surgical Process Model (gSPM) of the procedure. This Model is integrated as the scenario of the Virtual Environment.*

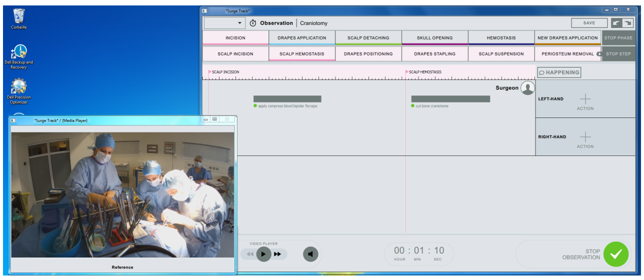

Figure 3: *The b<>com SurgeTrack data recording tool: observations taken post-operatively, using a video capture of a real case surgery.*

rative virtual environment for training surgical staff to the observed procedure. The generic procedure model can express variability in the procedure. The highest is the number of observed surgeries, the highest is the number of possible unfolding of the training scenario.

In this article we present three elements of our project

- The process and tools used to create a generic model of the surgical procedure (gSPM).
- The virtual environments tools we have developed, able to integrate the gSPM as a specification of the possible scenarios.
- How is made the connexion between the two models.

First, in Section 2, we propose an overview of the related work found in the literature. Secondly, in Section 3, we present our gSPM generation process, the virtual environments and scenarios models we decided to use and how they are connected. Finally, in Section 4, we talk about perspectives and future for our project.

2. Related Work

In this section propose an analysis of existing work from the point of view of the our project. In Section 2.1, we discuss the existing work in process model synthesis to generate generalised surgical process model. In Section 2.2, we propose an analysis of Collaborative Virtual Environments models, with a special interest in procedural training.

2.1. Generic Surgical Process Model

One fundamental hypothesis of our approach is that surgical procedural knowledge models can be computed from surgical cases observations. The difficulty is to create generic models representing the surgical procedure performed on a homogeneous population of patients. These generic models should represent the possible scenarios, that could be followed for a similar surgical case. Some initial works were published in this domain [NJS*11]. Such methods did not allow generalisation of the models, which represented the observed data only, requiring an exhaustive observation of all possible scenarios for a given procedure. In [NJS*11], a strategy was proposed for model execution, which consisted of the translation of a single surgical process model into a workflow and in the execution of this workflow by an open source workflow engine for Operating Room (OR) devices control only. This initial piece of work has several limitations. First, several aspects of the surgical procedure were not taken into account. For instance, objects constituting the environment were partially described and represented without associated numerical models. Collaborative aspects of the surgical procedure as well as exceptions were poorly handled. The other methods for computing generic models are usually restricted to the analysis of differences between populations or to a simple aggregation of all possible paths without any explanation.

The inference of finite automata from languages was proposed in [Ang87]. However, constructing a finite automaton exhibits neither causality nor independence between actions, both crucial to the intended use of a surgical process model. Second, when inferring finite automata, generalisation over the given procedure records can be regulated only by fixing the number of states, whereas we want generalisation in surgical process models to be regulated on a logical basis, i.e., based on the causalities extracted from the recorded patient-specific surgical process models. An alternative to the inference of finite automata is Petri net synthesis [BBD15]. One possibility would be to synthesise workflow nets from surgical operation records using van der Aalst's Alpha algorithm or other efficient algorithms implemented in the process mining suite PROM [VdAvDG*09]. These algorithms are targeted to derive process models in which every action produces items delivered as inputs to other actions. Unfortunately, surgical actions are subject to causal dependencies that cannot be reduced to producer/consumer dependencies. Among other available tools, Petrify [CKLY98] and Genet [CCK09] implement region-based synthesis algorithms. The former synthesises Elementary Net Systems, while the latter is also capable of synthesising Bounded P/T Nets. By examining effective surgical procedure records, and for reasons explained in Section 3.1, it appears that neither of these methods and tools can be applied successfully to the synthesis of surgical process models. Test and Flip nets [Cai13], and an efficient synthesis algorithm are introduced for this purpose in Section 3.1.

2.2. Collaborative Virtual Environment for Procedural Skills Training

Models used in Virtual environments for training are rarely generic enough to handle all the requirements from various applicative domains. However, [GGA09] stated they all have common needs including a reactive environment composed of behavioural objects and a description of the procedure or task to achieve (scenario). To be able to integrate the procedure model as the description of all possible scenarios, we must be able to relate the entities (i.e. actions, actors and items) referenced in the procedure model to virtual entities in the environment. These actions may be performed by the different actors but can also happen without their intervention (ex. power outage management during surgery).

In this Section, we propose an overview of the existing collaborative virtual environment models (Section 2.2.1) and collaborative scenarios models (Section 2.2.2). In Section 2.2.3, we state on the models we chose to use in our project.

2.2.1. Collaborative Virtual Environment Modelling

In virtual environment, the scenarios management defines an orchestration of the actions. These actions are behaviours involving one or more entities of the environement. As our generalised procedure model is based on real case observations, we propose to focus our efforts on virtual environment models able to provide the following features :

- All possible levels of collaboration, from perceptions to co-manipulation of objects (see [MAP99] for a detailed classification of collaboration levels), to be able to describe both individual actions and collaborative actions.
- A clear and simple model to define objects and actions, as close as possible to the observation model.

We define three families of models

- *Low levels behaviour models* [CKP95] [LD02], express sequencing and parallelism of actions using automata. However, they do not model the actors nor mechanisms required to model collaboration. It is possible to use them in multi-actors contexts, but it requires the use of complementary models. In general, they can be applied to a wide range of use cases, but require additional developments.
- *Smart-objects* [KT99] [BD04] [WH01], use the virtual object as an interface for the actions. Actions using an object are attached directly to it and their availability depends on the state of this object. This approach is efficient when it comes to imply one object in an action. However, it is difficult to define actions implying combinations of objects. One action for each combination of object must be defined. Co-manipulation of objects is also difficult to model.
- *Objects-relations* models [LC06] [MGA*07] [CTB*12] [LCL*13] [BGB*15] use an additional type of entity, the relations, to model the actions. Relations describe the involvement of several objects in a common behaviour. A relation is defined through two elements
  - A behaviour involving different objects and modifying their state
  - Pre-conditions on types and states of the objects involved in the execution of the relation

  Some objects-relations models are able to model callaborative actions up to co-manipulation of an object by multiple actors.

2.2.2. Collaborative Virtual Environments Scenarios Modelling

Specification of scenarios allows defining actions the actors will be able to perform, depending on the unfolding of the simulation. We define two families of solutions to this problem :

- *Predefined Scenarios models* [GMA07] [CTB*12] [BA06] [CGA15] rely on a complete description of all the possible scenarios during the simulation. Their formalisms are based on state machines or nets. Most of them are able to express temporal and causal constraints and a part of them can handle collaboration at different levels. Due to their formalism, they easily offer a graphical representation. This approach is more common in procedural training.
- *Emerging scenarios models* as proposed by [Szi03] [CLPC07] [LCL*13] rely on the more or less constrained behaviours of the actors. The scenario emerges during the simulation. These models do not propose solutions to ensure the sequencing of the actions. They are divided into two subfamilies. In the first one, as proposed by [Sha97], the behaviour of each actor is defined to react exactly to some specific states of the environment. In the second one, as proposed by [PMP01], the actor as to reach some goals and she, he or it as to decide his or her actions through action planning process. This approach in usual storytelling or in decision training.

2.2.3. Synthesis

objects-relations virtual environments models offer the features of the other two families of models (i.e. behaviour modelling, objects state manipulation) and propose some more (i.e. co-manipulations, generic actions). They are simplifying the modelling of the environment. As an example, a nurse can clean an instrument using any clean cotton piece available. This can be easily expressed trough relations without defining an action for each combination of instrument and cotton piece in the scene. Considering this, we focus on using this family of virtual environments models.

From our point of view, *pre-defined scenarios models* are more interesting than *emerging scenarios models*. They offer more control in the unfolding of the simulation. This is a key point on procedural training. Their formalisms allow graphical representation, easing their use in complex use cases. These models can be very difficult to be used to specify or to generate as scenarios are spread between the behaviours of all the entities of the environment.

We decided to use the #SEVEN [CGA15] scenario model as it is able to model intricate unfolding of actions and relies on a family of binary nets close to the Test and Flip net used in our gSPM model. These actions being actors' actions but also executed by the scenario engine itself without any intervention of the actors. #SEVEN does not make any assumption on the virtual environment with which it is integrated. However, #SEVEN is expendable and provides specific entities to be integrated within a #FIVE [BGB*15] collaborative virtual environment. #FIVE is an *objects-relations* model providing specific elements allowing reasoning on objects types and actions. Furthermore, action specification in #FIVE is close to the ontological model [GPP14] used in the data acquisition process. The joint usage of both #FIVE and #SEVEN provide a strong basis to our procedure model integration in collaborative virtual environment.

3. From observation to simulation

This section details our work allowing providing the virtual environment with a specification of scenarios derived from real case observations. In Section 3.1 we present the derivation process. In Section 3.2 we detail the functioning of #FIVE and #SEVEN, respectively the virtual environment model and the scenarios model we use in our project. Finally, in Section 3.3 we precise how the derived model of the procedure is integrated in the virtual environment through the scenarios model.

3.1. Generic SPM: Derivation using Process Mining from sets of observations

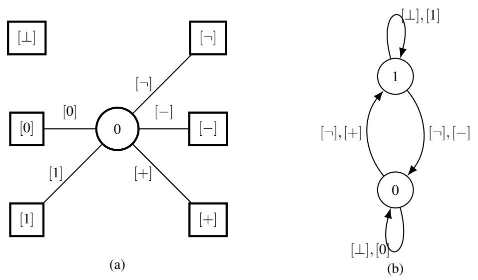

Figure 4: *(a) One place Test and Flip net and (b) its marking graph. There are five types of flow arcs:* $[\neg]$ *complements the marking of the place;* $[+]$ *sets the marking to* 1*, only when the place is marked* 0*;* $[-]$ *sets the marking to* 0*, and only when the place is marked* 1*;* $[0]$ *(resp.* $[1]$*) has no effect on the place. It tests whether the place is marked* 0 *(resp.* 1*), meaning that the transition can be fired only when the place is marked* 0 *(resp.* 1*).*

The second stage of our workflow derives a Generic Surgical Process Model (gSPM) of the observed procedure from the acquired observations. The data used to create a gSPM consist only in a very small set of observations and the resulting gSPM (Figure 5) can not possibly average the recorded behaviour, based on some statistics. The gSPM should nevertheless capture the "logic" of the type of surgical procedure being considered. In particular, it should allow for a larger variety of possible scenarios than those considered as input. Behaviour generalisation is a key feature of gSPM derivation: it should enable sequences of actions not observed in the input observation, but still consistent with a given type of surgical procedure. As for the gSPM presented in Figure Figure 5, the generalizations proposed by our derivation process are validated by expert surgeons.

Surgical procedures have several generic properties, that should be taken in consideration for the choice of formalism used to express gSPM. Manual inspection of several observations and exchanges with surgeons revealed several important features of surgical procedures, reviewed below:

- Surgical procedures are concurrent processes, during which several agents (surgeons, anaesthetists, nurses) perform technical actions, but also interact with one-another. The formalism used to capture gSPM should be expressive enough to capture the basic properties of concurrent processes [Ros10]. Namely, causality (an action can not be undertaken before another action has been completed), concurrency (a set of actions are independent, and can be performed in any order, or in parallel) and conflict (the occurrence of one action disables another action) are three properties that should be considered when synthesizing a gSPM.
- At the same time, there is no reason to use a very expressive formalism, for instance a *Turing complete* programming language. On the contrary, its expressiveness should be tailored to surgical procedures. For instance, counting (the capability to express that a sequence of actions should be repeated a precise number of times) is not required. The reason is that actions or sequences of actions that can be repeated twice, can also be repeated an arbitrary number of times. It is not uncommon to find, in observations, sequences of the form *rinse, incise, rinse, incise, ...*. The number of iterations is variable from one observation to another and depends on hidden parameters that are not taken into account in the gSPM. This includes the patient's morphology, and the surgeon's habit. The consequence is that counting is not required, but moreover, should not be allowed in process models. The reason is that we want the gSPM derivation method to generalise a set of observations and allow arbitrary iterations of repeated actions or sequences of actions, instead of putting a bound on the number of iterations.

Petri nets are perfectly suited to express concurrent behaviour [DR15]. They can capture causality, concurrency and conflict. They have been used with great success to model business processes, as exemplified by [vdA16]. They can be tailored to particular applications. In the case of surgical procedures, Test and Flip nets (Figure 4), a mild extension of Elementary Net Systems (nets with binary markings), have been chosen. The main reasons for this choice are that:

1. Counting is not expressible in Test and Flip nets.
2. Contrarily to Elementary Net Systems, Test and Flip nets can express repetitions of a single action.
3. Test and Flip net synthesis can be done in polynomial time [Sch96, Cai13]), contrarily to Elementary Net Systems, which synthesis problem has been proved to be NP complete [BBD15].
4. As explained below, the computation of a gSPM, generalising the observation can be formalized as an optimal synthesis problem.

The algorithm used to derive a gSPM from a set of observations is based on the theory of Regions [BBD15] and solves the following Test and Flip net synthesis problem (solved by [Cai13], using previous results by [Sch96]):

Test and Flip Net Synthesis Problem *Given an alphabet of actions* T *and a prefix-closed regular language* $A \subseteq T^\star$*, compute a Test and Flip net* N *such that the language of* N *is the least language of Test and Flip nets containing* A*:*

$$\begin{cases} A \subseteq \mathcal{L}(N) \\ \forall N',\, A \subseteq \mathcal{L}(N') \;\Rightarrow\; \mathcal{L}(N) \subseteq \mathcal{L}(N') \end{cases}$$

The algorithm reduces the Test and Flip Net Synthesis Problem to the resolution of several systems of linear equations in the

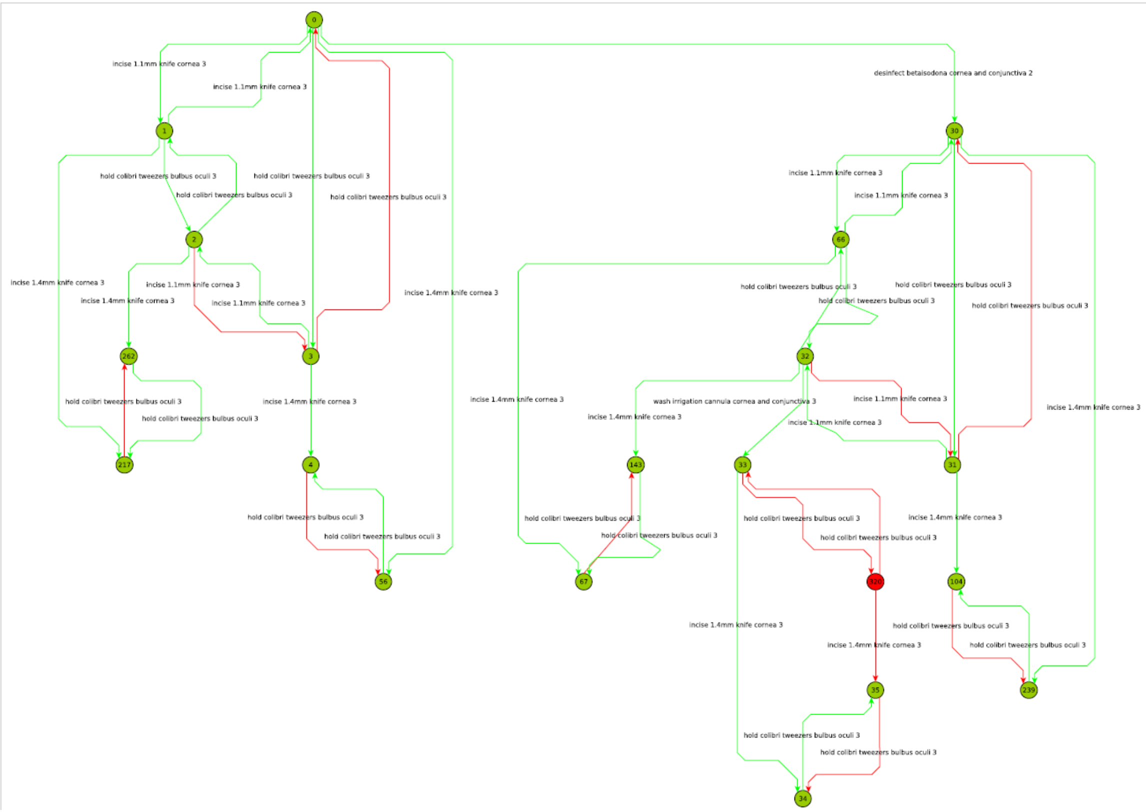

Figure 5: *Part of the gSPM derived from 19 observations of a cataract surgery. Green arcs and nodes: existing observations, red ones: generalisation. The validity of the generaliations have been verified by expert surgeons.*

Boolean ring. The toy example detailed in Figure 6, exemplifies the behaviour generalisation obtained with this algorithm. It scales up to large sets of observations, the limiting factor being the number of different types of actions. Experiments carried out on 19 observations of a cataract surgery, each consisting of about 200 actions (Figure 5), proved that our C++ implementation of the algorithm can cope with several hundred of different types of actions.

3.2. Models for collaborative virtual environments

Scenarios specification describe all the possible unfolding of the procedure. This specification is a description of possible sequencing of actions for the scenarios engine. However, it refers to actions, objects and actors that are entities of the virtual environment.

In this section, we briefly present #FIVE (Section 3.2.1) and #SEVEN (Section 3.2.2), the models we use in our project to respectively model the virtual environment and the possibles scenarios (i.e. the surgical procedure).

3.2.1. #FIVE, Framework for interaction in virtual environments

The #FIVE framework is two folded : it proposes an *object-relation model* and a *collaborative interactions model*. We will briefly present here the *object-relation model* as it is the main connexion between the scenarios engine and the environment. However, an overall presentation of #FIVE can be found in [BGB*15].

In #FIVE, a relation is an action in which one or more objects, or entities of the environment, are involved. A typical example of a relation is the relation *Pick*, involving an item, and a hand. The relation is defined by a type of relation (here the Pick relation type) and by constraints on the nature of the objects defined by #FIVE objects types. These types are assigned to virtual objects and entities and describe some properties. As an example, an object *bottle* can be at the same time a *Container* and a *Pickable*. Relations, describe the types required by several objects to be used in its execution. As an example, the relation *Pick* involve two different objects : a pickable one and a hand (that will picking the object). It means that any pair of objects *Pickable* and *Hand* existing in the environment can be involved in a relation *Pick*. As an example, it can concern any bottle (Container + Pickable) or any Forceps (Forceps + Pickable) along with any Hand of any actor in the scene.

A realisation is a specific instance of a relation, using a defined set of objects. Realisations have two purposes. First they can check the consistency between the state of the involved objects. As an example, it checks that the involved *hand* in a *pick* relation is empty. Second it realises the behaviour of the relation, by modifying the state of the involved objects.

More complex relations can be defined, using several combinations of types for each involved objects. As an example, a surgeon

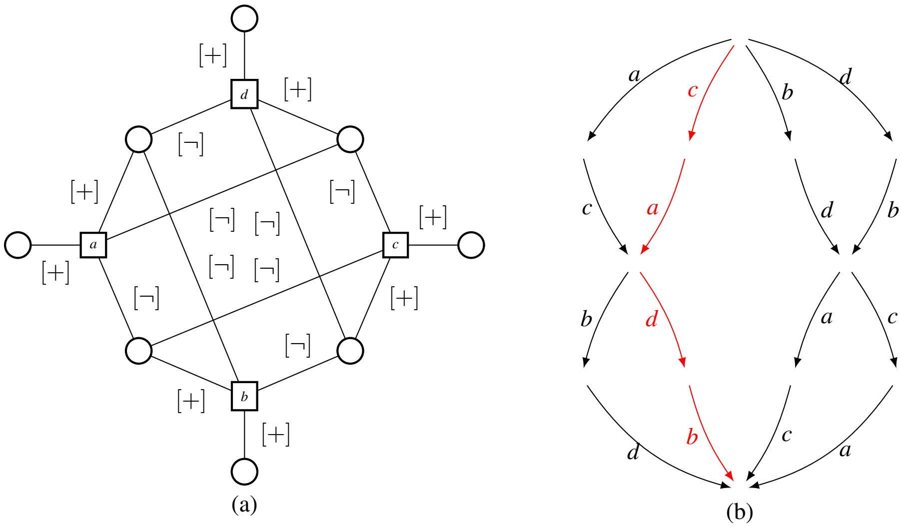

Figure 6: *Synthesis of a Test and Flip net from a toy example, consisting in three scenarios { acbd, dbca, bdac } (a) Synthesized Test and Flip net and (b) its gSPM graph. Remark that the gSPM enables a total of 8 different scenarios. The "logic" inferred by the synthesis algorithm is as follows: actions a, b, c and d can be carried out in any order, except that: if a is performed, c must happen next; if b is performed, d must happen next; and* vice-versa *if c is performed, a must happen next; if d is performed, b must happen next. Red edges have been obtained thanks to generalization and have never been observed.*

can use any of his/her hand to manipulate an instrument to act upon a specific area of the patient.

3.2.2. #SEVEN, Scenarios Engine for virtual environments

We propose the use of #SEVEN (Scenarios Engine for Virtual Environments) [CGA15], a model offering the features we require for the specification and the execution of scenarios in our virtual environments : expression of complex action sequencing, a formalism close to the gSPM and the ability to execute actions without the actors. #SEVEN focuses on the productivity as it is highly adaptable to the needs of the needs of the developer and to the nature of the virtual environment. The functioning of #SEVEN goes through an engine, executing a specification of scenarios defined in a specialised language. This language allows a specification of all the possible scenarios, defined automatically or by a specialist. A #SEVEN scenarios engine is an entity integrated in a 3D environment. Inputs of the engine provide it with data from the state of this environment. Outputs allow the engine to act on the environment by triggering actions.

As shown in Figure 7, #SEVEN is based on the *Perception-Decision-Action* principle.

- Inputs (perception) of the engine are managed by sensors, entities allowing checking high level conditions in the environment.
- Outputs (action) of the engine are managed by effectors, entities allowing triggering of actions in the environment.
- A decision system, the inner model, orchestrate behaviours of sensors and effectors.

The decision system uses a hierarchical safe Petri net based model, providing parallelism and concurrency, a graphical representation and abilities to describe a wide set of scenarios using

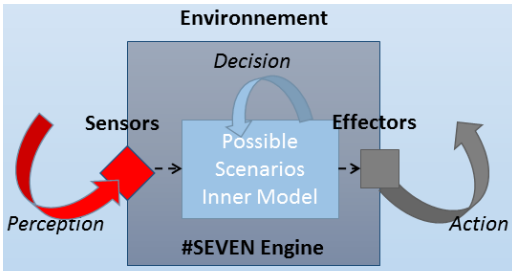

Figure 7: *Perception, Decision, Action loop used by #SEVEN to interact with the virtual environment.*

compact nets. The specification of the scenarios can be realised using the authoring tool (Figure 8) or by synthesis.

The specification of scenarios consist in describing the behaviour of the engine in the environment through its sensors and effectors. This specification is then used at startup by the engine to build an inner model of all the possible scenarios. This model is then maintained by the engine depending on time and on the state of the environment though the sensors. Changes in the state of the inner model triggers actions of the engine through its effectors. The inner model can be read by other entities integrated in the environment such as the actors, allowing them to know what must be done next. #SEVEN comes with a specific authoring tool, allowing to specify scenarios manually, and to modify a generated one, but also to follow and interact with its execution in the virtual environment (Figure 8).

Along with #FIVE, #SEVEN allows the specification and execution of scenarios with the features we defined in 2.2 :

- Complex sequencing of actions are managed through the sequencing features of #SEVEN and the generic approach of the action model offered by #FIVE.
- Actions modelled by #FIVE can be executed through #SEVEN's effectors.
- In the use #SEVEN make of the relation model of #FIVE, actors are integrated in the environment as objects. #FIVE allows the specification of collaborative actions and does not make any difference between real or virtual actors. Actions can be performed by any number of actors, and there are no constraints on their numbers.
- #FIVE models straightforwardly the actions and objects. These concepts are used as they are in the specification of the scenarios. Moreover, the scenario modelling relies on the graphic properties offered by Petri nets.

3.3. gSPM as specification of possible scenarios

The third step of our project workflow is to integrate the generalised procedure model (gSPM) as specification of possible scenarios in the virtual environment. The gSPM model and #SEVEN model are based on close binary nets formalism. However their formalisms are not fully compatible. The main issue being the translation of the complementation operator of the Test and Flip nets, making equivalent #SEVEN net explodes in term of number of places and transitions. However, a simple solution consist in translating the gSPM graph (Figure 6.b) itself into the #SEVEN formalism.

The translation of a gSPM graph, or reachability graph of the Test and Flip net, into the #SEVEN formalism is straightforward. It uses the events defining transitions from one state of the Test and Flip net to another. We refer as events the actions of the different members of the surgery team or the environmental happenings. They are used as parameters of the sensors (inputs) and effectors (outputs) in the #SEVEN net. Actions of the surgical staff are related to sensors while environmental happenings are related to effectors. Each arc of the graph is transposed to a transition each node is related to a place. Figure 9 shows the translation from the reachability graph of the Figure 6 to #SEVEN. In this example, the events

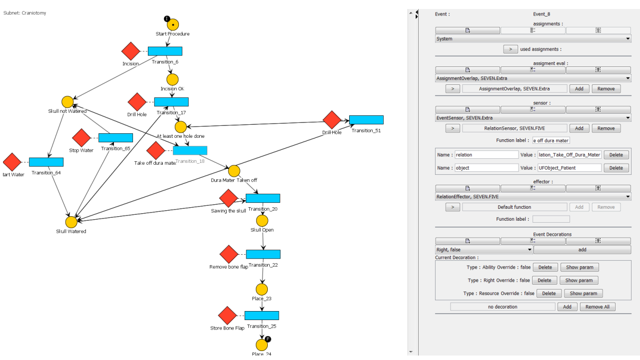

Figure 8: *#SEVEN graphical formalism and authoring tool: example scenario of a par of a surgical procedure.*

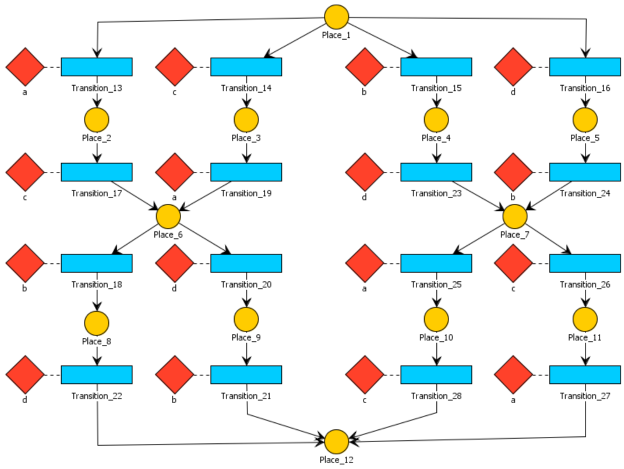

Figure 9: *Direct translation of the example of the Figure 6 in the #SEVEN formalism. The #SEVEN net is a direct translation of the reachability graph (Figure 6.b).*

a, b, c, d are considered as actions executed by the surgical staff. As such they are related to sensors in the #SEVEN net, meaning that the condition for going from a state to another is that this exact action is executed.

We have been able to execute simulations on various hardware configurations: from desktop computer using a standard display or HMD to a huge imersive room 1. Figure 10 shows a training simulation based on our tools.

Figure 10: *Our prototype collaborative virtual environment for the training of Scrub Nurses.*

4. Conclusion and Perspectives

This paper explained how we generate generalised surgical process models and use them as specification of possible scenarios for collaborative virtual environments for training. We use a theory of region based algorithm to derive a generalised Surgical Process model (gSPM), as a Test and Flip net, from real case observations. We then integrate this gSPM as the specification of all possible scenarios in a Collaborative Virtual Environment for training to the observed procedure using the #SEVEN scenarios model and the #FIVE virtual environment framework.

Our work will now be two-fold. First, we will continue data acquisition to provide more data to the generation process. It will allow us to ensure the consistency of our existing generated models and to create different generalised process models from other procedures. Second, we will use our existing virtual environment for training to assess its acceptance, efficiency and quality through experiments within medical staff (both experts and beginners) for training and rehearsal.

5. Acknowledgements

This work is part of the S3PM project, and has received French government support granted to the CominLabs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program (ANR-10-LABX-07-01). Authors thank IRT b<>com for the provision of the "Surge Track" software.

References

[Ang87] ANGLUIN D.: Learning regular sets from queries and counterexamples. *Information and computation 75*, 2 (1987). 2

[BA06] BROM C., ABONYI A.: Petri-nets for game plot. In *artificial intelligence and simulation behaviour convention* (2006), vol. 3. 3

[BBD15] BADOUEL E., BERNARDINELLO L., DARONDEAU P.: *Petri Net Synthesis*. Springer, 2015. 2, 4

[BD04] BADAWI M., DONIKIAN S.: Autonomous agents interacting with their virtual environment through synoptic objects. *CASA 2004* (2004). 3

[BGB*15] BOUVILLE R., GOURANTON V., BOGGINI T., NOUVIALE F., ARNALDI B.: # five: High-level components for developing collaborative and interactive virtual environments. In *SEARIS 2015, conjunction with IEEE Virtual Reality (VR)* (2015). 3, 5

[Cai13] CAILLAUD B.: Surgical process mining with test and flip net synthesis. In *Application of Region Theory (ART)* (2013). 2, 4

[CCK09] CARMONA J., CORTADELLA J., KISHINEVSKY M.: Genet: A tool for the synthesis and mining of petri nets. In *Application of Concurrency to System Design, 2009. ACSD'09. Ninth International Conference on* (2009), IEEE. 2

[CGA15] CLAUDE G., GOURANTON V., ARNALDI B.: Versatile scenario guidance for collaborative virtual environments. In *GRAPP'15* (2015). 3, 6

[CIRM06] COX M., IRBY D. M., REZNICK R. K., MACRAE H.: Teaching surgical skills—changes in the wind. *New England Journal of Medicine 355*, 25 (2006). 1

[CKLY98] CORTADELLA J., KISHINEVSKY M., LAVAGNO L., YAKOVLEV A.: Deriving petri nets from finite transition systems. *Computers, IEEE Transactions on 47*, 8 (1998). 2

[CKP95] CREMER J., KEARNEY J., PAPELIS Y.: Hcsm: a framework for behavior and scenario control in virtual environments. *ACM Transactions on Modeling and Computer Simulation (TOMACS) 5*, 3 (1995). 3

[CLPC07] CAVAZZA M., LUGRIN J.-L., PIZZI D., CHARLES F.: Madame bovary on the holodeck: immersive interactive storytelling. In *ACMMM* (2007). 3

[CTB*12] CHEVAILLIER P., TRINH T.-H., BARANGE M., DE LOOR P., DEVILLERS F., SOLER J., QUERREC R.: Semantic modeling of virtual environments using MASCARET. In *Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), 2012 5th Workshop on* (2012). 3

[DR15] DESEL J., REISIG W.: The concepts of petri nets. *Software and System Modeling 14*, 2 (2015). 4

[GGA09] GERBAUD S., GOURANTON V., ARNALDI B.: Adaptation in collaborative virtual environments for training. In *Learning by Playing. Game-based Education System Design and Development*. Springer, 2009. 3

[GMA07] GERBAUD S., MOLLET N., ARNALDI B.: Virtual environments for training: From individual learning to collaboration with humanoids. In *Technologies for E-Learning and Digital Entertainment*, vol. 4469 of *LNCS*. Springer Berlin Heidelberg, 2007. 3

[GPP14] GIBAUD B., PENET C., PIERRE J.: Ontospm: a core ontology of surgical procedure models. SURGETICA. 1, 3

[KT99] KALLMANN M., THALMANN D.: Modeling objects for interaction tasks. In *Computer Animation and Simulation*. Springer Vienna, 1999. 3

[LC06] LUGRIN J.-L., CAVAZZA M.: AI-based world behaviour for emergent narratives. In *International Conference on Advances in Computer Entertainment Technology* (2006). 3

[LCL*13] LANQUEPIN V., CARPENTIER K., LOURDEAUX D., LHOMMET M., BAROT C., AMOKRANE K.: Humans: a human models based artificial environments software platform. In *VRIC* (2013). 3

[LD02] LAMARCHE F., DONIKIAN S.: Automatic orchestration of behaviours through the management of resources and priority levels. In *International Joint Conference on Autonomous Agents and Multiagent Systems* (2002). 3

[LJ14] LALYS F., JANNIN P.: Surgical process modelling: a review. *International journal of computer assisted radiology and surgery 9*, 3 (2014). 1

[MAP99] MARGERY D., ARNALDI B., PLOUZEAU N.: A general framework for cooperative manipulation in virtual environments. In *EGVE*. 1999. 3

[McC97] MCCORMICK R.: Conceptual and procedural knowledge. *International journal of technology and design education 7*, 1-2 (1997). 1

[MGA*07] MOLLET N., GERBAUD S., ARNALDI B., ET AL.: Storm: a generic interaction and behavioral model for 3d objects and humanoids in a virtual environment. In *IPT-EGVE* (2007). 3

[NJS*11] NEUMUTH T., JANNIN P., SCHLOMBERG J., MEIXENSBERGER J., WIEDEMANN P., BURGERT O.: Analysis of surgical intervention populations using generic surgical process models. *IJCARS 6*, 1 (2011). 2

[PMP01] PAIVA A., MACHADO I., PRADA R.: Heroes, villians, magicians : dramatis personae in a virtual story creation environment. In *ACM IUI* (2001). 3

[Ros10] ROSCOE A. W.: *Understanding Concurrent Systems*. Springer, 2010. 4

[Sat96] SATAVA R. M.: Cybersurgeon: Advanced simulation technologies for surgical education. *BULLETIN-AMERICAN COLLEGE OF SURGEONS 81* (1996). 1

[Sch96] SCHMITT V.: Flip-flop nets. In *STACS'96* (1996). 4

[Sha97] SHAWVER D. M.: Virtual actors and avatars in a flexible user-determined-scenario environment. In *vrais* (1997), IEEE. 3

[Spe78] SPENCER F.: Teaching and measuring surgical techniques: the technical evaluation of competence. *Bull Am Coll Surg 63*, 3 (1978). 1

[Szi03] SZILAS N.: Idtension: a narrative engine for interactive drama. In *Technologies for Interactive Digital Storytelling and Entertainment* (2003), vol. 3. 3

[vdA16] VAN DER AALST W. M. P.: *Process Mining - Data Science in Action, Second Edition*. Springer, 2016. 4

[VdAvDG*09] VAN DER AALST W. M., VAN DONGEN B. F., GÜNTHER C. W., ROZINAT A., VERBEEK E., WEIJTERS T.: Prom: The process mining toolkit. *BPM (Demos) 489* (2009), 31. 2

[WH01] WILLANS J. S., HARRISON M. D.: Verifying the behaviour of virtual environment world objects. In *Interactive Systems Design, Specification, and Verification*, vol. 1946 of *LNCS*. 2001. 3