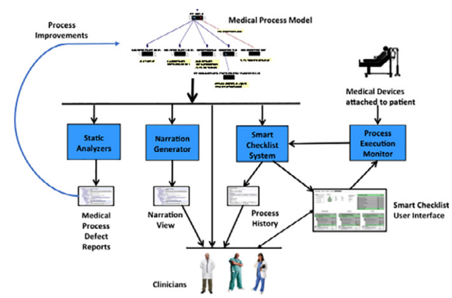
Process-Model-Driven Guidance to Reduce Surgical Procedure Errors: An Expert Opinion

Leon J. Osterweil, PhD, * *Heather M. Conboy, PhD,* * *Lori A. Clarke, PhD,* * *and George S. Avrunin, PhD*†

This paper explains how a detailed, precise surgical process model can help reduce errors by fostering better understanding, providing guidance during surgery, helping train newcomers, and by supporting process improvement. It describes the features that a process-modeling language should have in order to support the precise specification of such models.

Semin Thoracic Surg ■■:■■–■■

Keywords: Process modeling, Workflow, Error reduction, Safety



A detailed, precise surgical process model can help reduce errors in many ways.

Central Message

A detailed, precise surgical process model can help reduce errors by fostering better understanding, providing guidance during surgery, helping train newcomers, and by supporting process improvement.

Perspective Statement

Errors during cardiac surgery are a major source of adverse outcomes and death. Errors can be reduced by exploiting precise, detailed models of how surgeries should be performed. This paper addresses the characteristics that a modeling notation should have in order to form a sufficient basis for creating the kinds of surgical models that can be effective in reducing errors.

BACKGROUND AND INTRODUCTION

Modeling key processes and then using those models to support evaluation and subsequent improvements have facilitated error reduction in diverse areas of human endeavor. This approach, now called "continuous process improvement," was suggested by Taylor,[1] Deming,[2] and Juran[3] in the early 20th century and successfully applied to industrial and management processes. More recently, this approach has supported error reduction in domains as diverse as banking, elections, and, in medicine, chemotherapy,[4] emergency room operations,[5] and cardiac surgery.[6] Using process modeling and evaluation to support error reduction seems particularly promising in domains such as cardiac surgery where processes can be extremely complex.

Consider heparinization, a complex, critical task involving detecting and responding to many different contingencies where incorrect performance can increase the risk of strokes caused by blood clots. To reduce this risk, the anticoagulant drug heparin is administered to increase the activated clotting time (ACT). This key administration of heparin occurs before inserting the cannulae, connecting the cardiopulmonary bypass pump, and initiating cardiopulmonary bypass. It is a complex process, involving primarily the Surgery, Anesthesiology, and Perfusion specialty teams. If the target ACT is not achieved, a carefully prescribed sequence of contingency plans based on best practice guidelines[8] must be followed. Failing to do so accurately can result in adverse outcomes. This paper describes what a process-modeling notation should incorporate to provide adequate support for creation and evaluation of process models sufficient to represent complex, team-based processes such as heparinization, the example used throughout the rest of this paper.

This paper uses the term *process* to mean the collection of all recommended ways a product can be made or a task such as coronary artery bypass grafting can be done.[i] Processes typically involve *activities* (eg, heparinization or cannulation) performed by *agents* (eg, humans such as surgeons or devices

*Laboratory for Advanced Software Engineering Research (LASER), College of Information and Computer Science, University of Massachusetts Amherst, Amherst, Massachusetts

†Department of Mathematics, Laboratory for Advanced Software Engineering Research (LASER), College of Information and Computer Science, University of Massachusetts Amherst, Amherst, Massachusetts

Funding: This work was supported by National Institutes of Health (NIH) Award 1R/01HL126896, National Science Foundation Awards CNS-0831002, CCF-0905530, and CNS-1049738.

Conflicts of Interest: There are no conflicts of interest.

Address reprint requests to Leon J. Osterweil, PhD, Laboratory for Advanced Software Engineering Research (LASER), College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, MA 01003. E-mail: ljo@cs.umass.edu

[i] The term *workflow* used in business process management denotes a concept analogous to our concept of *process*.

1043-0679/$–see front matter
https://doi.org/10.1053/j.semtcvs.2019.02.030

such as cardiopulmonary bypass pumps), and applied to *artifacts* (eg, heparin doses or cannulae). *Process models* are abstract representations of processes, representing them with degrees of thoroughness, rigor, comprehensibility, and accuracy that vary depending on the goals of the modeling effort.

USES OF PROCESS MODELS

Figure 1 illustrates some ways a sufficiently precise and detailed abstract model of a real-world process can support error reduction. It shows an iconic representation (the details are explained in this article's supplementary material) of a process model at the top, with information flowing to the members of the surgical team, shown at the bottom, through different software systems that provide information about the process. We now describe these systems and benefits.

Shared Understanding

Errors can occur when different specialty teams (eg, Anesthesiology and Perfusion) or different members of the same team have different understandings of process details or different expectations about how the process should proceed, especially under non-normative circumstances. For example, if the target ACT has not been reached with the initial dose of heparin and an additional dose is given to increase the ACT, all members of all teams should be anticipating that the need to find a heparin alternative may be imminent, requiring each of them to have a clear and consistent understanding of the details of that process. A narration generation tool supports this kind of shared understanding by using the process model to produce a web page-like narration view (Fig. 1a), a clear detailed representation of the process designed to be studied, understood, and discussed by all team members, thereby helping avoid miscommunication, mal-coordination, and error. Figure 1 indicates that the model can also be made directly accessible to team members.

Education and Training

A clear, articulate process model can also be studied by newcomers to improve their process understanding. The narration

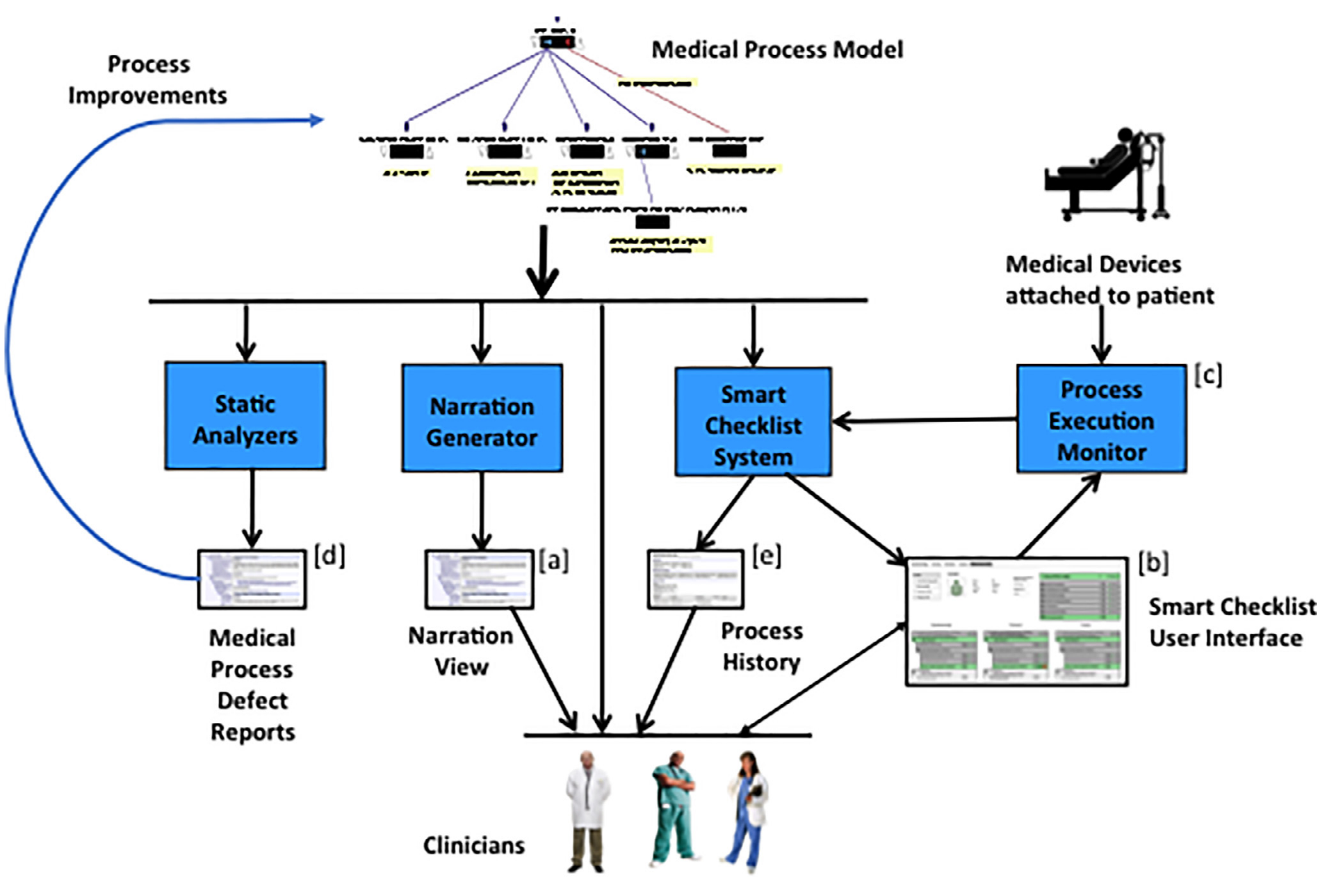


Figure 1. A detailed, precise surgical process model can help reduce errors in many ways including: by serving as the basis for manual and automated analyses that detect errors and support process improvement; by helping train newcomers to the surgical process so that they can function better as team members; by providing all surgical team members with a single, shared specification of the process that can resolve misunderstandings and foster better coordination; by helping drive the generation of detailed postprocedure documentation for future reference and study; and by fostering superior team communication and coordination during surgery by serving as the basis for presenting a timely accurate shared view of the current surgical process status. Bracketed letters identify features referred to by the text.

view, previously described as a vehicle for supporting team shared understanding, can also be studied, by newcomers to assure that they are sufficiently familiar with the details of the response of the surgical team they are joining. A sufficiently precise model could even be used to drive simulations (not shown in Fig. 1) that immerse both newcomers and more experienced teams in realistic situations, providing experience dealing with unusual situations, such as heparin resistance, which arise relatively infrequently and thus may be likely contexts for errors.

Communication, Coordination, and Guidance

Communication and coordination can be supported by showing process activities currently being performed, those recently performed, and those expected to be performed next. We have created an electronic smart checklist system (Fig. 1b) that concisely displays past, current, and impending activities, as well as agent and artifact information. Thus, all process performers see a consistent detailed summary of the activities being performed and the agents performing them, which improves communication and coordination and helps avoid errors, such as missing an important activity or failing to correctly coordinate with another team. Smart checklists also provide guidance by showing impending activities, including imminent decisions and choices, and responses to non-normative situations, thereby facilitating preparation for especially challenging situations. Guidance is also supported by real-time patient status information, gathered by a process execution monitor (Fig. 1c), which, combined with process activity completion data, provides an integrated view of the process state.

Process Improvement

The leftmost column of Figure 1 depicts a process improvement loop where automated static analyses of the model, such as Model Checking,[9] Fault Tree Analysis,[10] and Failure Mode and Effects Analysis,[11] generate reports (Fig. 1d) that identify defects and inefficiencies, leading to redesign of the actual process, and thus subsequently the model, to reduce those errors and inefficiencies. Thus, for example, in prior work, we analyzed a blood transfusion process in which a deadlock (ie, an "infinite wait") could occur when a nurse asked the blood bank for a unit of blood, and the blood bank, not knowing the patient's blood type, requested the patient's blood type from the nurse, who was unaware of the request. Deadlocks were broken when nurses phoned blood banks to inquire about the reason for the delay, but only after patients suffered potentially dangerous delays. To prevent such situations, the process was revised to prevent the possibility of this deadlock. Moreover, reanalyzing the revised process enabled verification that proposed changes did not create new defects or inefficiencies, making the improved process model a better basis for training, guidance, and coordination.

Process improvement also derives from study and analysis of process history reports (Fig. 1e), detailed historical traces automatically generated by the smart checklist as process performance proceeds. Automatically creating these traces reduces the amount of postprocedure documentation that must be created by the clinicians while improving the precision of these reports.

DESIDERATA FOR PROCESS-MODELING NOTATIONS

Here, we describe process-modeling notation characteristics that facilitate creating models that support the uses outlined above.

Rigorous Semantics

To avoid misunderstandings and assure that all process stakeholders have the same precise understanding of the process represented by a process model, the modeling notation should have rigorously defined semantics. An axiomatic system, such as some form of mathematics, is typically the basis for such semantics. This enables automated analyzers to use formal logic and reasoning to identify process defects and inefficiencies and to assure that simulators and guidance systems faithfully represent actual processes.

Selective Elaboration

Modelers include some process details in their models, and elide some, based on previous decisions about which process aspects seem more error prone and thus require more detailed elaboration. For example, a coarse-grained process model may specify only that cannulation should not be performed until and unless heparinization has succeeded. But a more fine-grained model, showing the details of verifying that ACT has been achieved, is needed to support preventing various erroneous ways to perform this verification. Thus, a process-modeling notation should support selectively specifying low-level process details as needed.

Broad Capabilities

A process-modeling notation that supports specifying a broader range of semantic issues supports reasoning about a greater variety of error-prone situations. For example, failing to detect and remove the possibility of timing errors (eg, not waiting a full 3 minutes to draw blood for testing after giving heparin) requires a process-modeling notation that supports specifying timing constraints. Errors resulting from inappropriate assignments of agents to activities (eg, assigning underqualified personnel to specialized tasks (eg, cannulating alternative sites, such as the femoral artery, in case cannulating the aorta is deemed unsafe) cannot be avoided if the modeling notation lacks facilities for specifying the characteristics that agents or tasks require.

Understandability

A process-modeling notation should incorporate features that make modeled processes readily comprehensible by all stakeholders. This is particularly important because domain experts (eg, surgeons, nurses) must understand the process models sufficiently well to be able to positively affirm that the models accurately reflect the actual processes. Models written in abstruse mathematical notations may offer the advantage of rigor, which might facilitate verification, but might be incomprehensible to many surgical domain experts who are unfamiliar with these notations. Pictures or diagrams are typically more comprehensible, but often lack rigorous semantics, making even informal validation impossible. This creates a tension between the needs for clarity and rigor. Providing alternative but consistent representations, such as the narration view, can help address this. The need for clarity sets up other tensions as well. For example, the sheer size of a highly detailed model of a process, such as coronary artery bypass grafting, could easily overwhelm process stakeholders, thereby destroying the clarity of the model. Incorporating abstraction and hierarchical decomposition into the notation, as described below, can help add clarity even in the face of size and complexity.

SPECIFIC NEEDS FOR SURGICAL PROCESS-MODELING NOTATIONS

This section describes important process-modeling notation features needed to support the requirements outlined above, especially for complex domains such as cardiac surgery. Unfortunately, many of these features, commonly found in modern programming languages, are absent from most process-modeling languages.

Hierarchical Decomposition

Surgical processes are sequences of high-level activities (eg, cannulation, separation from bypass, and closure) that must also be understood as comprising lower level activities. Effective process-modeling notations facilitate both high-level and low-level understandings. Typically, this is addressed using hierarchical decomposition, where the decomposition structure is explicitly visualized, so that process users can readily move between higher and lower level views. Many, but not all, modeling notations offer these facilities.

Activity Sequencing

Surgical processes are typically performed by teams that must do their activities in carefully specified sequences. Some activities must always precede (or follow) others under certain circumstances; some sequences may be allowed to proceed concurrently with others. A process-modeling notation must support the clear, precise specification of such sequencing. It must also support specifying choices and alternatives, indicating where and when such choices are allowable, and must also support specifying how surgical teams need to communicate with each other to assure their efforts are correctly coordinated and synchronized.

Exception Management

Surgical processes must be robust in facing diverse non-normative situations, which we refer to as *exceptions*. Exceptions vary in importance (ranging, eg, from incorrectly signed forms to postoperative detection of missing surgical instruments), arise in various ways, and happen at almost any time—indeed 1 exception may arise while another is being addressed (eg, finding a dead battery in a backup device, needed after a primary has failed). Because a surgical process model must specify how to deal with non-normative situations, a modeling notation must incorporate facilities both for specifying various kinds of exceptional situations and for how to respond to them. Exception handlers might need to repair damage (eg, repair a bypass graft), repeat an activity (eg, reverify ACT), or abort an activity (eg, discontinue heparinization). They usually aim to return process performance to a normative state. Some process notations attempt to meet these challenging needs with a simple if-then-else construct, typically requiring ungainly combinations of flags and nested if-statements. It is better to use specially designed exception-handling facilities, such as those found in modern high-level programming languages.

Abstraction

Similarly, facilities for specifying abstraction in modern programming languages offer important advantages to surgical process-modeling notations. Appropriate notations support modeling a frequently performed activity, such as verifying ACT, as an abstract subprocess skeleton, to be expanded differently in such different process contexts as during preoperative check as opposed to during an actual surgical procedure. This saves space by eliminating the need to expand the subprocess model at multiple sites, thereby making the overall process model more compact and more comprehensible. Despite the long-understood advantages of incorporating abstraction as a key feature of superior programming languages, this concept is almost never incorporated into process-modeling notations.

Human Choice and Judgment

Surgical processes must smoothly integrate the actions of human agents. Thus, a process-modeling notation should support representing when and how humans can make choices and judgments (although the possible choices may be restricted to reduce errors). Unfortunately, many process-modeling notations treat human and nonhuman agents identically, inevitably either disallowing the flexibility expected by the human participants or allowing unreasonable latitude for the behaviors of automated devices.

Resource Modeling

Important process errors, such as deadlocks and inefficiencies due to shortage of key resources (eg, surgical instruments, blood units) simply cannot be detected, and thus protected against, unless resource specification is incorporated into the selected notation. Very few process-modeling notations incorporate resource specifications.

AN EXAMPLE

An example of using rigorously defined notation in cardiac surgery process modeling is provided in supplementary material for this paper, which shows the Little-JIL[7] notation used to model part of a cardiac surgery process.

CONCLUSION

This paper summarizes our experience representing human-intensive, medical procedures with a process-modeling language. Our previous work demonstrated that the modeling and analysis (both manual reviews and automated analyses) of such process models can lead to better understandings of the process and a significant reduction in errors.[4] Here, we advocate also using the process models to provide offline training and online guidance. To evaluate this approach, we focus on life-critical, cardiac surgery procedures, which involve complex interactions among dedicated, highly trained teams of clinicians, and use this domain to describe the requirements for the process notation and various ways in which the resulting process models can be used to help reduce the occurrence of errors.

SUPPLEMENTARY MATERIAL

Supplementary material associated with this article can be found in the online version at https://doi.org/10.1053/j.semtcvs.2019.02.030.

REFERENCES

1. Taylor F: The Principles of Scientific Management. New York, NY, USA: Harper and Bros; 1911
2. Deming W: Out of the Crisis. Cambridge, MA, USA: MIT Press; 1982
3. Juran J: Quality Control Handbook. New York, NY, USA: McGraw-Hill; 1951
4. Mertens WC, Christov SC, Avrunin GS, et al: Using process elicitation and validation to understand and improve chemotherapy ordering and delivery. Jt Comm J Qual Patient Saf 38:497–505, 2012
5. Henneman PL, Shin SY, Brun Y, et al: Using computer simulation to study nurse-to-patient ratios in an emergency department. J Nurs Adm 45:551–556, 2015
6. Conboy HM, Avrunin GS, Clarke LA, et al: Cognitive support during high-consequence episodes of care in cardiovascular surgery. In: Proceedings of 2017 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA'17), New York, NY, USA: IEEE, 2017, pp 1–3
7. Cass AG, Staudt Lerner B, Sutton Jr. SM, et al: Little-JIL/Juliette: A process definition language and interpreter. In: Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000), New York, NY, USA: ACM, 2000, pp 754–757
8. Shore-Lesserson L, Baker RA, Ferraris VA, et al: The Society of Thoracic Surgeons, The Society of Cardiovascular Anesthesiologists, and the American Society of ExtraCorporeal Technology: Clinical practice guidelines—Anticoagulation during cardiopulmonary bypass. Ann Thorac Surg. 105:650–662, 2018. Co-published in Anesthesia & Analgesia and the Journal of ExtraCorporeal Technology
9. Clarke Jr EM, Grumberg O, Peled DA: Model Checking. Cambridge, MA, USA: MIT Press; 2000
10. Vesely VH, Goldberg FF, Roberts NH, et al: Fault Tree Handbook (NUREG-0492). US Nuclear Regulatory Commission; 1981
11. Stamatis DH: Failure Mode and Effect Analysis: FMEA From Theory to Execution. American Society for Quality; 1995