

Process Modelling of Aortic Cannulation in Cardiac Surgery: Toward a Smart Checklist to Mitigate the Risk of Stroke

Heather M. Conboy¹, Jason K. Maron², Stefan C. Christov¹, George S. Avrunin¹, Lori A. Clarke¹, Leon J. Osterweil¹, and Marco A. Zenati²

¹ Laboratory for Advanced Software Engineering Research, School of Computer Science, University of Massachusetts Amherst

² Division of Cardiac Surgery, Veterans Affairs Boston Healthcare System and Harvard Medical School

Abstract. Preventable adverse events related to surgery account for two thirds of hospital complications. Adherence to recommended processes of care has been suggested as a strategy to improve patient safety in surgery. This paper presents preliminary work that is exploring the use of a semantically rich process-modelling notation to describe and inform critical phases of common procedures in cardiac surgery. This work focuses on reducing strokes, a catastrophic and often preventable adverse event. The well-defined semantics of the process-modelling notation allow rigorous analysis techniques to be applied. In our work, model checking is applied to determine if the process as defined by the process model always adheres to event sequence requirements and fault-tree analysis is applied to determine where the process is vulnerable to performance failures. The results from these analyses lead to validated and improved process models that are then used to generate context-sensitive, dynamic “smart” checklists. Future work will evaluate whether the introduction of dynamic checklists based on these models will reduce the number and severity of errors in cardiac surgery.

1 Introduction

Preventable adverse events related to surgery account for two thirds of hospital complications, with 75% of errors leading to injury occurring inside the operating room (OR) [14, 29, 34]. There is now widespread recognition in the literature that many errors (latent or active) committed by the process performers within a given system can be attributed to inadequate attention to human factors [28]; there is also recognition that errors can be minimized by introducing cognitive artifacts or devices (e.g. procedural protocols) that help performers avoid errors [5, 34].

Standard checklists are examples of cognitive artifacts that were first introduced to improve safety in aviation in the 1930s. Accident rates, however, began to significantly drop only after the Naval Air Training and Operating Procedures Standardized (NATOPS), a procedural standardization program, was introduced [10]. Such procedure standardization seems particularly important for

improving coordination and communication in teams, by providing clear specification of who must do what, and when, especially in cases where non-nominal situations arise. The power of standardized procedural protocols has been clearly demonstrated in Critical Care Medicine by Pronovost, who showed, in a landmark study, that the prevalence of catheter-associated bloodstream infections could be eliminated using a standardized, evidence-based procedure for catheter insertion, compared to a rate of 11.5 per 1,000 central line days with usual practice [12, 20, 26]. In recent years, significant efforts have been made to introduce intraoperative checklists to standardize surgical procedures and team communication [19], but substantial resistance has been encountered from surgeons [17], a group of healthcare professionals with a culture deeply rooted in high standards of autonomous performance and a tradition of individualism.

On the other hand, because surgery is a technology-driven field in continuous evolution, the field of surgical process modelling has recently emerged to improve understanding and to support computer assistance in health care systems [16]. The introduction of minimally invasive and robotic surgery in the 1990s spurred standardization of complex procedures through their definitions as sequences of tasks [21, 24]. Specifically, a *surgical process* was defined as a “set of linked procedures or activities that collaboratively realize a surgical objective” and a *surgical process model* as a “simplified pattern of a surgical process that reflects a pre-defined subset of interest of the surgical process in a formal or semi-formal representation” [25]. Most surgical process modelling studies in the literature have been performed in the context of neurosurgery or endoscopy/laparoscopy [11, 16]. To the best of our knowledge, no surgical process models have been described in cardiac surgery.

In this paper, we describe a new research program in which we have begun to develop rich process models for key phases of common cardiac surgical processes (e.g. aortic valve replacement AVR and coronary artery bypass grafting CABG), with a focus on aspects of those surgeries known to be related to an increased risk of stroke [31]. A key aspect of our work is that we use a modelling language specifically designed to support concurrency, responses to unusual or non-nominal conditions, contention for resources, and dependencies on the flow of information and artifacts. These models have precisely defined semantics and are therefore suitable for formal, automated analyses to detect problems and vulnerabilities to human or device failures. Based on these validated models, we automatically generate context-aware, dynamic, smart checklists that provide surgical teams with guidance that is automatically tailored and adjusted as contingencies arise and contexts change. The next section provides background on the modelling language and associated analysis tools. Section 3, describes our preliminary work on modelling key phases of cardiac surgery.

2 Approach

As noted above, we need process models that are expressive enough to capture complex medical processes that involve such features as concurrency, responses

to a variety of unusual or non-nominal conditions, contention for resources, and dependencies on the flow of information and artifacts. We also need the process models to be represented in a notation that has well-defined semantics so that the models can be rigorously analyzed for potential safety violations and can be executed to provide information about the dynamic process state that is then reflected in the smart checklists. Most commonly used methods for describing processes (e.g. flow charts, decision diagrams, or even more sophisticated programming notations like Unified Modelling Language or Business Process Modelling Notation) are usually either not expressive enough to capture these features or not rigorous enough to be analyzed and then executed. Thus, we build upon the Little-JIL Process Improvement Environment [3], including the process modelling language [6], the static analyzers [7, 27, 35], and the execution engine [6]. The process modelling language was designed to capture the language features listed above. The static analyses allow us to identify potential problems that can arise even when the processes are executed as defined (such as when effective communication sequences are not properly embedded in the process), as well as when failures (such as human failure to correctly direct communications) are made. These analyses can also be used to evaluate the effect of proposed process modifications prior to their adoption, thereby reducing risks during actual surgical processes. After being favorably evaluated, these models can then be used to provide context-aware, dynamic guidance to process performers, helping them do the right thing and avoid doing the wrong thing (or failing to do right things) even when non-nominal situations arise. Static checklists cannot provide such context-sensitive feedback [17], and flow graph representations would be too large and complex to be helpful.

Constructing and validating such precise and expressive process models is labor-intensive, involving domain experts (e.g., surgeons, anesthesiologists, etc.) as well as experts in process modelling and analysis. But the investment in constructing such models is leveraged by their ability to support analyses that can provide feedback about possible errors, vulnerabilities, and inefficiencies in the process, guidance for process performers, and evaluation of process modifications.

Little-JIL: Space limitations prevent a detailed description of Little-JIL, but we illustrate some of its features by presenting a brief example here. Figure 1 shows part of a Little-JIL process model for the arterial cannulation phase of surgery. A Little-JIL step represents a task or activity and is shown as a central black bar, and steps are connected to each other by edges that represent both hierarchical decomposition and artifact flow. Decomposition edges emanate from the left side of the step bar, which also contains an iconic representation of the order in which the step’s substeps are to be executed. There are four step execution sequencing specifications: sequential (indicated by a right facing arrow), where substeps execute sequentially from left to right; parallel (indicated by an = sign), which specifies fork-and-join for its substeps; choice (indicated by a circle slashed through the middle), where any of the substeps can be chosen to be performed until one succeeds, and try (indicated by a right facing arrow with an X on its

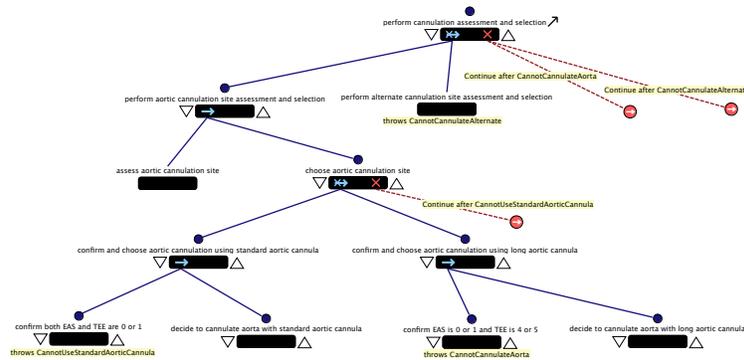


Fig. 1. Part of a Little-JIL process model

tail), where the substeps execute in left-to-right order until one of them succeeds. Lines emanating from the right of the step bar connect to exception handlers, steps that specify how to deal with specified exceptional conditions that may arise in the performance of any of the steps descendants. Each step contains an argument specification stating the artifacts used and created by the step, and a resource specification of the types of resources needed to perform the step. (The yellow notes in the figure show some of this additional information that would ordinarily not be visible in this view of the process model.) One resource is designated as the step’s agent, namely the human or non-human resource responsible for the performance of the step.

The process in Figure 1 starts with a try step, specifying that its substeps be executed in order from left to right until one succeeds. The first substep is *perform aortic cannulation site assessment and selection*. That step is a sequential step, and its substep *assess aortic cannulation site* would be executed first. That substep, whose elaboration is not shown here, involves the use of epiaortic ultrasound scanning (EAS) of the ascending aorta and trans-esophageal echocardiography (TEE) of the aortic arch, carried out by the surgeon and the anesthesiologist, respectively. The second substep, following the assessment by EAS and TEE, is *choose aortic cannulation site*. This is also a try step, so the left-most substep is executed next. That substep, *confirm and choose aortic cannulation using standard cannula*, is a sequential step, whose substeps are executed from left to right. The first substep checks whether the Katz scores from EAS and TEE are both 0 or 1. (This substep involves communication between the surgical and anesthesiology teams that is not shown in this figure.) If they are, the second substep, *decide to cannulate aorta with standard cannula*, is executed. If not, a non-nominal situation has been detected, and the exception *CannotUseStandardCannula* is thrown, meaning that execution of the current step terminates and control passes up the step hierarchy until a matching exception handler step is found. In this case, the handler simply continues execution

with the next substep of *choose aortic cannulation site*, and the step *confirm and choose aortic cannulation with long cannula* is executed. If the EAS and TEE Katz scores do not satisfy the criteria for aortic cannulation with a long cannula, the exception *CannotCannulateAorta* is thrown. Since the matching handler is attached to the *perform cannulation assessment and selection* step, this leads to termination of *choose aortic cannulation site* and execution of *perform alternate cannulation site assessment and selection*. (The elaboration of that step is not shown here.) In general, an exception handler is itself a step and, thus can be decomposed to an arbitrary level of detail and can throw exceptions itself. Little-JIL supports a variety of semantics for the return to nominal execution after the exception handler completes. Such extensive support for exception handling [18] seems important and relevant since non-nominal situations appear to be extremely common in medical processes, but usually cannot be represented clearly and precisely in commonly-used process modelling languages.

Analyzing the Models: Given a Little-JIL process model, we use model checking [8] to determine whether any possible execution of the process can violate any of a number of specified *properties*. These properties are typically requirements for the correct sequencing of process steps, such as “In the part of the execution following the first occurrence of event *S* and before the next occurrence of event *E*, each occurrence of event *B* must be preceded by an occurrence of event *A*,” and are typically expressed as finite-state automata or formulas in a suitable temporal logic. The properties serve as formal statements of the requirements that the process is designed to meet, and are intended to assure that, if each step in the process is executed correctly, none of these sequence requirements can be violated. Previous work with analyzing medical processes (e.g., [3, 23]) has shown that model checking can identify real problems in medical processes, including sequencing problems, where events could sometimes occur in an unintended order, and deadlocks, where different agents could each be waiting for the other to complete a task. In some cases, clinicians were aware that these problems sometimes arose but had not been able to identify their causes; in others, such as when a deadlock occurred, they had simply broken the deadlock by deviating from the prescribed steps, which caused a required safety check to be skipped. In this previous work, clinicians proposed modifications to the processes to avoid the problems, but, as when a proposed fix to a bug in a piece of software introduces new bugs, some of these clinician-proposed fixes introduced new problems that could, in at least some circumstances, lead to violations of other properties. The proposed changes can be evaluated by rechecking the properties on a modified model, without having to actually adopt untested, and possibly defective, versions of potentially life-critical processes.

Although model checking can evaluate whether the process model adheres to important safety requirements, it does not evaluate whether the process is robust against human or device failures. For example, model checking can assure that the recommended process always requires that the ventilator be started after weaning from CPB, but model checking does not provide any information about what happens if the ventilator is not correctly restarted when the process

model says it should be. Fault Tree Analysis (FTA) [33] and Failure Mode and Effects Analysis (FMEA) [32] are well-known safety analysis approaches that provide feedback about how resilient the process is to such failures. For a user-specified hazard, such as blood of the incorrect type being delivered from the blood bank, an FTA tool [7] can automatically derive a fault tree from the Little-JIL process model and then determine minimal cut sets, the minimal combinations of events (usually incorrectly performed steps) that could cause the hazard to occur. Conversely an FMEA tool can use the model [35] to show how the results of an incorrectly performed step can propagate to other steps, providing insight about possible hazards to be considered for FTA analysis [9]. Typically, the creation of fault trees and FMEA tables is done by humans and is, thus, labor-intensive and error prone. The use of automated tools leverages the effort already taken to create a verified process model to create fault trees and FMEA tables for a large number of potential hazards and faults.

These sorts of process models and analyses seem to have the potential to reduce process errors and improve safety. Mertens et al. [23] demonstrated a roughly 70% reduction in chemotherapy errors reaching the patient after the application of these process modelling and analysis methods.

Process Guidance: The analyzed process models are used to drive smart checklists to guide process performers [4]. A preliminary version of a generated smart checklist for part of the aortic cannulation assessment process (a necessary step common to all procedures in cardiac surgery using the heart-lung machine), is shown in Figure 2. This checklist is a view of the steps that the surgeon has performed, is performing, and is about to be asked to perform next.

The top of the figure shows patient-relevant information. The rows of text describe the steps in the process, where a green-shaded row indicates a step that is currently in progress and a gray-shaded row indicates a step that has completed (no future steps are shown in this figure). In this example, the “perform aortic annulation assessment and selection” step is in progress. Its first substep, *assess aortic cannulation site*, has been successfully completed, as shown by the gray background and green checkmarks on the lowest level substeps, and the second substep, *choose aortic cannulation site* is being executed. The first of its substeps did not complete successfully, as shown by the red X, and the step “confirm and choose aortic cannulation using long cannula” can now be started. (This step would not have been shown on the checklist unless *confirm and choose aortic cannulation using standard cannula* failed to be completed successfully.) If the substep

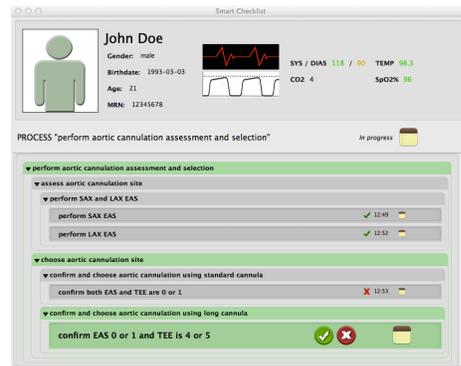


Fig. 2. Example of a generated smart checklist for part of the aortic cannulation assessment process

confirm EAS 0 or 1 and TEE is 4 or 5 completes successfully, the surgeon (or an assistant) would click the “completed successfully” button (the green button with the white checkmark). If the step cannot be completed successfully, the “failed to complete successfully” button (the red button with a white X) would be clicked. When either button is clicked, the completion status and time will be recorded and the checklist will update with steps that are now to be executed. As can be seen from this example, the amount of detail that is presented in a context-aware checklist depends on the structure (e.g. the step hierarchy) and the details captured in the process model. Annotations can be added to the process model to direct what information is actually displayed.

3 Modelling Cardiac Surgery

The incidence of perioperative stroke in cardiac surgery in the last 20 years has not decreased even in the face of improved surgical techniques and medical management [22]. The mortality associated with stroke ranges from 19%–32.8% versus 2.6%–4.9% for patients without a stroke, representing a 6-7 fold increase [2]. Morbidity associated with a perioperative stroke is responsible for a doubling of the length of stay in the intensive care unit and hospital as well as doubling of the cost of care [22]. The type of perioperative stroke associated with cardiac surgery is predominantly embolic [31]. Aortic manipulation can lead to thrombosis and embolism particularly with cannulation and clamping of the aorta during on-pump CABG or AVR. Disruption of significant aortic atherosclerotic plaques leads to perioperative stroke [15] and intraoperative measures during CABG or AVR to identify and avoid plaques, such as epiaortic ultrasound [30], reduce atheroembolic stroke risk during cross-clamping and cannulation. The ACCF/AHA 2011 guidelines recommend routine epiaortic ultrasound scanning prior to aortic manipulation during CABG to mitigate the risk of embolic stroke (class IIa, level of evidence B) [13]. Unfortunately, this evidence-based safe practice recommendation is not routinely implemented. A recent meta-analysis demonstrated that off-pump CABG was associated with a significant 30% reduction in the incidence of perioperative stroke (1.4 versus 2.1%) compared to on-pump CABG [1]. Off-pump CABG is recommended in place of traditional CABG in the setting of significant aortic atherosclerosis.

The surgical process model we have implemented in the Little-JIL language acts as a high-level decision framework for determining the appropriateness of central aortic cannulation for both CABG and AVR procedures. While surgeons have traditionally relied on finger palpation to ascertain the location and extent of calcification in the ascending aorta, our model eschews such inferior practice [30]. Instead we propose that the evidence dictating the choice of aortic cannulation site be derived from complementary use of epiaortic ultrasound in the ascending aorta and transesophageal echocardiography in the aortic arch [13].

We have used model checking to verify a small number of properties of our preliminary models, including that EAS always be used before aortic cannulation and that the long cannula only be used when Katz scores support that use. We

have also constructed a fault tree for the hazard “wrong cannula selected for aortic cannulation” to understand how errors in acquiring, communicating, and making use of information about the EAS and TEE cause that hazard. In doing so we assure that checklists generated from this process will provide surgery teams with guidance that conforms to evidence-driven best practices aimed at reducing the incidence of stroke and increasing patient safety.

In future work we will go further, incorporating specifications of how information is to be transferred among teams via verbal communication to assure that these exchanges provide a correct basis for belief formation and activity. For example, we will develop models and use the model checker to assure that the acquisition and successful communication to all essential parties of two values – Katz score is 0 or 1 in ascending aorta (via EAS) and Katz score is 4 or 5 in aortic arch (via TEE) always precedes the decision to implement a long cannula technique for aortic cannulation. We will also explore the integration of events from sensors and other medical devices into the checklist as well as approaches for presenting information about the steps of other team members to provide process performers with a more complete overview of the state of the process. We will also consider different ways of presenting information about past and even future steps that might be more useful to process performers. Further, we will use FTA and discrete event simulation to study the etiologies of communication breakdowns, use fault injection to identify risks incurred by faulty communication, and mitigate risks by incorporating additional checks that assure correct communication. Eventually, we hope to carry out experimental evaluations in a mock operating room. This work is intended to ensure that the dynamically-generated checklists provide useful guidance in assuring effective communication in the surgical suite, and to determine whether such checklists can help reduce the risk of stroke and improve patient safety.

4 Conclusions

In this preliminary work, we have described the use of a semantically-rich process modelling language (Little-JIL) to support the surgical team in the performance of a critical step (aortic cannulation) in cardiac surgery, a complex, error-prone sociotechnical system. Human factors experts have recognized the potential for human fallibility in complex systems where cardiac surgeons make an average of 3.5 errors per hour with many of these errors leading to injury to patients. Procedural standardization and routine implementation of evidence-based safe practices have been recommended to improve patient safety in surgery. We propose to explore a novel approach to optimize team performance using smart checklists.

Acknowledgements. This material is based upon work supported by the Medical Robotics & Computer Assisted Surgery Laboratory (MRCAS) with internal funding and the National Science Foundation under awards IIS-1239334 and CMMI-1234070.

References

1. Afilalo, J., Rasti, M., Ohayon, S.M., Shimony, A., et al.: Off-pump vs. on-pump coronary artery bypass surgery: An updated meta-analysis and meta-regression of randomized trials. *Eur Heart J* 33(10), 1257–1267 (2012)
2. Anyanwu, A.C., Filsoufi, F., Salzberg, S., Bronster, D.J., et al.: Epidemiology of stroke after cardiac surgery in the current era. *J Thorac Cardiovasc Surgery* 134(5), 1121–1127 (2007)
3. Avrunin, G.S., Clarke, L.A., Osterweil, L.J., Christov, S.C., et al.: Experience modeling and analyzing medical processes: UMass/Baystate medical safety project overview. In: 1st ACM Int. Health Informatics Symp. pp. 316–325. Arlington, VA (Nov 2010)
4. Avrunin, G.S., Clarke, L.A., Osterweil, L.J., Goldman, J.M., et al.: Smart checklists for human-intensive medical systems. In: Workshop on Open Resilient human-aware Cyberphysical Systems (WORCS-2012) (Jun 2012)
5. Calland, J.F., Guerlain, S., Adams, R.B., Tribble, C.G., et al.: A systems approach to surgical safety. *Surg Endosc* 16(6), 1005–14 (2002)
6. Cass, A.G., Lerner, B.S., McCall, E.K., Osterweil, L.J., et al.: Little-JIL/Juliette: A process definition language and interpreter. In: 22nd Int. Conf. Softw. Eng. pp. 754–757. Limerick, Ireland (2000)
7. Chen, B.: Improving processes using static analysis techniques. Ph.D. thesis, University of Massachusetts (2010)
8. Clarke, Jr., E.M., Grumberg, O., Peled, D.A.: Model checking. MIT Press, Cambridge (2000)
9. Dehlinger, J., Lutz, R.: Bi-directional safety analysis for product-line, multi-agent systems. In: Workshop on Innovative Techniques for Certification of Embedded Systems (ITCES'06). San Jose, CA (Apr 2006)
10. Dunn, R.F.: Six amazing years. RAGs, NATOPS, and more. *Naval War College Review* 64, 98–110 (2011)
11. Fargen, K.M., Velat, G.J., Lawson, M.F., Firment, C.S., et al.: Enhanced staff communication and reduced near-miss errors with a neurointerventional procedural checklist. *J Neurointervent Surg* 5(5), 497–500 (2013)
12. Hales, B.M., Pronovost, P.J.: The checklist: A tool for error management and performance improvement. *J Crit Care* 21, 231–235 (2006)
13. Hillis, L.D., Smith, P.K., Anderson, J.L., Bittl, J.A., et al.: 2011 ACCF/AHA Guideline for coronary artery bypass graft surgery: Executive summary: A report of the American College of Cardiology Foundation/American Heart Association task force on practice guidelines. *Circulation* 124(23), 2610–2642 (2011)
14. James, J.T.: A new, evidence-based estimate of patient harms associated with hospital care. *J Patient Saf* 9(3), 122–128 (2013)
15. Katz, E.S., Tunick, P.A., Rusinek, H., Ribakove, G., et al.: Protruding aortic atheroma predict stroke in elderly patients undergoing cardiopulmonary bypass: Experience with intraoperative transesophageal echocardiography. *J Am Coll Cardiol* 20(1), 70–77 (1992)
16. Lalys, F., Jannin, P.: Surgical process modelling: A review. *Int J CARS* 9, 495–511 (2014)
17. Leape, L.L.: The checklist conundrum. *New Engl J Med* 370, 1063–1064 (2014)
18. Lerner, B., Christov, S., Osterweil, L.J., Bendraou, R., et al.: Exception handling patterns for process modeling. *Trans. Softw. Eng.* 36(2), 162–183 (Mar 2010)

19. Lingard, L., Regehr, G., Orser, B., Reznick, R., et al.: Evaluation of a preoperative checklist and team briefing among surgeons, nurses, and anesthesiologists to reduce failures in communication. *Arch Surg* 143(12–17) (2008)
20. Lipitz-Snyderman, A., Needham, D.M., Colantuoni, E., Goeschel, C.A., et al.: The ability of intensive care units to maintain zero central line-associated bloodstream infections. *Arch Intern Med* 171(9), 856–858 (2011)
21. MacKenzie, C.L., Ibbotson, J.A., Cao, C.G.L., Lomax, A.J.: Hierarchical decomposition of laparoscopic surgery: A human factors approach to investigating the operating room environment. *Minim Invasive Ther Allied Technol* 10(3), 121–128 (2001)
22. McKhann, G.M., Grega, M.A., Borowicz, L.M., Baumgartner, W.A., et al.: Stroke and encephalopathy after cardiac surgery: An update. *Stroke* 37(2), 562–571 (2006)
23. Mertens, W.C., Christov, S.C., Avrunin, G.S., Clarke, L.A., et al.: Using process elicitation and validation to understand and improve chemotherapy ordering and delivery. *Jt Comm J Qual Patient Saf* 38(11), 497–505 (Nov 2012)
24. Munchenberg, J., Brief, J., Raczkowski, J., Worn, H., et al.: Operation planning of robot supported surgical interventions. In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS 2000)*. vol. 1, pp. 547–552 (2000)
25. Neumuth, T., Trantakis, C., Eckhardt, F., Dengl, M., et al.: Supporting the analysis of intervention courses with surgical process models on the example of fourteen microsurgical lumbar discectomies. *Int J CARS* 2(1), 436–438 (2007)
26. Pronovost, P., Needham, D., Berenholtz, S., Sinopoli, D., et al.: An intervention to decrease catheter-related bloodstream infections in the ICU. *New Engl J Med* 355(26), 2725–2732 (Dec 2006)
27. Raunak, M.S., Osterweil, L.J.: Resource management for complex and dynamic environments. *IEEE Trans. Softw. Eng.* 39(3), 384–402 (2013)
28. Reason, J.T., Carthey, J., de Leval, M.R.: Diagnosing the “vulnerable system syndrome”: An essential prerequisite to effective risk management. *Qual Health Care* 10(2), ii21–ii25 (2001)
29. Rogers, S. O., J., Gawande, A.A., Kwaan, M., Puopolo, A.L., et al.: Analysis of surgical errors in closed malpractice claims at 4 liability insurers. *Surgery* 140, 25–33 (2006)
30. Rosenberg, P., Sherman, S.K., Loffler, M., Shekar, P.S., et al.: The [influence of epiaortic ultrasonography on intraoperative surgical management in 6051 cardiac surgical patients. *Ann Thorac Surg* 85(2), 548–553 (2008)
31. Selim, M.: Perioperative stroke. *N Engl J Med* 356(7), 706–713 (2007)
32. Stamatis, D.H.: Failure mode and effect analysis: FMEA from theory to execution. *American Soc. Quality* (Mar 1995)
33. Vesely, W., Goldberg, F., Roberts, N., Haasl, D.: *Fault tree handbook* (NUREG-0492). U.S. Nuclear Regulatory Commission, Washington, D.C. (Jan 1981)
34. Wahr, J.A., Prager, R.L., Abernathy III, J.H., Martinez, E.A., et al.: Patient safety in the cardiac operating room: Human factors and teamwork: A scientific statement from the American Heart Association. *Circulation* 128(10), 1139–69 (2013)
35. Wang, D., Pan, J., Avrunin, G.S., Clarke, L.A., et al.: An automatic failure mode and effect analysis technique for processes defined in the Little-JIL process definition language. In: *22nd Int. Conf. Softw. Eng. Knowl. Eng.* pp. 765–770 (Jul 2010)