A Scare in the OR Highlights Value of Systems Engineering

Daniel McFarlane

Trying to Get Out of the Way

Suddenly, the operating room (OR) anesthesiologist looks alarmed and confused. Our relaxed discussion from one second ago still hangs in the air. I’ve become invisible as he scrambles to understand what is happening with his patient. I don’t have answers—I’m a first-time visitor to an OR. His concern is contagious, and I instantly try to get myself out of the way. He barks a loud command for help, and three of the OR staff immediately stop their work and run over. I’ve jammed myself up against the wall to try to become small. Without looking at me, he shoves me to one side to make more room. Something bad is happening. A bank of four (eight total) infusion pumps has lit up with bright red alarm notices. The patient’s vital signs beat out overhead on a large-screen high-definition television (HDTV). I see each member of the OR team glancing up nervously as the patient’s blood pressure begins dropping.

What am I doing here?! I’m 10 minutes into a scheduled 15-minute clinician-guided visit into an OR in which a patient is undergoing open-heart surgery. I am an invited outsider to this healthcare setting. My background includes 23 years’ experience in systems engineering research and development (R&D) for military command and control. The Association for the Advancement of Medical Instrumentation (AAMI) contacted me two years before and asked me to participate in steering discussions for alarm safety and systems engineering. I led advancements on these topics for the U.S. military, and AAMI wanted to leverage this, if possible, to help improve healthcare. A clinical lead (not related to my AAMI participation) has invited me to experience an OR setting for 15 minutes, so that I might understand the alarm safety and systems engineering challenges in this context. To prepare for this visit, I’ve changed into full OR garb (e.g., scrubs, hat, mask) for the first time. I’m nervous, but I’m pretending not to be because I’m grateful for the opportunity. Before entering the room, I started silently repeating to myself, “Don’t touch anything, don’t touch anything, don’t touch anything.”

My First Visit to an OR

Until five seconds ago, my OR visit seemed calm and low key. My clinical-guide ushered me to a spot just behind and to the left side of the anesthesiologist. He is facing the top of the patient’s head surrounded by a semicircular collection of medical equipment. The operation is already underway and the anesthesiologist is monitoring several ongoing processes that he started perhaps an hour previously. I see him keeping his eyes on his patient and scanning the multiple instruments in front of him while he casually describes what is happening. Everything seems under control, and I don’t feel as if I’m distracting him.

I notice that he is prioritizing his primary responsibilities; he maintains a visual scanning workflow as he talks. Intermittently, he pauses momentarily from describing the situation to check various details.
The scene looks like a miniature theater. A blue sheet that is hung as a screen is blocking my view of the patient’s body. The patient’s head and the top of the operating table show through at the bottom middle of the screen as if a character on a small stage. In the foreground on each side of this “screen” and “stage” are what look like high-tech parted theater curtains. The left curtain is a collection of towers with seven hanging bags of liquid medication and a complex embroidery of wires and clear tubes going through two banks of infusion pumps. To its left is what appears to be a respiratory console with tanks of gases, as well as a second large console that I don’t recognize. The right “curtain” includes a ventilator console, a large physiologic monitor covered with streaming data waveforms, another syringe-style infusion pump (the eighth), and some equipment that I don’t recognize.

As the anesthesiologist talks, I watch as he continues to scan this myriad of devices and their screens. I silently comment to myself that his visual scanning workflow is both dynamically changing depending on what he sees and habituated to a degree that it is happening at a nonconscious level. I count at least 15 different device screens, with a total of more than 200 buttons/knobs, and they don’t match. About half of these screens are large displays (≥20 inches) with complex tiled visual layouts and lots of text. About 40 different tubes, wires, hoses, etc., are streaming down free-form from about 7 feet high in a chaotic interweave, including a pile of wires on the floor. The audio space is crammed with a rich set of different noises, including beeps, whirrs, hums, and people talking. I see hand-written notes on medical tape stuck in various places. One says, “Keep me plugged in,” while others show reminders about names of medications.

As the anesthesiologist continues to describe what he is doing, I’m having three simultaneous impressions: 1) this physician obviously is working at a rare genius level of cognition, 2) he must have several thousands of hours of training to have habituated this job, and 3) this makeshift system is nakedly exposed to a seemingly overwhelming number of different types of risks of human error. From my experience with human error, I’m befuddled why everything seems to be working so well. . . .

Oh wait, something seems wrong. New beeps and flashes have started.

**Figure 1.** Artist rendering of infusion pump display showing battery run time at 0.5 hours.

---

**Critical Care**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
</table>
| A | **Epinephrine**  
  0.15 µg/kg/min |
| B | **Dobutamine**  
  10 µg/kg/min |
| C | **Heparin**  
  13 units/kg/h |
| D | **Esomeprazole** (continuous i.v.)  
  8 mg/h |

**Battery run time = 0.5 h**

---

**Crisis Ensues**

A controller for a bank of four infusion pumps is announcing a new alarm message. The anesthesiologist is standing right in front of the controller, sees it immediately, and points it out to me. The display indicates that the battery run time is at 0.5 hours (Figure 1), meaning that the battery this device has only 30 minutes of power remaining. The anesthesiologist comments how odd this is because the device is clearly plugged into a power strip. We both can see the controller’s...
The medication stopped for about 50 seconds, and during this time, the patient’s blood pressure went down by about half. The entire team looks very concerned.

One of them then grabs a live power strip hanging from the ceiling at about 4.5 feet, unplugs something else to make room, and plugs in the power. The infusion pump controller has not gone totally dark and has maintained its prior programming. The anesthesiologist now has to get to the infusion pump cluster and manually starts pressing buttons on each device to get the medications to start moving again. Of note, if the battery on the device had totally drained before the power was connected, the anesthesiologist would have had to program each pump configuration from scratch. This would have required substantial additional time.

During the entire episode, the patient’s blood pressure has been steadily decreasing on the anesthesiologist’s monitor and on the overhead large-screen HDTV. The medication stopped for about 50 seconds, and during this time, the patient’s blood pressure went down by about half. The entire team looks very concerned. The head surgeon and anesthesiologist have a rapid exchange and decide how much of one of the key medications has been missed. The anesthesiologist then grabs a new syringe, quickly draws the agreed amount from a bottle at hand, and injects it directly into the line. The patient’s blood pressure starts to return to its previous level. The patient was not harmed, and the OR team resumes its previous work as the operation continues. No one seems particularly shaken. However, I saw their worry during the crisis and believe that if the debugging had taken only a little longer, a different outcome may have occurred.

Analysis
It may be tempting to jump to simplistic conclusions, such as 1) “Only carelessness can explain a dumb mistake like having the power strip plugged into itself;” 2) “A simple checklist process would prevent this type of thing;” and/or 3) “No problem occurred; the infusion pump alerted appropriately and enabled the anesthesiologist to address the issue in time.”

However, I don’t think conclusion 1 or 2 is accurate. Also, although conclusion 3 is technically true, room for safety improvement may exist beyond what seems to be a last-line-of-defense alarms approach to preventing disasters. I was observing the event and did not see any signs of carelessness; however, my prior experience with safety informs a different
interpretation. As noted previously, I have 23 years of experience leading systems engineering for decision science and technology R&D for the U.S. Department of Defense (DoD; 14 years with Lockheed Martin Advanced Technology Laboratories and 9.5 years with the U.S. Naval Research Laboratory). I led creation of the new alarm safety solution deployed on U.S. Navy Aegis warships. I have studied many of the National Transportation Safety Board crash reports and the science of human error, and I have conducted several dozen field observation studies to assess team performance of critical missions.

I did not observe carelessness. Instead, I observed that the OR work environment is organized to facilitate the performance excellence of the key performers, especially the head surgeon and his/her supporting staff. Being in the OR during an operation was like watching a world-class orchestral performance. The lead surgeon and the anesthesiologist were master soloists leading the team through the seemingly impossible intricacies of a highly technical piece of music. It was awe-inspiring, and right on the “living-edge” of what genius-level master soloists can deliver with excellent supporting help. The large and diverse collection of tools that had been assembled seemed designed to afford maximal flexibility and configurability. These expert performers had ready access to all low-level details of all devices. Also, the devices seemed largely unencumbered by cross-integration or interdependencies that would hinder their individual dynamic configurability and control.

**Insights**

Relative to the event that I witnessed, I offer two summative opinions.

**Opinion 1: Individual Excellence Is Not Enough**

First, the OR team and work environment were organized to optimize performance excellence of a few key performers and the separate devices
Individual excellence may not be sufficient in an environment as complex as the operating room. Systems thinking can help manage the interoperation of the various components and actors in this high-risk setting.

that they use. This optimization of excellence included excellent hospital, excellent OR, excellent utilities, excellent surgical leaders, excellent staff, excellent skills, excellent motivation, excellent work attitudes, excellent work practices, excellent equipment, excellent medical resources (e.g., medications), excellent devices, excellent device flexibility, excellent access to low-level controls, and excellent configurability. All of the equipment looked new, extremely clean, well made, high quality, and expensive. The patient was receiving the best possible quality of the collection of individual people and devices available anywhere. I also saw, however, that individual excellence is likely not enough in such a complex environment. Even when excellence at the individual performer level is done well, this seems to not be completely effective in mitigating certain critical risks of catastrophic team-level errors.

Opinion 2: Attending to Low-Level Controls May Be a Distraction

My second summative opinion was that multiple hundreds of low-level control surfaces all were nakedly exposed to instant change. I use the phrase “control surfaces” to describe the entire collection of all the user interface mechanisms (hardware and software) across all devices that could be used to invoke function or change function configuration. These individual devices had been organized into a custom ad hoc system with complex interactions and interdependencies affecting the patient’s physiological “system.” Any single small change, such as a temporary pause of medication delivery, could have a large-scale impact on multiple different aspects of the overall “system.” Only the individual state of the various lowest-level devices was displayed. The states of the higher-level interactions and interdependencies was not visualized; they only existed in pieces within the minds of the physicians and staff. Dynamically assembling the several hundred state points and cognitively synthesizing them into actionable high-level situational awareness must require intense concentration and cognitive workload. The strain of this information processing effort must subtract from performers’ available cognitive resources to concentrate on the actual operation and patient. I suspect that a detailed cognitive task analysis would show that key OR leaders and staff are highly loaded with lowest-level information processing tasks and that this negatively affects the cognitive resources they have available for attending to the actual operation.

Errors as ‘System Bugs’

From observing this ad hoc OR system design, I inferred that the lead physicians must be ready for literally anything to happen at any time. The nakedly exposed low-level dynamic configuration potential and focus on low-level status information seem optimized to give the anesthesiologist the ability to respond to the unexpected—as if no one knows the future possible limits or direction of change for the patient. I believe this focus on low-level flexibility, however, has a cost in the potential for human error. To use an analogy: If every taxi included an exposed button for an ejection seat, would you feel safe hiring a ride?
My systems engineering perspective tells me that the configuration of the collection of OR devices is a kind of program. To use an analogy, software engineers delivering a complex program with hundreds of variables and sophisticated interdependent functions know better than to change the software while someone is using it to do a job. Software bugs typically occur whenever a software program is changed. The risk of inadvertently causing a new bug at run time is recognized within the software engineering community as a serious potential problem, and best practices do not allow changing software during use. The OR team, by contrast, seemed to me to routinely “reprogram” their system during use. The fact that I saw them experience a bug did not surprise me. The fact that the OR team was able to debug it so quickly was impressive.

Benefits of Systems Thinking To Healthcare
Systems engineering is a type of engineering that addresses how all the actors and components contributing to a complex endeavor will interoperate to achieve a common objective. For the OR example that I witnessed, the complex endeavor was to deliver improvement to this patient’s health through surgery—the common objective. In this system, an infusion pump is one component (of many) and the anesthesiologist is one actor (of many). A basic assertion of the systems engineering perspective is that it is not enough for the individual components and actors to perform with excellence. To achieve safety in a complex enterprise, an additional mechanism is required that facilitates and manages the interoperation of all the components and actors. From observing this OR, I believe that proven systems engineering concepts and technologies from the DoD could greatly improve patient safety.

An opportunity exists to greatly reduce the risk of human error in the OR by introducing a flexible system-level framework with automation for executing system-level policy models. This could off-load to automation a majority of the risk of inadvertently causing a new bug at run time is recognized within the software engineering community as a serious potential problem, and best practices do not allow changing software during use.
low-level information processing tasks that are currently being done in the heads of key OR people. Just because the usage policy model must remain flexible does not mean that execution of that policy model must be executed in the head of a busy surgical leader.

Such system-level automation could have alerted the anesthesiologist to the offending power-strip-plugged-into-itself bug before the surgery began.

I only saw one thing at the system level in the current OR: people. For example, an infusion pump does not have access to any contextual information about where it is, what is happening, who is its user, the user’s objectives, patient information, task goal, and possible interaction effects with other devices. The only thing an infusion pump “knows” is its own internal device-level task model and configuration. This is its “program”—to deliver user-specified drug X at rate Y unless one of conditions A, B, or C occur—and then react by doing X. Its data environment is isolated or siloed. I suggest that in addition to these current device-level programs or policy models, many system-level programs or policy models could be useful. System-level programs could include, for example: “When an infusion pump is being used in an OR during a surgery, it must have access to ‘live’ wall power. If this usage requirement is ever not true, generate an alert and deliver it to the relevant OR anesthesiologist in the following way . . . ”

Such system-level automation could have alerted the anesthesiologist to the offending power-strip-plugged-into-itself bug before the surgery began. In an envisioned system-level policy engine, the automation could have polled the OR room and determined the devices that were present. Perhaps radiofrequency identification tags on all devices could enable the room to respond with a list of device identifications. A future networked integrated data environment could have enabled the system to then poll each infusion pump in this OR and determine whether they were operating on wall power. The offending set of infusion pumps and their controller would have responded “no,” and the system-level policy then would have generated an alert to the anesthesiologist in this OR. Of note, this concept does not require devices to be any smarter about their context. The only needed device-level change would be for devices to be able to respond to basic status queries on a network.

In this envisioned approach, hospital leadership (including OR leaders) could dynamically collaborate on the specification of their custom system-level policy model. This would enable them to get whatever system-level automation services they decided would be useful. Most of the technologies required to realize this vision exist; they include flexible system frameworks, lightweight device integration mechanisms, policy model automation engines, and end-user development methods to enable hospital leadership to collaborate on the specification of policy models. Implementation of such a system could dramatically reduce the amount of low-level information processing that anesthesiologists, surgeons, and their team need to perform during operations. This reduction in cognitive workload would free up more cognitive cycles for the OR team to concentrate on the patient and avoid the distraction of having to do crisis debugging of configuration “bugs.”

Multiple organizations already have begun research into integrated clinical environments (ICEs) to improve healthcare safety. A full background review is beyond the scope of the current article. One such project is ICE STORM (Integrated Clinical Environments, Training, Operations, Research, Methods). Boosman and Szczerba1 describe the underlying systems engineering approach from a DoD perspective.

Reference
IT Collection

This new collection is your one-stop resource for 80001 and Medical Device Data System (MDDS) guidance.

Searchable and easy to use, it includes:
- 11 standards (80001-1, 80001 TIRs, 14971, TIR24971, TIR80002-1, TIR32, 62304, & SW87)
- Getting Started with IEC 80001 manual
- 2 webinars on 80001
- 3 webinars on MDDS
- 23 peer-reviewed articles

Order Code: ITCOL-CD
List Price: $480
AAMI Member Price: $268

To order, call 1-877-249-8226
or visit http://my.aami.org