Usability Engineering Recommendations for Next-Gen Integrated Interoperable Medical Devices

Paolo Masci and Sandy Weininger

Abstract

This article reports on the development of usability engineering recommendations for next-generation integrated interoperable medical devices. A model-based hazard analysis method is used to reason about possible design anomalies in interoperability functions that could lead to use errors. Design recommendations are identified that can mitigate design problems. An example application of the method is presented based on an integrated medical system prototype for postoperative care. The AAMI/UL technical committee used the results of the described analysis to inform the creation of the Interoperability Usability Concepts, Annex J, which is included in the first edition of the new ANSI/AAMI/UL 2800-1:2019 standard on medical device interoperability. The presented work is valuable to experts developing future revisions of the interoperability standard, as it documents key aspects of the analysis method used to create part of the standard. The contribution is also valuable to manufacturers, as it demonstrates how to perform a model-based analysis of use-related aspects of a medical system at the early stages of development, when a concrete implementation of the system is not yet available.

Integrated interoperable medical devices are a new generation (gen) of medical systems whose components can be seamlessly combined using standard communication networks. In this vision, medical devices will no longer be stand-alone elements but rather interoperable components of an integrated system. Clinicians will be able to create new medical systems simply by assembling interoperable devices. For example, combinations of medical devices may provide checks and balances that can be used to enhance the effectiveness and safety of the individual devices in the treatment of particular patient conditions.

In anticipation of the development and commercialization of interoperable medical devices, experts from regulatory authorities, industry, and academia have worked together to create recommendations for enabling safe and secure interoperability. These recommendations were recently published as a new standard: ANSI/AAMI/UL 2800-1: 2019.[1]

Usability engineering is one of the topics addressed in the interoperability standard. Usability engineering focuses on human-machine interaction and aims to ensure that an interactive system can be used effectively and safely. Usability engineering in medical devices is already challenging in the current generation of medical devices. Recent studies highlight that usability engineering issues affect nearly half of the design anomalies reported in software recalls[2] and that use errors resulting from usability engineering problems contribute to 28% of the adverse events reported for medical devices in the U.S.[3] When interoperable medical devices are integrated into systems, additional challenges will likely emerge. For example, integrated devices have multiple pathways for input and output that will increase the system's overall complexity. In addition, the clinician's role may change from operator of a single device to supervisor of multiple integrated devices, potentially introducing new opportunities for use errors that were not possible in the stand-alone devices.

This article demonstrates how an existing model-based analysis method[4] can be adapted to support usability engineering of next-gen integrated interoperable medical devices. The original method was conceived for stand-alone devices. Here, the method is applied to

***Paolo Masci**, PhD, is a senior research scientist at the National Institute of Aerospace in Hampton, VA. Email: paolo.masci@nianet.org*
Corresponding author

***Sandy Weininger**, PhD, is a senior engineer with the Division of Biomedical Physics of the Office of Science and Engineering Laboratories in the Center for Devices and Radiological Health of the Food and Drug Administration in Silver Spring, MD. Email: sandy.weininger@fda.hhs.gov*

interoperability functions. A concrete example is presented based on a prototype system for postoperative intensive care.

The adapted method presented in this work was successfully used in practice, for the development of the Interoperability Usability Concepts, Annex J, which is included in the ANSI/AAMI/UL 2800-1:2019[1] standard on medical device interoperability. It proved useful for systematically exploring potential latent design anomalies in interoperability functions that could lead to use errors and for guiding the identification of design recommendations that can be used by developers to prevent or mitigate the identified design anomalies.

Integrated Interoperable Medical System: Concepts and Definitions

An integrated medical system (hereafter referred to as "system") typically would include the following components:

- A set of interoperable medical devices (hereafter referred to as "devices"), each delivering a medical treatment or monitoring the patient's condition. Each device may provide an operator interface to allow the operator to interact with the device (e.g., to check the state of the device, program therapy parameters, or set alarm levels).
- A set of interoperable devices that are not medical devices but may also provide an operator interface.
- A system controller executing a set of applications (apps) necessary to integrate multiple devices. The controller can interact with multiple devices to implement automated functions such as safety interlocks, smart alarms, and clinical decision support systems. The controller can be equipped with an operator interface that displays the state of automated functions (e.g., whether a function is active/inactive), the state of interoperable medical devices connected to the controller, and a view of the data received/sent by the controller.
- One or more operators interacting with the devices and the system controller, in order to configure the delivery of the therapy and receive feedback about the health status of the patient.
- A patient receiving medical treatment. The patient might be connected to devices that administer a treatment (e.g., infusion pumps) and to devices that monitor physiological health indicators (e.g., pulse oximeters).

Example: The Integrated Clinical Environment

An example of an integrated medical system is depicted in Figure 1. It is an integrated clinical environment (ICE)[5,6] prototype for postoperative intensive care. The safety goal of the system is to administer, in a controlled manner, a prescribed infusion therapy to a patient and monitor the patient's vital signs. The patient's health is managed using two interoperable medical devices and an ICE app that integrates the two devices. An infusion pump administers pain relief medication intravenously. A pulse oximeter monitors vital signs data. The ICE app integrates and controls these two devices. It continuously checks physiological parameters produced by the monitor and, if the parameters move outside safe ranges, stops the infusion pump (e.g., respiratory depression can be triggered by an excess of pain relief medication) and alerts clinical staff.

Analysis Method

A model-based analysis method[4] is used to analyze the design of the ICE system and identify possible latent anomalies that could lead to use errors. A use error refers to "a situation in which the outcome of device use

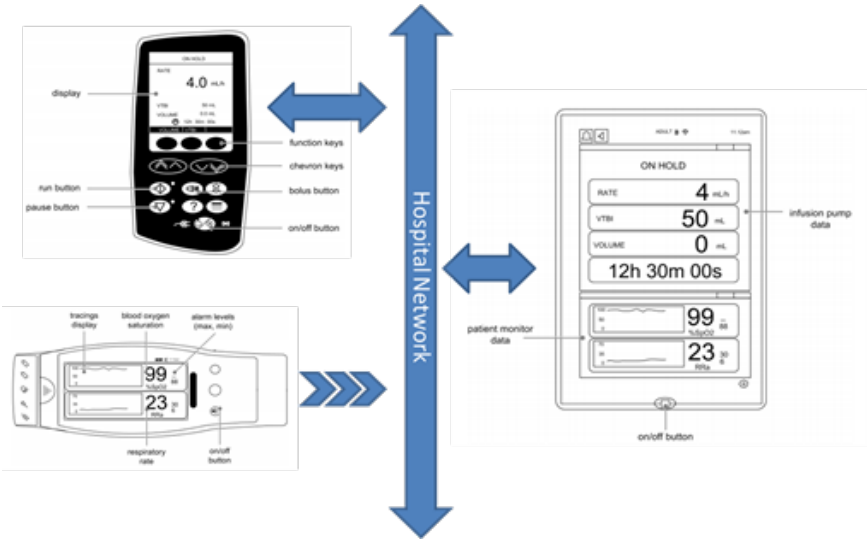

Figure 1. Integrated clinical environment.

was different than that intended, but not due to malfunction of the device."[7] In integrated medical systems, these errors can be triggered by user interfaces (UIs) that have not adequately considered interoperability. These errors can also occur in specific situations in which a combination of factors erroneously promote incorrect use of the device.

The analysis method represents the design of the sociotechnical system under analysis as a control process, in which controllers (either operators or technological elements) interact with controlled processes. In this view, hazards are interpreted as unsafe control actions, and use errors are identified by reasoning on possible unsafe control actions.[8] The main steps of the analysis method are as follows.

Step 1: Develop a Control Diagram

The purpose of the control diagram is to highlight the main functional components of the system and the interactions among them. The control diagram is depicted as a labeled directed graph. Nodes in the diagram represent technological components (e.g., a supervisor app, an interoperable device) or humans (e.g., operator, patient). Arrows between nodes represent control and feedback relations between components (e.g., an operator performing a user action on a device, feedback provided by a device to an operator, exchange of data and commands between devices, mechanical force). Figure 2 provides an example of a control diagram.

Step 2: Identify Foreseeable Use Errors

Foreseeable use errors are identified by considering the following general classes of unsafe control actions:

- UCA1. A user action required for safety is omitted.
- UCA2. A user action required for safety is performed incorrectly or at the wrong time/in the wrong order/for the wrong duration.
- UCA3. An unsafe user action is performed.

Step 3: Identify Design Anomalies

This step identifies possible issues in system design that could cause unsafe control actions leading to use errors. For each unsafe control action identified in the previous step of the analysis, focus on control loops involving the unsafe control action and reason about possible design anomalies in the components included in the control loop. A set of usability engineering design principles is used as guidance for making hypotheses about possible design anomalies. The principles capture best practices in interaction design, as described in usability engineering standards[9,10] and usability heuristics.[11,12]

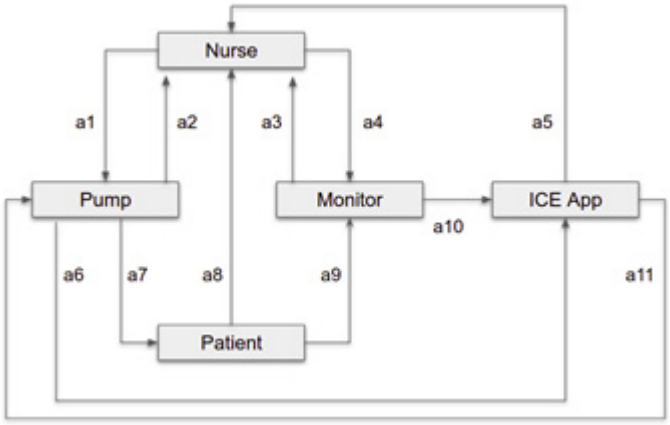

Figure 2. Control diagram of the integrated clinical environment (ICE) system.

Step 4: Generate Design Recommendations

This final step identifies design recommendations that can resolve or mitigate the identified design anomalies. The same set of usability engineering design principles used for the identification of design anomalies can be used as guidance for the definition of design recommendations.

The steps illustrated above are from the original method,[4] which is designed to support the analysis of stand-alone devices. Here, the method is adapted to focus on interoperability functions. To adapt the method to interoperable devices, step 3 is here modified by adding the following constraint on the control loops considered in the analysis: Control loops should include at least one operator and two or more technological components.

Results

This section presents the application of the analysis method to the ICE system, an interoperable medical system prototype

described in previous work.[5,6] A tool for checking possible ambiguities in natural language requirements[13] was also used during part of the analysis to improve the textual descriptions of hazards and design recommendations.

Control Diagram

The control diagram developed in this work for the analysis of use-related aspects of interoperability functions in the ICE system includes the following components (Figure 2):

- A nurse controls the infusion pump and the patient monitor (arrows a1 and a4 in Figure 2). The nurse can also interact with the UI of the ICE app to check remotely the status of the infusion pump and the patient monitor (a5).
- An infusion pump injects a pain relief medication into the patient (a7). The goal of the therapy is to provide a level of the medication high enough to address the patient's pain but low enough to avoid a respiratory depression. The front panel of the pump can be used by the nurse to check the state of the infusion (a2), program infusion parameters, and start/stop the infusion (a1). Through the network interface, the pump periodically sends the infusion status to the ICE app (a6) and receives commands from the ICE app (a11). Of note, modern infusion pumps can receive commands to program infusion parameters (but a nurse is required to verify the parameters and start the infusion at the pump), update infusion parameters of a running infusion by populating a new dosing parameter (titration), or populate a new volume to be infused for a subsequent bag. The ability to pause an infusion with a network command is currently only in prototype stage.
- A patient monitor records the patient's respiratory rate (RR) and blood oxygen level (peripheral oxygen saturation [SpO_2]; a9). The front panel of the patient monitor can be used by the nurse to check the monitored data (a3) and configure the monitor (a4). Through the network interface, the patient monitor sends the monitored data to the ICE app (a10).
- An ICE app implements a safety interlock mechanism that is designed to prevent patient harm and trigger smart alarms. Specifically, when the monitor data indicate the onset of respiratory depression, the ICE app triggers a smart alarm system and sends a command to the pump to interrupt drug delivery—the respiratory depression may or may not be caused by the infusion, but when a respiratory depression occurs, the injection of pain relief medication must always be paused. The nurse can interact with the ICE app through a UI that displays the app state (e.g., whether the safety interlock function implemented by the app is active), the infusion pump state (e.g., whether the pump is infusing or paused), the patient monitor state (e.g., the patient's respiratory rate and SpO_2 values detected by the monitor), and alarms generated by the pump, monitor, and ICE app (a5).
- A patient receives a medical treatment through the infusion pump (a7). The patient is constantly monitored through sensors (a9) and can be visually monitored by the nurse (a8).

Use Errors

The set of use errors is obtained by instantiating the three general classes of unsafe control actions (UCA1 to UCA3) for the system under analysis. Based on the functionalities of the ICE system being analyzed, the set of use errors considered for the analysis of interoperability functions is as follows:

- ERR-1. The nurse is unable to pause the infusion when the patient is in respiratory distress.
- ERR-2. The nurse is unable to start/resume the infusion or starts/resumes the infusion after a delay.
- ERR-3. The nurse erroneously disconnects or misconfigures the monitor.

Design Anomalies and Design Recommendations

This section presents a selection of design anomalies and corresponding example design recommendations obtained with the model-based analysis. The presented

selection does not capture all possible anomalies in the ICE system. The design recommendations are for illustrative purpose and do not represent requirements currently mandated by regulatory authorities. This content is meant to give the reader an understanding of how the analysis works and what kind of results can be obtained.

Design anomalies and design recommendations are grouped by use errors. The following information is presented for each use error:

- A design anomaly that could trigger the use error.
- The unsafe control loop in the control diagram that was considered when reasoning about the design anomaly, as well as a reference to the usability engineering principles that were used to identify the design anomaly. The full list of principles is provided in the supplemental material for this article (available at www.aami.org/bit).
- A scenario describing the design anomaly and the safety consequences.
- A set of design recommendations that can mitigate the considered use error.

ERR-1. The nurse is unable to pause the pump.

- **Design anomaly 1-1.** The front panel of the pump is erroneously designed to become disabled under certain conditions.
  - **Unsafe control loop.** The nurse tries to control the pump using the front panel of the pump, but the pump erroneously disables the front panel when connected to the ICE app (Figure 3).
  - **Principle.** Availability of controls (P11; control widgets necessary for safe operation are available on the UI at the right time [see supplemental material]).
  - **Scenario.** The nurse needs to pause the infusion because the patient needs assistance for other emergencies. Controls on the front panel of the pump are erroneously not available (e.g., because the pump disables the front panel when the ICE app is connected to the pump). As a result, the nurse fails to pause the infusion using the controls provided on the front panel of the pump.
  - **Recommendation.** The pump provides authorized users with means to take control of the pump at any time, regardless of the integration status with the ICE app.
- **Design anomaly 1-2.** The ICE app is erroneously designed to become inactive under certain conditions.
  - **Unsafe control loop.** The ICE app does not receive data from the monitor, feedback provided by the ICE app suggests everything is OK, and the ICE app does not send stop infusion command to the pump when necessary (Figure 4).
  - **Principles.** Availability of feedback (P3; feedback reporting important information or events should be visible and timely and should

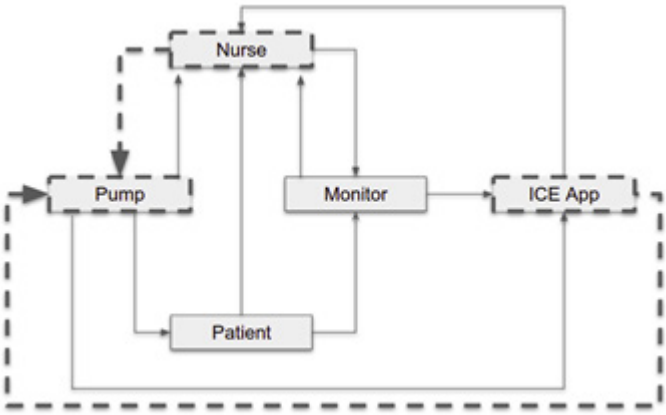

Figure 3. Control loop considered when reasoning about design anomaly 1-1. Dashed lines indicate the control relations and the components included in the control loop. Abbreviation used: ICE, integrated clinical environment.

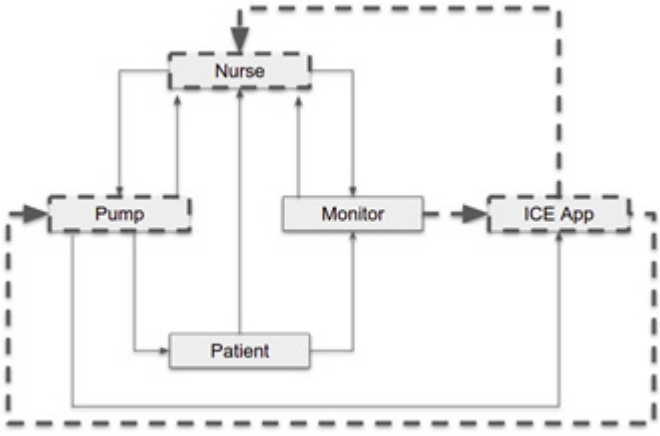

Figure 4. Control loop considered when reasoning about design anomaly 1-2. Abbreviation used: ICE, integrated clinical environment.

provide correct information) and salience of feedback (P4; feedback should be prominent and easy to locate on the UI).
  - **Scenario.** The ICE app becomes disabled and does not send commands to the pump to stop the infusion when the monitor indicates the onset of respiratory depression. The nurse fails to understand that the ICE app is disabled because nothing on the ICE app interface clearly indicates that the app is inactive (e.g., warnings are reported by the app but are not prominent or easy to notice or are displayed only temporarily).
  - **Recommendation.** When the ICE app becomes disabled, the app alarms and requires the user to take action.
- **Design anomaly 1-3.** The monitor erroneously sends historic data not representative of the patient's current condition to the ICE app.
  - **Unsafe control loop.** The monitor incorrectly sends old data to the ICE app, feedback provided by the monitor suggests everything is OK, and the ICE app does not send stop infusion command to the pump when necessary (Figure 5).
  - **Principle.** Availability of feedback (P3).
  - **Scenario.** The patient monitor is stuck on an outdated measurement value. Because of this, the ICE app is unable to detect the onset of respiratory depression and will not pause the infusion if the patient's condition deteriorates. The nurse fails to identify the problem because neither the front panel of the patient monitor nor the app interface indicate the problem.
  - **Recommendation.** Devices have a means of detecting the reception of data that is not current, including, for example, monitors based on time stamps or other metadata. Operators are alerted when the situation cannot be automatically recovered.
- **Design anomaly 1-4.** The monitor erroneously sends invalid data to the ICE app.
  - **Unsafe control loop**. The monitor sends incorrect data to the ICE app, feedback provided by the monitor suggests everything is OK, and the ICE app does not send stop infusion command to the pump when necessary (Figure 5).
  - **Principle.** Complexity of feedback (P2; feedback for frequent actions or important events [e.g., system failure or patient-related emergencies] should not require observing and understanding multiple information resources).
  - **Scenario.** Because of a software anomaly, one or more measurements and/or alarms transmitted from the patient monitor to the ICE app are invalid, e.g., negative values for respiratory rate and/or alarms. The nurse fails to notice the problem by looking at the patient monitor display, because SpO_2 and respiratory rate measurements are correct on the patient monitor display. Even though in principle the nurse could detect the problem by looking at the values displayed on the app interface and comparing them with those displayed by the patient monitor, this is in practice not feasible (e.g., the app interface and the patient monitor are in two different physical locations in the ward).
  - **Recommendation.** The ICE app automatically checks integrity and validity of data provided by controlled devices and reports critical errors to the user. Validity in this context

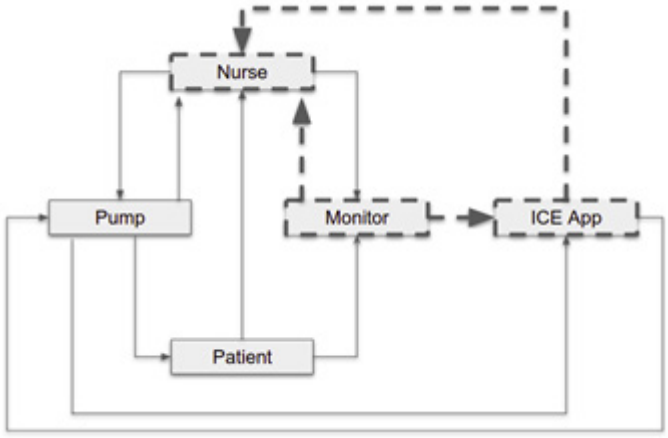

Figure 5. Control loop considered when reasoning about design anomalies 1-3 and 1-4. Abbreviation used: ICE, integrated clinical environment.

means checking that a SpO_2 measurement is physiologically within an accepted range. Integrity means checking the absence of data corruption (e.g., through cyclic redundancy check codes).

ERR-2. The nurse is unable to start/resume the infusion, or starts/resumes the infusion after a delay.

- **Design anomaly 2-1.** The monitor fails to link to the ICE app.
  - **Unsafe control loop.** The monitor is unable to send data to the ICE app, feedback provided by the monitor suggests everything is OK, the nurse tries to start the infusion using the front panel of the pump, and the ICE app erroneously sends a stop command to the pump (Figure 6).
  - **Principles.** Availability of feedback (P3) and complexity of feedback (P2).
  - **Scenario.** Unbeknownst to the nurse, the patient monitor fails to link to the ICE app. Because of this, the ICE app may send a pause infusion command or fail to initiate the closed-loop therapy protocol when the nurse tries to start the infusion therapy. The nurse fails to understand the situation at the bedside, because only the patient monitor and the pump are in sight and nothing on the user interface of the patient monitor indicates this problem.
  - **Recommendation.** The user interface of the monitor indicates when the connection link between the monitor and the ICE app is down.

ERR-3. The nurse erroneously disconnects or misconfigures the monitor.

- **Design anomaly 3-1.** Alarm thresholds are not reported up front in the monitor and/or in the ICE app.
  - **Unsafe control loop.** Monitor does not send alarms to the ICE app, and feedback on the user interface of the monitor and of the ICE app suggests everything is OK (Figure 7).
  - **Principles.** Availability of feedback (P3) and salience of feedback (P4).
  - **Scenario.** The nurse fails to notice that alarm thresholds in the monitor are too low and could prevent a prompt detection of the onset of respiratory depression. The problem may go unnoticed because, for example, alarm thresholds are not reported up front in the monitor and/or in the ICE app.
  - **Recommendation.** Authorized users are able to visually check, at any time, alarm thresholds and other critical settings of a controlled device from both the controlled device and the user interface of an ICE app.
- **Design anomaly 3-2.** All monitor alarms are erroneously disabled.
  - **Unsafe control loop.** The nurse erroneously pauses alarms indefinitely, the monitor does not send alarms to the ICE app, and feedback provided by the monitor and the ICE app suggests everything is OK (Figure 8).
  - **Principles.** Forgiveness for erroneous control actions (P10; safety interlocks

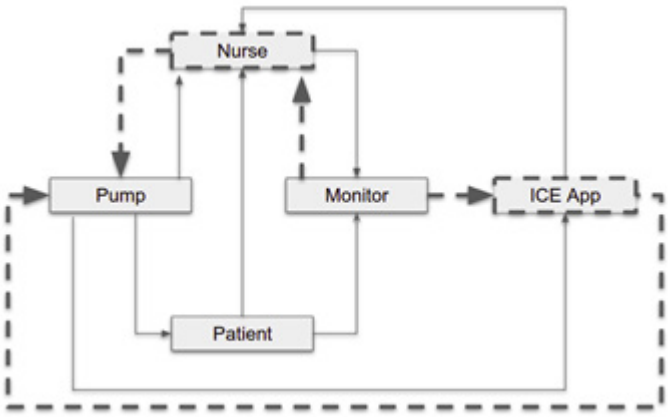

Figure 6. Control loop considered when reasoning about design anomaly 2-1. Abbreviation used: ICE, integrated clinical environment.

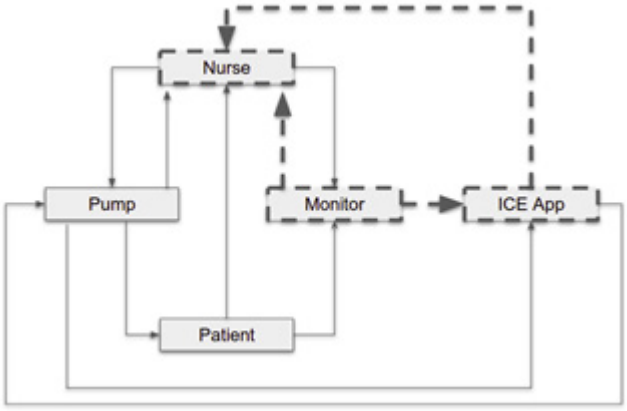

Figure 7. Control loop considered when reasoning about design anomaly 3-1. Abbreviation used: ICE, integrated clinical environment.

are available to prevent accidental activation of critical controls or block/mitigate foreseeable use errors) and salience of feedback (P4).

- **Scenario.** The patient monitor starts alarming because the SpO_2 sensor gets displaced or falls off. The nurse reaches the bedside, suspends the SpO_2 alarm, repositions the SpO_2 sensor, checks that the monitor correctly measures the SpO_2 value, and leaves. Unbeknownst to the nurse, the SpO_2 alarm has been suspended indefinitely. Because of this, the safety logic of the supervisor is also suspended. (The ICE app relies on the functionalities of the patient monitor to detect alarms.) Neither the front panel of the patient monitor nor the app interface clearly indicate that alarms are disabled and that the safety logic of the ICE app is suspended.
- **Recommendation.** Component manufacturers comply with IEC 60601-1-8[14] to consider alarm signal inactivation states and duration, the front panel of the patient monitor and the app interface indicate when alarms are disabled, and the user interface of the ICE app indicates when the safety logic of the app is suspended.

- **Design anomaly 3-3.** The ICE app is erroneously designed to pause the infusion when the nurse disconnects the patient monitor from the network.
  - **Unsafe control loop.** The nurse unplugs the monitor from the network, the ICE app erroneously treats the disconnection as an alarm, and the ICE app erroneously sends a pause infusion command to the pump (Figure 9).
  - **Principles.** Predictability of controls (P9; the UI provides means to help the user anticipate the effects or the consequences of control actions), availability of feedback (P3), and salience of feedback (P4).
  - **Scenario.** The nurse unplugs the monitor from the network. Unbeknownst to the nurse, the disconnection triggers an alarm in the ICE app, which in turn automatically triggers a pause command being sent to the infusion pump.
  - **Recommendation.** Interoperable devices provide a functionality to safely disconnect the device from the network. Disconnection of a controlled device from the network does not silently trigger critical control actions on other controlled devices. If disconnection of a controlled device affects a closed-loop safety function, the ICE app should actively alert the clinician that the closed loop delivery/protection is suspended.

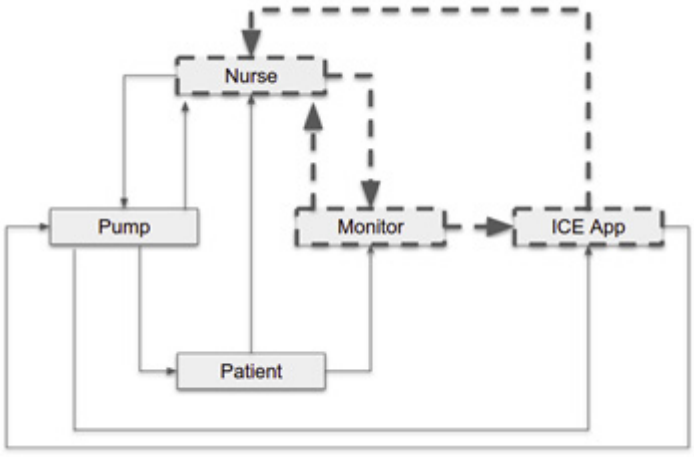

Figure 8. Control loop considered when reasoning about design anomaly 3-2. Abbreviation used: ICE, integrated clinical environment.

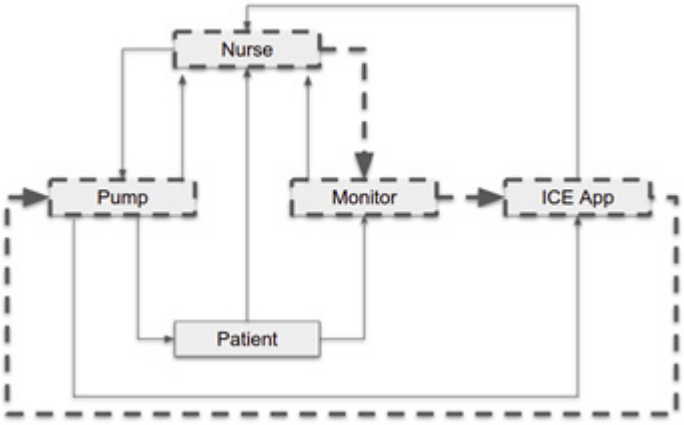

Figure 9. Control loop considered when reasoning about design anomaly 3-3. Abbreviation used: ICE, integrated clinical environment.

A synthesis of the analysis results described in this section for the ICE system is presented in Table 1.

Discussion

The presented example demonstrates how to apply an existing model-based analysis

Hazard Description	Use Error	Root Causes in System Design
The patient enters respiratory depression or is in excessive pain.	The nurse is unable to stop the pump	The front panel of the pump is erroneously designed to become disabled in certain scenarios.
		The ICE app is erroneously designed to become inactive in certain scenarios.
		The monitor erroneously sends historic data not representative of the patient's current condition to the ICE app.
		The monitor erroneously sends invalid data to the ICE app.
	The nurse is unable to start/resume the infusion, or starts/resumes the infusion after a delay.	The monitor fails to link to the ICE app.
	The nurse erroneously disconnects or misconfigures the monitor.	Alarm thresholds are not reported up-front in the monitor and/or in the ICE app.
		All monitor alarms are erroneously disabled.
		The ICE app is erroneously designed to pause the infusion when the nurse disconnects the patient monitor from the network.

Table 1. Synthesis of the analysis result for the integrated clinical environment (ICE) example.

method to interoperable medical systems. The analysis facilitates a systematic exploration of possible use errors and design anomalies from the early stages of the development process. The same process can be used as is throughout the entire development life cycle of the device. It aligns well with the recommended risk management process for medical devices (as described in ANSI/AAMI/ISO 14971:2019[15]) and helps developers improve their confidence that all major problems have been taken into account and appropriately addressed.

However, multivendor medical devices with interoperability functions, such as the ones considered in this work, are not yet available on the market. As such, understanding of how clinical staff can deploy and use this new generation of medical devices in practice remains limited. This may create blind spots in the analysis performed this early in the development process, as possible uses and misuses of the technology may substantially deviate from developers' expectations. The analysis needs to be iterated throughout the development process and refined with additional information available at later stages of the development process.

The set of usability engineering principles considered in this analysis is reported in the supplemental material. The set covers frequent safety concerns recorded for medical devices. It is not exhaustive and is meant to provide an illustrative example rather than a complete analysis. As such, developers can consider the presented set as a starting point, as opposed to a final target. For example, the set of principles can be extended to cover additional concerns that developers may identify throughout the development life cycle.

The focus of the analysis presented in this work has been on accidental use errors, as opposed to intentional malicious behavior. Although the presented analysis method is general, its effectiveness in addressing security-related concerns needs additional validation.

A different analysis method could have been adopted to identify usability engineering recommendations. Classic usability evaluation methods based on user studies usually are not an option at the early stages of development, as they build on user evaluations and user studies that require the availability of a functional prototype, and such prototypes are still not available. Notably, the presented analysis results can inform user studies carried out later in the development life cycle, when functional prototypes become available.

Classic hazard analysis methods (e.g., preliminary hazard analysis,[16] root cause analysis[17]) heavily rely on group brainstorming exercises and lack guidance on how to address use errors. In contrast, the method presented in this work is guided by control diagrams that capture relevant functionalities of the system under analysis, and a systematic process for using the control diagrams to reason about use errors and causal factors in system design. Other systematic analysis techniques (e.g., fault tree analysis,[18] hazard and operability analysis,[19] failure mode and effects analysis[20]) focus on component failures or deviation from reference parameters and tasks. Each of these techniques can be used for the analysis of use-related aspects of a system and can be used to complement analysis results obtained with the method presented in this work.

System theoretic process analysis (STPA)[8] is a model-based analysis method that represents the system as a control structure and uses a set of guide words to identify possible unsafe actions that could lead to hazards. The method covers the analysis of use errors but offers limited tools for reasoning about latent design anomalies in system design that could trigger use errors.

The analysis adopted in this work modifies STPA by adopting usability engineering principles as a means to guide the identification of possible use errors and their causal factors in system design. In fact, the original STPA analysis considers only three broad classes of causal factors: issues with feedback, issues with mental models, and issues with external information. These general classes are too coarse and provide little mental scaffolding for the identification of specific possible use-related problems in system design.

STPA extensions[21] have been proposed by others who use cognitive models for assessing the role of humans in complex automated systems. The cognitive models try to explain why humans behave the way the do. In contrast, the method presented in this work extends STPA by using established human factors engineering principles, which provide a more direct explanation of how design aspects of an interactive system can trigger use errors.

Conclusion

This article described the application of a model-based method for the analysis of usability engineering in next-gen interoperable medical systems. The method enabled a systematic analysis of possible latent design anomalies in interoperability functions that could lead to use errors. The obtained results were used to inform the development of the usability engineering recommendations included in the first edition of ANSI/AAMI/UL 2800-1:2019.[1]

Acknowledgments

Stefania Gnesi (ISTI-CNR), Michael Harrison (Newcastle University), John Hatcliff (Kansas State University), Scott Thiel (Hologic, Inc.), and Yi Zhang (Massachusetts General Hospital) provided useful feedback for the presented example.

Disclaimer

The mention of commercial products, their sources, or their use in connection with material reported herein is not to be construed as either an actual or implied endorsement of such products by the Department of Health & Human Services.

References

1. ANSI/AAMI/UL 2800-1: 2019. *Standard for Safety for Medical Device Interoperability*. Arlington, VA: Association for the Advancement of Medical Instrumentation.
2. Zhang Y, Masci P, Jones P, Thimbleby H. User interface software errors in medical devices: study of U.S. recall data. *Biomed Instrum Technol*. 2019;53(3):182–94.
3. Knisely BM, Levine C, Kharod KC, et al. An analysis of FDA adverse event reporting data for trends in medical device use error. *Proceedings of the International Symposium of Human Factors and Ergonomics in Healthcare*. 2020;9:130–4,
4. Masci P, Zhang Y, Jones P, Campos JC. A hazard analysis method for systematic identification of safety requirements for user interface software in medical devices. In: *Proceedings of the 15th International Conference on Software Engineering and Formal Methods, Trento, Italy, September 2017*. New York, NY: Springer; 2017.
5. Weininger S, Jaffe MB, Rausch T, Goldman JM. Capturing essential information to achieve safe interoperability. *Anesth Analg*. 2017;124(1):83–94.

6. ASTM F2761-09. *Medical Devices and Medical Systems—Essential safety requirements for equipment comprising the patient-centric integrated clinical environment (ICE)—Part 1: General requirements and conceptual model*. West Conshohocken, PA: ASTM International; 2009.
7. Food and Drug Administration. Human factors considerations. www.fda.gov/medical-devices/human-factors-and-medical-devices/postmarket-information-device-surveillance-and-reporting-processes. Accessed Aug. 30, 2021.
8. Leveson NG. *Engineering a Safer World: Systems Thinking Applied to Safety*. Cambridge, MA: MIT Press; 2016.
9. ANSI/AAMI HE75:2009. *Human Factors Engineering—Design of Medical Devices*. Arlington, VA: Association for the Advancement of Medical Instrumentation.
10. ANSI/AAMI/IEC 62366-1:2015+AMD1:2020. *Medical devices—Part 1: Application of usability engineering to medical devices*. Arlington, VA: Association for the Advancement of Medical Instrumentation.
11. Norman D. *The Design of Everyday Things*. New York, NY: Basic Books; 2002.
12. Nielsen J. *Usability Engineering*. San Francisco, CA: Morgan Kaufmann; 1993.
13. Bucchiarone A, Gnesi S, Fantechi A, Trentanni G. *A tool for the analysis of natural language requirements. In: Proceedings of the 2010 ACM Symposium on Applied Computing (SAC), Sierre, Switzerland, March 22–26, 2010*. New York, NY: Association for Computing Machinery; 2010.
14. IEC 60601-1-8:2006+AMD1:2012+AMD2:2020. *Medical electrical equipment—Part 1-8: General requirements for basic safety and essential performance—Collateral Standard: General requirements, tests and guidance for alarm systems in medical electrical equipment and medical electrical systems*. Geneva, Switzerland: International Electrotechnical Commission.
15. ANSI/AAMI/ISO 14971:2019. *Medical Devices—Application of Risk Management to Medical Devices*. Arlington, VA: Association for the Advancement of Medical Instrumentation.
16. Ericson CA. *Hazard Analysis Techniques for System Safety*. Hoboken, NJ: John Wiley & Sons; 2015.
17. Ishikawa K, Lu DJ. *What Is Total Quality Control?* The Japanese Way. Hoboken, NJ: Prentice Hall; 1985.
18. Lee WS, Grosh DL, Tillman FA, Lie CH. Fault tree analysis, methods, and applications: a review. *IEEE Transactions on Reliability*. 1985;R-34(3):194–203.
19. Kletz TA. *Hazop and Hazan: Identifying and Assessing Process Industry Hazards*. Boca Raton, FL: CRC Press; 2001.
20. Stamatis D. *Failure Mode And Effect Analysis*. Milwaukee, WI: American Society for Quality; 2003.
21. France ME. *Engineering for humans: a new extension to STPA* [doctoral dissertation]. Cambridge, MA: Massachusetts Institute of Technology; 2017.
22. National Patient Safety Agency, National Health Service. *Design for Patient Safety*. Leeds, UK: National Health Service; 2007.