Supporting Medical Device Adverse Event Analysis in an Interoperable Clinical Environment: Design of a Data Logging and Playback System

David Arney[1], Sandy Weininger[2], Susan F. Whitehead[1], and Julian M. Goldman[1],[3]

[1] MD PnP Program CIMIT, Cambridge, MA
[2] U.S. Food and Drug Administration CDRH/OSEL/DESE
[3] Departments of Anesthesia and Biomedical Engineering, Massachusetts General Hospital, Boston, MA

1 Introduction

It is often difficult or impossible for clinicians or regulators to find root causes of failures when adverse events happen. This problem will only get worse as healthcare organizations integrate more and more devices into their information systems in order to accomplish meaningful use of electronic medical records and to meet objectives for improved patient safety [3]. Moreover, the risk of liability is one of the factors in the reluctance of medical device manufacturers to make their devices interoperable. [4] This is becoming increasingly relevant to health care providers due to the recent FDA ruling on Medical Device Data Systems (MDDS)[5], which will require applicable hospitals to register as device manufacturers. Risk management standards including ISO/IEC 80001 [6] require users to mitigate many of the risks associated with interconnecting medical devices. A data logging and playback system can address many of these needs. Data loggers for interoperable systems should capture commands, device connections and disconnections, physiologic and technical alarms, physiologic data from patients, and other information about the status of devices. In this paper, we explore the issues involved in designing such a logging system and present some preliminary solutions.

Healthcare delivery organizations need a data logger for network-integrated devices that can capture the data needed for effective adverse events analysis. The purpose of the data logger is to record low-level device data (e.g. button presses and physiological data values) from individual medical devices, along with location information and data about the status of the medical device network, in an open, standardized, and time-synchronized manner. It is impossible to trace back to the origins of interactions between devices that can cause serious hazards to patients without a coordinated, time-synchronized log of all of the data sent by all of the devices in the system. This complete data record offers a more complete event picture than the highly filtered and processed data that goes into the EMR.

Challenges must be overcome at each step of adverse event analysis. Simply locating devices is frequently difficult, and unless devices are immediately sequestered following an incident their internal data log may be overwritten or deleted. Much data is entered manually, raising the problem of retrospective documentation where the clinician enters a value from memory or enters what they reconstruct it might have been. Reported times in the records may come from the clock on the wall, a device, or the clinicians wristwatch; current recording systems are not time synchronized. Thus, even something as simple as the start time of surgery or the time an infusion was started may be different in the nursing record versus the device internal log versus the anesthesia record.

Analysis of medical adverse events is commonly a manual process, whether a patient is affected or the event is an unexpected device interaction that does not directly impact patient care. The analysis team must locate and sequester the devices involved, get data out of the devices in a format they can use, analyze the data to figure out what happened, and then produce reports detailing their findings. Interviewing of clinicians is an integral part of the process. Adverse event reports are usually captured individually, on a case-by-case basis. Hazard analyses of medical devices and systems are based on collections of these reports, and the safety of devices is predicated on these hazard analyses. Adverse events may occur even if all of the devices act in accordance with their specifications; if this happens repeatedly it may be an indication that something needs to change– possibly the device specifications, other processes, or the device's use environment. After an event is analyzed and the root cause is found, it is usually impossible to know how widespread the problem is. Because detailed logs are not kept, is not possible to look back at similar situations in the past to see if a similar chain of events occurred that could indicate other undetected adverse events or near misses. Adverse event analysis thus must operate on a series of disconnected, anecdotal, individual cases rather than being able to apply epidemiological principles to consider device failures across populations.

A recorder that logs data from all of the medical devices attached to a patient has the potential to facilitate radical improvements in patient safety, with the added benefit of simplifying troubleshooting of network-related problems. Event logs and adverse event analysis entail a cross-cutting effort across clinical engineering, IT, compliance, biomedical engineering, quality assurance, and clinical care in the OR, ICU, and other settings. We believe that better collection and more accurate documentation of adverse events will lead to safer medical devices and systems in the future.

2 Design

Our design is based on the ICE (Integrated Clinical Environment) architecture from the ASTM F2761-09 standard [2]. Medical devices associated with an single, high-acuity patient are all connected through an ICE Network Controller that contains a data logger. Figure 1 shows the general architecture of an ICE system. As medical device interface capabilities improve, more device data will be available to the data logger. F2761-09 requires logging of "user interaction with devices" – e.g., button presses – that will help add context to events to facilitate analyses of usability problems.

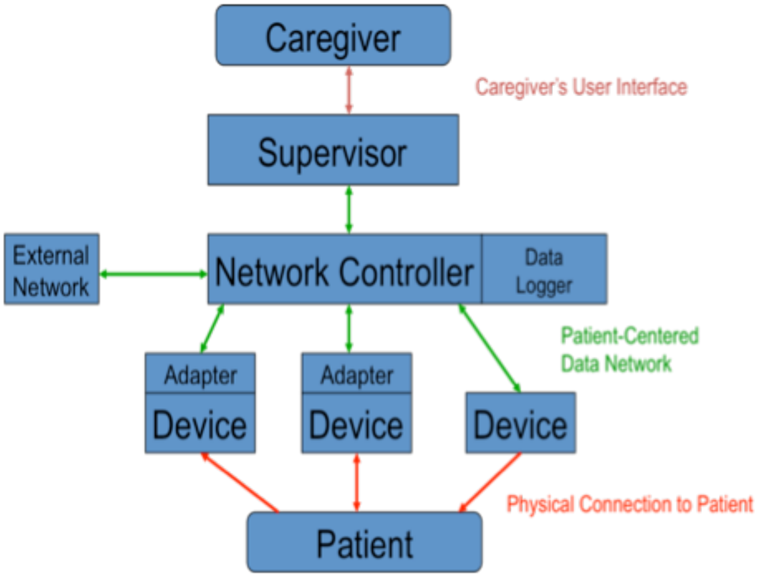


Fig. 1. ASTM F2761-09 ICE Architecture Overview

Types of logging The event recorder will be useful for analyzing adverse events and near misses with patients as well as debugging interactions between multiple medical devices (such as bedside monitors and remote alarm systems) or between medical devices and other IT systems (e.g. the EHR). We anticipate that this data will also be extremely useful for developing advanced clinical algorithms and analyzing patient outcomes. Log data for debugging network interactions will typically be much more detailed than that used for clinical event analysis. For instance, when a pulse oximeter transmits SpO_2 on the network, a log for clinical data would create a single entry for that data value. A log for debugging the network would record the request for the data, each of the likely multiple packets comprising the data transmission, and the acknowledgement message. Thus, debugging logs are a superset of, and take up much more space than, clinical data logs. A logging system should include options to allow users to select how detailed they want the logs to be. Data compression algorithms may be used to reduce the size of the log files provided that they do not lose data in the process.

Timestamps and Logical Clocks The ICE network controller will contain a real-time clock set using the network time protocol (NTP). Synchronization of the network controller clock, and information about the accuracy with which it was set, will be entered in the log as it happens. Data from devices on the network will be entered in the log along with a sequence number (described below) and a timestamp from the network controller clock. The network controller will not attempt to set the device clocks or adjust the time they report, though some supervisor applications may adjust device clocks when possible. The data logger will record both network controller time (NTP time) and the times the individual devices report.

Often, the clock time of messages is not as important as the sequence in which they are sent. Not all devices have clocks, and many devices that do have clocks only report the time to the nearest minute, or in the best case, second. This is too coarse-grained to properly order messages at the network controller level, and we cannot use the network controller clock to order them because messages may take varying times to travel through the network. This issue of ordering messages in a distributed system is a well-known problem in computer science, and the usual solution is to use logical clocks. Implementations such as Lamport clocks [7] or vector clocks [8] are applicable. We propose using vector clocks, where each device adapter on the network transmits a set of counter values with each transmission. This will allow analysis and playback programs to correctly establish causal ordering between messages even in cases where timestamps are not useful or available. Research results obtained through this analysis will inform an emerging Federal initiative on improving the timestamp accuracy of medical device data in the EHR.

Format of Device Data Devices on the ICE network will transmit data using a standard format. ICE part 1 [2] does not specify this format, leaving its definition to future parts of the standard. It is expected that devices will encode their data using a well known ontology, though it is not necessary for all devices in the system to use the same ontology. Candidates include SNOMED, HL7, and 11073. One function of the playback and analysis software is to assist clinicians in categorizing adverse events. FDA CDRH uses event problem codes and evaluation codes to classify the device problems in associated with an adverse event. These codes are harmonized with ISO TS 19218 and there are plans to integrate these codes into SNOMED and to work with IEEE 11073 to incorporate the codes into these two codes sets to create a global vocabulary to report device problems.

We assume that the data logger playback application and supervisor applications will be able to interpret the ontologies used by connected devices. If this is not the case, the applications will at least be able to notify the user that the device is unsuitable for the application (in the case of the supervisor) or that the playback program cannot handle the data log. The data logger will record raw network traffic even when it can not interpret the contents.

Each data transmission from a device includes the unique device identifier (UDI) as specified by the FDA, a logical timestamp as described above, the data from the device encoded in that device's ontology of choice, and a checksum used to test if the data is corrupted in transmission. Where possible, existing adverse event ontologies will be used, such as the device problem and evaluation codes of the FDA's ★ *MDR* ★ system.

Security and Trustworthiness of the Log When problems arise in systems whose components come from multiple manufacturers, it can be difficult to convince an individual manufacturer to take responsibility. The event recorder log provides a vendor-neutral record of transactions on the network that can be shown as evidence to device manufacturers.

We anticipate that the log from the recorder will be an important legal record as well as a clinical and engineering tool. This means that the data in the record must be trustworthy, and any tampering with the record must be obvious. To address these concerns, we give each log entry an individual sequence number and cryptographic signature in addition to a tagging it with the time the message was received at the network controller. The sequence number makes it obvious if a record is missing from the sequence, and the signature allows verification that the content of the record has not been changed.

Analysis and Replay of Log Data An event log is only useful if it can provide relevant information to users. Turning the raw data in the log into useful information is the job of the replay program. This program should be able to open the data log from the event recorder, check it for consistency by examining the signature of each entry, and provide the user with a set of tools for analyzing the data. The log serves two general purposes: it will support analysis of adverse events involving multiple devices and it will allow system developers to view low-level data for debugging their applications. These purposes require different playback tools and techniques. We call the first use clinical log playback and the second use debugging playback.

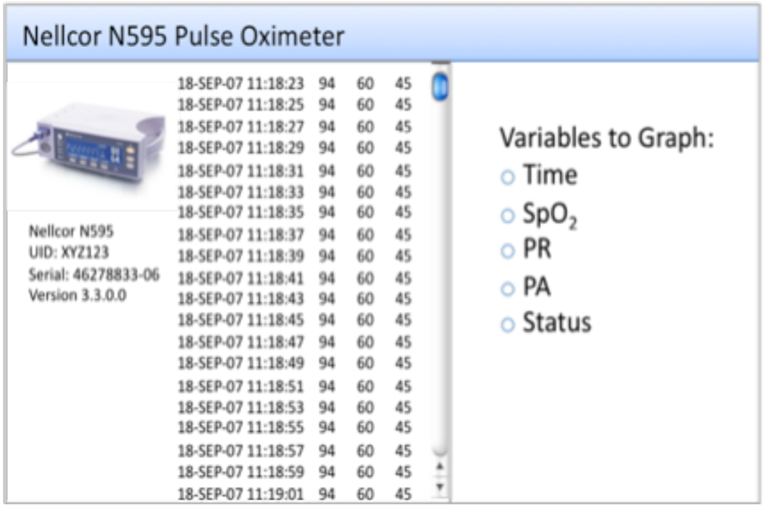
The clinical log playback tool will allow analysts to build an interactive timeline of logged data and events and to link text from clinician interviews in the appropriate places. Location information will be automatically included in the timeline when it is available in the record.

Analysts will need to be able to view the sequential data stream from a specific individual device and the interleaved sequences from multiple devices. In addition to the textual display, the program will be able to build a graphical timeline of data values and events from the devices. Because clinician narratives are an important part of the adverse event analysis process, the playback program will allow analysts to display narrative text beside the logged data, tag sections of the entries with times, and mark entries in the graphical timeline.

A typical session using the tool will have these steps:

1. Copy the log from the network controller to the computer running the playback tool.
2. Open the log in the tool and, for each device of interest, select the variables or items to appear on the graphical timeline. This step is illustrated in Figure 2.

3. Add clinician narratives from interviews. Manually mark the narrative with times given by the interviewed clinician. The tool will support vague times like “between 9 and 9:30 am” as well as descriptions like “between when the alarm went off the first and second time”.
4. The user can view a synchronized timeline of events and produce text or graphical output to help analyze the sequence of events and produce reports.



Nellcor N595 Pulse Oximeter

Nellcor N595
UID: XYZ123
Serial: 46278833-06
Version 3.3.0.0

```
18-SEP-07 11:18:23   94   60   45
18-SEP-07 11:18:25   94   60   45
18-SEP-07 11:18:27   94   60   45
18-SEP-07 11:18:29   94   60   45
18-SEP-07 11:18:31   94   60   45
18-SEP-07 11:18:33   94   60   45
18-SEP-07 11:18:35   94   60   45
18-SEP-07 11:18:37   94   60   45
18-SEP-07 11:18:39   94   60   45
18-SEP-07 11:18:41   94   60   45
18-SEP-07 11:18:43   94   60   45
18-SEP-07 11:18:45   94   60   45
18-SEP-07 11:18:47   94   60   45
18-SEP-07 11:18:49   94   60   45
18-SEP-07 11:18:51   94   60   45
18-SEP-07 11:18:53   94   60   45
18-SEP-07 11:18:55   94   60   45
18-SEP-07 11:18:57   94   60   45
18-SEP-07 11:18:59   94   60   45
18-SEP-07 11:19:01   94   60   45
```

Variables to Graph:

○ Time
○ SpO_2
○ PR
○ PA
○ Status

Fig. 2. Device Data Selection User Interface

Figure 3 shows a mock-up of the clinical data playback user interface. The center of the display shows graphs of the device data that the user has selected and marks on the timeline for chosen events from narrative descriptions. There are two timelines along the bottom of the screen because the device data comes from two devices. The red “shift” button to the right of the timeline will allow the user to move the timelines forward and backward with respect to one another. The user can also let the system align events automatically using the clock and timestamp data in the log, but the manual option will be particularly useful for showing narrative events or events from manually entered or paper records. The user can also play back the data, either in real-time or at increased or reduced speeds.

Debugging playback typically involves too much data to view graphically. If system developers want to see a graphical display, they can use the clinical playback program or graph the data in another application. We expect that system developers will use standard tools like Matlab and protocol analyzer software to examine the files, and we will support this by exporting the data

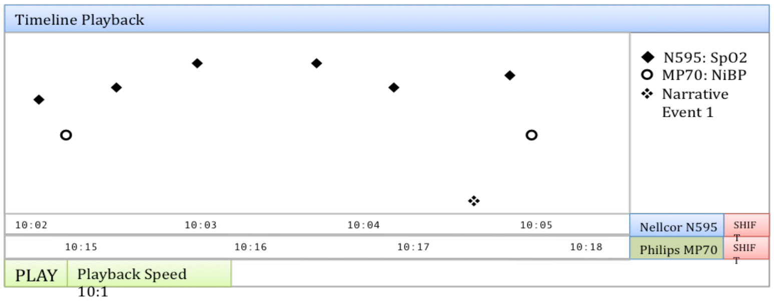


Fig. 3. Mock-up of Clinical Playback User Interface

in appropriate formats. The playback tool for debugging will allow these users to select which data they want and pick an output format. The tool will also support down-sampling the data to reduce the size of the files and the strain on the analysis tools.

3 Future Work and Conclusion

Since no existing medical devices provide data at the resolution we want to support, we will add this capability to the Generic Infusion Pump [1], an open-source infusion pump project supported by the FDA. This prototype device will be useful for requirements gathering and for testing the prototype system. We will connect existing devices in the MD PnP Interoperability Lab [9], including pulse oximeters, two models of Dräger ventilators, and a Philips patient monitor, although these legacy devices will not transmit high-resolution low-level data such as key presses.

Our data playback and visualization applications will be an improvement in current practice regardless of data source and even if the devices are not connected to an ICE network. A comprehensive log might make it possible to have greater contextual information to better understand the sequence of actions involved in an adverse event and hence more accurately and meaningfully report to FDA under the MDR regulations. The visualization and playback application will be useful with current hospital adverse event analysis workflow, although some data would have to be entered manually or converted from the devices proprietary formats.

Time synchronization and management of medical device clocks is becoming widely recognized as a barrier to acquiring accurately time-stamped EMR data for meaningful use and adoption of device data into EMRs. Without accurate timestamps, the information is of limited use for adverse event analysis.

We plan to explore integrating our data logger with the ASTER-D project. This project involves pulling patient history data and other relevant data from the EHR, applying event codes, and automatically transmitting an event report to the FDA. We may also be able to feed data into a Clinical Medical Device Management System.

We will produce general-purpose tools, but it will be useful for our development to focus on some concrete use cases. We will work with our MGH and FDA collaborators to identify appropriate and rich use cases, and our preliminary discussions have already identified unintended intra-operative awareness under anesthesia as an interesting case. This use case involves data from many different devices, hand-written and computer entered case notes, and interviews with clinicians. When we are able to weave this data together into a coherent picture of what happened during a particular case, we will be well on our way to finishing the general purpose tools.

References

1. D. Arney, R. Jetley, P. Jones, I. Lee, and O. Sokolsky. Formal methods based development of a PCA infusion pump reference model: Generic infusion pump (GIP) project. In *Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability (HCMDSS-MDPnP 2007)*, pages 23 – 33, 2007.
2. ASTM F2761-09. ASTM F2761-09, New Specification for Equipment in the Integrated Clinical Environment - Part I: General Requirements for Integration.
3. S. L. Brown, R. A. Bright, and D. R. Tavris. *Medical Device Epidemiology and Surveillance*. Wiley, 2007.
4. FDA Center for Devices and Radiological Health (CDRH). Workshop on medical device interoperability. http://www.mdpnp.org/FDA_Interop_Workshop.php, January 2010.
5. Food and Drug Administration. Medical devices; medical device data systems. http://www.federalregister.gov/articles/2011/02/15/2011-3321/medical-devices-medical-device-data-systems, February 2011.
6. ISO 80001-1:2010. *Application of risk management for IT-networks incorporating medical devices – Part 1: Roles, responsibilities and activities*. ISO, Geneva, Switzerland, 2010.
7. L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558565, 1978.
8. F. Mattern. Virtual time and global states of distributed systems. In *Workshop on Parallel and Distributed Algorithms*, page 215226, Chateau de Bonas, France, 1988. Elsevier.
9. MD PnP. Medical Device Plug and Play Program. http://www.mdpnp.org.