

Plug-and-Play for Medical Devices: Experiences from a Case Study

David Arney, Sebastian Fischmeister, Julian M. Goldman, Insup Lee, Robert Trausmuth

Medical devices are pervasive throughout modern healthcare, but each device works on its own and in isolation. Interoperable medical devices would lead to clear benefits for the care provider and the patient, such as more accurate assessment of the patient's health and safety interlocks that would enable error-resilient systems. The Center for Integration of Medicine & Innovative Technology (www.CIMIT.org) sponsors the Medical Device Plug-and-Play Interoperability program (www.MDPnP.org), which is leading the development and adoption of standards for medical device interoperability. Such interoperable medical devices will lead to increased patient safety and enable new treatment options, and the aim of this project is to show the benefits of interoperable and interconnected medical devices.

The authors envision a future where interoperable medical devices can be seamlessly integrated. This integration could operate through a plug-and-play mechanism and work as follows:

1. The clinician decides on the treatment plan for the patient at hand.
2. An algorithm selects the optimal set of monitoring and treatment devices to execute the plan, considering, for instance, measurement precision, interoperability, and redundancy.
3. The nurse assembles the treatment system.

At the moment, steps two and three are technically not feasible, because out-of-the-box medical devices cannot communicate with each other and are not interoperable. This also severely limits the choices for the first step, because the clinician must take the availability and quality of only single devices into consideration when deciding on the treatment plan.

In this work, we present a prototype of a system supporting seamless integration and thereby enabling steps two and three. The prototype also acts as a research platform for the ASTM standard for the interconnected clinical environment (ICE)¹ based on Ethernet. Medical devices and other components were extended through custom logic on an external dongle and connected

through a standard Ethernet switch. A version of this demo was presented at the American Society of Anesthesiologists annual meeting in 2007, where it won first place in the scientific exhibits, and the system as described here was presented at the Healthcare Information and Management Systems Society (HIMSS) 2008 and the 2008 CIMIT Innovation Congress.

The selected scenario for our work consists of a patient connected to a patient-controlled analgesia (PCA) infusion pump and several patient monitors. PCA infusion pumps are involved in a large number of adverse events, and current safeguards such as drug libraries and programmable limits are not adequate to address all the scenarios seen in clinical practice.⁷ We aim to create a safer PCA pump system by integrating the pump with patient monitors and a supervisor program that can stop the infusion and sound an alarm if the patient experiences an adverse event.

The prototype as presented in the exhibits provides a simple interoperability layer for the medical devices used in the scenario and allows for seamless integration of the supported medical devices. Future iterations of the prototype will be embedded in the ICE system and include more complex interoperability layers supporting, for example, membership functionality, security features, and use other standards such as HL7, the ISO/IEEE 11073 family of standards, or the Integrating the Healthcare Enterprise (IHE) standards on device data.

The Proposed System

Figure 1 shows the components of the PCA demo system. The system is centered around an individual patient and the devices connected to him or her. For this demo, the devices were a PCA infusion pump, two pulse oximeters, and a simulated respiratory rate monitor. The patient was represented by a simulator capable of providing signals to the patient monitors and the supervisor computer ran a demo program that presented selected scenarios and controlled the output of the patient simulator. Each of the pulse oximeters was capable of measuring oxygen

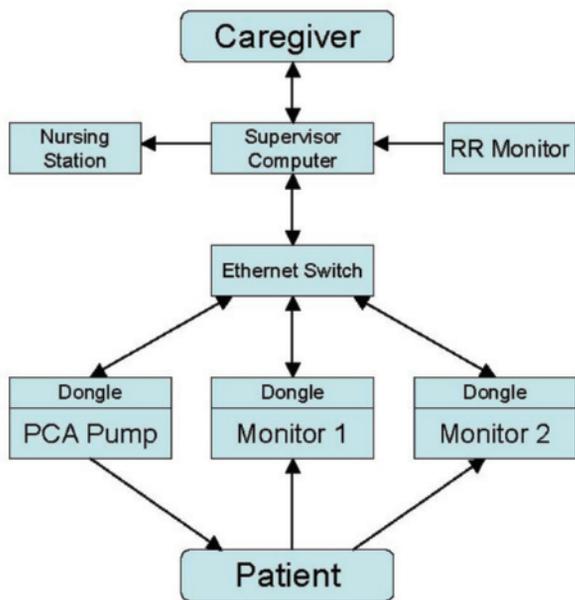


Figure 1. Demo system overview.

saturation (SpO₂) and heart rate (BPM). These measurements, along with a simulated respiratory rate (RR) from the respiratory rate monitor, were used as input for the demo scenarios. The nursing station was represented by a small box that could display a red, yellow, or green light to indicate alarm, warning, or normal condition.

Requirements

In the process of creating the prototype, we also established an initial set of requirements for seamless integration of medical devices. This initial set can act as input for subsequent parts and extensions of the ASTM ICE standard.¹ We refined the requirements during the process of building the prototype and the following summarizes the most important ones for this audience:

Remote Access

All devices connected to the medical network must provide an interface to remotely access the device from the supervisor. This interface can vary depending on the device class, and our requirements included three different classes: (1) treatment devices, (2) monitoring devices, (3) infrastructure-support devices. Treatment devices directly influence the patient’s physiologic parameters. In general this includes all devices that deliver medication, such as infusion pumps, but it also includes mechanical devices such as pacemakers or ventilators. The class of monitoring devices encompasses all passive devices that merely read the patient’s physiologic parameters but have

no significant influence on them. This class includes blood pressure, oxygen saturation, or heart rate monitoring. Finally, the class of infrastructure-support devices comprises all equipment that keeps the infrastructure working. This includes, for example, the supervisor and the data logging facility.

The interface for a treatment device exposes functions for controlling the intensity of the treatment, for turning the device on and off, and for acquiring status information. In the case of our PCA pump, the intensity control consisted of stopping the infusion and sounding an alarm. The interface for monitoring devices only needs to provide status information. Both of our pulse oximeters periodically report measurements via a serial interface. Finally, we have as yet no specific requirements for the infrastructure support devices, because all these components are not yet standardized. Future requirements for the supervisor will focus on the user interface and the connection between the supervisor and the network controller.

Workflow Models

Protocols are prescribed sets of steps to specific actions in patient care. They usually capture the best practices and pathways and play an important role in the daily routine of caregivers. Such work protocols are typically captured in workflow models that describe the sequence of actions taken by the clinical staff when setting up a specific treatment. For example, to set up the closed-loop PCA infusion pump system the clinical staff first must plug in all devices, then acknowledge the devices on the supervisor node, and finally remotely activate the devices. The ICE prototype must provide support for such workflow models, and there are two main aspects to this support. First, it must provide the right human-computer interaction abstractions so the clinical staff can understand workflows and execute them in the correct order. Second, the MD PnP infrastructure must be able to parse workflows and ensure that the workflow is correctly and completely executed. In our prototype, we concentrate on the second requirement.

Timing as First Class Citizen

Medical devices perform time-critical tasks. Some of them require high temporal precision such as measuring pulse transit times or controlling intra-aortic balloon pumps; others only require low precision such as sending a stop signal to the PCA pump in case of an overdose.

Nevertheless, the underlying infrastructure must provide means to execute such time-critical tasks right from the beginning. Otherwise, the infrastructure will become obsolete once it starts to proliferate. The support infrastructure of the medical network must therefore provide means for computing end-to-end guarantees for delays in the system. For example, it must provide all the data necessary to compute how long it will take from the pulse oximeter reporting an alarm condition until the PCA pump stops pumping fluid.

Certification

Some treatment devices perform safety-critical operations since a patient could die as a result of an incorrect treatment due to a malfunctioning treatment device. This was clearly demonstrated by the example of the Therac-25 linear accelerator system.⁶ Treatment devices must therefore be subjected to rigorous verification and validation (V&V) to ensure safety and must be approved by regulatory agencies, such as the U.S. Food and Drug Administration (FDA), before being put on the market. V&V and approval of interconnected devices pose new challenges since there is no longer a single manufacturer of the system and the final configuration can not always be predicted before the individual devices are sold. This situation is different if the manufacturer of the treatment device designed it with the intention of a defined control interface in mind.

PCA PnP Case Study

Interconnected devices enable more complex alarms than standalone devices. To show this, we chose scenarios which would cause false positive and negative alarms in standalone devices and showed how we would detect and correct the false alarms in an interconnected system.

A false positive alarm is one where the system sounds an alarm when no real fault is present. This causes the caregiver to waste resources responding to situations that do not require intervention and, in extreme cases, may cause caregivers to ignore alarms. False negative alarms are conditions where the patient has a problem but the monitor does not detect it and no alarm is sounded. A monitor may also sound the alarm too late.

A false positive may occur if, for example, a patient is resting and his or her heart rate goes below the threshold set on the monitor. We can try to mask this kind of false alarm by requiring that the signal stay low for a period of time before sounding an alarm, but this delays caregiver

response for all real alarms as well. Instead, we need to correlate the heart rate and SpO₂ from the patient monitor with data from other monitors (such as respiratory rate and other data from a capnograph) to get a better picture of the patient's overall status. Rather than triggering an alarm based only on heart rate measured by the pulse oximeter, we can also look at the other data that is available and only alarm if they also show abnormal conditions.

System

The prototype system consisted of the PCA pump, two pulse oximeters, and the supervisor node (see Figure 2). The nurse alarm and the respiratory rate monitor were directly connected to the supervisor. A time-division multiple-access (TDMA) driven Ethernet forms the communication backbone of the medical network.

We selected all system components so they satisfy the requirement for remote controllability. The patient monitors already had a serial output port that they used to publish their status and patient information. The PCA infusion pump was more difficult since there are currently no PCA infusion pumps that allow external control of their settings. We used an experimental pump system based on the Generic Infusion Pump.² We hope in the future to modify an existing pump to enable external control and are discussing this possibility with pump manufacturers.

The system uses time-triggered computation and time-triggered communication. This means that devices execute actions and communicate at predefined points in time. To ensure this, each device connects to a dongle that shields the network from the device and the device from the network.⁴ The dongle acts as the gateway to the medical network, implements the plug-and-play functionality, and handles the communication layer interface. During operation, the medical network layer is completely transparent for the medical devices because they only interact with the dongle. The dongle synchronizes all information across the system at predefined rates and maintains a consistent state.

The following scenario outlines how the system works in operation. The pulse oximeter is a monitoring device and it reports new values every second. The oximeter dongle transmits the latest measurements with a frequency of 1Hz to the supervisor. The supervisor collects information from all monitors and decides the treatment dose. The supervisor dongle communicates the treatment



Figure 2. Case study setup.

dose to all treatment devices at a frequency of 1Hz. The PCA pump is a treatment device and it keeps on delivering treatment only as long as it periodically receives a go signal from the supervisor.

We preallocate communication bandwidth for the individual dongles and use temporal isolation to eliminate collisions on the shared communication medium. Thus, we can precisely calculate the end-to-end delay from the oximeter reporting a hazardous condition until the pump stops delivering medication.

The dongles, together with the supervisor, support workflows, and the ability to implement precisely timed actions allows us to implement time-sensitive workflows as well. Consider a general workflow with the following three steps: (1) enable monitoring devices, (2) wait for the first reading, (3) if ok, start delivering medication. The nurse need not follow this workflow precisely because it is embedded in the dongles. Instead, the nurse simply connects all the devices, selects the right workflow at the supervisor node, and then presses start. The pump dongle will keep the pump turned off until it receives a go signal from the supervisor. The supervisor will not send this go signal before it has received the first reading from the monitor devices and before all readings are fine. And finally, the monitoring devices will not report readings before the nurse presses the start button on the supervisor.

The final step in the system is to understand whether it works as specified. For this, we used software and communication abstractions that facilitate formal verification and thereby create evidence for a potential certification. Formal verification of our system allows us to statically check properties of the systems such as: will the pump dongle ever turn on the pump before it has received a go

signal? An in-depth discussion of the formal verification part is beyond the scope of this column and we point the interested reader to other works.^{5,2}

Examples

The closed-loop integration of monitoring devices with treatment devices allows us to lower the work burden of the clinical staff by reducing the number of false alarms produced by the monitoring devices. In the following we present a concrete case, showing how our prototype prevents a false alarm and can take safety precautions in the presence of a real alarm.

Consider a patient experiencing a drop in the SpO₂ reading. This drop can be the result of a variety of reasons including the common occurrences of the measuring clip falling off or the patient fiddling with the sensor. At this point an isolated system will sound an alarm and notify the clinical staff to check on the patient. If the drop is transient, then this will be a false alarm and the notification will have been unnecessary.

Our prototype system will not sound an alarm resulting from transient drops because it can integrate readings from multiple monitoring devices and has more data available to make the decision. The pulse oximeter transmits the SpO₂ reading to the supervisor. At the same time, the respiratory rate monitor reports its readings. The supervisor can now fuse the data of these two monitoring devices and use a smart algorithm that, for example, will sound an alarm only if the SpO₂ value remains low for a predefined period of time and the respiratory monitor also reports abnormal readings. This will reduce the number of false alarms, assuming that the patient continues breathing while fiddling with the sensor or after the sensor falls off. The algorithm might report a warning to the nurse to check the patient if time permits.

Furthermore, our prototype system can take safety precautions in case an alarm appears. If we assume that the SpO₂ drop is permanent and that the respiratory rate monitor also reports abnormal readings, then the workflow can specify to terminate the treatment and the supervisor will command the pump to stop delivering medication. This simple safety interlock between the monitoring and the treatment devices can safeguard a significant number of patients from a lethal overdose.^{8,3,9}

Conclusion

One major element of the case study is the medical network. Its key property is to provide bounded com-

munication delays for the participating devices. While the theory on bounding communication delays is rich, the practical solutions do not provide acceptable latencies and flexibility when basing communication decisions on the application state. This is required to ensure future tight integration of safety checks into the dongle to protect the patient from transient and permanent hardware and software failures. We built our own communication chip with Ethernet technology that provides the required latencies and flexibility for the medical network. The current version of this chip works with a predefined set of devices and we are working on extending it to support any device that is plug-and-play compatible. Adding complexity to the medical network introduces new hazards, so our design must include mechanisms to mitigate those hazards.

We are also working to support smarter treatment algorithms. In the current implementation, the supervisor shuts off the pump in the presence of a permanent respiratory depression based on a preset threshold. We envision using smart algorithms that integrate data from multiple sensors together with the patient history and statistical data from other patients to calculate the threshold for each individual. The next generation of our case study is aimed at use in a hospital setting where it collects data from real patients. This data will be used to develop better algorithms for detecting when a patient is receiving an overdose.

The case study implementation presented here has served several roles. It is a research platform for developing infrastructure for prototype interoperable systems, a tool for presenting the ideas of medical device plug-and-play, and a means of exploring the clinical patient safety applications enabled by interoperable systems. We have presented results in each of these areas and we expect that our future work will continue to advance infrastructure for medical device interoperability and applications for improving patient safety. ■

References

1. ASTM F2761 2009, New Specification for Equipment in the Integrated Clinical Environment—Part I: General Requirements for Integration.
2. Arney D, Jetley R, Jones P, Lee I, Sokolsky O. Formal methods based development of a PCA infusion pump reference model: Generic Infusion Pump (GIP) project. In HCMDS5-MD-PNP '07: Proceedings of the 2007 Joint Workshop on High Confidence Medical Devices, Software, and Systems and Medical Device Plug-and-Play Interoperability, pages 23–33, Washington, DC, 2007. IEEE Computer Society.
3. Barker SJ, Novak S, Morgan S. The Performance of Three Pulse Oximeters During Low Perfusion in Volunteers. In Annual Meeting of the American Society of Anesthesiologists, chapter A409. ASA, 1997.
4. Fischmeister S, Lee I, Trausmuth R. Hardware Acceleration for Verifiable, Adaptive Real-Time Communication. In Proc. of the 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Hamburg, Germany, 2008.
5. Fischmeister S, Sokolsky O, Lee I. A Verifiable Language for Programming Communication Schedules. *IEEE Transactions on Computers*, 56(11):1505–1519, November 2007.
6. Leveson NG. *Safeware: System Safety and Computers*. Addison-Wesley Professional, New York, NY, 1995.
7. Nuckols TK, Bower AG, Paddock SM, Hilborne LH, Wallace P, Rothschild JM, Griffin A, Fairbanks RJ, Carlson B, Panzer RJ, Brook RH. Programmable infusion pumps in ICUs: An analysis of corresponding adverse drug events. *J General Int Med*. 23(Supplement 1):41–45, January 2008.
8. Shah N, Chitkara A, Miller JM, Taleghani A. Motion Induced Failure of Pulse Oximeters (POs) -Failure Rates and Recovery Times in Human Volunteers. In Annual Meeting of the American Society of Anesthesiologists, chapter A881. ASA, 2005.
9. Shah N, Estanol L. Impact of Motion and Low Perfusion on SpO₂ & Pulse Rate in Three New Generation POs in Volunteers. In Annual Meeting of the American Society of Anesthesiologists, chapter A1433. ASA, 2006.

David Arney and Insup Lee are with the Department of Computer and Information Science at the University of Pennsylvania. Julian M. Goldman is with the Departments of Anesthesia and Biomedical Engineering, Massachusetts General Hospital, and the Center for Integration of Medicine & Innovative Technology. Robert Trausmuth is with the University of Applied Sciences Weiner Neustadt, Austria. Sebastian Fischmeister is with the Department of Electrical and Computer Engineering at the University of Waterloo.