Using machine-to-machine / “Internet of Things” communication to simplify medical device information exchange

Matthias Görges, Guy A Dumont
Electrical & Computer Engineering Department
University of British Columbia, Vancouver, Canada
Email: mgorges@cw.bc.ca, guyd@ece.ubc.ca

Christian L Petersen, J Mark Ansermino
Department of Anesthesiology, Pharmacology and Therapeutics
University of British Columbia, Vancouver, Canada
Email: cpetersen@cw.bc.ca, anserminos@yahoo.ca

***Abstract*—A new push-based mechanism of information exchange between medical devices using the MQ Telemetry Transport (MQTT) protocol for the Internet of Things (IoT) is presented. The proposed implementation allows any medical device to become interoperable without knowledge of the existence of other devices, and provides a simple, scalable solution to the current lack of medical device connectivity. Our approach facilitates the use of shared information in ways that were not envisioned by the original device manufacturer, thus enabling the continued independent improvement of patient safety and information integration throughout the life cycle of devices. We have implemented a VitalBridge interface running on Wifi-enabled Beaglebone Black embedded devices. This allows existing legacy devices to connect to this new communication framework, and use it to implement an integrated mobile monitoring and team communication platform for the intensive care unit.**

***Keywords*—medical devices; interoperability; mq telemetry transport protocol; machine-to-machine communication; publish-subscribe information exchange**

I. Introduction

Despite rapid advances in computer networking and connectivity between devices, little, if any, progress has been made in the medical setting that is of any practical use. Medical devices remain disconnected from each other, and unable to share critical information that could potentially prevent adverse events or allow the use of decision support tools. In this paper we explore the use of the MQTT messaging protocol to enable voluntary information exchange between medical devices using a new push-based strategy.

A. Harm due to preventable medical errors and limitations in monitoring technology

In the complex and stressful environment of the intensive care unit (ICU), about 1% of patients will suffer a serious adverse event every day (120 adverse events in 79 patients in 1490 patient-days) [1], [2]. Such events are more common in the ICU than in any other hospital setting, and as many as 45% are deemed preventable. The variety of medical devices (physiological monitors, ventilators, infusion devices), and the complexity of the technology used, contribute to errors and adverse events [3]. Despite major advances in sensor and device technology, the means to extract important features/data from these electronic devices to alert healthcare workers of possible risks and adverse events are weak. At present, most medical devices use the single-sensor, single-indicator paradigm [4], where each device indicates relevant changes in patient status via its own separate display and alarm; information is not shared among devices. These systems have many limitations and frequent false alarms are nuisance causing rather than helpful in improving the care of patients [5]–[7].

B. Lack of information integration and technology interoperability

Healthcare team members and patients in the intensive care unit, operating rooms, or even general patient wards, would benefit significantly from harnessing new technology to develop an integrated system capable of increasing situational awareness, reducing false alarm rates, and providing early warning of patient deterioration [8]. Data integration using clinical dashboards [9] or single indicators combining multiple variables [10] shows promise for improving patient care. Advances in medical data visualization have led to a reduction in medical errors and provider workload (both physical and mental) as well as improvements in medical decision-making [11], allowing clinicians to detect an adverse event, determine a diagnosis, or make a treatment decision more rapidly [12]. Here, interoperable medical devices would allow more accurate assessment of the patient’s health and offer opportunities for safety interlocks, leading to more error-resilient systems [13].

Aspects of the interoperability of medical devices have been described in the Patient-Centric Integrated Clinical Environment (ASTM F2761-2009) standard [14]. The functional environment described in this standard lends itself well to an IoT publish-subscribe implementation. There are many different protocols available for such machine-to-machine (M2M) communication, all with different strengths and weaknesses.

For example, the Data Distribution Service (DDS) protocol has previously been considered for medical device integration in the Medical Device “Plug-and-Play” Interoperability Program at Massachusetts General Hospital [15]. DDS is a mature protocol developed for military applications, and is certainly a candidate for this type of use. However, the available DDS implementations have commercial license restrictions, and the

M. Görges is holding a Canadian Institutes of Health Research postdoctoral fellowship.

complexity of the protocol makes it unsuited for deployment on most embedded devices.

In contrast, the MQ Telemetry Transport (MQTT) protocol, a simple M2M communication standard developed by International Business Machines Corporation, offers a scalable architecture that can be implemented on limited hardware as well as on unreliable networks [16]. MQTT allows subscribers to indicate interest in topics, and then be notified of updates automatically, preventing inefficient polling schemes. MQTT implementations are available under the liberal Berkeley Software Distribution license, which makes royalty-free commercial usage possible.

Due to the adoption of MQTT by the Eclipse Foundation as the choice for a general purpose open source M2M messaging protocol [17], the popularity of MQTT already exceeds DDS 20 fold in Google Trends, while DDS remains of interest mainly in niche military and similar complex industrial applications. In this paper we therefore consider the use of MQTT for medical device communication. The MQTT protocol is well suited for this purpose as it is scalable, unencumbered by commercial licensing restrictions or requirements, and a good fit for the ASTM interoperability standard [14]. Indeed our approach retraces several elements of the standard.

C. State of the art of data extraction

Currently, most medical devices operate independently and are not connected to a network. The notable exceptions are physiological vital signs monitors in the ICU and operating room (OR), which employ central stations for short term (72 hr) record keeping and alarm escalation. Also infusion pumps may allow the downloading of drug libraries (with dose limits) over wireless networks. Laboratory devices, which tests clinical specimens often make their measurements available to some central location or push/pull data from an electronic medical record (EMR). However, these are beyond the scope of this paper.

Hence, to obtain data from a medical device the end user has to either employ a software development kit provided by the manufacturer and download the data from the central station's database or connect directly with the desired communication device, typically using a serial connection (see Fig.1a). The other alternative is to employ a passive network tap [18] with vendor-specific parsers to retrieve monitoring data from the network. All of these solutions require the user to implement vendor specific communication protocols, and to pull the data from the individual devices.

With the increasing prevalence of EMRs, specialized companies now provide hardware solutions to obtain data from third-party systems. These solutions implement the native protocol of the device, communicating with it via a serial cable, and feeding a limited subset of the obtained data into the EMR using their proprietary communication protocol [19] (see Fig.1b). However, all of the above approaches have a multitude of limitations, including a) difficulty of obtaining data, b) the need for specialized software and/or hardware to communicate with medical devices, c) limitations in data frequency and resolution, especially when it comes to downloading raw waveform data, and d) problems with data synchronization.

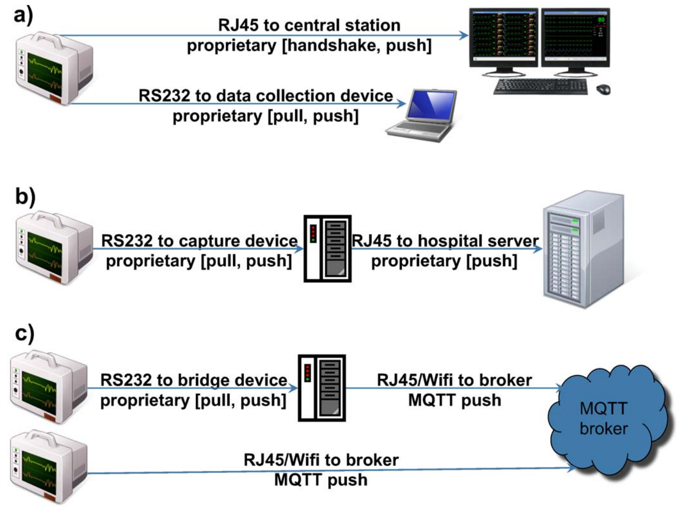

Fig. 1. Data export from medical devices in the past, present, and future. Part a (top) shows the serial based data extraction for research purposes or network connection to a central monitoring station. Part b (middle) shows the use of a data integration device pushing data into the hospital's electronic data warehouse, while Part c (bottom) shows the information flow using MQTT push messages for legacy devices, or future medical that will natively speak MQTT.

D. Previous work, motivation and proposed solution

In order to make better use of collected patient data, the Pediatric Anesthesia Research Team (PART) at BC Children's Hospital in Vancouver, Canada has previously developed a data access node and accompanying wireless real-time display of multi-bed patient data for mobile phones called telePORT. TelePORT operates independently of the commercial patient monitor network [20] to improve information exchange, and simplify communication between anesthesia team members. This solution has been successfully running at BC Children's hospital for the last year [21]. However, to expand its use into the ICU, multiple previously standalone devices, such as infusion pumps and mechanical ventilators, needed to be connected, which prompted a re-evaluation of our connectivity strategies.

In an ideal world, medical data would be easily available to anyone authorized to access it. Hence, a shift from a pull-based data model to a push-based data model is desired. This paper describes an integrated data integration framework and its prototype implementation.

II. Information Exchange Framework

For information exchange we employed Mosquitto [22], an open source message broker that implements the MQ Telemetry Transport protocol (MQTT) [23]. An overview of the communication structure, which is required to support legacy devices, is shown in Fig.1c and Fig.2.

A. Communication devices

Switching from a pull-based messaging system, wherein a data collection device requests data from a medical device, to a push-bashed information exchange framework, wherein the devices voluntarily publish data without prompting, has

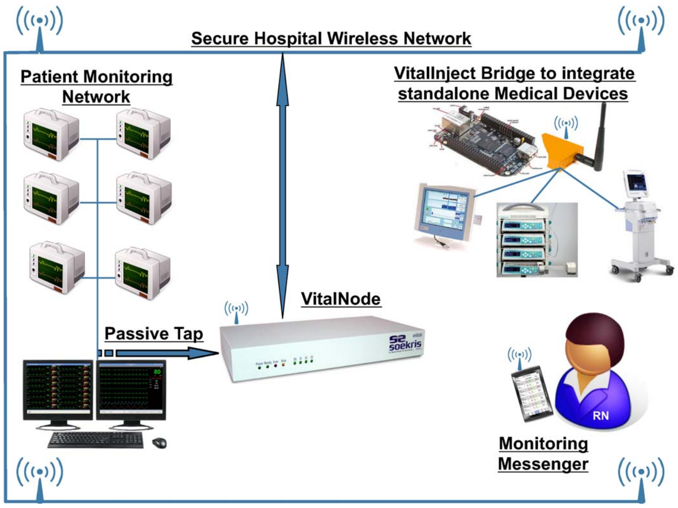

Fig. 2. Information Flow using legacy devices. Data is collected, processed, and integrated by the VitalNode device, by passively tapping existing monitoring networks, and integrating data from standalone devices using a wireless bridge. This data is then made available for display in the Monitoring Messenger mobile application.

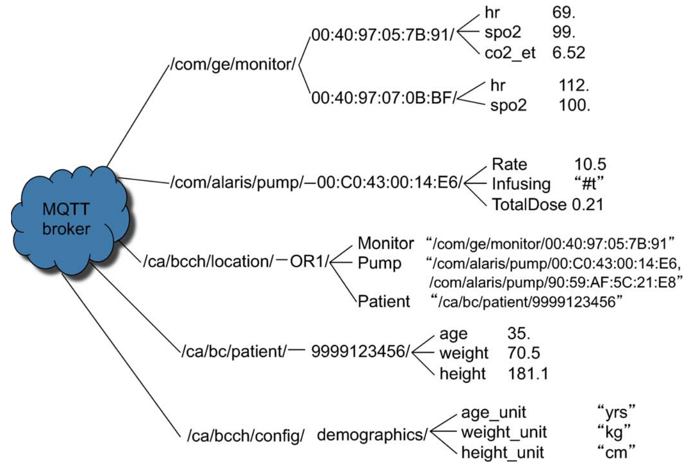

Fig. 3. Namespace example for the ICU Messenger application. This example shows a combination of three elements from the device category, each with some example values, as well as example entries for location, patient, and configuration namespaces.

multiple advantages: It is simple to implement, reduces network traffic, and most importantly allows the use of data not envisioned as useful when the medical device was originally purchased/implemented. An example of this would be to provide clinicians with easy access to high-resolution clinical data to drive quality improvement and improved patient outcome, e.g. for surgical quality improvement [24].

1) Message broker: The VitalNode server runs an MQTT message broker, to which all clients publish their data. It supplies a server certificate signed by a trusted certificate authority, and performs user/password authentication of all clients. Authentication is used to prevent namespace collisions by allowing clients to only publish data in their respective name spaces, and to control access to potentially sensitive data. All communication between the message broker and the client is automatically encrypted by the MQTT protocol using a commonly used and validated OpenSSL implementation.

The VitalNode platform will also allow integration of data obtained from the passive tap, for which monitoring network specific parsers need to be written. It then publishes the information in the vendor's name space following the proposed naming convention (see subsection II-B2).

2) MQTT enabled devices: These devices speak MQTT natively, and only require the hostname of the message broker, as well as namespace specific username and password to be set upon device setup. Devices do not need to be aware of their locations, and are freely movable within the hospital entity, as location to device mapping is performed in a different part of the namespace by its governing organization.

3) Legacy devices: In order to integrate legacy devices into the MQTT-based communication structure, a bridging device VitalBridge is used. This bridging device communicates serially or via Ethernet with the medical device, and publishes all received data to the message broker using a wired or wireless network connection. It requires implementation of a broad number of proprietary monitoring drivers and runs on a small embedded platform, such as a BeagleBone Black (BeagleBoard.org Foundation, Richardson, TX).

B. Namespace proposal

In order to allow easy access to medical device information while avoiding collisions between devices and preserving private name spaces for manufacturers a combination of a hierarchical naming system, similar to the Domain Name System, and the ISO 11073-10101 for device classifications, is proposed. [TLD] as top level domain, and [SLD] as second level domain, uniquely identify each communication providers' personal namespace. At a minimum four main categories of devices, locations, patients, and configurations, are required, while certain applications may add their own private namespace as needed. An example is shown in Fig.3.

1) Patient namespace: This namespace follows the /[TLD]/[SLD]/patient/[patient identifier]/*, e.g. /ca/bc.gov.health/patient/NNNNNNNNNN/*, naming convention and provides access to demographic information, such as date of birth, weight, height, names, and potentially billing information including addresses and phone numbers. The [patient identifier] is a unique identifier, assigned to each patient, by a government entity, such as the Medical Service Plan of British Columbia, and is used to link the patient to devices both as input variables used during setup and to export when new variables are measured.

2) Device namespace: All connected devices have a dedicated physical topic subspace in which they generate data. The proposed namespace convention here would be /[TLD]/[SLD]/[basecategory]/[serial number or MAC address]/*, e.g. /com/ge/monitor/121313/*. The [basecategory] uses one of the ISO 11073-10101 device categories:

- pump (e.g. a syringe pump)
- monitor, (e.g. a physiological vital signs monitor)
- system (e.g. a ventilator)
- generator (e.g. a heater)

- regulator (e.g. a drug infusion controller)
- analyzer (e.g. a blood gas analyzer)
- calculator (e.g. an ideal body weight calculator)
- filter (e.g. a dialysis system)
- meter (e.g. a scale)
- stimulator (e.g. pacemaker)
- interface, a non-standard category (e.g. an information display)

A unique [serial number], if exported by the device, or the devices' media access control address [MAC address] is used to differentiate multiple instances of the same category of device, which are frequently encountered in the hospital setting. Using the MAC address as MQTT's client identifier, for which only one connection is accepted by the broker allows for namespace collision prevention (see also II-C below).

Within this name space the device manufacturer would publish information, such as time stamped measured values, raw data, alarm messages, and its device state.

3) Location namespace: Location topics are used to refer to physical devices as follows: /[TLD]/[SLD]/location/[location identifier]/[base category], e.g. /ca/bccw/location/OR1/pump, which would contain a comma separated list of devices of this category present in a given location, such as /com/alaris/device/pump/123,/com/alaris/device/pump/234. The location governing entity would control the location space and update locations of devices and patients, as they move throughout the hospital.

4) Configuration namespace: Configuration topics are used to assign commonly used settings throughout a location entity. An example for this would be standardized drug concentrations for premixed medications, or age-appropriate default alarm limits for vital signs monitors. This namespace is organized as /[TLD]/[SLD]/config/[config identifier]/* and additional layers of nodes are added by the populating entity as needed.

C. Security, authentication and access control

The standard mosquito MQTT broker includes authentication through a simple username/password mechanism, and can encrypt communication using the industry standard OpenSSL cipher suite. While this prevents unauthorized access to the broker, a finer level of access control is required to prevent accidental or malicious contamination of the name space of each connected device. This can be accomplished by using the unique MAC address of each device as the broker client ID in combination with the access control list (ACL) feature of the mosquito broker. For example, the mosquito ACL entries

```
topic read /+/+/monitor/+/+
pattern write /+/+/monitor/%c/+
```

will allow any connected device read access to the device monitor namespace, while restricting write access to the device itself, as %c expands to the unique client ID. In addition, central to device integration is the ability for intelligent agents connected to the broker to set parameters on the individual devices. This can also be managed through ACL, by allowing access based on user names and implementing a writable set sublevel to the device namespace. For example,

```
user pumpcontrol
topic write /+/+/pump/+/set/+
```

will provide an agent with the pumpcontrol user name the ability to set the infusion rate of pump devices. This scheme can be combined with the location namespace to further limit control to devices in a particular location.

D. Device synchronization and system time

A crucial element to successful device integration is to ensure that timekeeping is consistent on all connected devices. Maintaining the same system time on all MQTT clients simplifies merging of data sources and allows accurate forensic logging. For this purpose we propose that the broker platform include a Network Time Protocol (NTP) time server, optionally synchronized against an external pool of official time servers.

Additionally, the broker platform time is pushed in ISO 8601 time format to the broker under a universal topic "/system/time". Connecting client devices are encouraged to run a local NTP client to ensure minimal overhead and clock jitter. However, some devices may not have the ability to run an NTP client, for example if the MQTT client is implemented on a microcontroller or similar minimal embedded system. These devices must then pull the time from the broker directly to maintain synchronicity.

III. USE CASE AND IMPLEMENTATION EXAMPLE

The use case described here is for the ICU Monitoring Messenger. This application aims to address information integration challenges using a mobile device (tablet computer or smartphone) which allows health care team members to continuously monitor critically ill patients on or off-site. In order for the mobile application to display the patient's historical vital signs trends alongside currently administered medications (see Fig.4), it requires information from several devices including the patient monitor, mechanical ventilator, and infusion pumps.

A. Hardware implementation

For this project we combined our existing passive network tap technology [18], which obtains data from the Intellivue monitoring network (Philips Healthcare, Andover, MA) connected to the VitalNode, an embedded Linux server (Soekris Engineering Inc., Scotts Valley, CA), and two VitalInject boxes for each patient, one for the G5 mechanical ventilator (Hamilton Medical Inc., Reno, NV), and one for the Alaris Medley infusion pump (CareFusion Corporation, San Diego, CA). All of these devices feed their information to the MQTT broker, also running on the same physical Vitalnode box.

For the mobile client we used Google Nexus 7 devices (Google Inc., Mountain View, CA), or similarly sized tablet computers, which fit well into a typical scrubs' pocket. All devices use the hospital's secure wireless network infrastructure, or wired network, where available, to connect to the MQTT broker.

Fig. 4. Overview screen of telePORT [21] and medication vital signs interaction screen prototype for the ICU Monitoring Messenger [25]. The left subplot shows an overview of all operating rooms the user subscribed to, each with an anesthetic phase indicator (room being empty of procedure stage), three vital sign values (mean arterial blood pressure, heart rate, and end-tidal carbon dioxide concentration) colored in the same fashion as displayed on the patient monitor and corresponding trends, as well as the number of pages, reminders, or system generated alerts received during the last 30min. The right subplot shows a more detailed trend screen for one patient, in which vital sign trends (heart rate in the top plot and blood pressure in the middle plot) are presented alongside medication infusion rates (bottom plot). This allows the healthcare provider to explore potential causality for the patients' decrease in heart rate and blood pressure, with the change in the bold lined drug infusion rate.

B. Software implementation

Software for all components other than the MQTT broker, is developed using LambdaNative [20], a cross-platform, open source development environment written in Scheme, which is available at http://www.lambdanative.org/.

The VitalBridge client software is a simple application, which instantiates all currently implemented patient monitor and pump drivers and automatically detects the medical device to which it is currently connected. After a successful handshake with the device, it generates the MQTT namespace using the manufacturer's domain and its own MAC address, after which each updated data store variable, received from the medical device, is published to the MQTT broker.

The VitalNode device starts a passive packet-capture, and parses the observed network traffic for known monitor communication protocols. If these are encountered, it generates the MQTT namespace similarly to the VitalBridge server, however, using the MAC addresses of each of the sending monitors instead of its own address. Additional information, such as system-generated alerts or reminders, as well as health care team communication about this specific patient, is published in a private patient specific namespace by the VitalNode server.

The ICU Monitoring Messenger client subscribes to selected medical device information topics from the MQTT broker by combining information from the patient's namespace, such as physical location, and the location namespace, such as which infusion pumps and which monitors are located in that location. Voice over IP calls are negotiated and performed directly between the devices.

C. Challenges encountered

The biggest challenge is the need to implement complicated vendor-specific medical device drivers, for each new device to be added to the system. This requires access to a physical device for testing as well as the companies' proprietary protocol specification, for which a non-disclosure agreement may or may not be required. This challenge would be overcome if medical device manufacturers were to adopt support for a push-based data model, ideally using MQTT, in order to comply with the increased interoperability requirements [26].

Additionally, there are challenges with the hospital's wireless network, including obtaining access to the network, and support for seamless handoff between wireless access points, which requires support and buy-in from the hospital's IT department. Solutions for this problem remain currently unknown. However, it is clear that with the ever-increasing number of health informatics opportunities, the barriers for entry will decrease as key stakeholders start to acknowledge the importance of decision support systems in order to improve patient safety.

Finally, there is need for additional research into using integrated data to optimize patient outcome and safety, while presenting it to health care providers in the most intuitive and relevant way for a given context or situation. However, without this data being available the actual implementation challenges are moot. Hence simple, open, and secure access to medical device data, like the one proposed in this paper need to be implemented first.

IV. Conclusion

The recent advances in the infrastructure of the Internet of Things and the wide acceptance of device-device communication standards such as the MQTT protocol, offer a proven scalable framework on which to build safe medical interoperability solutions that can be disseminated broadly to the health care community.

The proposed messaging scheme is simple, feasible to implement, and provides enterprise wide data availability through a switch from poll-based data models to a push-based data model free of commercial license restrictions.

This open-source push-based approach for the first time enables community-driven development and implementation of new patient safety initiatives, and facilitates integration of devices independently of manufacturer commitment. This solves the age-old problem of improvements in medical device integration being contingent on the collective agreement between manufacturers, something that continues to be elusive and extremely difficult to obtain.

We plan to continue work in this field, and to demonstrate how practical closed-loop hospital systems can be successfully implemented using this technology.

References

[1] J. M. Rothschild, C. P. Landrigan, J. W. Cronin, R. Kaushal, S. W. Lockley, E. Burdick, P. H. Stone, C. M. Lilly, J. T. Katz, C. A. Czeisler, and D. W. Bates, The Critical Care Safety Study: The incidence and nature of adverse events and serious medical errors in intensive care., *Crit Care Med*, vol. 33, no. 8, pp. 1694–700, Aug. 2005.

[2] Y. Donchin, D. Gopher, M. Olin, Y. Badihi, M. Biesky, C. L. Sprung, R. Pizov, and S. Cotev, A look into the nature and causes of human errors in the intensive care unit., *Crit Care Med*, vol. 23, no. 2, pp. 294–300, Feb. 1995.

[3] ECRI Institute, PHYSIOLOGIC MONITORING SYSTEMS, *Health Devices*, vol. 10, no. October, pp. 310–340, 2013.

[4] L. Goodstein, Human Detection and Diagnosis of System Failures, in *Human detection and diagnosis of system failures.*, 1st ed., J. Rasmussen and W. B. Rouse, Eds. Boston, MA: Springer, 1981, pp. 433–449.

[5] E. M. Koski, A. Mkivirta, T. Sukuvaara, and A. Kari, Frequency and reliability of alarms in the monitoring of cardiac postoperative patients, *Int J Clin Monit Comput*, vol. 7, no. 2, pp. 129–33, Apr. 1990.

[6] M. Imhoff and S. Kuhls, Alarm algorithms in critical care monitoring., *Anesth Analg*, vol. 102, no. 5, pp. 1525–37, May 2006.

[7] M. Borowski, M. Görges, R. Fried, O. Such, C. Wrede, and M. Imhoff, Medical device alarms, *Biomed Tech (Berl)*, vol. 56, no. 2, pp. 73–83, Mar. 2011.

[8] A. Garland, Improving the ICU: part 2., *Chest*, vol. 127, no. 6, pp. 2165–79, Jun. 2005.

[9] M. Egan, Clinical dashboards: impact on workflow, care quality, and patient safety., *Crit Care Nurs Q*, vol. 29, no. 4, pp. 354–61, 2006.

[10] L. Tarassenko, A. Hann, and D. Young, Integrated monitoring and analysis for early warning of patient deterioration., *Br J Anaesth*, vol. 97, no. 1, pp. 64–8, Jul. 2006.

[11] M. Görges and N. Staggers, Evaluations of physiological monitoring displays: a systematic review., *J Clin Monit Comput*, vol. 22, no. 1, pp. 45–66, Feb. 2008.

[12] M. Tentori and J. Favela, Activity-aware computing in mobile collaborative working environments, in *Proceedings of the 13th international conference on Groupware: design implementation, and use*, 2007, pp. 337–353.

[13] D. Arney, S. Fischmeister, J. M. Goldman, I. Lee, and R. Trausmuth, Plug-and-play for medical devices: experiences from a case study., *Biomed Instrum Technol*, vol. 43, no. 4, pp. 313–7, 2009.

[14] ASTM Committee F-29.21, 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, 2009.

[15] J. M. Goldman, S. Whitehead, D. Arney, R. Schrenker, P. Mattegunta, J. Plourde, D. Alonso, and A. Lenco, Medical Device 'Plug-and-Play' Interoperability Program, *MD PnP Program*, 2014. [Online]. Available: http://www.mdpnp.org/. [Accessed: 03-Apr-2014].

[16] A. Stanford-Clark and A. Nipper, MQ Telemetry Transport, *MQ Telemetry Transport*, 2013. [Online]. Available: http://mqtt.org/. [Accessed: 03-Apr-2014].

[17] N. O'Leary, Paho - Open Source messaging for M2M, *Eclipse Paho's MQTT*, 2014. [Online]. Available: http://www.eclipse.org/paho/. [Accessed: 03-Apr-2014].

[18] M. Görges, C. Petersen, and J. M. Ansermino, Capturing vital signs for research in a multi-bed monitoring environment, in *Proceedings of the 2011 Society for Technology in Anesthesia Annual Meeting*, 2011, p. A32.

[19] B. McAlpine and D. Vankampen, Clinical engineering and information technology: working together to implement device integration., *Biomed Instrum Technol*, vol. 45, no. 6, pp. 445–9, 2011.

[20] C. L. Petersen, M. Görges, D. Dunsmuir, M. Ansermino, and G. A. Dumont, Experience report: functional programming of mHealth applications, *Proc 18th ACM SIGPLAN Int Conf Funct Program - ICFP '13*, pp. 357–62, 2013.

[21] M. Görges, T. Coelho, P. Thomas, C. L. Petersen, G. A. Dumont, and J. M. Ansermino, The Portable Operating Room Tracker (telePORT) - Analysis of system usage and identification of areas for improvement, in *Proceedings of the 2014 Society of Technology in Anesthesia Annual Meeting*, 2014, p. A32.

[22] R. Light, Mosquitto - An Open Source MQTT v3.1/v3.1.1 Broker, 2014. [Online]. Available: http://mosquitto.org/. [Accessed: 19-Mar-2014].

[23] D. Locke, MQ Telemetry Transport (MQTT) V3.1 Protocol Specification, Armonk, NY, 2010.

[24] M. Görges, J. M. Ansermino, and S. D. Whyte, Using routinely collected physiologic vital signs for surgical and anesthesiology quality improvement projects, in *Proceedings of the 2014 Society of Technology in Anesthesia Annual Meeting*, 2014, p. A23.

[25] M. Görges, P. P. Morita, C. M. Burns, and J. M. Ansermino, Mobile Patient Monitoring for the Pediatric Intensive Care Unit: Work Domain Analysis and Rapid Prototyping Results, *2013 IEEE Int Conf Syst Man, Cybern*, pp. 3765–70, 2013.

[26] Committee on Data Standards for Patient Safety, Patient Safety: Achieving a New Standard for Care. Washington, DC: National Academy of Sciences, 2004, p. 550.