31st Annual International Conference of the IEEE EMBS
Minneapolis, Minnesota, USA, September 2-6, 2009

A Platform for Testing and Comparing of Real-Time Decision-Support Algorithms in Mobile Environments

Maxim Y. Khitrov, Matthew Rutishauser, Kevin Montgomery, Andrew T. Reisner, and Jaques Reifman

Abstract—**The unavailability of a flexible system for real-time testing of decision-support algorithms in a pre-hospital clinical setting has limited their use. In this study, we describe a plug-and-play platform for real-time testing of decision-support algorithms during the transport of trauma casualties en route to a hospital. The platform integrates a standard-of-care vital-signs monitor, which collects numeric and waveform physiologic time-series data, with a rugged ultramobile personal computer. The computer time-stamps and stores data received from the monitor, and performs analysis on the collected data in real-time. Prior to field deployment, we assessed the performance of each component of the platform by using an emulator to simulate a number of possible fault scenarios that could be encountered in the field. Initial testing with the emulator allowed us to identify and fix software inconsistencies and showed that the platform can support a quick development cycle for real-time decision-support algorithms.**

I. Introduction

The U.S. military has long been interested in advanced decision-support capabilities for combat casualty care, in which an automated computer algorithm processes available vital-signs data, offers field medics early and accurate information about the state of the casualty, identifies the need for life-saving interventions and immediate evacuation, and advises on appropriate therapeutic actions [1]. While progress has been made, the technology does not yet exist as a reliable, established capability. Only a handful of decision-support algorithms have been tailored for early assessment of trauma casualties in a chaotic pre-hospital environment, where vital-signs data are of intermittent availability and of varying quality [2]. And, to date, these algorithms have only been evaluated retrospectively [2], [3]. Before such capabilities are widely disseminated, algorithms must be optimized, and their effectiveness prospectively demonstrated, as medics are unlikely to embrace decision-support algorithms that make frequent mistakes or that only make obvious recommendations. This requires the availability of a testing platform that integrates real-time vital-signs data collection with immediate analysis.

Here we describe a novel plug-and-play platform for real-time data collection and prospective testing of decision-support algorithms in pre-hospital, clinical setting. The proposed system, which offers the ability to quickly (e.g., within minutes) insert a new decision-support algorithm into actual clinical operations, offers major benefits. First, algorithm developers can receive rapid feedback based on real-world tests and make multiple iterative improvements, as the proposed system promotes iterative design cycles with minimal expense and time. Second, the system can simultaneously compare competing analysis methods, promoting good technologic approaches and identifying inferior ones, so that future development resources are invested appropriately. Ultimately, the system illuminates the capabilities and limitations of standard vital signs for decision support and offers a benchmark against which future monitors and sensors should be compared.

The proposed platform is based on Welch Allyn's (Skaneateles Falls, NY) Propaq Encore 206-EL monitor [4] and Roper Mobile Technology's (Tempe, AZ) ruggedized Switchback personal computer (PC) [5]. The Propaq is mounted on top of a small cage, which houses the PC and acts like a pedestal (Fig. 1). The PC is interfaced to the Propaq, allowing it to archive physiologic data collected by the Propaq and run investigational algorithms implemented in MATLAB [6]. This system is intended for deployment on Boston Medflight (civilian) air ambulances en route to the Massachusetts General Hospital (MGH). Initially, caregivers will not be shown results of the algorithms, so the testing will not alter clinical care.

II. Methods

A. Requirements

The process of collecting and analyzing time-series physiologic data in real-time imposes strict requirements and

Manuscript received Apr. 02, 2009. This work was supported by the Combat Casualty Care and the U.S. Army Medical Department, Advanced Medical Technology Initiative funded by the Telemedicine and Advanced Technology Research Center (TATRC) of the U.S. Army Medical Research and Materiel Command (USAMRMC), Fort Detrick, Maryland. The investigators adhered to the policies for protection of human subjects as prescribed in Army Regulation 70-25, and the research was conducted in adherence with the provisions of 45 CFR Part 46.

M. Y. Khitrov is with the Bioinformatics Cell, TATRC, USAMRMC, Frederick, MD 21702 USA (e-mail: mkhitrov@bioanalysis.org).

M. Rutishauser is with Intelesense Technologies, Milpitas, CA 95035 USA (e-mail: matt@intelesense.net).

K. Montgomery is with Intelesense Technologies, Milpitas, CA 95035 USA (e-mail: kevin@intelesense.net).

A. T. Reisner is with the Bioinformatics Cell, TATRC, USAMRMC, Frederick, MD 21702, and with the Massachusetts General Hospital Department of Emergency Medicine, Boston, MA 02114 USA (e-mail: areisner@partners.org).

J. Reifman is a Senior Research Scientist and Director of the Bioinformatics Cell, TATRC, USAMRMC, ATTN: MCMR-TT, Bldg. 363 Miller Drive, Fort Detrick, MD 21702 USA (corresponding author; phone: 301-619-7915; fax: 301-619-1983; e-mail: jaques.reifman@us.army.mil).

Fig. 1. Assembled prototype of the proposed system (under development in collaboration with Intelesense Technologies, Milpitas, CA). Propaq 206 vital-signs monitor “A” is mounted on top of the protective cage “B,” which contains the ruggedized Switchback PC.

Fig. 2. Protective cage detached from the Propaq. LEFT: Bottom view of the cage, showing the touch screen of the ruggedized Switchback PC. RIGHT: Propaq 206 vital-signs monitor.

limitations on hardware and software. Hardware components must have a small footprint to fit inside limited helicopter space and be light enough for direct attachment to the Propaq. We identified the bottom of the Propaq to be the only available location for attachment of a pedestal-like cage to house and protect the ruggedized PC (Fig. 1). The challenge in this design was to account for the inevitable stresses of real-world use, such as withstanding a drop of several feet with the weight of Propaq on top and being exposed to heat, cold, rain, mud, and other elements. Furthermore, adequate ventilation was needed to cool the PC components inside without exposing them to environmental hazards. This was achieved by creating openings in the cage as far away from the ground as possible.

We identified four primary requirements for the software components in our platform. 1) The software must reliably capture and store data as they are streamed in real-time via a serial interface from the Propaq to the PC. Random disconnects, corrupt data packets, and transfer delays must all be handled gracefully. Under no circumstances can the software interfere with the normal function of the monitoring equipment. 2) The captured data should be made available for analysis as quickly as possible in a format that accurately represents wall-clock time, and properly accounts for missing data caused by communication faults. It is also essential for all physiologic measurements to be accurately synchronized on a common timeline, regardless of when each sensor is first connected to the patient. 3) As the primary goal of this platform is to test and compare various physiologic data analysis methods, it must provide facilities for quickly adding, removing, and modifying algorithms before sending the system back into the field. 4) The need to modify existing algorithms for compatibility with the system and the effort to iteratively enhance previously tested algorithms should be minimal. Accordingly, the software environment for data analysis should support a large library of mathematical functions and allow for quick prototyping.

B. System Design

Fig. 2 illustrates the proposed platform consisting of two main components: a Propaq Encore 206-EL monitor with the Acuity Port option and a ruggedized Switchback PC running Microsoft’s Windows XP operating system with 2GB of memory and a 32GB solid state drive. We connected Propaq to the PC through an RS-232 to Universal Serial Bus (USB) adapter, and used a USB flash drive to transfer data logging software, analysis algorithms, backups of captured data, and analysis results to and from the Switchback. This design allows for quick and easy update of all software systems in the platform, thus supporting requirement 3.

Fig. 3 shows the three main software modules residing in the PC: Controller, Shell, and Analysis. The Controller application opens the serial port, establishes connection with the Propaq, and records received data into log files [7]. Each log file contains data for one monitoring session, i.e., one patient. Propaq uses packets to transmit information at predetermined rates. Waveform packets, transmitted every 88ms, contain the electrocardiogram (ECG), photoplethysmograph, carbon dioxide, and respiratory waveforms. Numeric packets, sent once per second, contain

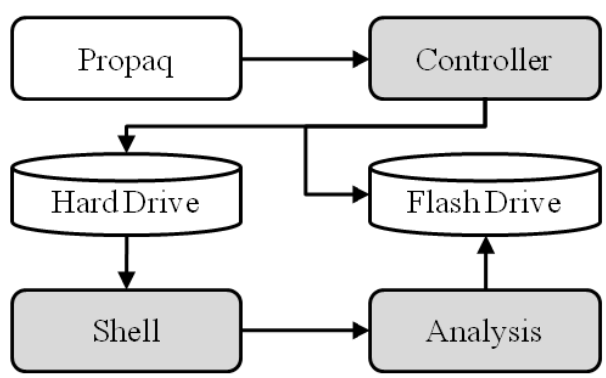

Fig. 3. Overview of the data path implemented in our software. Data from Propaq are received by the Controller, mirrored to the hard drive and Universal Serial Bus (USB) flash drive, read from the hard drive by the Shell, and provided for analysis. Analysis results are written back to the USB flash drive.

Propaq-calculated heart rate (HR), blood pressure, temperature, oxygen saturation, and respiratory rate. Log files record the contents of each packet and the time when that packet is received (requirement 1). By time stamping the packets as they arrive, we can later replay any previously monitored session in simulated real-time conditions.

When a new session is started, Controller launches the Shell application. Shell reads the current log file and passes vital-signs data for processing to the Analysis module. Because many of our existing algorithms for data qualification and decision support were written in MATLAB and because MATLAB supports requirement 4, we used it as the foundation for the Analysis module. Shell communicates via the MATLAB Application Programming Interface (API) to make newly received data available to all analysis algorithms in the form of constant-frequency vectors for each monitored variable. To maintain the frequency and common timeline for every variable, Shell inserts the appropriate number of NaNs, i.e., "Not a Number," which are special floating-point values used to represent lost data, into the vector to replace information from packets that were corrupt, missing, or temporarily delayed (requirement 2). This abstraction eliminates most complications associated with real-time data representation.

We also implemented a mechanism to separate received vital-signs data from Propaq into per-patient sessions. Because the Propaq may be left on without a patient attached, and between patients, the Controller's ability to establish a connection was not sufficient to identify the start of a new patient session. A new session is started when the Controller receives a numeric packet containing HR values within a pre-defined valid range, such as 10-300 beats per minute. This prompts the creation of a new data log file and the start of the Shell and the Analysis modules. From here on, Shell reads the data log as it is appended by the Controller, passes new data to MATLAB, and triggers execution of the analysis software. This process is executed every five seconds or longer, depending on whether the last iteration has been completed, giving each algorithm the ability to run and produce a result. The end condition for a session is triggered when no valid HR is received for the previous five minutes. This may be a combination of HR values being outside of the valid range for five minutes, which happens when the sensors are detached, or no packets arriving from Propaq, possibly indicating that the monitor was turned off.

III. Results

To test the platform prior to field use, we developed a Propaq emulator that simulates physiologic data by using mathematical functions or by reading plain-text files for each time-series variable. The emulator transmits data to the PC with 100- or 200-series Propaq communication protocols, thus testing the implementation of the entire data path from the Controller to the Shell to the Analysis module. Most importantly, we can use the emulator to introduce multiple types of errors in the transmitted data stream in a deterministic fashion, allowing us to test the reliability of data capture and the accuracy of data representation according to requirements 1 and 2. A key element of the testing is to assure that the time series sent by the emulator is the exact time series that is made available to the Analysis module. Although Propaq is also able to simulate certain physiologic parameters through its in-service operation mode, this was insufficient for testing because of the inability to alter generated data and simulate failure scenarios.

To test the first requirement of reliable data capture and correct handling of unexpected interruptions, we programmed the emulator to periodically corrupt transmitted packets, ignore received commands, and disconnect the serial link. Controller and Shell have several mechanisms for dealing with such eventualities. Corruption of packets was detected by using two-cyclic redundancy check (CRC16) bytes appended to the end of every 200-series packet, or one checksum byte appended to a 100-series packet. Depending on the level of corruption, it may still be possible to extract valid data from the packet, but the general way to handle such cases was to discard the packet and wait for the next one.

Short breaks in communication, lasting at most five seconds, were detected by the use of sequence numbers transmitted along with every numeric and waveform packet. Sequence numbers are incremented by one for each following packet and wrap around to zero after 255. Therefore, if the difference between sequence numbers of two consecutive packets was greater than one, Controller and Shell knew exactly how many packets were lost in the transfer. Shell used this information to fill-in the appropriate number of NaNs in the MATLAB vectors, thus satisfying the second requirement for accurate data representation. Long communication breaks, which require the connection between Controller and Propaq to be reestablished, reset sequence numbers to zero making them unusable in this case. To decide how much data were lost over a long communication break, we used known packet rate information to make an estimate. With an average rate of 88ms for waveform packets and 16 ECG data points per packet, Shell would estimate that 3,200 ECG data points were lost if the break lasted for 17.6 seconds. The same principle would apply to all other variables, taking into account their individual frequencies.

To further test the second requirement of accurate data representation and alignment, we had the emulator skip the transmission of one or more numeric and waveform packets, simulating their loss. This is similar to the corrupt packet scenario described previously, where Shell used either the sequence numbers or the duration of the break, depending on the specific test case, to replace lost data with NaNs. We also simulated the case where some of the Propaq-recorded variables were missing in a packet, representing the situation where sensors were attached to a patient at different times with some delay in between [8]. To accurately align the start time of each variable once the first data point was received, Shell computed the number of NaNs that need to be inserted at the beginning of the vector by dividing session duration up to that point by the variable's sampling rate. This ensured that current and future data points would be correctly aligned with all other variables.

Finally, we assessed the performance of the Analysis module by testing a set of analysis algorithms that will be field tested in the first deployment of the platform. These included algorithms for estimating the reliability of HR computed from ECG waveforms and for extracting ECG features, such as beat-to-beat intervals, approximate entropy, and sample entropy, to determine heart rate variability (HRV). By comparing the input data with what was made available by Shell to the algorithms, we assured the correct functioning of data transmission throughout the various components of the platform. By duplicating the algorithms' results with those obtained off-line, we guaranteed that the Analysis module was performing as intended. A similar procedure will be used in the future, when real-time analysis results are available, to fine-tune algorithms and re-run them in simulation mode on the same input data.

IV. Discussions

Unlike any prior system, the proposed plug-and-play platform provides an efficient means to prospectively test novel decision-support algorithms in real-time during actual clinical operations, allowing for a rapid development cycle. We are in the process of obtaining Institutional Review Board (IRB) approval for use of the platform on Boston Medflight air ambulances en route to MGH. The vital-signs data and analysis results will be recorded and stored in the Switchback PC for off-line review and comparison with patient history and outcome. This will allow us to assess the performance of the implemented platform in a clinical setting and, more importantly, to evaluate the value of different decision-support algorithms. Because the algorithms will be running in the background, there will be no new IRB permission necessary when a different investigational algorithm is plugged into the system for testing.

Initially, we will test algorithms developed by our group that estimate the reliability of vital-signs data by providing a quality index for every HR [9], respiratory rate [10], and blood pressure record [11]. We will also assess the performance of our ensemble decision-support classifier, developed to predict trauma patients with major hemorrhage [2]. As these algorithms were developed for off-line analysis, they are currently being modified for real-time applications. Moreover, we will also test newer algorithms that, from their onset, were developed for real-time use, such as extraction of ECG waveform features and calculation of HRV.

The platform is currently restricted to use with Propaq 206 and MATLAB. Future development plans include adding support for other medical monitors and an API for analysis programs written in C, C++, and Python. Use of compiled MATLAB code is also under consideration. Following the initial deployment phase, a graphical user interface will be added to the platform to make analysis results available to caregivers in real time. They, in turn, will use this interface to input additional information to the analysis routines, such as the type and time of performed life-saving interventions (e.g. resuscitation fluids, intubation, and cardiopulmonary resuscitation). This information should further improve the performance of the decision-support algorithms.

Disclaimer

The opinions and assertions contained herein are the private views of the authors and are not to be construed as official or as reflecting the views of the U.S. Army or of the U.S. Department of Defense. This paper has been approved for public release with unlimited distribution.

References

[1] R. W. Hoyt, J. Reifman, T. S. Coster, and M. J. Buller, "Combat medical informatics: present and future," in *Proc. AMIA Symp.*, San Antonio, TX, 2002, pp. 335-339.

[2] L. Chen, T. M. McKenna, A. T. Reisner, A. Gribok, and J. Reifman, "Decision tool for the early diagnosis of trauma patient hypovolemia," *J. Biomed Inform.*, vol. 41, pp. 469-478, Jun. 2008.

[3] C. A. Rickards, K. L. Ryan, W. H. Cooke, S. A. Romero, and V. A. Convertino, "Combat stress or hemorrhage? Evidence for a decision-assist algorithm for remote triage," *Aviat. Space Environ Med.*, vol. 79, pp. 670-676, Jul. 2008.

[4] "Propaq Encore Monitors," Internet: www.welchallyn.com/products/en-us/x-11-ac-100-0000000001101.htm [Mar. 26, 2009].

[5] "Switchback," Internet: www.ropermobile.com/products/switchback [Mar. 26, 2009].

[6] "The MathWorks," Internet: www.mathworks.com [Mar. 26, 2009].

[7] A. Gandsas, K. Montgomery, D. McKenas, R. Altrudi, and Y. Silva, "In-flight continuous vital signs telemetry via the Internet," *Aviation, Space and Environmental Medicine*, vol. 71, pp. 68-71, Jan. 2000.

[8] L. Chen, T. McKenna, A. Gribok, and J. Reifman, "LAM: A landscape matching algorithm for respiratory data alignment," in *Proc. SPIE*, Orlando, FL, 2006, pp. 62351B:1-10.

[9] C. Yu, Z. Liu, T. McKenna, A. T. Reisner, and J. Reifman, "A method for automatic identification of reliable heart rates calculated from ECG and PPG waveforms," *J. Am. Med. Inform. Assoc.*, vol. 13, pp. 309-320, May-Jun. 2006.

[10] L. Chen, T. McKenna, A. Reisner, and J. Reifman, "Algorithms to qualify respiratory data collected during the transport of trauma patients," *Physiol. Meas.*, vol. 27, pp. 797-816, Sep. 2006.

[11] A. T. Reisner, L. Chen, T. M. McKenna, and J. Reifman, "Automatically-computed prehospital severity scores are equivalent to scores based on medic documentation," *The Journal of TRAUMA Injury, Infection and Critical Care*, vol. 65, pp. 915-923, Oct. 2008.