The 28th International Conference on Distributed Computing Systems Workshops

Cyber-Physical Medical and Medication Systems*

Albert M. K. Cheng
Real-Time Systems Laboratory
Department of Computer Science
University of Houston
TX 77204-3010, USA
cheng@cs.uh.edu

Abstract

Medical and medication devices are real-time systems with safety and timing requirements. They range from hard-real-time, embedded, and reactive systems such as pacemakers to soft-real-time, stand-alone medication dispensers. Many of these devices are already connected to computer networks, especially in hospital intensive-care units, so that patients' conditions detected by sensors can be monitored in real-time at remote computer stations nearby or at other sites. However, remote adjustment of medical devices' output and actuation is typically not allowed due to safety concerns. This article discusses a number of issues such as verification that must be resolved in order to allow cyber-physical operation of medical devices. In particular, we propose using formal methods, self-stabilization, and (m,k)-firm scheduling to allow the safe cyber-physical operation of a medical ventilator, a life-critical reactive device to move breathable air into and out of the lungs of a patient with respiratory difficulties, with the ultimate goal of speeding-up the recovery of the patient.

1. Introduction

Medical and medication devices (MMDs) are increasingly controlled by computer systems with hardware and software components. Many of these devices are real-time systems with safety and timing requirements. They range from hard-real-time, embedded, and reactive systems such as pacemakers to soft-real-time, stand-alone medication dispensers [17]. To facilitate monitoring of these devices and patients, these devices are increasingly connected to computer networks so that their functions and patient conditions can be remotely displayed at nearby workstations and/or physician offices at other sites. However, remote operations in addition to just monitoring remain absent due to safety concerns. Figure 1 shows an embedded real-time system [12] with the decision and control component D and environment being monitored and controlled T.

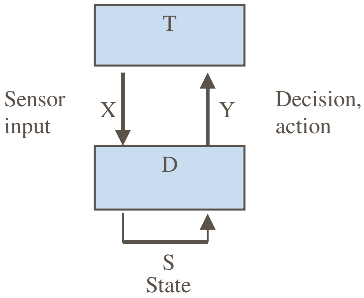

Figure 1. An embedded real-time monitoring/control system.

Figure 2 shows in more detail the components of an embedded real-time monitoring and control system node attached to a network for remote control and communication. This network allows cyber-physical operation of this control node via a remote user interface.

This article discusses a number of issues in developing cyber-physical medical and medication systems (CPMMDs) that allow remote monitoring as well as function actuations. To accomplish cyber-physical operations, rigorous and potentially novel real-time scheduling and verification technologies must be applied to guarantee their safety and performance. Our guiding principle is that we introduce real-time systems technology into MMDs only if this technology improves the safety and performance of existing devices.

There are many ways to apply real-time systems technology to MMDs. Here, we focus on using the

---

*This work is Supported in part by the National Science Foundation under Award No. 0720856, a 2006 grant from the Institute for Space Systems Operations, and GEAR Grant No. I092831-38963.

1545-0678/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDCS.Workshops.2008.67

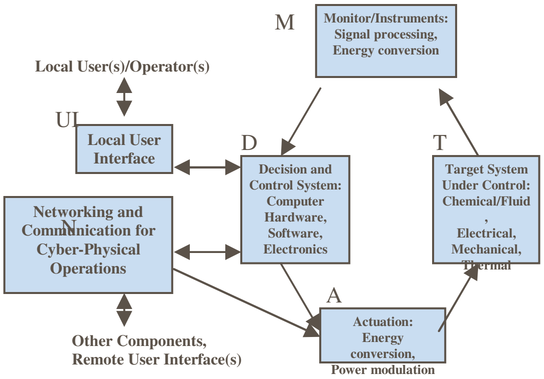

Figure 2. Components of an embedded real-time monitoring and control system node connected to a network.

(m,k)-firm constraints as a guide to increase the Quality-of-Service (QoS) and stability of MMDs ranging from those with hard real-time constraints to soft real-time constraints. In particular, we use an adaptive ventilator system as a motivating example to illustrate the approaches for making it cyber-physical and for improving its performance which translate into speedier recovery for the patient. One goal is to allow a clinician (doctor or respiratory therapist) who cannot be physically with the patient all the time to remotely vary the ventilator assist rate by adapting it to the respiratory capability of the patient.

The patient's breathing capability is measured by stable performance over an interval of time. One way to model this is to employ (m,k)-firm constraints, where the deadlines of m jobs must be satisfied in a sequence of k jobs, with m <= k. For our ventilator application, k represents the length of the above interval expressed as the number of total breathes by the patient and ventilator assist, and m is the number of breathes by the ventilator. Here, m can vary from 0 (no ventilator assist) to k (full ventilator assist). As the patient improves his/her breathing capability, m becomes smaller, and thus the ventilator assist becomes lower. Therefore, we have a sequence of (m,k)-firm constraints where m is a varying parameter over time.

Another goal is to ensure fault-tolerant cyber-physical operation of the ventilator system. The ventilator itself is an already hard-real-time system with built-in backups for all safety-critical components such as the power supply and the oxygen tank. Typically, a backup processor is ready to run all critical tasks in the event of the primary processor failing. If the backup processor has lower performance than the primary processor, it would still meet the deadlines of all critical tasks such as the ventilator assists but it may not be able to satisfy the deadlines of non-critical tasks, such as refreshing the digital displays of the system's parameters and settings. Employing (m,k)-firm scheduling would allow us to ensure a higher QoS derived from the non-critical tasks. However, allowing cyber-physical operation of the ventilator leads to more possible points for delays or failures. We propose two ways to make a cyber-physical ventilator as dependable as a physical one via formal methods and self-stabilization.

The remainder of this article is organized as follows. In section 2, we describe a life-critical medical device with an embedded real-time system, a medical ventilator. Then in section 3, we show how (m,k)-firm scheduling has the potential to enhance a state-of-the-art ventilator into an adaptive system. Formal methods [24-27] and self-stabilization [28] are two ways to allow safe cyber-physical operation of the ventilator system, making it possible for the doctor or respiratory therapist to control the ventilator remotely and thus more frequently adjust the ventilator settings as to improve the recovery of the patient.

The purpose of this article is not to provide results of the application of real-time scheduling technologies to medical systems, but to outline areas in which there is potential for safety and performance improvements some of which are being carried out by the author's research group.

2. Motivating System: Medical Ventilator

A medical ventilator [19,20] is a device for mechanically move breathable air into and out of the lungs of a patient with respiratory difficulties, such as the inability to breathe on his own or breathing insufficiently caused by one or more ailments. The ventilator enables the exchange of oxygen and carbon dioxide, a function performed by normally functioning lungs. The main components of a ventilator include air and oxygen supply tanks, a compressible air reservoir, a variety of valves and tubes, and a disposable or reusable patient circuit.

The ventilator pneumatically compresses the air reservoir several times per minute to deliver normal room air or a mixture of air and oxygen to the patient. The percentage of the oxygen in the inspired gas can be set to 21 (normal air) to 100 (pure oxygen). As a result of the lungs' elasticity, the patient passively exhales when overpressure is released, releasing the air via the one-way valve contained in the patient circuit.

The ventilator is attached to the patient via an endotracheal tube connected through the mouth to the trachea (larynx intubation) or a tracheotomy canula (a more comfortable and practical approach requiring minor surgery if connected for more than two weeks). An embedded system monitors and controls a currentgeneration ventilator in order to dynamically adjust air flow and pressure appropriate to the real-time needs of the patient. This makes the ventilator more comfortable to and tolerable by the patient.

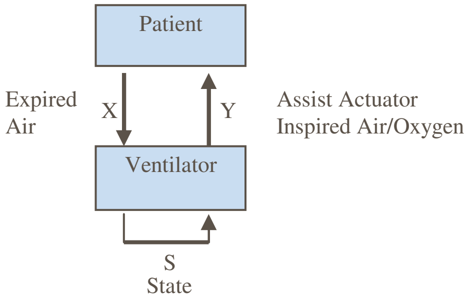

Figure 3. High-level diagram of ventilator control system.

The ventilator monitors: (1) patient-related parameters such as air flow, pressure, and volume from sensors attached to the patient and to the connection between the ventilator and the patient as described above; (2) ventilator functions such as power failure, air leakage, pneumatic unit, and mechanical problems; (3) backup power supply; and (4) oxygen tanks.

A clinician (doctor, respiratory therapist, or nurse) can physically set the ventilator to provide full support where the patient does not initiate breaths, or to provide partial support in which case the patient's inspiratory efforts trigger some or all the breaths delivered by the ventilator. Our objective in this work is to make it possible for the safe cyber-physical operation of the ventilator so that the clinician does not have be physically present with the patient. This would allow more frequent and thus more attentive operation of the ventilator. This would also allow the efficient remote operation of a number of ventilators, especially important when some of the ventilators are located in rural clinics or homes.

In the assist-control mode of ventilation, the clinician manually sets a minimum respiratory rate and volume of gas (or inspiratory pressure) for the patient. After initiation, patient inspiratory efforts can trigger the ventilator at a higher rate as a result of pressure or flow triggering. The ventilator rate is usually set at between 8 and 12 breaths per minute (BPM) initially for optimal assist-control rate support.

The normal BPM for a healthy person is 12 [18]. The rate is of course higher if the person is performing physical exercises. Therefore, it is desirable to maintain a combined BPM of 12 from the ventilator-assist A and the patient's natural breathing B: A + B = 12 BPM. Actually, the operator '+' above does not mean a simple addition but rather a combination. The relationship between A and B is more complex and modeled by differential equations [21], but for this article it suffices to use this equation to illustrate our approach. Star and bus network topologies can be used to connect ventilators to a remote operator. Other options include wireless.

3. Cyber-Physical Ventilator System

In current practice, the clinician manually adjusts the assist BPM and other output parameters on a periodic basis (usually once a day or in a fixed interval of several hours). Once this assist rate is set, the ventilator attempts to adapt its pressure and flow characteristics of the air delivered to the patient while keeping this preset assist BPM. However, the ventilator does not dynamically adjust the assist BPM even if the patient's breathing capability becomes stronger as a result of medication and other factors. In this case, there is an overuse of the assist provided by the ventilator. The patient has to wait until the clinician revisits in the next period in order for the assist rate to be adjusted. Therefore, it is desirable for the clinician to adjust this assist rate cyber-physically, which can be done more frequently without having the clinician to travel to the patient's site.

The objective is to make the ventilator's assist rate adaptive to the breathing performance of the patient in while maintaining a combined BPM of 12. If the breathing capability of the patient is improving (even slightly), this more frequent cyber-physical adjustment of the ventilator performance would potentially shorten the assistive period, allowing the patient to have a speedier recovery. It is important to note that the patient's breathing capability should not be measured by transient performance but by stable performance over an interval of time. We believe that this can be readily modeled by the (m,k)-firm model with k being the length of this interval expressed as the number of total breathes by the patient and ventilator assist, and m being the number of breathes by the ventilator. Here, m can vary from 0 (no ventilator assist) to k (full ventilator assist). As the patient improves his/her breathing capability, m becomes smaller, and thus the ventilator assist becomes lower. Therefore, we have a sequence of (m,k)-firm constraints where m is a varying parameter over time.

As a hard-real-time system, the embedded computer has sufficient processing power and memory to handle worst-case demands of the ventilator monitoring and control tasks. As a result, it seems that there is no need for overload management using (m,k)-firm scheduling or other approaches. However, to ensure even higher fault tolerance in the unlikely event that some processing components fail and thus the remaining and functioning components may be overloaded with the same task set, it is desirable to apply (m,k)-firm scheduling and our proposed QoS maximization technique [16]. In most ventilator systems, a backup processor is ready to run all critical tasks in the event of the failure of the primary processor. If the backup processor has lower performance (for instance, a slower CPU speed) than the primary processor, it still guarantees the satisfaction of the deadlines of all critical tasks such as the ventilator assists but it may not be able to meet the deadlines of non-critical tasks, such as those for refreshing the digital displays of the system's parameters and settings. Employing (m,k)-firm scheduling would allow us to ensure a higher QoS derived from the non-critical tasks and to achieve greater performance stability. We next describe two approaches to make cyber-physical medical systems such as the adaptive ventilator safe for actual operation.

Even if the embedded control system of the ventilator is not overloaded, the control system's operation may benefit from the application of (m,k)-firm scheduling. Currently, the clinician utilizes heuristics to adjust the ventilator based on the patient's condition [22]. However, the relationship between the patient's perceived medical condition and the appropriate ventilator settings is nonlinear and very complex, with parameter and state variations over time. To automate this control procedure, conventional methods such as proportional-integral-derivative (PID) control is often inadequate. Researchers have proposed using rule-based expert systems and fuzzy logic control algorithms to manage this control process [22,23].

We are investigating whether dynamically varying the assist rate can be modeled by varying the parameter m in the (m,k) constraint, treating k as the window or interval length. In this model, $m = 0$ means there is no ventilator assist, and $m = k$ means there is full ventilator support. Furthermore, we are studying whether patterns can be used to provide better control of the air flow and pressure settings in real-time.

Our work in applying formal methods and self-stabilization to medical and medication systems is very preliminary; however, we believe that these technologies hold great promise in making cyber-physical operation of these medical systems safe, thus benefiting patients and expediting their recovery.

References

[1] M. Hamdaoui and P. Ramanathan. A dynamic priority assignment technique for streams with (m,k)-firm deadlines. IEEE Transactions on Computers, 44: 1443-1451, Dec 1995.

[2] G. Koren and D. Shasha. Skip-over: Algorithms and complexity for overloaded systems that allow skips. RTSS 1995, pages 110-117.

[3] R. West and C. Poellabauer. Analysis of a window-constrained scheduler for real-time and best-effort packet streams. IEEE RTSS 2000, pages 239-248.

[4] P. Ramanathan. Overload management in real-time control applications using (m,k)-firm guarantee. IEEE Transactions on Parallel and Distributed Systems, 10(6):549-559, Jun 1999.

[5] D. B. Seto, J.P Lehoczky, L. Sha, K.G. Shin. On task schedulability in real-time control systems. RTSS 96, pages 13-21.

[6] D. B. Seto, J. P. Lehoczky, L. Sha. Task period selection and schedulability in real-time systems. IEEE RTSS 98, pages 188-198.

[7] Q. Quan, X. Hu. Enhanced fixed-priority scheduling with (m,k)-firm guarantee. IEEE RTSS 2000, pages 79-88.

[8] J. W. S. Liu. Real-time systems. Prentice Hall, 2000

[9] J. P. Lehoczky, L. Sha, and Y. Ding. The rate-monotonic scheduling algorithm: Exact characterization and average case behavior. IEEE RTSS 1989, pages 166-171.

[10] G. Buttazzo, M. Spuri, F. Sensini. Value vs. deadline scheduling in overload conditions. RTSS 1995, pages 90-99.

[11] M. R. Garey and D. S. Johnson. Computer and intractability: a guide to the theory of NP-Completeness. Freeman, New York, 1979.

[12] A. M. K. Cheng. Real-time systems: scheduling, analysis and verification. Wiley-Interscience, 2002. 2nd printing, 2005.

[13] S. K. Baruah and J. R. Haritsa. Scheduling for Overload in Real-Time Systems. IEEE Trans. Computers, 46: 1034-1039, 1997.

[14] S. Baruah, J. Haritsa, and N. Sharma. On line scheduling to maximize task completions. RTSS 1994, pages 228-237.

[15] S. A. Brandt. Performance analysis of dynamic soft real-time systems. IEEE IPCCC 2001, pages 379-386.

[16] J. Lin and A. M. K. Cheng. Maximizing Guaranteed QoS in (m,k)-firm Real-time Systems, Proc. 12th IEEE-CS International Conf. on Embedded and Real-Time Computing Systems and Applications, Sydney, Australia, Aug. 2006.

[17] P.-H. Tsai et al. Compliance Enforcement of Temporal and Dosage Constraints, Proc. IEEE RTSS 2006.

[18] D. E. Larson, EIC, Mayo Clinic Family Health Book, William Morrow & Company, New York, 1990.

[19] F. Lemaire. Mechanical Ventilation, Translation of "La ventilation artficielle," 2nd ed. Paris: Masson, Editeur, 1991.

[20] D. R. Hess and R. M. Kacmarek, Essentials of Mechanical Ventilation. New York: McGraw-Hill. 1996.

[21] T. Fernandot, J. Packert and J. Cade. On-line Estimation of Patient and Ventilator Respiratory Work, IEEE-EMBC 1995.

[22] D. S. Nelson, J. H. Strickland, and T. C. Jannett. Simulation of Fuzzy Control for Management of Mechanical Ventilation Respiratory Rate in Assist-Control, IEEE/EMBS 1997.

[23] R. Felber. Automation of Ventilator Control for Hyperbaric Oxygen Therapy, IEEE 2004.

[24] S. Andrei and A. M. K. Cheng. Faster Verification of RTL-Specified Systems via Decomposition and Constraint Extension, RTSS, Rio de Janeiro, Brazil, December 2006.

[25] S. Andrei, W.-N. Chin, A. M. K. Cheng, and M. Lupu. Automatic Debugging of Real-Time Systems Based on Incremental Satisfiability Counting, IEEE Transactions on Computers, Vol. 55, No. 7, pp. 830-843, July 2006.

[26] S. Andrei and A. M. K. Cheng. Optimization of Real-Time Systems Timing Specifications, Proc. 12th IEEE-CS International Conf. on Embedded and Real-Time Computing Systems and Applications, Sydney, Australia, Aug. 2006.

[27] A. M. K. Cheng. A Survey of Formal Verification Methods and Tools for Embedded and Real-Time Systems, Journal of Embedded Systems, Issue 1, 2006.

[28] A. M. K. Cheng. Self-Stabilizing Real-Time Rule-Based Systems, Proc. 11th IEEE Symp. on Reliable Distributed Systems, pages 172-179, Houston, Texas, Oct. 1992.