University of Pennsylvania
ScholarlyCommons

Departmental Papers (CIS) | Department of Computer & Information Science

11-11-2010

GSA: A Framework for Rapid Prototyping of Smart Alarm Systems

Andrew King
University of Pennsylvania, kingand@seas.upenn.edu

Alex Roederer
University of Pennsylvania, roederer@seas.upenn.edu

David Arney
University of Pennsylvania, arney@seas.upenn.edu

Sanjian Chen
University of Pennsylvania, sanjian@seas.upenn.edu

Margaret Fortino-Mullen
University of Pennsylvania, Margaret.Fortino-Mullen@uphs.upenn.edu

See next page for additional authors

Proceedings of the 1st ACM International Health Informatics Symposium (IHI '10), Arlington, Virginia, USA, November 11-12, 2010
http://doi.acm.org/10.1145/1882992.1883069

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/cis_papers/468
For more information, please contact repository@pobox.upenn.edu.

GSA: A Framework for Rapid Prototyping of Smart Alarm Systems

Abstract
We describe the Generic Smart Alarm, an architectural framework for the development of decision support modules for a variety of clinical applications. The need to quickly process patient vital signs and detect patient health events arises in many clinical scenarios, from clinical decision support to tele-health systems to home-care applications. The events detected during monitoring can be used as caregiver alarms, as triggers for further downstream processing or logging, or as discrete inputs to decision support systems or physiological closed-loop applications.

We believe that all of these scenarios are similar, and share a common framework of design. In attempting to solve a particular instance of the problem, that of device alarm fatigue due to numerous false alarms, we devised a modular system based around this framework. This modular design allows us to easily customize the framework to address the specific needs of the various applications, and at the same time enables us to perform checking of consistency of the system.

In the paper we discuss potential specific clinical applications of a generic smart alarm framework, present the proposed architecture of such a framework, and motivate the benefits of a generic framework for the development of new smart alarm or clinical decision support systems.

Keywords
design, verification, health informatics, smart alarm, clinical decision support, model driven development, system synthesis, middleware, medical systems, PRECISE_paper, PRECISE_CPS_Medical

Comments
Proceedings of the 1st ACM International Health Informatics Symposium (IHI '10),
Arlington, Virginia, USA, November 11-12, 2010
http://doi.acm.org/10.1145/1882992.1883069

Author(s)
Andrew King, Alex Roederer, David Arney, Sanjian Chen, Margaret Fortino-Mullen, Ana Giannareas, C. William Hanson III, Vanessa Kern, Nicholas Stevens, Jonathan Tannen, Adrian Viesca Trevino, Soojin Park, Oleg Sokolsky, and Insup Lee

This conference paper is available at ScholarlyCommons: http://repository.upenn.edu/cis_papers/468

GSA: A Framework for Rapid Prototyping of Smart Alarm Systems

Andrew King, Alex Roederer, David Arney, Sanjian Chen, Margaret Fortino-Mullen[*], Ana Giannareas[†], C. William Hanson III[‡], Vanessa Kern, Nicholas Stevens, Jonathan Tannen, Adrian Viesca Trevino, Soojin Park[§], Oleg Sokolsky, Insup Lee
University of Pennsylvania
{kingand,roederer,arney,sanjian,anarg}@seas.upenn.edu
{vkern,nicholjs,tannen,aviesca,sokolsky,lee}@seas.upenn.edu
{Margaret.Fortino-Mullen,Bill.Hanson,Soojin.Park}@uphs.upenn.edu

ABSTRACT

We describe the Generic Smart Alarm, an architectural framework for the development of decision support modules for a variety of clinical applications. The need to quickly process patient vital signs and detect patient health events arises in many clinical scenarios, from clinical decision support to tele-health systems to home-care applications. The events detected during monitoring can be used as caregiver alarms, as triggers for further downstream processing or logging, or as discrete inputs to decision support systems or physiological closed-loop applications.

We believe that all of these scenarios are similar, and share a common framework of design. In attempting to solve a particular instance of the problem, that of device alarm fatigue due to numerous false alarms, we devised a modular system based around this framework. This modular design allows us to easily customize the framework to address the specific needs of the various applications, and at the same time enables us to perform checking of consistency of the system.

In this paper we discuss potential specific clinical applications of a generic smart alarm framework, present the proposed architecture of such a framework, and motivate the benefits of a generic framework for the development of new smart alarm or clinical decision support systems.

1 Introduction

Within the modern hospital room it is common practice to continually monitor patient vital signs with a variety of medical devices attached to the patient. These medical devices can alert medical professionals to changes in the patient state. Many medical devices can be configured with threshold alarms: alarms that activate when a certain vital sign crosses a predefined threshold. These alarms can be vital for the timely detection of reversible emergency states [6, 13]. Unfortunately, current devices with threshold based alarms are limited in the following ways:

1. The alarm is only raised when the predefined threshold is crossed.
2. Each vital sign monitor is independent of other devices and oblivious to information from those devices.
3. The monitors typically do not leverage patient context information or general medical knowledge.

These shortcomings result in many false positive alarms [22] (which fatigue caretakers [14]) as well as fail to provide the medical professionals with more useful and interesting information about the current state of the patient [5, 18]. A reduction in the number of false alarms has been shown to improve patient safety [14]. It is easy to imagine that more capable alarm and clinical decision support systems (CDSS) could be built by combining physiologic data streams from different medical devices as well as other sources (such as the patient's EHR.) Indeed, over the last several years, companies such as CareFX [8], Cerner [4], GE [9] and Phillips [24] have built information technology products that attempt to provide a more cohesive view of a patient's state. Typically these systems collect patient vitals from different discrete devices and either record them in an electronic health record or push them to a centralized monitor (such as a large computer monitor in the patient's room or a command center staffed by medical professionals). A 'smart' alarm or CDSS takes this concept further; the real-time medical information streaming from the different devices would be combined with information from the patient's health record in order to automatically suppress irrelevant alarms, predict trends in the patient's status, and possibly even intelligently coordinate the actions of medical devices (closed loop control). While the potential benefits of these types of systems seem promising, there are a number of challenges the research community will face. Sittig et-al [23] has identified 10 of these 'Grand challenges' of CDSS. We have chosen to focus on two of the aformentioned 10 challenges: 1) How do we disseminate best practices in CDSS design, development, and implementation? and 2) How do we create an architecture for sharing executable CDSS modules and services? In

[*]Author's affiliation: University of Pennsylvania Health System
[†]Ana Giannareas, Vanessa Kern, Nicholas Stevens and Adrian Viesca Trevino designed the CABG smart alarm prototype and investigated its efficacy.
[‡]CMIO (Chief Medical Information Officer), University of Pennsylvania Health System
[§]Author's affiliation: Department of Neurology, University of Pennsylvania School of Medicine

previous work [2] authors Arney, Sokolsky, and Lee worked with the FDA to define a framework (now known as the Generic Infusion Pump) to aid researchers and industry in understanding how to build a safe infusion pump and is now used as an example by the FDA Infusion Pump Initiative [7]. We propose that a Generic Smart Alarm (GSA), similar in spirit to the GIP, could provide a valuable framework for researchers to understand the issues concerning the design, implementation, and testing of smart alarms and Clinical Decision Support Systems. In this paper we:

1. describe the various applications of 'smart alarm' and medical decision support systems we have investigated;
2. summarize the various approaches we have explored to realize these systems and the challenges faced in doing so; and
3. describe the Generic Smart Alarm architecture that we believe will facilitate the experimentation and research necessary to ultimately apply smart alarms and CDSS effectively in the clinical setting

2 Specific Applications

2.1 False Alarm Suppression for CABG Patients

Post-operatice patients after artery bypass graft (CABG) surgery are at particular risk of physiologic instability [20]. Continuous monitoring of a combination of common vital signs in the intensive care unit allows for detection of physiologic changes so practitioners may intervene in a timely manner and prevent complications. In many cases, however, individual thresh-hold alarms can produce a large number of false positives. For example, the fingerclip sensor of a pulse-oximeter can fall off the patient or react to the artificial lighting of the care environment to produce erroneous readings. These sorts of basic failures can be mitigated by a system which is able to monitor multiple vital signs and distinguish data artifacts (such as disconnected leads which lead to sharp vital sign drops) from true patient distress.

To this end, we have implemented a rule-based system which combines four major vital signs commonly monitored in the ICU: heart rate (HR), blood pressure (BP), blood oxygen saturation (SpO_2), and respiratory rate (RR). To do so, we interviewed ICU nurses to determine ranges for classifying each vital sign as a member of some collection of fuzzy sets (classifying, for example, a blood pressure between 50 and 60 as a mix of 'Low' and 'Normal', between 60 and 100 as 'Normal', and between 100 and 107 as being a mix of 'Normal' and 'High'. See Figure 1). Afterward, nurses determined rules that identified combinations of these vital sign statuses which would be cause for concern. An example fragment of the rule table is shown in Table 1. The 'smart' alarm worked by monitoring a patient's four vitals, classifying those into fuzzy sets, and searching the rule table for the corresponding alarm level to output. Combining vital signs in this way produced a very large reduction (57.13 percent over 1,451 hours of recorded CABG patient data) in the number of false alarms generated without suppressing any true alarms. Classification of vital signs using fuzzy sets helped to overcome the difficulty of establishing a ruleset customized to each patient's baseline vital signs. Fuzzy set classifiers can easily be modified to address a specific patient with, for example, a very low 'Normal' resting heart rate, without rewriting the entire rule set.

2.2 Decision Support for Neurocritical Care

For many treatable ominous conditions in the neurocritical care unit (seizure, stroke, etc), time to diagnosis is critical. The patient's

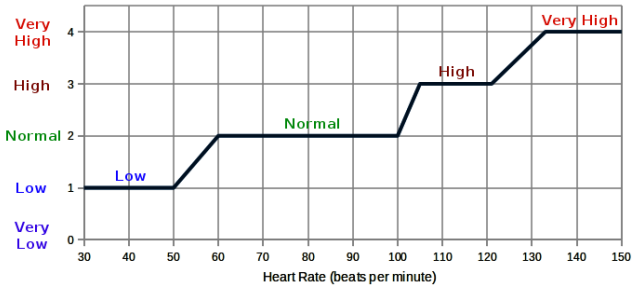

Figure 1: Sample fuzzy set classification for Heart Rate

BP	HR	SPO_2	RR	Alarm Level
Normal	Normal	Normal	Normal	0
High	Normal	Normal	Low	1
High	Low	Normal	Normal	2
Very Low	Normal	Normal	High	3
High	High	Low	High	2

Table 1: Small subset of the rule set.

outcome depends on the ability and experience of the caregiver to recognize patterns in nonspecific changes of multiple sensors, as well as recognize long-term trends in patient status by summarizing changes over multiple days of data. The utility of simple threshold-based alarms to guide this sort of pattern recognition is limited [21]. There are more illustrative interpretations of physiologic data that require computational aid for human comprehension. The incorporation of these tools into clinical care has not been conclusively demonstrated.

Additionally, despite the lack of definitive evidence for many of newer monitor-guided therapies (brain oxygen, microdialysis, etc), the actionable nature of their associated threshold-based alarms are enticing to neurointensivists and traumatologists. They lead to additional tests and empiric treatment of the many potential causes of changes in monitor values - in a temporizing manner - until (if and when) a definitive etiology is discovered. Each additional test and therapy is associated with some cost and/or potential risk of harm to the patient.

To this end, we recognized the possibility of utilizing multiple, low cost/low risk patient monitors to reduce the abundance of unnecessary alarms, thereby reducing reactionary temporizing treatments. Additionally, there exists the opportunity for integration of statistical methods and machine learning techniques to detect long term trends in available patient data. These trends are often difficult for medical experts to detect due to the quantity of data available from disparate sources and the short amount of time available to process it before it becomes irrelevant. Integrating and processing multiple data streams would enable the discovery of significant features of the data that would improve prognostic and diagnostic ability.

2.3 Closed Loop Control of PCA Pumps

A Patient Controlled Analgesia (PCA) pump is a device designed to provide a prescribed bolus dose of pain medication to a patient when the patient presses a button. PCA pumps are associated with a large number of adverse events [12, 15]. The most common type of adverse event is oversedation [19]. An excessive dose of the analgesic can cause neurologic depression which may lead to respiratory depression and eventually respiratory distress. In extreme cases the patient may not be able to breathe adequately, leading to

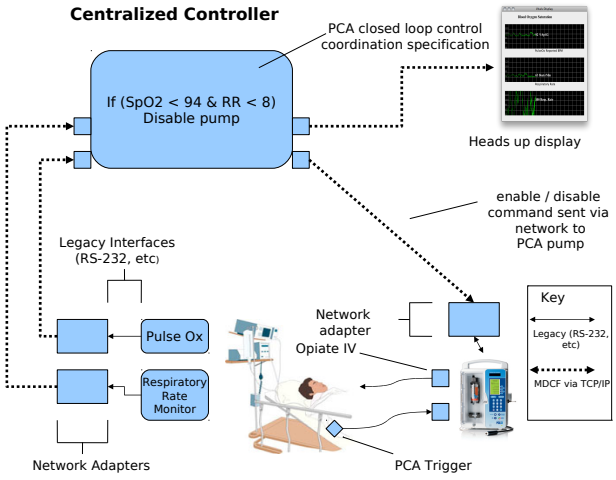

Figure 2: Multi-vital PCA pump control to prevent overdose

death. Overdoses may have many causes including pump configuration errors [10], the use of the wrong concentration of drug, drug interactions, and PCA-by-proxy (a caretaker or family member requests a dose on the patient's behalf). We have previously investigated systems that integrate one or more patient vital signs (such as RR, HR, and SpO_2) in order to determine if a patient is in respiratory distress and, if so, to disable the PCA pump for the duration of the distress state [16, 1]. The general topology of the system consists of patient monitors streaming data to a central controller. The central controller uses the data to determine if the patient is in respiratory distress, and if so, remotely disables the PCA pump, closing the loop between sensors and devices.

3 Challenges

3.1 Choice of Decision Model

'Intelligent' medical systems such as those described in Section 2 will have difficulty seeing widespread use in the field unless the research community can reach a scientific understanding of how various decision support approaches behave in a given domain, as it has been shown that no single technique is likely to be appropriate in all domains [3]. Ultimately this scientific understanding should be a product of analyses and experiments performed by the research community at large. We envision that providing a common generic framework to researchers will facilitate collaborative investigation and aid in overcoming these problems.

3.2 Interoperability

A major challenge of such medical systems is the requirement that they must interface with a broad range of devices developed by different manufacturers. The GSA architecture must have the ability to capture patient data streaming from vital sign monitors and other medical devices in the hospital room. Additionally, certain application domains require the GSA to output to a device (*e.g.*, a PCA pump or output display), and receive feedback from that device. Thus, the GSA architecture must be able to incorporate modules which enable interdevice communication.

3.3 Flexible Application of Existing Inference Approaches

Another major difficulty is that many promising approaches to machine inference do not apply directly to the numerical data streams commonly encountered in medicine (*e.g.*, discrete event bayesian networks, decision trees, or even the table based system described in Section 2.2 require coarse, high level events, while patient data is often encountered in the form of numerical data and waveforms). Therefore some sort of data pre-processing, such as fuzzy set classification, must occur to use real-time vitals in an "intelligent" medical system. More interestingly, these preprocessing steps may be different for separate patient populations (*e.g.*, a high heart rate in a febrile patient is a normal physiologic response without need for intervention. A high heart rate in a post-operative CABG patient may signify hypovolemia or an arrhythmia, both of which would require immediate interventions) or even between patients (*e.g.*, an athlete's resting heart rate is different than that of the average population). Realistic "intelligent" medical systems should take these differences into account and provide a way for practitioners to easily configure the system for a given patient or patient population. Since the core of the inference system itself would likely be resource intensive to curate (*e.g.*, generated using large datasets or extensive surveys of domain experts), the patient centric configuration parameters should be independent of the core intelligence if possible.

4 Architecture

We propose that the best way of solving the challenging problems discussed in Section 3 is through the development of a general architecture which is flexible enough that it enables us to rapidly prototype systems which are reconfigurable, and thus can address the problems in different ways, and subject these systems to verification and validation to determine the most effective (and patient safe) solution.

The example applications from Section 2 share architectural commonalities: raw data from medical devices is processed in successive stages; the final stage of processing is some form of machine inference algorithm; this algorithm produces a high level result (e.g., an alarm is raised, prognostications with respect to patient state is provided, or a command is transmitted to a medical device to alter its operating mode). In order to address the challenges described in Section 3 and due to the conceptually compartmentalized nature of the tasks in each application domain, we propose that a Generic Smart Alarm take the form of a fully configurable and flexible processing pipeline, where different preprocessing tasks can be linked with appropriate inference tasks, and these inference tasks with various outputs.

4.1 Pre-Processing Components

A smart alarm implementation using the GSA would consist of a collection of components which sequentially process and analyze input data. These plug-ins can be categorized according to their behavior and computational demands:

1. **Lightweight** components perform simple transformations on their input. Examples: the fuzzy set classifiers described in Section 2.1 or a task that applies a FFT to a waveform.

2. **Stateful** components maintain state from previous processing actions. Examples would include a task that maintains the running average of an input stream or a task that maintains trending information for a patient vital.

3. **High-Level** components typically take abstract inputs (i.e., not raw data) and compute some result. Examples would include inference mechanisms, such as a Bayesian Belief Network (BBN) or a rule table as described in Section 2.1.

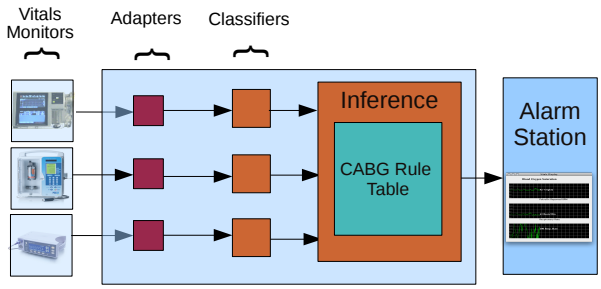

Figure 3: GSA configured as a smart alarm for CABG patients

4.2 Modular Inference Components

An inference components designed to work with the GSA should also be modular and customizable for different care contexts just as the low level processing tasks are customizable for specific patients. We believe that this feature would be most helpful for researchers who are trying to determine what inference algorithm or expert system is most effective in a certain context. As an example, the rule table inference used in the CABG smart alarm should be configurable with arbitrary rule tables. Likewise, if a BBN was used, the inference task hosting the BBN should have the ability to utilize an arbitrary BBN. Researchers could then directly compare different inference approaches within an otherwise identical system.

4.3 Pipeline Configuration

The GSA framework, combined with a palette of appropriate components, and a configuration which specifies how data flows from one components to another would be sufficient to instantiate a specific smart alarm for a given application. The following section provides example GSA configurations for the applications described in section 2.

5 Example Configurations

5.1 Smart Alarm Rule Table

The rule table smart alarm (Figure 3) consists of a fuzzy set classifier for each vital sign and a rule table expert system. Raw data is classified by fuzzy set classifiers, which can be parameterized for each patient, allowing medical professionals to tune the alarm for a specific patient. The classifications are forwarded to the rule table and appropriate alarm level is determined.

5.2 Decision Support for Neurointensive Care

Decision support systems, which we are interested in applying in a neurointensive care context, may need to be significantly more complex than the smart alarm system based solely on rule tables in order to capture the complexity of the domain in which it is to advise. We envision a system by which the architecture is adapted to include two major intelligent components: The first monitors the input vital sign data for long-term trends identified by medical experts as helpful in practice but difficult to identify manually due to the large amount of computation required to do so. These are output to a display program where they can serve as alerts or notifications to medical professionals. The second component receives input from the original vital sign monitors, the trends currently detected by the trend detection mechanism, and any pertinent patient data from linked medical databases, and outputs a series of possible medical conditions, alerts, or suggestions to a display. Response by medical professionals can be received by the display, which should also function as an input device. The medical professional could, for example, select one of the diagnoses to view the evidence that lead the module to suggest it, confirm the diagnosis to refine the device's decision making process, or enter more information not available to the machine, such as visual patient inspections performed by nurses, to improve the accuracy of the system.

5.3 Sensor Fusion and Smart Control for PCA Pumps

Post-CABG patient care commonly involves PCA. In this setting a smart controller for a PCA pump could be configured by taking the pipeline for the rule table smart alarm and extending it with an extra processing step that takes as input alarm level and then determines whether to send a signal to disable the PCA pump.

6 Verification and Validation

A high level distinction can be made between the two different types of correctness questions a designer of a smart alarm or CDS system should answer if the system in question will ever be used in a clinical context. The first question asks 'Does the overall system behave as specified?' The second question asks 'Does the high-level inference model effectively accomplish what we believe it does?' We believe the GSA could help researchers and system designers to address both issues.

The first question is primarily concerned with low-level system behavior. Consider the CABG smart alarm example from Section 2.1. If the RR monitor updates the system with new values at a much lower rate than the HR, SpO_2 and BP monitors, then the system must apply some form of event correlation in the preprocessing stage. For example, one correlator may wait to forward the other signals until the RR value updates. A different correlator could forward events as fast as the fastest vitals monitor and simply reuse most recent values for vitals which update more slowly. In essence, a given GSA configuration, along with formalized properties of the individual components, forms a system model which directly supports component-based model driven development (MDD). The component-based MDD approach to software system design has been successfully applied to the development of reliable distributed avionics systems [11] and could provide similar benefits for distributed medical systems.

The toolset required to answer the second question will be different from those used to answer the first. The effectiveness of the inference model (i.e. the decision tree, rule table, bayesian network, etc) in patient care will likely have to be determined by 'in-silico' trials against a virtual (simulated or pre-recorded) patient population. Hand curated models such as the rule table or an expert system should be analyzed for both completeness (all possible inputs have an output) and consistency (a input does not have more than one output). The GSA explicitly divorces the inference model from the underlying implementation in order to make it easier for designers and domain experts to validate the inference model with tools and approaches appropriate for the specific type of inference model in question.

7 Conclusion and Future Work

In this paper we have presented common challenging scenarios medical professionals face in the hospital room, scenarios in which we believe that the introduction of computer science techniques including system integration, machine learning, and sensor fusion could lead to major improvements in patient care. We have explained our position on what we perceive to be an effective way of surmounting many of the uncertainties inherent in solving challenging medical problems.

By constructing a flexible, generic software framework which

allows synthesis of patient information and application of machine learning techniques, uncertainties about the best solutions to specific application challenges can be addressed through experimentation.

Additionally, considering the formal verification process necessary when building safety critical systems, we regard the generic architecture approach to be advantageous, as once the infrastructure has been certified and approved, it easily be reused for different applications. If an application requires the creation of new, previously uncertified processing tasks, only those new tasks should have to be certified, potentially reducing the time to market for a new smart alarm.

We have implemented a prototype of the GSA on the Medical Device Coordination Framework (MDCF) middleware [17]. This prototype includes a palette virtual medical device components, adapter components for some real medical devices, pre-processing components, and an inference component capable of loading arbitrary rule tables. We intend to develop more inference components capable of loading different types of statistical or decision models (e.g. Bayesian Belief Networks) as well as formalize the behavior of each component's implementation. We will also produce detailed GSA instantiations addressing the specific applications described in Section 2, and begin testing the usefulness of the systems in improving critical care, with the hope that ideal configurations will emerge from the repeated testing and refinement process and via collaboration with the greater research community.

8 References

[1] D. Arney, S. Fischmeister, J. M. Goldman, I. Lee, and R. Trausmuth. Plug-and-play for medical devices: Experiences from a case study. *Biomedical Instrumentation & Technology*, 43:313–317, 2009.

[2] D. E. Arney, R. Jetley, P. Jones, I. Lee, A. Ray, O. Sokolsky, and Y. Zhang. Generic infusion pump hazard analysis and safety requirements version 1.0. Technical report, University of Pennsylvania, February 2009. Department of Computer and Information Science Technical Report No. MS-CIS-08-31.

[3] D. Bates, G. Kuperman, S. Wang, T. Gandhi, A. Kittler, L. Volk, C. Spurr, R. Khorasani, M. Tanasijevic, and B. Middleton. Ten commandments for effective clinical decision support: Making the practice of evidence-based medicine a reality. *Journal of the American Medical Informatics Assocation*, 10(6):523–530, 2003.

[4] Cerner CareAware. `http://www.cerner.com/public/Cerner_3.asp?id=29157`, 2008.

[5] Clinical Alarms Task Force. Impact of clinical alarms on patient safety. Technical report, American College of Clinical Engineering Healthcare Technology Foundation, 2006.

[6] Clinical Alarms Task Force. Impact of clinical alarms on patient safety. *Journal of Clinical Engineering*, 32(1):22–33, 2007.

[7] FDA infusion pump initiative. `http://www.fda.gov/MedicalDevices/ProductsandMedicalProcedures/GeneralHospitalDevicesandSupplies/InfusionPumps/default.htm`, 2010.

[8] CareFX FusionFX products. `http://www.carefx.com/products/`, 2010.

[9] GE Healthcare Interoperability. `http://www.gehealthcare.com/usen/interoperability/index.html`, 2010.

[10] M. Grissinger. Misprogram a PCA Pump? It's Easy! *P&T*, 33(10):567–568, October 2008.

[11] J. Hatcliff, W. Deng, M. B. Dwyer, G. Jung, and V. Ranganath. Cadena: An integrated development, analysis, and verification environment for component-based systems. In *in Proceedings of the 25th International Conference on Software Engineering*, 2003.

[12] R. W. Hicks, V. Sikirica, W. Nelson, J. R. Schein, and D. D. Cousins. Medication errors involving patient-controlled analgesia. *American Journal of Health-System Pharmacy*, 65(5):429–440, March 2008.

[13] M. Imhoff and S. Kuhls. Alarm algorithms in critical care monitoring. *Anesthesia and Analgesia*, 102(5):1525–1536, 2006.

[14] E. H. J Edworthy. Fewer but better auditory alarms will improve patient safety. *Qual Saf Health Care*, 14:212–215, 2005.

[15] Joint Commission. Preventing patient-controlled analgesia overdose. *Joint Commission Perspectives on Patient Safety*, page 11, October 2005.

[16] A. King, D. Arney, I. Lee, O. Sokolsky, J. Hatcliff, and S. Procter. Prototyping closed loop physiologic control with the medical device coordination framework. In *ICSE Companion*. ACM, 2010.

[17] A. King, S. Procter, D. Andresen, J. Hatcliff, S. Warren, W. Spees, R. P. Jetley, P. L. Jones, and S. Weininger. An open test bed for medical device integration and coordination. In *ICSE Companion*, pages 141–151. IEEE, 2009.

[18] S. Lawless. Crying wolf: False alarms in a pediatric intensive care unit. *Critical Care Medicine*, 22:981–985, 1994.

[19] P. E. Macintyre. Safety and efficacy of patient-controlled analgesia. *British Journal of Anaesthesia*, 87(1):36–46, 2001.

[20] M. Mullen-Fortino and N. O'Brien. Caring for a patient after coronary artery bypass graft surgery. *Nursing*, 38(3):46–52, March 2008.

[21] A. Otero, P. Felix, S. Barro, and F. Palacios. Addressing the flaws of current critical alarms: a fuzzy constraint satisfaction approach. *Artificial Intelligence in Medicine*, 47(3):219 – 238, 2009.

[22] S. Siebig, L. Juhls, M. Imhoff, U. Gather, U. Scholmerich, and C. Wrede. Plug-and-play for medical devices: Experiences from a case study. *Crit Care Med*, 38(2):451–455, 2010.

[23] D. Sittig, A. Wright, J. A. Osheroff, B. Middleton, J. M. Teich, J. S. Ash, E. Campbell, and D. W. Bates. Grand challenges in clinical decision support. *Journal of Biomedical Informatics*, 41:387–392, 2008.

[24] Phillips VisiCUE. `http://www.healthcare.philips.com/main/products/patient_monitoring/products/eicu/index.wpd`, 2010.