MODEL-BASED DATA PROCESSING, VALIDATION, AND ABSTRACTION FOR INTELLIGENT REAL-TIME PATIENT MONITORING AND CONTROL

Benoit M. Dawant[1], Eric J.Manders[1], and Daniel P. Lindstrom[2]
[1]Department of Electrical Engineering
P.O. Box 1662 Sta. B; Nashville, TN 37235.
e-mail: dawant@vuse.vanderbilt.edu
[2]Department of Pediatrics, Neonatology Division
Vanderbilt University

INTRODUCTION

Intelligent patient monitoring and control can be conceptually decomposed into four layers [Coi93]: (1) the signal level concerned with the acquisition and processing of the raw data, (2) the validation level that should eliminate noise and artifacts from the raw data, (3) the signal-to-symbol transformation level which maps features detected in the signals to various clinical conditions and (4) the inference level which integrates information from various sources to derive possible diagnoses and control actions or to predict the patient's evolution. Although a fair amount of work has been dedicated over the years to addressing questions and solving problems associated with each of these layers, most of this work has been done in isolation and very little has been done to conceptualize and formalize the interaction between layers and to provide an integrated framework for intelligent monitoring tasks. The main reason may be that intelligent monitoring tasks require real-time data interpretation, a concept that lies at the intersection of two vastly different disciplines: artificial intelligence (AI) and signal processing.

Historically, the AI community has focused on developing reasoning and knowledge representation schemes and methods, often assuming the availability, in a symbolic form, of the information on which reasoning is based. The signal processing community, on the other hand, has focused on developing more and more sophisticated numerical methods for the extraction of features (the data-to-symbol transformation) from individual signals without making use of contextual information to guide the selection of algorithms or adapt parameters needed by each individual algorithm. It is our position that new generations of monitoring systems will require bridging the gap between these disciplines and permit a tight interaction between signal processing and reasoning techniques. A similar approach has been taken in systems like Guardian [Hay92] or the Intelligent Cardiovascular Monitor (ICM) [Fac91] in the medical domain. In nonmedical domains, PREMON and its successor SELMON [Doy91], and MIMIC [Dvo91] are most closely related to our own work. The distinctive characteristic of our system called SIMON (an acronym for Signal Interpretation and MONitoring) is the attempt to use contextual information to dynamically adapt the processing and the abstraction of incoming data. SIMON is also capable of selecting and scheduling signal processing, abstraction, and validation tasks to optimize the relevance and the overall recency of the information it provides.

METHOD

SIMON consists of three main components implemented as independent processes that communicate with each other through shared memory and message passing: (1) a model, (2) an adaptive, reconfigurable data abstraction (DA) module, and (3) a display. This paper focuses on the DA module and on its interaction with the model.

The model characterizes patient status in terms of clinical states. For each state, it generates the expected evolution of the monitored variables and state transition conditions. These are expressed with a high level descriptive language and communicated to the DA module. The

DA module synthesizes signal processing methods to both follow the evolution of the monitored variables and detect state transition conditions. State transition conditions may specify the behavior of several variables and temporal relations between these behaviors. Because the DA module shields the model from any incoming data as long as variables follow their expected course or state transitions conditions are not met, the set of constraints on the data which specify normal evolution and state transition conditions is referred to as the *reporting* conditions. When new data reach the model, either because the variables do not follow their normal course or state transition conditions have been met, the state of the patient is reassessed and new reporting conditions are generated. This, in turn, modifies the signal processing strategy previously established by the DA. Information can also be sent directly to the model by the user to permit state transition caused by events not observable by the monitoring equipment.

In addition to reporting conditions, the model can issue state-dependent *monitoring* conditions. These include normal ranges for the monitored variables, detection thresholds, relative importance of the monitored variables, or frequency at which variables should be observed. The last two monitoring conditions are used to schedule and select tasks when real-time constraints necessitate making a choice between tasks to be executed. Finally, the model is also capable of specifying state-dependent display configuration information to focus the user's attention on critical events.

STRUCTURE OF THE DA MODULE

The DA module is organized around three main components: a data hierarchy composed of measurement and quantity objects, fault and artifact models, and a scheduling block. Each measurement object corresponds to a sensor or analysis device and quantity objects correspond to variables and parameters in the model. Relations between measurements and quantity objects are captured in a network structure.

Measurement objects are designed as semi-independent intelligent signal processing modules with two major components: procedural knowledge and control knowledge. The procedural knowledge consists of a library of digital signal processing (DSP) algorithms capable of detecting features from the raw signal. The control knowledge captures heuristics required for selecting the most appropriate algorithm for the extraction of a feature within the context defined by the monitoring and reporting conditions sent by the model.

Quantity objects also contain both procedural and control knowledge. Procedural knowledge is used to combine information from several measurement sources, when necessary. The control knowledge consists of heuristics used to select, within context, the best among possible sources of information about the quantity. At any given time, the preferred source of information for a quantity is called the Current Best Measurement Source (CBMS).

Fault and artifact objects model possible causes of data contamination. Faults model instrumentation problems, artifacts model events which are not of physiologic origin or are not part of the patient model (e.g., a motion artifact).

The scheduling block is central to the operation of the DA module. It is responsible for prioritizing the various data processing, validation, and abstraction functions performed by the DA module. The DA module is capable of using contextual information provided by the model to dedicate resources to the crucial decision support functions thus preserving the system's functionality even when faced with episodes of computational overload.

FLOW OF INFORMATION IN THE DA MODULE

When a new datum reaches the system, it is dispatched to the appropriate measurement object and a sequence of steps are initiated. First, the related quantities are informed that new information is available. When a quantity is informed that a related measurement object has been provided with new information, it can decide to either use that information or to ignore it temporarily (depending whether or not the information comes from its CBMS). Once a

quantity decides to use a source of information, it requests a feature extraction operation to the measurement. The quantity is also responsible for the validation of the information provided by the measurement. It first determines whether the value is within its normal range. If so and the value meets the reporting conditions, a data vector is sent to the model for patient status reassessment. If the value is outside of normal limits, a sequence of data validation steps are triggered. First, fault models then artifact models are activated to explain discrepancy between observed and expected values. If faults or artifacts can explain the discrepancy, the measurements and quantities they affect are labelled as unreliable, a fault or artifact monitoring procedure is initiated, and incoming data points are discarded as long as the fault or artifact is active. If no fault or artifact can explain the discrepancy, it is assumed that the data is valid and that the unexpected value is due to a change in patient status. A data vector is then sent to the model to reestimate the patient state.

SCHEDULING OF OPERATIONS IN THE DA MODULE

To permit their scheduling, all operations on the data (i.e, feature extraction, data validation, artifact and fault monitoring, etc.) are considered as independently schedulable tasks which are put on a task list. We use dynamic scheduling to constantly determine which task to execute first to maximize both the relevance of the operation and the recency of the results. The scheduling algorithm uses criticality, precedence relations, timing requirements, and availability of resources. Parameters affecting the scheduling algorithm can be dynamically modified by the model via monitoring conditions to allow state-dependent behavior. More about our current scheduling scheme can be found in [Daw93].

EXAMPLE

To illustrate the modelling of a particular monitoring task and the operation of SIMON, we have used a portion of the data provided for this symposium.

A. Problem statement and description

We have used four signals acquired with three instrumentation devices: saturation acquired with a pulse oximeter, heart rate and mean blood pressure acquired with an arterial pressure sensor, and the heart rate provided by an ECG monitor. We have also defined two faults: an arterial line fault and an ECG lead fault. We have not defined faults for the saturation signal nor do we have defined artifact models (the lack of detailed information on the data set prevented us to do so). It should also be noted that the arterial heart rate is affected by quantization noise and cannot be used as a valid measurement source for accurate heart rate values. It can, however, be used for data validation purposes as explained below.

Based on visual analysis, we have interpreted the data as describing a patient in either of two states: normal assisted ventilation (AV) and bag and mask ventilation (BMV). Lacking detailed information on the patient, we have been unable to model the long term response of the patient to episodes of BMV, to modifications in the ventilator setting, or to drug injection. For illustration purposes, we have thus limited our model to the expected short term response of the patient to either the beginning or the end of episodes of bag ventilation.

Our application is thus characterized by four measurements, three quantities, two states, and two faults. All the knowledge required to describe this application is captured and represented with a declarative language in what is referred to as the domain file (i.e., one domain file is defined for each application). This file is parsed and the runtime system is automatically instantiated. This completely shields the end user from the intricacies and the complexity of the system.

A typical section of data extracted from the 12 hours recording is shown in figure 1. It shows two episodes of bag ventilation marked by vertical solid lines.

B. Selected reporting conditions

The complete data set contains 6 episodes of BMV ranging in duration from 7 to

18min. All episodes have been used to empirically determine reporting and monitoring conditions. We have also assumed that the beginning and end of BMV episodes were manually entered.

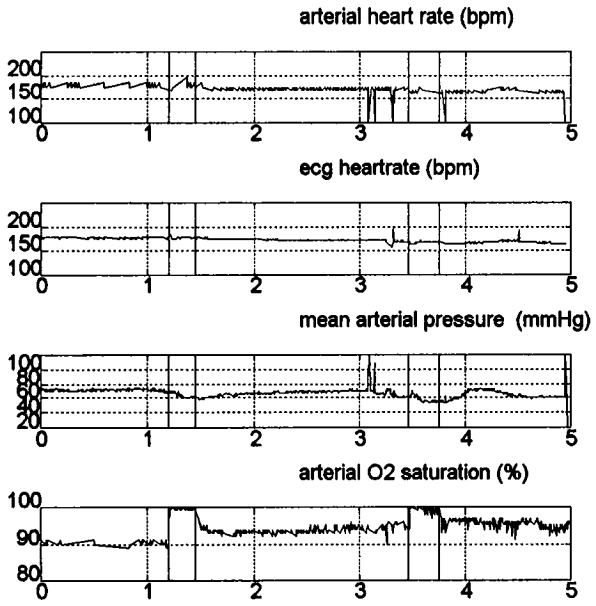

Figure 1: A five hours section of data. Shown from top to bottom are the arterial heart rate, the ECG heart rate, the arterial blood pressure and the arterial saturation. Episodes of bag ventilation are marked with vertical lines on all four measurements.

The expected response of the patient to the onset of BMV episodes has been defined in terms of two quantities, saturation and heart rate. We have observed a fast increase in saturation to 100% followed by a plateau and a decrease in arterial pressure. It was observed that saturation and blood pressure did not respond with the same time constant. Based on the provided data set we have chosen the following values: increase of saturation to 100% within 3min of the beginning of BMV and a drop of 5mmHg in the blood pressure within 9min. After the saturation has reached its maximum it is also expected to remain within a [97%, 100 %] window for the duration of the episode; blood pressure is expected to decrease throughout the episode. The response of the patient to a return to AV has been described as a rapid reduction in saturation and an increase in blood pressure, also within specified time intervals.

C. Selected monitoring conditions and data validation strategy

Our data validation strategy relies on information redundancy and correlation. For the heart rate, we have used the two independent data sources that were provided. We have also assumed that a fault on the arterial line will both affect the arterial pressure and the arterial heart rate values. The normal range for saturation has been set to [85%, 97%] for AV, and to [85%, 100%] for BMV. The normal range for heart rate has been set to [150bpm, 190bpm] and for blood pressure to [50mmHg, 85mmHg]. These are not changed between states.

The ECG lead fault model is activated when the heart rate goes outside its expected range. A fault on the ECG data channel is confirmed when the ECG heart rate is flagged as abnormal while the arterial heart rate remains within normal limits. The arterial line fault model is activated when the blood pressure goes outside its normal limits. A fault is confirmed if the arterial heart rate is simultaneously found to be outside normal limits. This is consistent with our assumption that a fault on the arterial line should affect both the arterial blood pressure and the arterial heart rate measurements.

D. Operation of the system

The system's operation is now illustrated on a portion of the data shown in figure 1. Because the lack of space precludes detailing the complete sequence of actions, a few representative events have been selected.

time:[0:00]; regular ventilation. Monitoring conditions for saturation, blood pressure, and heart rate are set to [85%, 97%], [50mmHg, 85mmHg], and [150bpm, 190bpm]. The reporting conditions are defined as "report when not steady". I.e., the model should be informed if any of the monitored variable is found to be not steady. Here steadiness has been defined in

terms of short term trends.

time:[1:12]; beginning of an episode of BMV (this is manually specified). The model specifies the reporting conditions associated with BMV for saturation and blood pressure, as well as normal limits for saturation. These are passed to the DA module.

time:[1:14]; response of the patient to the onset of BMV. Saturation reaches the 100% value for the first time, normal conditions are not violated. Both the saturation and the blood pressure follow their expected trend. The model is thus shielded from incoming data.

time:[1:27]; return to normal ventilation (again, this is manually specified). The model resets the reporting and monitoring conditions for ventilator based ventilation and specifies the expected transient response.

time:[1:35]; response of the patient to the interruption of BMV. Both the saturation and the blood pressure follow their expected trends, i.e., saturation has dropped below 95% within the specified time window and the blood pressure has began to increase within the expected time window.

time:[3:05:24]; the blood pressure value goes outside its normal limits.This triggers a fault detection procedure on the measurement from which the blood pressure is derived (here the arterial line). The fault model is activated and the heart rate value measured by the same arterial line is examined. It is found to be outside its normal range and the fault is confirmed. Because there is no alternative source of information for blood pressure, blood pressure values are labelled as being currently unavailable. A fault monitoring algorithm is triggered to detect the end of the fault.

time:[3:05:45]; the fault monitoring algorithm previously triggered on the arterial line detects the end of the fault. Computation of the blood pressure from the arterial blood pressure measurement is reactivated and blood pressure values are available again.

time:[3:08]; a second fault is hypothesized on the arterial line and the same procedure as the one described above is initiated. The arterial line measurement returns to normal within one minute.

time:[3:19:03]; the heart rate exceeds its normal range (recall that heart rate values are normally obtained from the ECG signal). This value is compared to the value provided by the arterial line. It is found that no arterial heart rate value is available at that time. The system cannot rule out a bona fide event and the model is informed. Being underdeveloped, the model cannot interpret that information and it sends an alarm to the display.

time:[3:27]; beginning of a second episode of BMV.

ACKNOWLEDGEMENTS

This research is supported, in parts, by NIH grant HL-38354 and by the Center for Biomedical Informatics, Vanderbilt University.

REFERENCES

[Coi93] Coiera E., "Intelligent Monitoring and Control of Dynamic Physiological Systems", *Artif Intel Med* 5(1):1-8, Feb 1993.

[Daw93] Dawant B.M., Uckun S., Manders E. J., and Lindstrom P. "The SIMON project. Model-based signal acquisition, analysis, and interpretation in intelligent patient monitoring", *IEEE EMB*, 12(4):82-91, 1993.

[Doy91] Doyle R.J. and Fayyad U.M. "Sensor selection techniques in device monitoring", *Proc 2nd Ann Conf on AI, Simulation, and Planning in High Autonomy Systems*, pp. 154-163, Cocoa Beach, FL., Ap 1991.

[Dvo91] Dvorak D.L. "Monitoring and diagnosis of continuous dynamic systems using semi-quantitative simulation", Ph.D. dissertation, Computer Science Dept. University of Texas at Austin, 1992.

[Fac91] Factor M, Gelertner D.H., Kolb C.E., Miller P.L. et al. "Real-time data fusion in the intensive care unit", *IEEE Computer*, 45-53, Nov. 1991.

[Hay92] Hayes-Roth B., Washington D, Hewett R, Collinot A, et al. "Guardian: a prototype intelligent agent for intensive-care monitoring", *Artif Intell Med* 4(2):165-185, 1992.