Monitor-Driven Data Visualization: SmartDisplay

James Fackler and Isaac Kohane, Children's Hospital and Harvard Medical School

Exhaustive display of all available clinical data, particular in data-rich environments like the intensive care unit, can easily overwhelm the ability of clinicians to comprehend the clinical status and evolution of their patients and may reduce their ability to detect pathological trends in a reliable and timely manner. SmartDisplay is a system we have designed that restricts the data sets displayed to time-lines of those parameters that are relevant to the patient context and to the particular care provider. The relevance criteria are provided by monitoring programs which may range in complexity from simple threshold alarms to full-fledged diagnostic engines. SmartDisplay can specify which parameters to display and the time intervals during which they should be displayed.

INTRODUCTION

In intensive care units, it is not unusual for a patient to be instrumented with a half-dozen probes generating over twenty signals sampled very frequently (between every few milliseconds to every minute). The same patient will also have fluid flux noted every hour or half hour. A number of clinical measurements and laboratory studies will be obtained every day or every few hours. Clinically significant physiological effects may occur over seconds (e.g. increased heart rate) to weeks (increased creatinine clearance). Knowledge of the data trends may provide the clinician with an opportunity to observe (even over multiple caregivers) such effects and therefore change therapy accordingly. A graphical summary of this data may permit rapid communication of these trends [1].

Most commercially available bedside critical-care monitors only provide two types of data display: a view of all signals monitored over the last 30 to 60 seconds and a summary view that allows the user to scroll (sometimes at varying levels of temporal resolution or granularity) through the last few hours to days worth of data. Such capabilities are often grossly inadequate to communicate effectively to the clinician what important events may have happened or are happening to the patient. To begin with, the data displayed on these monitors contains, at the most, 25% of the data which can be obtained on-line and in real-time [2]. Furthermore, a patient will be in the ICU for days or even weeks. Consequently, simply displaying a scrolling window over a two-week history of each measured parameter is likely to be unhelpful.

The nature of the data visualization requirements can be perhaps best considered in the context of the monitor responsibilities of an ICU nurse. Typically, he will be intermittently watching the monitors for evidence of current or impending cardiopulmonary pathology and therefore will be interested in only the last few minutes of monitored data (e.g. the heart rate as measured from the ECG). However, there are some parameters that are worth tracking over the entire period the patient was monitored. For instance, prior to administering a specified dose of a potentially toxic, renally-cleared antibiotic, knowledge of how much of the antibiotic was administered over the past week and at what level of serum creatinine would serve as a check to avoid erroneous dosing. The question this begs is: which parameters should be displayed and what period of time should each parameter display cover? This paper describes one methodology for answering this question and some examples of its use in critical-care data sets obtained at Children's Hospital.

The intuition that underlies the approach we have taken is that existing decision support programs that are capable of monitoring primary or "raw" bedside data provide important clues as to which parameters are relevant to display and when. Whether these monitoring programs implement simple boundary or threshold alarms or provide full-fledged differential diagnoses, the presumption is that they have been engineered to flag relevant data items or collections of data in a timely manner. Furthermore, even if the specifics of a hypothesized fault are incorrect, simply displaying the data that triggered the hypothesized fault may serve as a useful alert of a current or impending pathological trend or event. The central contribution of the research described here is in providing a language and an interpreter to translate the outputs of a wide range of monitoring programs into relevant data displays using off-the-shelf display technologies. We call this language and its interpreter SmartDisplay.

Related Work

Significant work has been accomplished in transforming numerical data into novel, concise visual metaphors that summarize large data sets (e.g. [1,3,4]. The investigations of Cousins and Kahn [5]. are closest to our own interests. They has been particularly influential in the design of the display layout or formatting components of the SmartDisplay language.

DESIGN ASSUMPTIONS

In the design of SmartDisplay we have made some simplifying assumptions about the format

0195-4210/94/$5.00

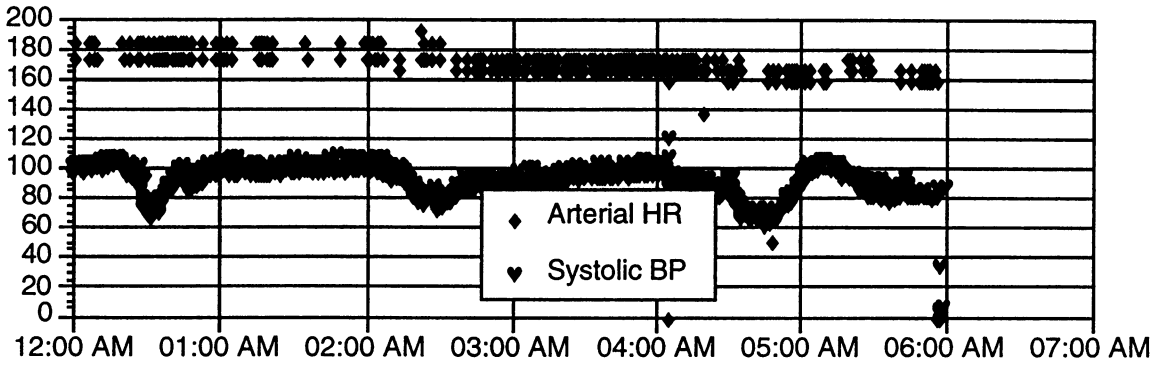


Figure 1: Six hour plot of heart rate and systolic blood pressure both measured via intra-arterial catheter.

of the data displayed. If these assumptions prove to be too limiting, we can revise them subsequently.

First, we assume that the clinician will be viewing clinical data trends on a rectangular screen of fixed area. Within the rectangular area are one or more horizontal lanes, vertically stacked, each with potentially independent temporal granularity or scale along the abscissa. Each horizontal lane or *time-line* can be independently labeled with a legend. One or more parameters can be plotted within each horizontal lane with the parameter value determining the position of a point or bar along the ordinate of a time-line.

Second, we assume that SmartDisplay will not control when a SmartDisplay specified set of time-lines (a *display set*) is executed. Nor does it control for what length of time the executed display set will remain on the screen before it is updated. These decisions are dependent on the particular class of monitoring programs that trigger the execution of a display and therefore require control logic that is outside the scope of SmartDisplay.

Third, we assume that display sets of interest will vary with the clinician (nurse, respiratory therapist, physician) observing the patient and the particular patient.

Finally, SmartDisplay should be able to support all the classes of monitoring programs listed below and be able to generate the displays described for each class.

Threshold/boundary alert

Display: If pH < 7.2 at time *t1* then display pH and pCO2 for a 24 hour interval prior to *t1*.

Single or Boolean combination of simple filters:

Display: Whenever: mean Heart Rate (HR) < 80 for at least 60 seconds and mean Systolic Blood Pressure (SBP) > 200 for at least 2 minutes then display Diastolic Blood Pressure (DBP) from the beginning of the interval of HR < 80 to the end of the interval of SBP > 200.

Automated abstraction engines

Programs such as Shahar's RÉSUMÉ [6] and Russ' TCS [7] automatically generate abstractions of primary data over time and parameter value. Display: For all abstracted intervals of increasing diastolic blood pressure, display heart rate during these same intervals.

Pattern-driven trend-detection engines

Some programs such as TrenDx [8] and DIAMON-1 [9] distinguish between competing knowledge-engineered archetypal patterns of parameter variation over time by comparing the degree to which primary data match or fit these archetypal patterns. Display: For the leading archetypal pattern engineered to detect falling blood pressure associated with manually-assisted ("hand-bagging") ventilation, display the heart rate during the interval from 2 minutes prior to the drop in blood pressure until the end of "hand bagging".

SPECIFICATIONS FOR THE SMARTDISPLAY LANGUAGE

We will motivate the specification for the SmartDisplay Language by working through an example monitoring task on patient data obtained from the Children's Hospital Multidisciplinary Intensive Care Unit. Figure 1 illustrates a graph of HR, SBP and DBP over a six hour period of a patient with Adult Respiratory Distress Syndrome.

SmartDisplay requires that for each relevant output or *trigger* of a monitoring program the following tuple or *display set* should be defined:

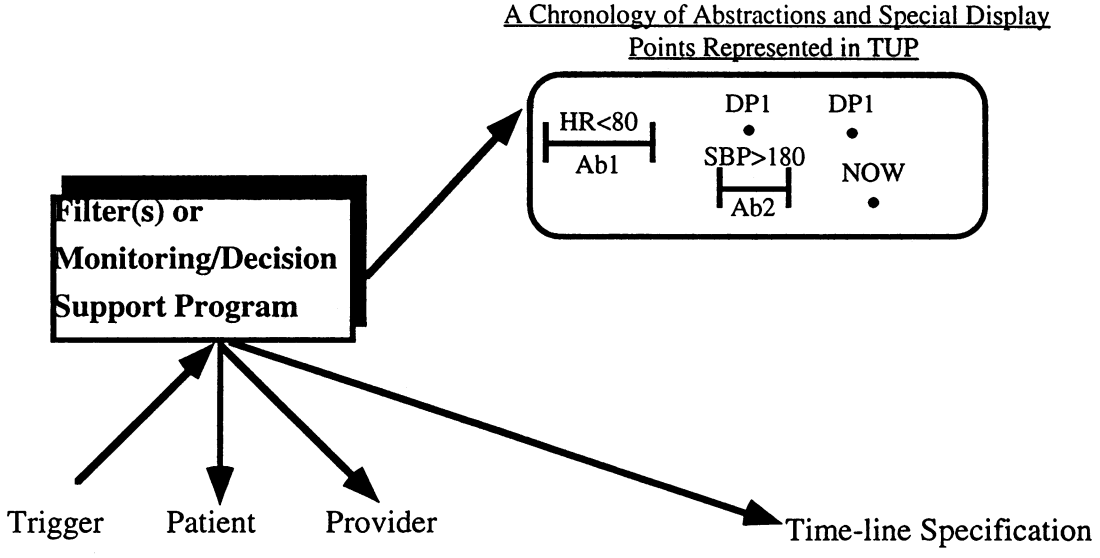


Figure 2: SmartDisplay Display Set Specification

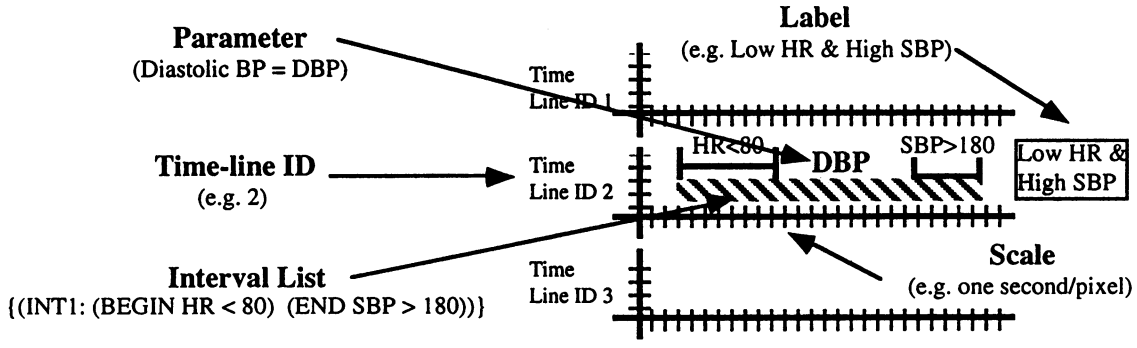
{Provider, Patient, Chronology, Trigger, Time-Line Specification}

where *provider* specifies the class of provider for which the display set is appropriate, and *patient* the patient for which the display set is appropriate. *Trigger* provides a pointer to the output of a monitoring program that triggered the execution of this display set. For example, if the following pair of filters: mean Heart Rate (HR) < 80 for at least 60 seconds and mean Systolic Blood Pressure (SBP) > 200 for at least 2 minutes generated an alert in a monitoring program, the alert would constitute the trigger of the display set. *Chronology* specifies a set of partially ordered points and/or intervals and their temporal relationships. This partial order is expressed in the temporal representation language of the Temporal Utility Program (TUP) [10]. The intervals and points represented in the chronology are referenced in the Time Line Specification which controls what actually appears on the clinician's display screen. Chronologies must be generated by any monitoring program designed to communicate with the SmartDisplay interpreter. In the example we have used, the chronology would include the duration and relative order of the intervals of HR < 80 and SBP > 200. In addition to the points and intervals in the chronology the SmartDisplay interpreter also recognizes references in the Time-Line Specification to two privileged time points: NOW (the present, in real-time) and TimeTriggered (the time the trigger was issued by the monitoring program). Figure 2 illustrates the components of the display set tuple.

A time-line specification determines how a graphing program will display each parameter along the horizontal lanes or time-lines described above. Time-line specifications are lists of the following form:

{(TL-ID, parameter,label, scale,{interval list})}

where *TL-ID* uniquely identifies a horizontal lane or time-line on the fixed rectangular display. *Parameter* specifies which parameter is to be plotted, *label* specifies a textual annotation for the time line. Labels are most useful if they bear a direct relationship to the alert message of the trigger. The *interval list* describes those times during which the specified parameter should be displayed. The intervals in the interval list are specified with respect to the interval endpoints and other points of the TUP chronology generated by the triggering monitoring program. *Scale* is the suggested time-scale or temporal resolution for that time-line. It can be overridden by the SmartDisplay interpreter if it does not permit the display within the fixed width of the display area of parameter values during times covered by the interval list. Figure 3 diagrams an instance of a time-line specification for the pair of HR and SBP filters described above. Figure 4 diagrams the time-line generated by the example display set tuple.



Time-line Specification

Figure 3: Time-Line Specification.

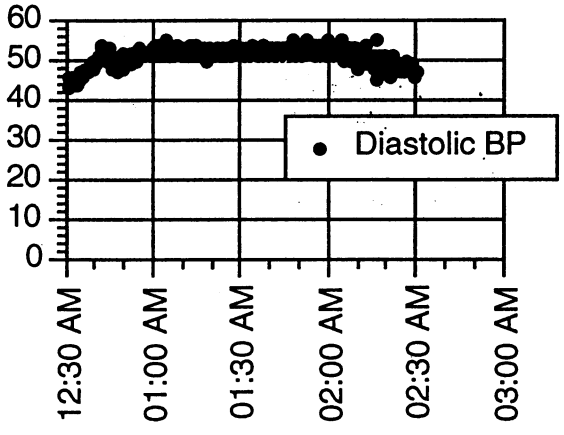


Figure 4: Result of Execution of Example Display Set.

Although the example we have used is contrived and quite simple, the SmartDisplay language enables the specification of a wide range of displays. Multiple parameters can be plotted across each time-line in multiple (disjoint or overlapping) intervals and temporal granularity can vary across time-lines. This would enable a knowledge engineer to write a display set that, when triggered, could show along one time-line the immediate hemodynamic effects of a drug and along another time-line the entire history of the intervals during which the patient received infusions of that drug.

IMPLEMENTATION STATUS

Earlier in the development of the display language we attempted hand-simulations of the operation of the SmartDisplay interpreter using TrenDx as the monitoring program [11]. These simulations led to the current definition of the SmartDisplay language. We are currently implementing the SmartDisplay interpreter. The program is currently able to plot displays whose chronologies only contain intervals with fixed endpoints. That is, there cannot be temporal uncertainty in the displayed intervals. One important implementation decision we have to resolve is whether to use the time-line display source code that Dr. Michael Kahn has kindly provided (from his earlier work on this subject [5]) or instead have the SmartDisplay interpreter generate graphing commands that are then executed by a commercial graphics package. Several graphing/plotting software packages available on personal computers now support some level of interprocess communications (e.g. OLE or AppleEvents). However, we have yet to determine if these packages can provide the SmartDisplay interpreter with sufficient control of the display through the available interprocess communications protocols.

CONCLUSION

We have described a language, SmartDisplay, that is intended to exploit the operation of a wide range of monitoring programs to focus the attention of clinicians onto a relevant subset of the available measured patient parameters by displaying them over specified intervals.

As it is currently defined, the SmartDisplay language has some significant limitations. First, it requires that a knowledge engineer select the relevant intervals and parameters and encode the display sets for each monitoring program intended to work with the SmartDisplay interpreter. Second, the language does not have any provision to direct the synthesis of a single display for each parameter when it is specified in multiple display sets (e.g. when two monitoring programs specify the display of heart rate).

Also, we have yet to answer several important questions. These include:

• Does SmartDisplay improve the rate at which clinicians accurately detect pathological processes? Is any change in performance related to the amount of data presented compared to commercial display systems?

• What are the control issues regarding the duration and update frequency of each display? How do these issues depend upon the nature of the monitoring programs?

Answers to these questions require testing SmartDisplay in conditions closely approximating clinical practice. In the short term however, we are working on completing the implementation of the SmartDisplay interpreter.

References

1. Tufte, E. R. *Envisioning Information*; Graphics Press: Cheshire, CT, 1990.

2. Fackler, J. C.; Kohane, I. S. Integration of sparse data from a clinical workstation with continuous data from bedside monitors. In: *Sixth Annual IEEE Symposium on Computer-Based Medical Systems.* Ann Arbor, Michigan: IEEE Computer Society Press, 1993:289-294.

3. Cole, W. G.; Stewart, J. G. Metaphor graphics to support integrated decision making with respiratory data. *International Journal of Clinical Monitoring and Computing* 1993, *10*, 91-100.

4. Wenkebach, U. Visualization of large datasets in intensive care. *Proceedings of Sixteenth Annual Symposium on Computer Applications in Medical Care.* 1992, 18-22.

5. Cousins, S. B.; Kahn, M. G. The visual display of temporal information. *Artificial Intelligence in Medicine* 1991, *3*, 341-357.

6. Shahar, Y.; Tu, S.; Musen, M. Knowledge acquisition for temporal abstraction mechanisms. *Knowledge Acquisition* 1992, *1*, 217-236.

7. Russ, T. A. Using hindsight in medical decision making. *Computer Methods and Programs in Biomedicine* 1990, *32*, 81-90.

8. Haimowitz, I. J.; Kohane, I. S. An Epistemology for Clinically Significant Trends. In: *National Conference on Artificial Intelligence.* Washington, D.C.: 1993:176-181.

9. Steimann, F.; Adlassnig, K. P. Clinical monitoring with fuzzy automata. *Fuzzy Sets and Systems* 1994, *61*, 37-43.

10. Kohane, I. S. Temporal reasoning in medical expert systems. In: *MEDINFO 86/Fifth World Congress on Medical Informatics.* R. Salomon, B. Blum and M. Jørgensens, Eds., Washington, D.C.: Elsevier Science Publishers, 1986:170-174.

11. Fackler, J.; Haimowitz, I. J.; Kohane, I. S. Knowledge-based Data Display Using TrenDx. In: Technical Report SS-94-01: *AAAI Spring Symposium: Interpreting Clinical Data.* Palo Alto: AAAI Press, 1994: 37-41.