

# An Integrated Data Management Approach to Manage Health Care Data

Diogo Guerra<sup>+</sup>, Ute Gawlick<sup>x</sup>, Pedro Bizarro<sup>+</sup>

CISUC/University of Coimbra<sup>+</sup>, University of Utah Health Sciences Center<sup>x</sup>

deag@student.dei.uc.pt, Ute.Gawlick@hsc.utah.edu, bizarro@dei.uc.pt

## ABSTRACT

Intensive Care Unit data management systems suffer from three problems: data and meta-data are spread out in different systems, there is a high rate of false positives due to default thresholds, and data mining predictions are not available in a timely manner. This proof-of-concept demonstration, based on the Intensive Care Unit environment of the University of Utah Health Sciences Center, presents a system that: i) integrates in one place historical data, events, rules, and data mining models; ii) is highly customizable letting users create or change rules; and iii) identifies possible future risks by performing data mining in soft-real-time. Using simulated inputs, we show the complete system working, including writing and editing rules, triggering simple alerts, prediction of cardiac arrests, and visual explanation of predictions

## Categories and Subject Descriptors

D.2.11 [Software Architectures] Domain Specific Architectures.

## Keywords

Sensors, Rules, Alerting, Event Processing, Data Mining

## 1. INTRODUCTION

Modern medical institutions have electronic devices that continuously monitor the vital signs of patients such as heart rate, cardiac rhythm, blood pressure, and many others. Monitors trigger alerts for critical values above or below predefined thresholds. Due to the generic configuration of those thresholds, the devices fire for the same critical values regardless of patient condition, demographics, and alarm history. That is, the monitoring middleware typically does not distinguish between:

- A patient with or without a cardiac condition;
- A male baby or a female senior with high heart rate;
- A patient that started to have high fever and a patient that has had high fever more than 30 minutes.

Ignoring those differences leads to missed alarms, a very high rate of false alarms and alert fatigue. This demonstration pre-sents a highly customizable system that lets users edit and create personal (per patient, per doctor, per patient-doctor), composable or overriding rules. In addition, the system applies data mining

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DEBS'09, July 6–9, 2009, Nashville, TN, USA.

Copyright 2009 ACM X-XXXXX-000-0/00/0004...\$5.00.

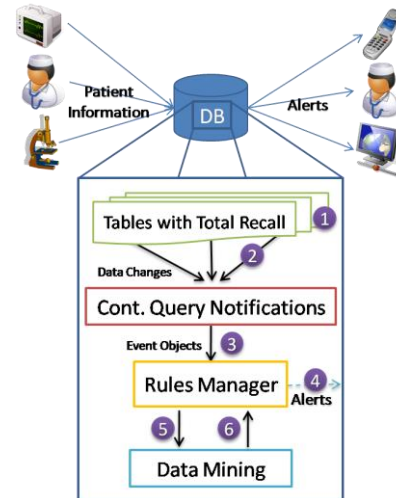


Figure 1 - Architecture

models in real-time and predicts probabilities of cardiac arrests and respiratory failures as new information arrives.<sup>1</sup>

## 2. COMPONENTS and ARCHITECTURE

The goal of the prototype was to develop a single, integrated, low-cost CEP system able to perform different kinds of processing in a centralized and integrated system. The prototype uses Oracle technologies (Total Recall, CQN, Rules Manager, and Data Mining) which have been developed as independent components and not as parts of an integrated system for event processing. Part of the challenge and motivation to build this demo was using these technologies in a seamless way. These technologies and their contribution are described below.

### 2.1 Oracle Components

**Oracle Total Recall** [3] (TR) is a technology developed to transparently handle the versioning of records. Those historical versions can be accessed with the AS OF or VERSIONS BETWEEN clauses. An example with AS OF is:

```
-- Select temperature 5 mins ago
SELECT temp FROM patients
```

<sup>1</sup> The authors would like to acknowledge for the success of this project to Dieter Gawlick for the coordination and for making available all the technical support needed. This work was supported to a great extent by Oracle staff: Aravind Yalamanchi, Pablo Tamaya, Srinivas Vemuri and Venkatesh Radhakrishnan. Also Dr. Sean Mulvihill, Dr. Edward Kimball, and Dr. Jeffrey Lin at University of Utah Health Sciences Center.

```
AS OF TIMESTAMP
systimestamp - INTERVAL '5' MINUTE);
```

**Oracle Continuous Query Notification** [2] (CQN) is a technology that allows the database engine to notify clients about new or changed data. Unlike database triggers, CQN notifies about changes in committed data only. CQN allows fine-grain notifications with SQL queries which specify what data changes should be notified.

**Oracle Rules Manager** [1] (RM) is a rules engine inside the database. It works based on events represented by objects and matches those objects to previously defined rules. Rules can identify sequence of events, patterns based on aggregations, non occurrence of events, and more. RM also allows user-defined: callbacks, conflict resolution policies, and consumption policies.

**Oracle Data Mining** [4] (DM) is data mining engine embedded in the database. In this project, the data mining model takes as input the patient laboratory information and outputs the predicted risk of patients having a cardiac arrest or respiratory failure in the following 24 hours. The models use a 725-patient training set from the University of Utah Health Sciences Center.

## 2.2 Architecture

All the information persists in database tables. These tables are TR enabled and automatically keep history of changes for each record (1 in Figure 1). CQN monitors changes on sensor states (2 in Figure 1), generates corresponding events objects and sends them to be consumed by RM (3 in Figure 1). Then, RM evaluates the rules and triggers actions to alert doctors and nurses through dashboards and mobile communication channels (4 in Figure 1). Some rules also trigger calls to data mining models (5 in Figure 1) to apply predictions in real time. The results are sent back to the Rules Manager (6 in Figure 1), which then determines if those results need to be pushed to staff as well.

## 3. CUSTOMIZATION

Physicians can edit, or add alerts in the Alerts section of the application. The alerts have multiple targets: for all patients, for one patient, for all patients of a specific physician, or for a specific patient-physician pair. The physicians create personal rules by editing generic rules. The alerts can be changed easily with forms or with the help of sliders (Figure 2). The patient history chart is also displayed allowing better decision making.

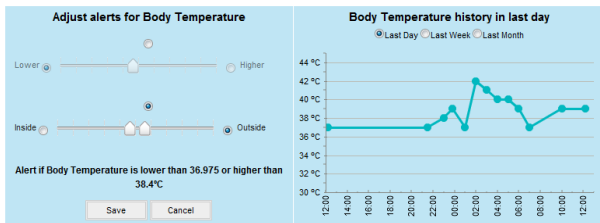


Figure 2 - Definition of alerts for vitals

### 3.1 Composability of Rules

The system success depends mostly on the way how rules can be expressed and how complex scenarios can be implemented. The rules are written in XML and can be easily composed. Figure 3 shows part of a description (in medical technical language) to identify possible cardiac arrest situations. The complete rule takes

more than a page to be described in technical English is implemented in Rules Manager with about 50 rules.

- a. *Persistent or rapidly progressing hypotension*
  - i. MAP, SBP or DBP in serious/critical range >30min
  - ii. MAP, SBP or DBP change from serious to critical (...)
- b. *Arrhythmia*
  - i. **Asystole (AYST)**
  - ii. **Supraventricular tachycardia (SVT)**
  - iii. **Arterial Fibrillation (AFIB)**
  - iv. **Arterial Flutter (AFLUT)**
  - v. **VFIB (Ventricular Fibrillation)**
- c. **GCS <=8**
- d. **Critical low or high temperature**

Figure 3 - Technical definition of a complex rule

An alert should fire when one red or three orange or five black statements are true. This is accomplished with a hierarchical, composable approach where rules signal further event objects into the system with their priority (color). Other rules monitor for those types of events and determine if a composable rule fires.

## 4. INTEGRATION WITH DATA MINING

The system uses data mining models to score the probability of Cardiac Arrest or Respiratory Failure for each patient as new laboratory information arrives. The system also displays the percentage contribution of each factor for the overall probability. Figure 4 is an example of a prediction of a Cardiac Arrest with 96% of probability. The physician can look at the chart and see that the negative fractions are values that are normal, or don't contribute for the condition of Cardiac Arrest and the positive ones are the ones that contribute to the cardiac arrest.

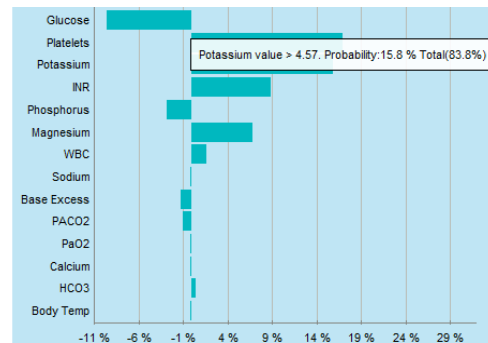


Figure 4 - Prediction explanation

## 5. DEMONSTRATION OUTLINE

In this demo the main features of the system will be demonstrated. The users can update data and rules, watch the alerts fire depending on the patient and doctor, and be alerted from data mining models when high risk situations may occur.

## 6. REFERENCES

- [1] Oracle® Database Rules Manager and Expression Filter Developer's Guide 11g. <http://download.oracle.com/docs>. Accessed May 2009.
- [2] Oracle® Database PL/SQL Packages and Types Reference 11g. <http://download.oracle.com/docs>. Accessed May 2009.
- [3] Oracle® Database Advanced Application Developer's Guide 11g. <http://download.oracle.com/docs>. Accessed May 2009.
- [4] Oracle® Database PL/SQL Packages and Types Reference 11g. <http://download.oracle.com/docs>. Accessed May 2009.