

A Rule-based decision framework for Medical Sensor Networks

Sowmya Manjanatha
Computer Science Dept
Boston University
sowmya@cs.bu.edu

Azer Bestavros
Computer Science Dept
Boston University
best@cs.bu.edu

Mark Gaynor
School of Management
Boston University
mgaynor@bu.edu

Steve Moulton
School of Medicine
Boston University
smoulton@bu.edu

1. Introduction

In the future, continuous vital sign data collection will be an integral part of every patient's electronic medical record. The sensors that collect this information will become smaller, wireless and potentially more heterogeneous. They will need to be smart, scalable and extensible to accommodate large patient populations. In this paper, we propose a general rule-based system that enables automated vital sign processing through the SnBench framework [1]. The SnBench framework provides scalability through distributed event management. The rule-based engine described in this paper enables adaptive event processing necessary to satisfy a wide array of patient needs.

The vast majority of vital sign monitoring instruments and patient support devices (e.g. ventilators, IV pumps and glucose monitors) function in isolation with little or no communication with electronic medical records or downstream systems. These devices display but do not process, save or integrate vital sign data with patient care information. A large amount of timely and potentially useful patient information is therefore discarded and never acted upon.

The development of automated response algorithms offers the possibility of providing more timely and precise therapy, with the potential for better outcome. As an example, autonomous control of ventilator oxygen concentration (FiO₂) using pulse oximetry (SpO₂) as a signal could reduce oxygen exposure to preterm infants and decrease the incidence of retinopathy of prematurity. A nearly infinite number of other types of fine ventilatory, IV fluid and medication adjustments are theoretically possible, based on subtle changes in a patient's physiology or response to treatment.

With the realization that there will be growing numbers

of continuously monitored patients, data aggregation, storage and management becomes a challenging task.

We believe that a rule-based system running in the application layer can provide the toolset necessary for adapting to a patient's changing conditions and supporting a wide variety of sensor platforms. The SnBench platform was built with heterogeneous architectures and distributed computing in mind. Therefore, SnBench provides the right framework to enable scalability of the rule-based system.

2. Design and Implementation

The SnBench is positioned between a medical sensor network and the actuator network as shown in Figure 1 (A). The SnBench can be visualized as a network computer consisting of two key distributed components namely *Sensor Resource Manager (SRM)* and *Sensor Execution Engine (SXE)*. The *rule-based engine* resides in the *SXE*. Sensor events are delivered to SnBench through *Sensor Services* that are responsible for translating sensor data to *STEP graphs*. The *Sensor Services* can be provided by such devices as a *Sensor Gateway* [2]. A *STEP graph* is a directed acyclic graph (DAG) composed of step nodes that represent functions, values, communication sockets etc [1]. The *STEP* program operates on values with the help of execution units in *SXE* named *Opcodes*.

Rules are logical units that are programmed into the *SXE* rule-based engine via *STEP graphs*. A rule-base is obtained by grouping several rules together. A rule fires and causes an action to be performed when the attributes of a sensor event match with that rule. Figure 1 (B) illustrates how a rule gets executed on the *SXE*.

We used the JESS Engine [5] for creation and management of rules within the *SXE*. JESS is similar to and uses

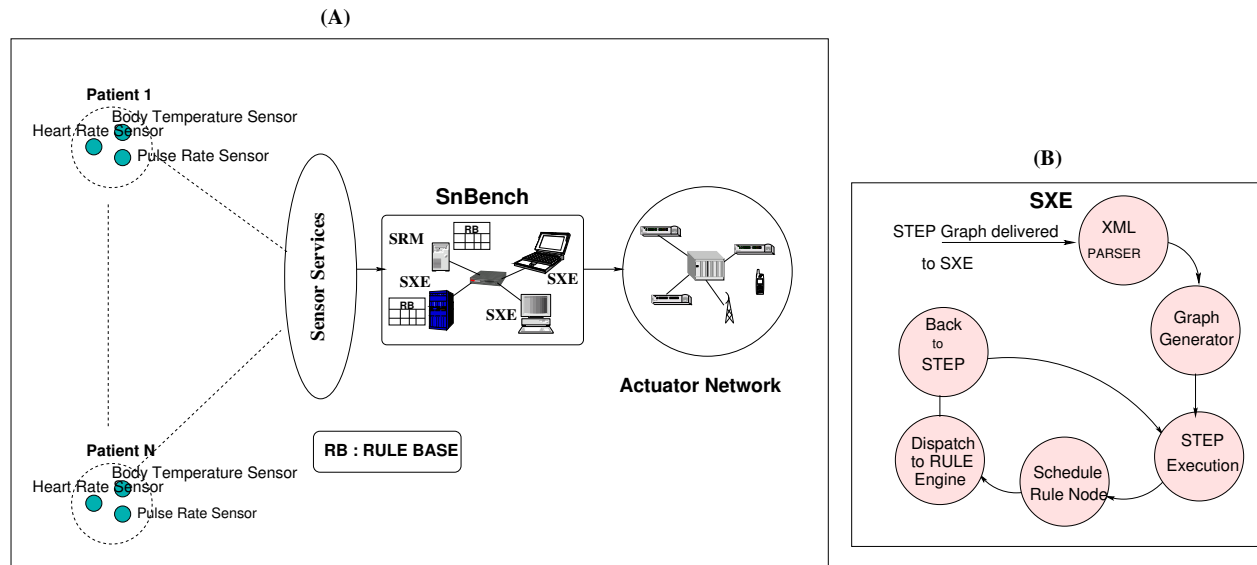


Figure 1. (A) SnBench architecture, (B) Rule Execution Cycle on SXE.

concepts from ruleML [5]. JESS is fully integrated with the Java 1.5 package and provides a number of JAVA APIs. Because SnBench is completely implemented using the JAVA standard platform, integrating JESS with SnBench was natural. Hence, we preferred the use of JESS over the ruleML package. The JESS module uses a forward chaining rule construct in addition to the Rete algorithm [6] for efficiency. JESS rule constructs are provided as input to the RuleInterpreter opcode on SXE. The RuleInterpreter module uses JESS APIs [5] to evaluate the rule expression and return the result back to the SXE.

We have built a prototype of the rule engine on the SXE using Jess 7.07b release of Java API. Programs are delivered to the SXE as STEP execution files or STEP Graphs. A new opcode namely RuleInterpreter was added into the SXE. RuleInterpreter opcode extracts Rule Node information from the STEP Graph.

The advantage of using smart sensors is their ability to communicate with each other and process information at the edge of the network, thereby collecting and reporting contextual information to various types of healthcare applications. Herein we have shown how a rule based engine can provide a flexible and easily extensible framework for processing, managing and contextualizing vital sign information from a sensor network.

Acknowledgements

This work was partially supported by NSF ACI Award #0330244, NSF CNS Cybertrust Award #0524477, CNS NeTS Award #0520166, CNS ITR Award #0205294, and EIA RI Award 0202067.

References

- [1] M. J. Ocean, A. Bestavros, and A. J. Kfoury, "Programming and Virtualization framework for distributed multitasking sensor networks", *Usenix*, 2006.
- [2] S. Baird, S. D. Haggerty, D. Myung, M. Gaynor, M. Welsh, and S. Moulton "Communicating Data from Wireless Sensor Networks Using the HL7v3 Standard", *International Workshop on Wearable and Implantable Body Sensor Networks*, April, 2006.
- [3] D. Malan, T. F. Jones, M. Welsh, and S. Moulton "CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care", *International Workshop on Wearable and Implantable Body Sensor Networks*, April 2004.
- [4] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar "Next Century Challengers: Scalable Coordination in Sensor Networks", *Proceedings of the ACM Mobicom*, August 1999.
- [5] E. J. F. Hill, "Jess, the Rule Engine for the Java Platform", <http://www.jessrules.com/jess/docs/70/>, Sandia National Laboratories, September 2006.
- [6] R.B.Doorenbos, "Production Matching for Learning Systems", *PhD Thesis*, Computer Science Department, Carnegie Mellon University, January 1995.