Improving the Knowledge Base by Integrating Vital Sign Data into Pre-Hospital Patient Care Records

Mark Gaynor, Dan Myung, and Steve Moulton

Abstract

This paper discusses improving the medical knowledge base by integrating real-time vital sign data into pre-hospital patient care records, such as in ***iRevive*** an EMS application developed for medical first response which is sensor enabled to automate the collection of fine grained vital sign sensor data. ***iRevive*** integrates emerging technology such as rule-based documentation, decision support, integration of wireless/wired medical sensors, and mobile wireless devices to enhance the capabilities of emergency medical responders. The development of ***iRevive*** is in the context of a design science to aid in the integration of Radio Frequency Device data (such as RFid and motes) into medical IT applications.

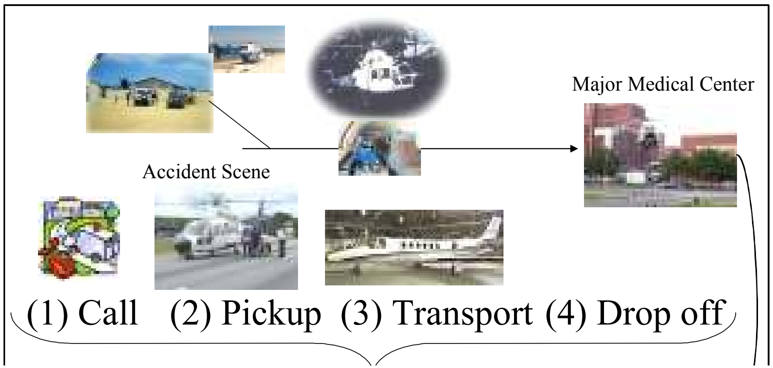
1 INTRODUCTION

This paper describes a pre-hospital emergency medical response application called ***iRevive***, which creates electronic patient care records (EPCRs) containing real-time vital sign data that is automatically collected in parallel with on-going paramedic observations and treatments. The benefits of integrating sensor data into IT applications such as ***iRevive*** include richer data to increase the medical knowledge base, improved patient care due to better situational awareness of the EMT, and more complete and consistent documentation enabling more effective QA/QI. However, because the technology of RF identification and sensing devices is rapidly changing, the best method of integration remains largely unknown; there is continued uncertainty in the choices between technology, architectures, and algorithmic solutions. This paper discusses our experience in making some of these choices.

We believe that real-time vital sign capture and information processing will play an important role in future healthcare systems while, at present, most vital sign data is displayed and discarded, in a closely monitored setting, such as a trauma resuscitation or in the operating room, vital sign data may be collected and manually entered into an electronic medical record every five or ten minutes. In an intensive care setting, once an hour usually suffices, and on the wards, once every four to six hours is the norm. The point is, a great deal of patient care information is being thrown away without regard to the potential usefulness of this information which may be collected at increased frequencies if automated.

We are working with a major critical care transport service to create a mobile pre-hospital electronic patient care system, called ***iRevive***. A high level view of the ***iRevive*** system is illustrated in Figure 1. This figure illustrates the phases of a typical airmedical mission: request for service, patient assessment, patient transport, patient delivery, documentation completion, and data transfer. Data capture begins when the transport service receives a request for patient transport. Upon arrival at the scene, nurses and paramedics are briefed by first responders. A wireless vital sign sensor is placed on the patient and the patient is assessed; ongoing treatment continues, additional treatment may be carried out and, time permitting, the documentation process is started. The data capture phase ends when patient care is transferred to the physicians and nurses at the receiving hospital or trauma center. The final phase requires completing all necessary documentation of the mission and transferring this data to the organization's database for billing, storage and future data retrieval.

The next section of this paper describes ***iRevive,*** the application we are building. Then we discuss design choices we faced to include real-time vital sign data in the ***iRevive*** EPCR. The following section discusses a sensor gateway architecture we developed to allow ***iRevive*** to consume data from a range of medical sensors from off-the-shelf devices to research oriented smart sensors that have the ability to process information locally.



2 *iRevive*

A future full function version of the iRevive application is illustrated in Figure 2. The arriving medic places wireless vital sign sensors on one or more patients to automate vital sign capture by the application. Each medic is equipped with a ruggedized tablet PC that captures and displays this real-time sensor data and allows the manual documentation of observations and treatments. Manual data entry is guided by a set of rules that enforce consistent and complete capture of data. Captured data is used to support the quality assurance/quality improvement (QA/QI) process, billing and research. Each transport vehicle is equipped with a base station linking local technicians, command centers, and destination hospitals. This WAN linkage will enable global allocation of resources as well as increased awareness of the condition of incoming patients at destination hospitals.

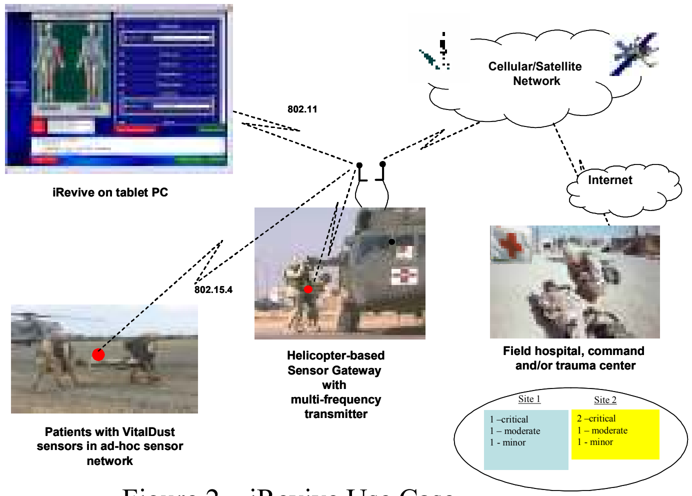


Figure 2 – iRevive Use Case

A unique feature of ***iRevive*** is the integration of fine-grained real-time vital sign data with manually recorded human observations and interventions. As sensor data streams into ***iRevive***, it is correlated with manually entered patient data to create a time-line of events. This integrated data offers a time-synchronized view of vital signs in parallel with observed changes in patient condition. It will enhance the ability of first responders to provide more effective care as well as increase the ability of researchers to gauge the effectiveness of in field interventions in the context of long term outcomes. This is critical to the advancement of evidence-based medicine as the automatic collection of fined-grained vital sign data provides more complete and consistent data than either paper based systems or electronic systems without sensor integration because data collected by manual entry is error prone and inconsistent.

Central to the ***iRevive*** architecture is a sensor gateway [Baird et al.] that aggregates data from multiple sensory devices. The aggregated data is available for consumption by different applications including the documentation component of ***iRevive***. The data is delivered from the gateway to applications based on standards that including web services and HL7v3. These web services permit exchanging data in a manner that is compliant with the HL7v3 (Health Level 7[1] version 3) messaging standards for exchanging medical information. The sensor gateway architecture provides a fixed API to a heterogeneous group of sensors by de-coupling the physical sensors from the application consuming the sensor data.

3 Design Choices

During the design and development of ***iRevive*** and other sensor enabled applications our team faced numerable technological, architectural, and algorithmic choices. We found that choice in sensor technology drastically affected later design options. There has been much hype about how simple, passive, and inexpensive RFID tags will revolutionize IT applications in various areas such as supply chain management [Asif and Mandviwalla 2005; Joglekar and Rosenthal 2005], transportation [RFID 2002], and access control, but little discussion about the logical evolution of these simple RFD tags to include data storage, processing power, and sensing ability.

3.1 Technology

There were many choices of technology for building the prototype wireless sensor (Viteldust) for use with ***iRevive***. Two important capabilities are the processing power of the sensing device and the flexibility of the processing device to interface with a variety of sensors. Our taxonomy [Gaynor et al. 2007] is to divide processing ability into three classes: none; a static level built into the hardware; or dynamic. The sensor flexibility ranges from none to a static set of sensors; and finally, to a generic interface acting as a dynamic sensor platform for experimentation. A device with no processing ability and without sensor integration represents the traditional simple RFID tag that responds to a read command with its identification, usually a 96bit EPC product code. Smart sensors such as motes [Hill 2003] that are programmable (having development environments and operation systems), and generic sensor interfaces for flexibility in sensor choice represent the other extreme which was the technology we used. The middle ground is static sensors and/or processing and is most appropriate for the efficient turn-key type solutions that do not need flexibility in use, or the ability to evolve. The processing and sensing capabilities of RFID tags define their range of usefulness for possible IT applications: as you increase processing and sensing ability, the RFIDs are more useful since they can be used in all infrastructure and architectural deployments.

3.2 Architecture

After choosing a technology, the next decision is the spatial relationship and interaction between sensors and readers, which include a range of possible architectural implementations [Gaynor et al. 2006]. The spatial taxonomy is based on mobility of the tag/sensors and readers and is broader than the taxonomy discussed by Tilak [Tilak 2002]. This mobility architecture defines the dynamic spatial relationship between the tags and their readers and has the following type of relationships:

- Fixed/Fixed – Both the RFD and reader are fixed. The advantages of wireless in this configuration are primarily cost and wirelessness. One example application are machines in a shop that can be instrumented with wireless sensors to predict when preventive maintenance is required by monitoring the temperature of critical ball bearings or vibration characteristics.

---

[1] http://www.hl7.org/

- Fixed/Moving - Fixed readers interrogating mobile RFDs is a typical application for tracking inventory; retail point of sales applications; or access management where fixed readers verify access based on smart RFid cards; Ski lift access [RFID 2002]; luggage handling; and object tracking.
- Moving/Fixed – Moving readers query fixed tags. This reversal of Fixed/Moving architecture has a reader moving over a fixed grid of tags. Examples of applications are inventory, train location tracking, or finding stored files [RFID 2002].
- Moving/Moving – When readers and tags are both moving. One example of this is a proximity monitoring system used on tankers in the North Sea to detect impending collisions when two ships are within very close range of each other. EMS triage applications such as ***iRevive*** fit into this quadrant. Roving medics triage and treat moving patients with real-time location and vital sign information from medical sensors.

The relationship architecture describing the interaction between tags and readers has the following taxonomy:

- Master/D-slave representing the traditional simple passive RFid tag with the reader interrogating a dumb slave RFD.
- Master/S-slave keeps the master/relationship between tag and reader, but allows the tag to execute business rules on the RFD, which greatly enhances flexibility (for example the tags ability to execute security protocols).
- Peer-to-Peer relationship is when the RFDs and readers are the same, allowing RFDs to form ad-hoc mesh networks . This is a hot area of computer science research.

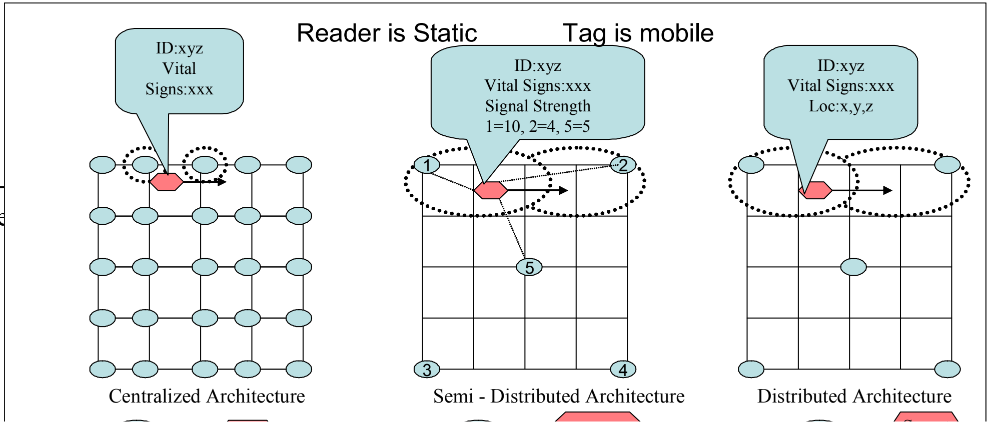
For a particular application there are often many choices, for instance, suppose we wish to integrate patient position into a medical application and have decided to use a fixed set of readers located inside a hospital to track mobile tags on patients and equipment. Figure 3 illustrates three different architectural choices. [2]

Dumb Tag – This fits the Master/Slave paradigm with a dumb tag responding to a reader's request. Since each reader has a short range and there are many readers placed in a pre-determined grid, the location of the dumb RFD tag is easy to locate. This method does not scale well in large areas. This is the left side of Figure 3.

Semi Smart Tag – A smarter tag enables a Master/S-slave infrastructure because the more capable tag can sense and broadcast the radio frequency signal strength received from all readers within range similar to some emergency 911 location finding technologies in cellular networks. This information can be used by the readers to compute the most likely location of the tag. The main benefit of this architecture is a less dense grid of readers. This is the middle of Figure 3.

Smartest Tag – The infrastructure on the right utilizes smart RFDs with an integrated GPS receiver that relays its location to the infrastructure. These smartest tags work well with both the Master/S-slave and peer-to-peer interaction. In this architecture, the GPS sensor is replaceable with the ultra-sound cricket [Smith et al. 2004] location tracking system.

These many architectures for a location tracking application demonstrate that even for simple applications, the choice of technology (including sensors), and architecture is unclear.



---

[2] We

3.2.1 Other Architectural Decisions

In addition to the relationship and mobility architecture, there are two other architectural choices designers may need to address. First, the management structure of data, whether it is centralized or distributed, needs to be considered. Second is the issue of where to filter sensor data, if that is required. The data filtering options range from the sensor to the application. These architectural choices grow out of the end-to-end arguments that were the foundation of the early design of the Internet [Clark and Blumenthal 2002; Gaynor 2003; Gaynor and Bradner 2004].

4 Sensor Gateway Solution

Recognizing the potential usefulness of vital sign information processing in future healthcare applications, as well as the need to link the sensor data into the ***iRevive*** application we built a sensor gateway called VitalTrac [Baird et al. 2006]. This gateway uses HL7 v3 messaging and web services to define messaging interactions between a data client and its sensor data server. This standard based approach gives application developers a fixed target with respect to controlling and consuming data from real-time sensors, while giving designers the flexibility to experiment with sensors from many different vendors. Currently, the gateway communicates with the off-the-shelf Welch Allyn Propaq (the unit used by our partner BMF), or the smaller Nonin AVANT® 4100, as well as several research oriented sensors such as 10Blade's Vitaldust [Gaynor et al. 2004] and Harvard's CodeBlue [CodeBlue 2007] sensor network infrastructure. The Propaq provides pulse oximetry, systolic blood pressure, diastolic blood pressure, and end-tidal CO_2 data to the ***iRevive*** application, which is more information that the Nonin that only transmits traditional pulse oximetry. However, the Nonin is far less expensive than the Propaq, much smaller, and wireless. The unique quality of the sensor gateway is that the application is de-coupled from the sensor providing the data. This approach promotes easy application development as there are advanced software development environments such as Microsoft's .NET or SUN J2EE that are tailored towards web services based applications.

Our Sensor gateway removes the complexity of proprietary and cumbersome protocols for exchanging and controlling physiological sensor data. The gateway architecture enables heterogeneous applications to access data from heterogeneous sensors with a standard based API that mitigates the complexity of integrating sensor data away from the application. This architecture allows application designers to focus on how to use real-time data, rather than the often complex protocols that most sensors adhere to for data exchange. Our SOA Sensor gateway de-couples applications from sensors via the application of emerging standards to exchange medical vital sign data.

The seamless exchange of electronic healthcare information among heterogeneous interoperable systems is an important goal for future healthcare information systems. To facilitate interoperability, a not-for-profit organization called Health Level-7 (HL7) was created in 1987. HL7 develops specifications, the most widely used being a messaging standard that enables disparate healthcare applications to exchange clinical and administrative data [Health Level 7 1997]. The latest version of the HL7 specification, version 3.0, departs from its predecessors by using XML messaging as its foundation. The semantics of the XML messages are defined by a Reference Information Model (RIM). This model is a group of already-defined classes, data structures, and vocabulary. By adhering to concepts found within the RIM, healthcare information systems can communicate more easily.

Our implementation acts as a gateway between the HL7 world and other, proprietary or non-standard wireless sensor network protocols, such as those implemented by CodeBlue. A client creates a query, packages it into an HL7 XML message, and sends it to the server. The server unpacks the message, reviews the contents, and sends a query to the wireless network according to the parameters specified in the message.

5 Conclusion

Integrating sensor and RF identification into medical IT applications has the potential to enable a new breed of applications which sense and respond to dynamic environments. The new technology of processors, RF radios on a chip, and small sensors will usher in this new application paradigm that promotes situational awareness with improved data quality. Building sensor enhanced applications and integrating sensor data into legacy applications is an important area for further research because of the strategic advantage that comes from the integration of multimodal.

Acknowledgements

We wish to thank the following organizations for helping fund our research: NIH grant (NIH 1 R41 RR018698-01A1), NSF (PFI-0227879, ACI-0330244, IIS-0529798).

6 REFERENCES

1 Asif, Z. and Mandviwalla, M. “Integrating the Supply Chain with RFID: A Technical and Business Analysis,” Communications of the Association for Information Systems; 2005, 393-427

2 Baird S, Dawson-Haggerty S, Myung D, Gaynor M, Welsh M, Moulton S. Communicating data from wireless sensor networks, IEEE Workshop on Body Sensor Networks. 2006.

3 CodeBlue Project. Matt Welsh, PI. http://www.eecs.harvard.edu/~mdw/proj/codeblue/, 2007.

4 Clark, D, and Blumenthal, M, “Rethinking the design of the Internet: The end-to-end arguments vs. the brave new world,”. TRPC August 10, 2000.

5 Gaynor M, Sensor/RFid Design Theory, working paper, 2006.

6 Gaynor M, Myung D, Winkler D, Ganesan S, Moulton S. An intelligent pre-hospital patient care system, Accepted to the International Journal of Electronic Healthcare 2007

7 Gaynor, M., Welsh, M, Moulton, S, Rowan, A, LaCombe, E, and Wynne, J, Integrating Wireless Sensor Networks with the Grid IEEE Internet Computing, special issue on the wireless grid, July/Aug 2004.

8 Gaynor, M., Bradner, S. A Real Options Metric to Evaluate Network , Protocol, and Service Architecture, Computer Communication Review(CCR), Oct 2004.

9 Gaynor, M, Network Service Investment Guide: Maximizing ROI in Uncertainty Markets, Wiley, 2003.

10 Health Level Seven. 1997 - 2005 Health Level Seven, Inc. http://www.hl7.org/

11 Hill, Jason, System Architecture for Networked Sensors Ph.D. Thesis, UC Berkeley, May 2003.

12 Joglekar N. and Rosenthal S., ‘Experimentation with RFID usage in supply chains’, POMS Chronicle, vol.12, no.1, first-second quarter 2005.

13 Smith, Adam, Hari Balakrishnan, Michel Goraczko, Nissanka Priyantha,Tracking Moving Devices with the Cricket Location System, Proc. 2nd USENIX/ACM MOBISYS Conf., Boston, MA, June 2004.

14 Tilak, Sameer, ‘A taxonomy of Wireless Micro-Sensor Network Models’, Mobile Computing and Communication Review, vol. 6 (2), 2002.