Interoperability of Medical Applications and Devices

Mark Gaynor (Boston University, School of Management), Dan Myung (10Blade), Amar Gupta (University of Arizona), James Rawn (Harvard Medical School), Steve Moulton (University of Colorado, Denver)

Abstract

This paper discusses several types of interoperability for medical applications and devices: the ability of applications to run on any platform; how modularity affects interoperability; and data exchange between heterogeneous applications. It draws from experience with Internet standardization to find lessons for creating standards for interoperability.

1. Introduction

This paper defines interoperability and then discusses how lessons from standardization of the Internet can aid interoperability of medical devices and applications. Several examples of medical applications are presented to illustrate three types of interoperability. These are: the ability to run the application on any platform (platform interoperability); the ability to interchange module sub-systems within an integrated application (modular interoperability); and the capability to exchange data (data interoperability) with other applications including information from medical devices containing sensors.

The current state of interoperability between medical applications is poor. In a world where international banking institutions exchange complex financial data, where individuals routinely shop online with relatively secure protocols, Boston Medical Center (BMC) and the Brigham and Women's Hospital (BWH) cannot electronically exchange basic patient information. Large academic medical centers, including BMC and BWH, typically have a combination of systems from vendors and home grown efforts. These result in "Island" like data infrastructures; data exchange within the organization is possible, but sharing information with outside organizations is significantly more difficult.

For broad acceptance and implementation of cross-organizational platform, modular and data exchange interoperability, there must be unambiguous standards at many layers above the traditional Internet TCP/IP set of protocols. These higher layer standards are currently emerging and will enable semantic understanding of data exchanged between heterogeneous applications, provided vendors comply with them.

The standardization of the Internet is the prime example of how interoperability between a wide group of vendors worldwide can work. Starting with simple standards and evolving these standards as they are used became a winning strategy when compared to the Open Systems Interconnection (OSI) model of specifying complex and complete standards from the beginning. The stunning success of Internet and related standards such as email, HTML and web services are good models for standardization of medical applications.

There is a significant need for standardization of output and interoperability of the many medical devices that measure physical vital sign information including pulse rate, blood oxygen saturation, blood pressure, end tidal CO2 content and multi-lead ecgs. While some standards exist for medical devices [Lesh 2007], most vendors use proprietary protocols that make interoperability impossible. Our experience building a sensor enabled ***iRevive*** application illustrates how sensor gateway architecture can standardize data exchange between applications and physical sensors.

Two medical applications are described in the context of the different types of interoperability defined in this paper. ***iRevive*** is an out-of-hospital patient documentation application designed for Emergency Medical Care. While not platform independent, the data exchange and storage in ***iRevive*** is based on extensible mark-up language (XML) schemes. Next, the Brigham and Women's Hospital Cardiac Surgery Electronic Medical Record (***BCSEMR***) is a web based in-hospital patient documentation system developed at Harvard's BWH. It enables data collection and reporting for patients undergoing cardiac surgery. The application is platform independent and uses XML templates to pull data from other IT applications at BWH.

iRevive is presented as an example of a platform dependent application with the infrastructure to exchange information with many heterogeneous applications. This pre-hospital documentation application runs in the Microsoft Windows environment. It is not currently web based because out of hospital emergency applications must operate without continuous network connectivity. ***iRevive*** contributes towards interoperability in two ways: it allows automated capture of real-time vital sign information into the EMR, and it serves as a data mediator enabling conversations between numerous emerging standards and many legacy proprietary applications. ***iRevive*** has the ability to produce a more complete and consistent EMR, which is then capable of exchanging information from the EMR with many other applications, including the National Trauma Registry of the American College of Surgeons (NTRACS) and other commercial applications.

Platform and data exchange interoperability is illustrated with ***BCSEMR***. It was developed in-house because vendors did not offer data exchange interoperability, nor the ability to acquire specific data elements from other in-house applications. Because ***BCSEMR*** is based on XML data exchange, it is capable of exporting subsets of data for outcomes measurement, quality improvement and clinical research.

The next section of this paper explores motives for the creation of an interoperable EMR. This is followed by a discussion on types of interoperability and lessons learned from the standardization of the Internet. The balance of this paper discusses ***iRevive*** for pre-hospital and ***BCSEMR*** for in-hospital documentation.

2. Motivation for Interoperability

The benefits of converting from a paper chart to an electronic medical record are dependent upon the implementation. At the most basic level, an EMR provides increased legibility and distributed access. Increasing value can be achieved by structuring data in a way that facilitates knowledge acquisition and data interoperability. Implemented optimally, an EMR application should improve communication, enhance clinical decision making, improve compliance with documentation and treatment standards, minimize redundancy, enable context specific information presentation, integrate clinical documentation and billing functions, and facilitate quality improvement and clinical research. These benefits are more likely to be realized when clinical information is captured as standardized data elements (rather than as free text) and stored in a way that allows the use of standardized retrieval methods. Unfortunately, data interoperability between heterogeneous medical IT infrastructures has been almost non-existent to date, despite the obvious advantages stemming from the ability to exchange data between applications that cross functional and organizational boundaries. Incompatibilities between applications occur both at the encoding level when differences in data structures create data silos and at the semantic meaning level where there is a lack of common standards for representing clinical data. Standards that can help facilitate interoperability at the semantic level of the EMR are discussed below.

3. General Interoperability

Interoperability has different meaning to different people. In this paper we discuss three important aspects of interoperability: platform; modular; and data exchange[1]. Applications restricted to a single platform such as Linux, Apple's OS X, or Microsoft's Windows/XP/Vista are called platform dependent; applications that can run on many (or all) operation systems are independent of the underlying platform. Other applications are built with a monolithic architecture where many application sub-systems are intricately linked with proprietary technology, while other applications have architecture with well defined modules and clear interfaces based on standards. Applications that create and exchange information based on closed and vendor proprietary standards stifle information sharing, while applications that store and exchange data based on open standards promote distributed data exchange because they display data interoperability. The next three subsections further define platform, modular, and data interoperability.

3.1 Platform Interoperability

An application is interoperable at the operating system (OS) level if it can run on similar hardware (i.e. a laptop or desktop computer) independent of the OS. Applications that are platform independent are better at meeting the needs of heterogeneous groups of users because they place fewer restrictions on users' choices of hardware and software.

Platform independent applications can either be web browser based or Virtual Machine (VM) based such as Java applications. These technologies can create applications that are interoperable at both the software and hardware level, albeit with some limitations. For example, applications with complex data display or entry will be difficult to use on devices with small screens or limited keyboards. Web based applications work with any OS that has a web browser. This includes

[1] These are not the only types of interoperability; however, they represent three important aspects of medical device and application interoperability.

Microsoft Windows OS, Mac OS X, and Linux. These OSs have several choices of web browsers, including Microsoft's IE, Mozilla's Firefox, or Apple's Safari[2]. Web browsers are also available on PDAs and cell phones. VMs such as Java provide a non-Internet technology to create applications running on any OS that has the VM. This includes traditional OSs on laptops as well as PDAs and some cell phones. Today's technology provides tools for platform interoperability.

3.2 Modular interoperability (i.e. best of breed Vs turnkey)

Modular interoperability provides the ability to break a complex system down into well-defined modules that communicate with well-specified interfaces. Baldwin and Clark [Baldwin 1999] explained the dramatic impact of modularity on the computer industry when IBM introduced the first modular computer system. This modularity in computer components transformed the industry by spawning new companies that made these now interchangeable parts, such as disk drives. In a software system, modular interoperability allows different application components to be interchanged, as long as the defined module interface is adhered to. Software modularity, in which complete systems are built from interoperable modules, enables rapid and flexible development.

Web services with XML encoded information allow the development of modular software applications and, as a result, are emerging as the standard for data exchange across distributed sets of heterogeneous applications [Graham 2002]. Web services enable Service Oriented Computing (SOA) by defining what services are available (via Universal Description, Discovery and Integration (UDDI) service) and how to invoke services (via the WSDL standard). By combining standard languages that semantically define the data, with messaging protocols that describe the flow of messages between modules, complex applications can be coordinated across many functional areas of an organization.

Modular interoperability within and between hospitals could have a dramatic impact on the efficiency of healthcare systems and services. Some teaching hospitals such as BWH have expensive, complex, home-built applications that have some degree of integration. Unfortunately, vendors such as Meditec create applications composed of many sub-systems including prescription order entry (POE), billing and insurance functions, lab order and results, and EMR. These turnkey systems, with their tightly integrated components, give users only one choice: take the complete system, or nothing. Given the complexity of the various sub-systems in a typical hospital no single vendor will have the best solution for all departments. Modular interoperability would give users the option of picking the best vendor for each area or function.

3.3 Semantic Data Interoperability

Data format interoperability is the ability to exchange information between applications with semantic meaning [Leah 2007]. This is the top layer of interoperability because it assumes that applications can exchange data via standards, e.g. IP, TCP, and HTTP. There are many good examples of this type of interoperability; banks, for example, are able to wire money across both organizational and international boundaries. The Open Financial Exchange (OFX) XML based protocol is designed to exchange data between financial institutions, businesses and consumers.

Technologies for semantic data exchange exist at two different levels: the data encoding layer and the semantic meaning layer. XML is becoming the standard for encoding virtually all types of information. XML provides encoding structure, but it needs language standards to specify what the XML tags mean and how the data is represented. These languages are domain specific languages that carefully define the set of items that can be defined, the attributes of the items, and the range of values these items take. Existing medical vocabularies (including ICD-9 codes and SNOMED) can partially assume this role; they are, however, incomplete and there is a lack of consistency between vocabularies. Combining transport, web services, and vocabularies would allow applications to more easily exchange data across both functional and organizational boundaries.

4. The Internet

There is much to learn about interoperability from the success of the Internet Engineering Task Force (IETF) in creating internet standards that promote interoperability at the transport and application layers. Organizations commonly share information on web pages across organizational and international boundaries. Email is regularly

[2] It is possible (but not wise) to build web based applications that only work with one browser because vendors often extend standard protocols with propriety extensions.

exchanged between hundreds of millions of users. Internet connectivity is becoming a commodity and its stunning and rapid successful standardization provides many lessons for interoperability above the transport layer.

4.1 Rough Consensus and Running Code

One tenet of Internet standardization is "rough consensus and running code" as described by Bradner [Bradner 98] in Request For Comment (RFC) 2418, which discusses general guidelines for Internet standards within the IETF. Roughly, this means that not only do most people need to agree with the standard, but it also must be implemented. The combination of these methodologies promotes standards that are easily implemented and not bloated with extra features The IETF is unlike organizations such as the International Telecommunications Union (ITU) or the International Organization for Standardization (ISO) that tend to develop complex standards overloaded with features, as the OSI protocol suite demonstrated [Gaynor 2003; Gaynor and Bradner 2004] when, despite the predictions of pundits, it failed to capture the networking market.

The main goal of an IETF Internet working group in developing a new standard is to achieve rough consensus among working group participants. Rough consensus is a somewhat vague idea – it means most but not all. According to the working group guidelines in RFC 2418: "51% of the working group does not qualify as "rough consensus" and 99% is better than rough. It is up to the Chair to determine if rough consensus has been reached". The rough consensus approach tends to produce smaller, simpler standards because they don't need to address the needs of every individual, organization or country involved in the standardization process. One example of this is seen when comparing the ITU's H.323 to the IETF's Session Initiation Protocol (SIP); both are voice over IP protocols that allow distributed communications, however H.323 (developed by the ITU) is complex because of the many requirements imposed by member countries. This is in contrast to SIP, which is relatively simple. SIP seems to be the clear winner, illustrating the value of rough consensus.

The IETF requirement of running code is to ensure the feasibility of implementing a proposed standard. This common sense requirement promotes standards that are "reasonable to code". The combination of Rough Consensus and Running Code promotes lean standards with working prototypes.

4.2 End-2-end

The end-2-end principle is a large part of the design philosophy behind the Internet [Seltzer 84]. According to the end-2-end principle, networks should provide only the simplest of services. The end systems should have responsibility for all applications and any state information required for these applications. The idea is to keep the network simple, and build any needed complexity into the end, or edges, of the network. By providing the basic building blocks instead of complex network services, the network infrastructure will not constrain future applications. The end-2-end argument is one of increased innovation, and the proof of its validity is the success of the Internet.

The creators of the Internet believed in the end-2-end argument, and the basic Internet Network and Transport layer protocols IP, TCP, and UDP are examples of its application. The network layer protocol IP guarantees little; however, this simplicity translates into flexibility below at the device layer and above at the transport layer. Different applications will demand different services and the network should not constrain these choices. The end-2-end argument helped the designers of the Internet promote the development of applications by users because of the flexibility it gave developers.

4.3 Testing Interoperability

There are several ways to test internet standards for interoperability in both early and later phases. Having two independent development teams working on an implementation of the standard without contact, and then testing interoperability is a good indication of unambiguousness of the standards. By demonstrating interoperability between their implementations without any communication between the groups prior to testing demonstrates a clear standard. Effective in later standardization phases is vendor interoperability or "bakeoff" testing. In later phases, when many vendors are selling products supporting the standard, a "bakeoff" such as for voice over internet protocol (VoIP) SIP products[3], may be useful.

[3] http://www.callcentermagazine.com/GLOBAL/stg/commweb_shared/shared/article/showArticle.jhtml?articleId=8700868&pgno=6

Interoperability "bakeoffs" provide test beds to evaluate which vendors have successfully implemented the emerging standard in the context of interoperability with products from other vendors. Both these methodologies provide tests of the completeness and clarity of the standard

4.4 Lessons Learned

Table 1 below summarizes several lessons from the standardization of Internet protocols that organizations involved in creating and harmonizing health care related standards will benefit from.

Table 1 – Lessons From Internet Standards

Lessons	**Result**
Rough Consensus	Simple standards
Running Code	Standards that can be implemented
End-to-end Infrastructure	Increased innovation
Independent Implementations	Clear and complete standards
Bakeoff	Vendor compliance to standard, Clear and complete standards

Although internet IETF standards have led to much success with transport and application development, there standards do not attempt to assign semantic meaning to the data that is sent from application to application across the Internet. Semantic and data encoding standards such as Health Level-7 (HL7) and XML are emerging from other standard development organizations; these are discussed in the following section.

5. Standards for interoperability

5.1 Service Oriented Architecture

Both ***iRevive*** *and* ***BCSEMR*** are based on a service-oriented architecture providing the necessary agility and flexibility to link with the various internal and external healthcare systems that are essential to an independent, critical care environment. Both implement this SOA architecture utilizing web services standards [Gaynor et al 2002] [Graham 2002] such as Extensible Markup Language (XML) for data encoding [XML 2007] [Sokolowski and Dudeck, 1999], Simple Object Access Protocol (SOAP) [SOAP 2003] to define message envelopes, and Web Services Description Language (WSDL) [WSDL 2001] service description. Both ***iRevive*** and ***BCSEMR*** exchange information via web services, which provides an API that is easy to program, thus promoting data exchange.

5.2 The Healthcare Information Technology Standards Panel

The importance of common standards to exchange medical information cannot be over emphasized. This topic was recently summarized in a report from the July 2006 hearing on the Functional Requirements for a Nationwide Health Information Network. The Healthcare Information Technology Standards Panel (HITSP) has defined a "Minimal Data Set" to describe many types of medical information at many levels. This minimal data set includes the semantic meaning of medical information that can be requested, acceptable responses, and the structure messages should take when exchanging information between applications. The recommendation from the HITSP committee includes a group of standards that harmonize many heterogeneous standardization efforts into a manageable group, in order to promote interoperability that will improve treatment and reduce costs. Our work is mindful of these emerging standards.

5.3 Emerging EMR Standards

The many components of an EMR complicate the interoperability of health care applications. The key modules of a typical in-hospital EMR are administrative systems, clinical documentation, laboratory, radiology, pharmacy, and physician order entry. These areas have overlapping and competing standards, all of which have been developed by different organizations (e.g. HL7, CEN, and ASTM). Examples of overlapping terms include 11 different ways to define and spell "Total cholesterol" [Stanford and Thornton]]. Below is a summary of some of the more important standards:

- International Classification of Disease (ICD) [ICD-9 2006] is published by the World Health Organization[4]. ICD is primarily used to identify a disease or problem for billing purposes.
- Systematized Nomenclature of Medicine (SNOMED) [SNOMED 2006] was developed by a division of the College of American Pathologists to provide a "comprehensive, multi-axial, controlled terminology" [Stanford

[4] www.eicd.com/EICDMain.htm

and Thornton] for indexing an entire medical record. SNOMED-CT (Clinical Terms) specifies the core file structure of SNOMED medical terms.

- Logical Observation Identifiers, Names, and Codes (LOINC) [LOINC] is used to identify individual laboratory results, clinical observations, and diagnostic study observations.
- Health Level 7 (HL7) [HL7] is a messaging protocol for exchanging health care information. It includes several vocabularies, such as patient demographics. Unfortunately, there is poor backward compatibility between early and later versions of HL7; thus, there is poor compatibility between vendors who support different versions of the same standard.
- National EMS Information System (NEMSIS) [NEMSIS 2007] is a standard for pre-hospital care endorsed by the National Highway Traffic and Safety Administration (NHTSA). This is a domain focused standard for out of hospital emergency medical services.

HITSP is reconciling these often overlapping and inconsistent standards to enable a consistent elemental description of the EMR. It is adopting LOINC for assessing a patient's condition. This instrument consists of a set of questions and allowable answers. One example of using LOINC is the creation of an instrument for evaluating patients admitted to elderly care facilities [Kramer 2004]. The questions and answers specified by LOINC use the vocabulary of other standards (e.g. HL7, SNOMED, and ICD-9) to semantically define the meaning of each message. HL7 is then used as the messaging standard for defining the flow of messages being exchanged.

6. Applications

This section presents two examples of medical documentation systems that illustrate different types of interoperability. ***iRevive*** is a pre-hospital non-web based patient documentation application that is built to run under Microsoft's Windows operating system. It captures real-time sensor data and manually entered observations and treatments under field conditions. Two essential components of ***iRevive*** are its sensor gateway and data mediator. The sensor gateway de-couples vital sign sensors from the application, while the data mediator translates between proprietary protocols to enable ***iRevive*** to exchange semantically understood information with legacy hospital IT systems. ***BCSEMR*** is an in-hospital documentation system that is web browser based and platform independent. It uses XML templates to exchange information within the hospital for patient care. It also exports de-identified patient data to the Society of Thoracic Surgeons Adult Cardiac Surgery Database, to allow sharing of outcomes data.

6.1 iRevive

Figure 1 illustrates how the ***iRevive*** application [Gaynor 07] will be used by a typical EMS service provider. The arriving medic places wireless vital sign sensors on one or more patients. Each medic is equipped with a ruggedized tablet PC that captures and displays the real-time sensor data and allows manual documentation of observations and treatments. Data capture is automated for vital sign data. This data entry is guided by a set of rules that enforce consistent and complete data capture. Medics are linked to the transport vehicle or aircraft via an 802.11 wireless infrastructure that provides situational awareness to local and regional providers, so that other may better anticipate patient care needs. Each transport vehicle is equipped with a base station that links local technicians, command centers, and destination hospitals. This WAN linkage enables increased awareness of the condition of incoming patients at the destination hospital. During patient transport, ***iRevive*** continues to capture both sensor data and data recorded by medical personnel. The ***iRevive*** application enables EMS service providers to create and transfer complete electronic patient care records, composed of automated, time-stamped vital sign information and manually entered human observations and interventions.

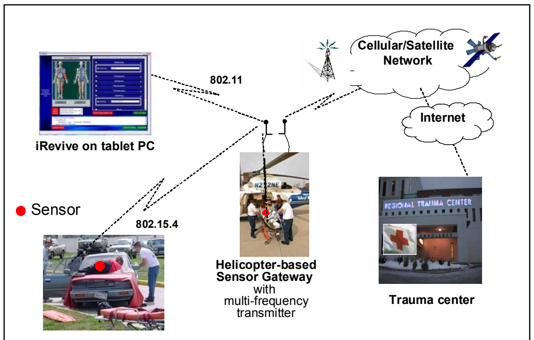

Figure 1 – iRevive Use Case

6.3 Sensor Gateway architecture and details

We believe that real-time vital sign capture and information processing will play an important role in future healthcare systems. 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 frequently suffices, and on the wards, once every four to six hours is the norm. A significant amount of patient care information is being thrown away without regard for the potential usefulness of this information.

Recognizing the potential usefulness of vital sign information processing in future healthcare applications, as well as the need to link sensor data in the ***iRevive*** application, we built a sensor gateway called VitalTrac [Baird et al 2006]. The 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 for controlling and consuming data from real-time sensors, while giving designers the flexibility to experiment with sensors from different vendors. Currently, the gateway communicates with the off-the-self Welch Allyn Propaq cardiopulmonary monitor, the smaller Nonin AVANT® 4100 pulse oximeter, as well as several research oriented sensors including 10Blade's Vitaldust sensor [Gaynor et al 2004] and Harvard's CodeBlue [Lorincz et al 2004] sensor network infrastructure. The Propaq provides pulse oximetry, systolic blood pressure, diastolic blood pressure, and end-tidal CO_2 data to the ***iRevive*** application; the Nonin only transmits traditional pulse oximetry. However, the Nonin is far less expensive than the Propaq, much smaller, and wireless.

Our Sensor gateway removes the complexity of proprietary and cumbersome protocols for exchanging and controlling physiological sensor data. The application is de-coupled from the devices that produce sensor data, thus simplifying the integration of vital sign data into web services based applications. This architecture allows application designers to focus on how to use real-time data, rather than the often-complex protocols that most devices adhere to for data exchange.

6.3.1 Data Exchange

Enabling data exchange between pre-hospital systems and the hospitals they serve is a difficult problem requiring a large number of complex transformations. Our approach uses a mediator to exchange information (developed by our colleagues at the University of Arizona in cooperation with researchers at 10Blade, Inc). This approach reduces the number of translations between pre-hospital systems (m) and hospital systems (n) from (m x n) transformations to (m + n) transformations, using ***iRevive*** as the overarching schema. To achieve this, ***iRevive*** was designed as a flexible superset of schemas, which can morph into any of the other component schemas. The high level architecture is illustrated in Figure 2. Transformation into a common data format that is a superset of all the overlapping standards is far more scalable than translating each standard to every other standard. This data mediator approach facilitates the sharing of data with other pre-hospital in-hospital systems, by requiring translation to and from the new standard, not the n (or m) translations for a non-mediated scheme.

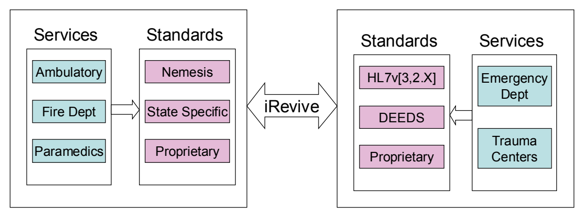

Figure2 – Data Exchange Via Mediator

Our method is flexible in the context of deployment architecture. Figure 3 illustrates that either a centralized or a distributed version of the data mediator is possible. The centralized model has one mediator that communicates with all data producers and users as depicted in Figure 3 (a). The distributed model places a wrapper around each application which converts to and from each standard as illustrated in Figure 3 (b). Our methodology also allows hybrid infrastructure, for example: each organization could have a local mediation server. The flexibility to use a range of architectures from distributed to centralized[5] is one important aspect of our mediation infrastructure.

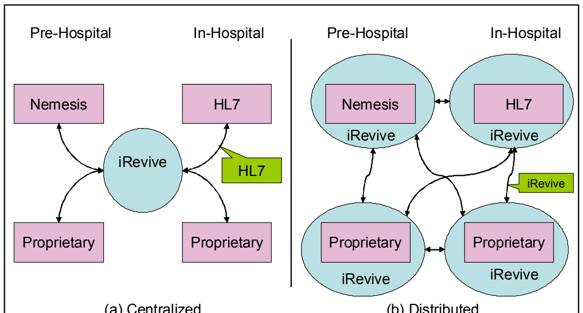

Figure 3 – Centralized Vs Distributed Infrastructure

[5] See Gaynor's book [Gaynor 2003], and Gaynor and Bradner's [Gaynor and Bradner 2004] for an argument about the value of this flexibility.

There are advantages and disadvantages to both models:

- Centralized
  1. More efficient management
  2. Easy to monitor
- Distributed
  1. Controlling your own data
  2. Robust to failure

One key problem in the implementation of EMRs is the non conformance to standards by vendors, teaching hospitals and research institutions. Both BMC and Partner's Health Care have partially integrated IT infrastructures; however, BMC and Harvard have a limited ability to exchange information with semantic meaning. This lack of interoperability tremendously limits the value to all users of EMRs. Network economics [Katz and Shapiro 1985] tell us that greater interoperability creates greater value. For example, in the networking world a single phone network in which all users can connect with all other users is more valuable to each user than two distinct phone networks where users can only connect to users in their own network. If, however, the two different phone networks are interoperable, then the value of the two distinct networks to each user might exceed[6] that of the single network. Similarly, medical IT infrastructure is more valuable to all users if data transfers between heterogeneous applications happen with semantic meaning intact.

6.4 A Mediating Schema Approach for Data Translation

Several mediator based schema integration mechanisms have been proposed in previous literature [Wiederhold, 1993; Gupta, 1989]. We have extended the previously proposed heterogeneous data translation mechanisms to suit the context of healthcare. The issues specific to heterogeneous data translation in healthcare include the following types:

1. The data elements and semantics of the patient care records can be well defined in specialized contexts. For example, standards such as NEMSIS (National EMS Information System) and DEEDS (Data Elements for Emergency Department Systems) specify the core data elements of a patient care record in the context of pre-hospital care and emergency department care respectively.
2. Privacy is a major concern to clinical research in linking electronic patient care data between emergency medical services (EMS) systems and the hospitals they serve.

[6] This argument is an extension of Gaynor and Brander work on innovation of network based services.

6.4.1 Schema Incompatibilities

Several schema incompatibilities were encountered during the schema matching and data translation process. We present a summary of some of the incompatibilities using the [Reddy et al. 1994] framework in Table 2.

Table 2 - Schema Incompatibilities

Type of Conflict	Description	Example
Naming	Different vocabulary for similar concept.	**ref_pulse Vs pulse_rate**
Key	Different keys to identify a record.	SSN Vs drivers license # to ID a patient record
Missing Data	Different data captured	Home phone Vs emergency contact phone #
Level of Abstraction	Different granularity	4 locations for pulse in pre-hospital Vs one location in hospital ED
Scaling	Different size attributes	25 Vs 50 chars for patient name
Accuracy	Different measurement units	Seconds Vs minutes
Incompatible coding	Different Codes	ICD 9 Vs SNOMED

6.5 Web Based Cardiac Surgery Electronic Medical Record

The Division of Cardiac Surgery at Brigham and Women's Hospital (a Harvard Medical School teaching affiliate) has developed a web based electronic medical record, primarily in response to state reporting requirements. The application is integrated with a clinical database to allow automated capture of data elements from other clinical applications. Free text data entry is minimized and clinical information is therefore entered in a structured way. (See Figures 4 - 8).

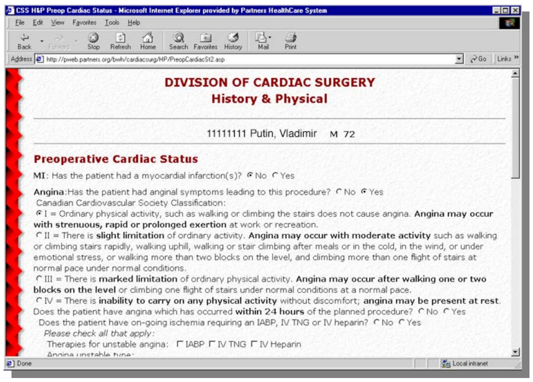

Figure 4 – Web Based Data Entry Form

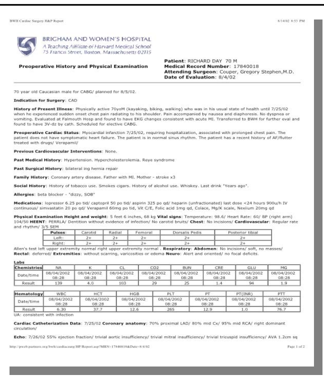

Figure 5 – Preoperative History and Physical Examination

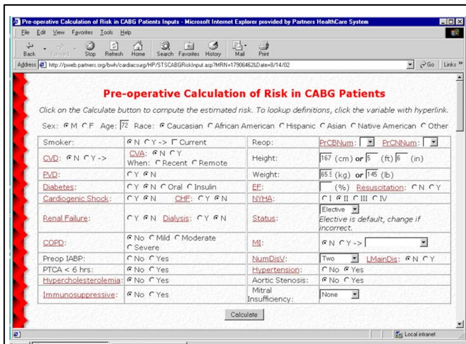

Figure 6 – Pre-operative Risk Calculation

The capture of clinical data elements at the point of care enables consistent, comprehensive, and legible documentation. Structuring data entry and forcing the capture of specific data elements insures that key data points for treatment and diagnosis, reporting requirements, quality improvement and clinical research are not lost. Because the application captures the data elements of the Society for Thoracic Surgeons Adult Cardiac Surgery Database, real time estimates of individual and aggregate mortality and morbidity can be calculated.

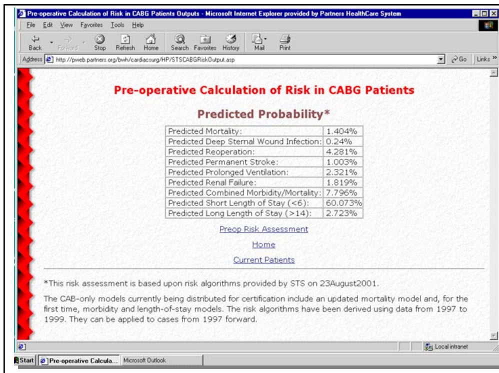

Figure 7 – Predicted Risk Probability

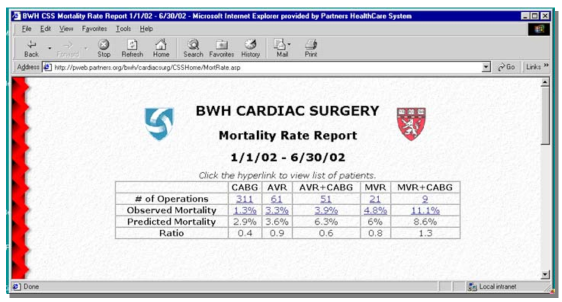

Figure 8 – Mortality Rate Report

One of the strengths of an EMR that uses structured text for data capture is the ability to use the same information in different clinical contexts. One example is a preoperative checklist generated automatically from information captured in the History and Physical and from web services.

Partners Healthcare has invested significant resources in the development of web services for clinical applications. The BCSEMR makes extensive use of XML calls to hospital clinical information systems. Services utilized include user authentication, patient demographic information, laboratory values, medications ordered, blood products issued, preadmission medication lists, operating room schedule information, coronary angiography results, and patient location. These services facilitate patient care by making information available in contexts where they are most relevant.

Because BCSEMR is compliant with web standards, it can run on multiple platforms without modification, including portable devices and computers utilizing a variety of operating systems and browsers.

6.6 Application Summary

Table 3 summarizes the interoperability aspects of iRevive and BCSEMR

Table 3 – Application Interoperability		
	iRevive	BCSEMR
Platform	No	Yes
Modular	Yes	Yes
Data Exchange	Yes	Limited

7. Conclusion

This paper discusses several types of interoperability (platform, modular, data exchange) using two medical applications as illustrative examples. Several lessons learned from Internet standardization are presented to emphasize the importance of developing and embracing health information standards. Our future work includes: developing a framework to evaluate interoperability; expanding lessons from the Internet to develop a set of recommendations to HITSP; along with more rigorous evaluations of iRevive and BCSEMR in the context of interoperability.

8. References

1. Abiteboul, S., Cluet, S. and Milo, T. (2002) “Correspondence and translation for heterogeneous data” Theoretical Computer Science, 275.
2. Baird S, Dawson-Haggerty S, Myung D, Gaynor M, Welsh M, Moulton S. Communicating data from wireless sensor networks using the HL7v3 standard. In: Proceedings of the 2006 IEEE International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2006).
3. Baldwin, C. and Clark, K. (1999). Design Rules: The Power of Modularity. Cambridge, MA: MIT Press.
4. DEEDS, “Data Elements for Emergency Department Systems”, http://www.cdc.gov/ncipc/pub-res/deedspage.htm
5. Bradner, S., "IETF Working Group Guidelines and Procedures", BCP 25, RFC 2418 (www.ietf.org), September 1998.
6. Gaynor, M., Network Service Investment Guild: Maximizing ROI in uncertainty markets. Wiley, 2003.
7. Gaynor, M., Bradner, S. A Real Options Metric to Evaluate Network , Protocol, and Service Architecture, Computer Communication Review(CCR), Oct 2004
8. Gaynor, M., Iver, Bala, Wyner, George, and Freedman, Jim, Web Services Tutorial AMCIS 2003 in Tampa, FloridaXML "www.xml.org," 2002.
9. 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.
10. Lesh, Kathy, Weininger, Sandy, Goldman, Julian, Wilson, Bob, and Himes, Glenn, Medical Device Interoperability, Joint workshop on high confidence medical devices, software, and systems, (HCMDSS) and medical device plug-and play (MD PnP) interoperability, Cambridge, MA., June 25, 2007
11. Gaynor M, Myung D, Ganesan S, Moulton S. An intelligent pre-hospital patient care system. International Journal of Electronic Healthcare, 2007.
12. Graham, S., et al. Building Web Services with Java: Making Sense of XML, SOAP, WSDL and UDDI, Sams, Indianapolis, 2002.
13. Gupta, A. (ed.) (1989) "Integration of Information Systems: Bridging Heterogeneous Databases" IEEE
14. Haas, L., Miller, R., Niswonger, B., Roth, M., and Wimmers E. (1997) “Transforming Heterogeneous data with database middleware: Beyond Integration” IEEE Data Engineering Bulletin
15. HL7, Health Level Seven. 1997 - 2005 Health Level Seven, Inc. http://www.hl7.org/
16. ICD-9, http://www.cdc.gov/nchs/icd9.htm, 2007.
17. Katz, M. and Shapior, C. Network Externalities, Competition and Compatibility, American Economic Review 75:3:424-440.
18. Kramer, Andrew, Bennett, Rachael, …, Case Studies of Electronic Health Records in Post-Acute and Long-Term Care, U.S. Department of Health and Human Services, Aug. 18, 2004
19. NEMSIS, National EMS Information System “NHTSA Version 2.2 Uniform Pre-hospital Dataset” http://www.nemsis.org.
20. Seltzer, J, and Reed, and Clark, D. 1984, “End-To-End Arguments in System Design.” ACM Transactions in Computer Systems 2, 4 , Nov: 1984.
21. SNOMED, http://www.snomed.org/, 2007
22. SOAP, The World Wide Web Consortium. SOAP Version 1.2 Part 1: Messaging Framework. W3C Recommendation 24 June 2003. . http://www.w3.org/TR/SOAP
23. R, Sokolowski, and Dudeck, J XML and its impact on content and structure in electronic health care documents, Proc AMIA Symp, 1999.
24. Standord, Jean and Thornton, Pamela, Electronic Health Records Overview, MITRE technical report.
25. Walker, J., Pan, E., Johnston, D., Adler-Milstein, J., Bates, D. W. and Middleton, B. (2005) "The Value Of Health Care Information Exchange And Interoperability." Health Affairs.
26. WSDL, The World Wide Web Consortium. “Web Services Description Language (WSDL) 1.1.” W3C Note 15 March 2001. http://www.w3.org/TR/wsdl
27. Wiederhold, G. (1992) "Mediators in the architecture of future information systems" IEEE Computer, 25, pp38-49
28. XML, The World Wide Web Consortium. “Extensible Markup Language” (XML). http://www.w3.org/XML, 2007.