The Brave New World of Systems Interoperability in Healthcare

By Tom Leonidas, Jr., P.E.

The Electronic Medical Record (EMR) creates a positive impact on quality of care by creating a single accessible and credible source for the patient’s historical record. One of the next ‘big things’ in healthcare delivery is using the EMR platform to create automation of routine tasks that occur in a hospital with a goal of streamlining the workflows that support patient care. Imagine a physician entering a prescription order at the patient bedside through an Integrated Bedside Terminal that is automatically dispensed in the hospital pharmacy. Then a Automated Guided Vehicle (AGV) is automatically notified and picks up the patient medications, captures the elevator and then delivers the medications to the nursing station on the patient floor; or the patient’s room. Far-fetched? Not really. There are leading edge hospitals that are either looking at or designing this kind of technology because they see the benefit in terms of freeing up caregivers and creating efficiencies in workflow.

The EMR acts as the initiator of these actions, but it does not physically or logically integrate all of the systems together that are required to execute the desired automation. Something must act as the ‘glue’ to connect individual systems together, pass information between systems and then confirm that that the desired instructions have been executed. In other words, the systems must ‘interoperate’ with each other.

Interoperability is the ability of one system to use the information or parts of another system and vice versa to work in concert with each other to perform a certain function or act. In essence it creates leverage between information systems and traditional building systems. Interoperability and integration of building systems with applications such as the electronic medical record are starting to gain a lot of attention as ways to automate routine tasks, improve workflows and increase the quality of clinical outcomes. We hear about portals, smart rooms and context aware environments where events are triggered by...
user input, tracking systems and the like. Interoperability has strong potential to do all of these things but this is a brave new world in the area of healthcare technology; and without careful thought and planning, comes the potential for expensive pitfalls that can take the benefits of systems interoperability off course.

What are the benefits?

Simply stated, the benefits of linking building systems, communication systems and information systems together is to improve clinical workflows while increasing the quality of the workflow. Increasing quality of the workflow and quality of care implies reduction in operating costs. With the average earnings for US hospitals at razor edge levels and the challenge of providing care at declining Medicare reimbursement levels, every tool in the toolbox counts toward creating and maintaining financial health. While the financial and business side of interoperability gives tangible monetary reward, the equal or greater than equal intangible benefits include reduced errors, creating greater access to healthcare for patients, and the ability to provide dynamic research and feedback into processes and patient buying methods. This feedback can be used to optimize clinical care as well as enhancing both the patient and caregiver experience.

What are the possibilities?

The possibilities are endless and crafting interoperability use cases can be likened to being a kid in a candy store. There are two broad categories of interoperability approach: system-to-system and application driven interoperability.

The most basic and traditional uses of interoperability are system-to-system and are for sharing data between systems and applications so as to avoid duplicating entry of the same data. Examples of this include:

- Integration with nurse call and communication systems
- Integration of access controls with the building automation system
- Integration of the access control system with the timekeeping and staff scheduling system
- Integration of medical equipment alarms with pagers, nurse call and other communication systems

Each of the above potentials is what we define as ‘use cases’ that define or map to a certain clinical workflow within the circle of patient care. The more basic and traditional use cases noted above are viewed as closed systems or system-
to-system integration that typically use proprietary methodologies to connect systems.

More sophisticated and application driven interoperability use cases have the highest potential for making impacts on healthcare delivery. Possibilities are unique to each healthcare provider but some of the use cases can include:

- Integrating patient registration with way finding to direct patients to their destination
- Automated distribution of patient lab test results to caregivers via smartphone, text message
- Automated patient monitoring alerts sent to caregivers through predefined threshold settings via medical equipment or by the electronic healthcare record
- Automated communications of clinical orders via a unified communications systems
- Automated dispensing and physical delivery of medication via automated guided vehicles (robots) from an initial input via the computerized physician order entry (CPOE) system
- Schedule based building controls based on patient procedure schedules defined by the electronic medical record system; for instance, turning on the surgery HVAC in each operating room based on the surgery schedule of the day.
- Patient bedside initiation of a video call via the nurse call system to the nurse’s mobile handheld device
- Patient initiated control of their room environment such as room temperature via through bedside controls
- Menu based educational and on-demand content via web portals or patient bedside controls

Integration of home medical equipment monitoring with the electronic medical record to monitor and alert the caregiver of a variation in acuity level of the patient condition

Application driven interoperability is more broad based and is used to create more of an enterprise networked interoperability among many systems; not just a few system-to-system cases. Application driven interoperability allows for building a broad and open platform that will more readily adapt and accept future technologies that may desired to be integrated for interoperability.
What are the pitfalls?

The biggest pitfall is not having a clear focus or business case of why you want to create interoperable systems or what it is that you wish to accomplish by doing so. The second biggest pitfall is that once you do know the why and the business case, is not having a well thought out plan or standards developed. By not figuring out these two core and key issues, you will spend a lot of capital dollars, put in systems that may or may not fit your needs and in the end will not accomplish your goals. You may also in the process create such a complicated system that it creates a large long-term cost of ownership if you are dealing with systems that don’t seamlessly integrate together; or if you create a system that uses proprietary methodologies for integration.

First, understand what you want to accomplish and why; then create the business case that shows it bring a positive return on investment. After you clear these steps, then set out to create standards for an open source platform and a strategy of how systems will work together to accommodate the use cases that you can define now and ones that you will define in the future.

What methodologies are available?

If the goal is to create a small scope of connected systems to accomplish basic information sharing and some event driven actions between a small number of systems, then it may be more cost effective to create system-to-system direct interfaces through the use of a common communication protocol such as IEEE HL7 interface or through specific middleware that works to connect specific systems.

If the desire is to create an open architecture platform that has the ability to share data between many systems and create action-reaction events from applications such as the electronic healthcare record, then it is more beneficial to deploy a common information bus where each separate systems ‘plugs in’ to that bus and has the ability to talk to any other system on the information bus.

One proven technology to create a common information bus is what is termed an Enterprise Service Bus or ESB for short. The ESB is essentially a software based middleware that acts as a multi-lingual translator between systems. ESB
technology has been around for decades and is commonly used in the business application world. For example, ESB technology is what manages the millions of stock trades per second in the New York Stock Exchange.

In Figure 1 - ESB Based Interoperability, the ESB acts as the central integration hub between systems. The arrows between systems and the ESB are called Application Programming Interfaces (APIs) that act as adapters that allow each individual system to speak in a common language with the ESB. The APIs are protocol based and can be written as web services interface (HTTP), JAVA, SOAP, HL7 or a variety of other protocols or programming language. In the model shown in Figure 1, user authentication is provided by Active Directory.

The ESB provides a simplistic, single integrated approach and acts as an open source platform. This type of system is scalable in that as new systems are added in the future, one simply writes and API to connect it to the ESB. The planning phase of the interoperability deployment would define how each disparate system acts individually and in concert with others so that the ESB knows what information and actions need to be transferred between systems to accommodate specific use cases.

**Key Factors to Think About**

The ESB is not the only component of the system to think about. One must also decide how users are authenticated between systems and what permission levels are assigned to users for each system. The model shown in Figure 1 uses MS Active Directory as the core authentication tool. Other core components are the Unified Communications platform which provides messaging from event driven actions to users and vice versa. The building automation system and materials management systems are also key systems if one desires to have event driven automation that utilizes or controls building systems.

It is important to look at each specific system whether it is existing or new, and discern its individual capabilities to understand the scope of how integration can be achieved within an overall interoperability ecosystem. It is also important to understand full architecture of the system and that it likely includes mirrored redundant server systems to achieve high availability of the interoperability platform.

Finally, the cost of the system both initial capital and long-term cost of ownership is important to understand. There is the initial capital cost for hardware, software for both a primary system and redundant system. There is initial cost to write application programming interfaces, refine use cases, as well as testing and commissioning of a fully functioning system. There are ongoing
recurring costs that involve licensing, hardware refresh that can range between 20% - 25% per year of the initial capital cost.

The overall goal is create as simplistic of a system as possible that provides an open source platform to easily adapt systems and technologies, and provide the flexibility to easily create and deploy use cases as they are defined and desired. Keeping things simple as well as choosing proven technologies and manufacturers will keep the long-term cost of ownership optimized.

**How should you proceed?**

- Define the business case or cases for integration - what is the benefit desired and how does it optimize the work flow and process of delivering healthcare?
- Define a strategy of approach, how it needs to scale in the future, and define a set of common standards
- Define individual use cases that enhance clinical workflows and map them out
- Refine individual use cases by testing them out with actual users; get feedback as to how to simply the integration and what actual information users need from systems
- Design the integration platform with as few individual components as possible; use proven technologies and manufacturers
- Define a budget and expectations that match the budget, as well as the long-term cost of ownership of the system
- Use an RFP process to retain the services of a Systems Integrator that understands both building systems and the clinical applications side of the hospital
- Use pilot installations to test out and refine use cases before full deployment

---

*Tom Leonidas is Vice President of Wood Harbinger, Inc. He has over 30 years of designing healthcare facilities, building systems, technologies and technology infrastructure, as well as providing integration and interoperability strategies to help clients optimize the delivery of healthcare.*