

# Architectural Choices and Challenges of Integrating Electronic Patient Questionnaires into the Electronic Medical Record to Support Patient-Centered Care

Barbara J Moore MD, Stephan Gaehde MD MPH, Clayton Curtis MD PhD  
VA Boston Healthcare System, Jamaica Plain, Massachusetts

## Abstract

*Strategies to deliver guideline-concordant, patient-centered care during office visits sometimes impose conflicting demands on clinicians. One way to help relieve time-constrained visits and to improve the patient-centeredness of care may be through patients electronically self-reporting data that flow automatically into an EMR note for clinician confirmation or editing, relieving physicians of some data entry and rote history-gathering tasks, thus freeing up time to allow clinicians to focus on significant issues and patient concerns while also increasing the likelihood that necessary data are gathered and available for decision-making.*

*We developed a prototype to enable such data gathering and integration into the EMR. The lack of consistent provision of interfaces by vendors for sending data into EMRs and the idiosyncrasies of any particular EMR in the context of a particular organization's IT infrastructure and policies pose architectural choices and challenges that healthcare organizations embarking on such IT projects may need to consider.*

## Background

Access to medical care is a major concern in the United States. Patient access requires both appointment availability and visits of sufficient time to accomplish visit goals. Yet in response to shrinking reimbursement for clinical services, providers get less time per patient as organizations attempt to insulate budgets from declining revenue. This produces visits that may fall short of addressing all aspects on patient and clinician agendas.<sup>1</sup> Simultaneously, mandates for reporting metrics, reducing error, fulfilling guidelines and satisfying patients are growing exponentially as the time available to accomplish these goals evaporates.<sup>2</sup> Because these forces often work in opposition, they may hamper healthcare delivery. To successfully

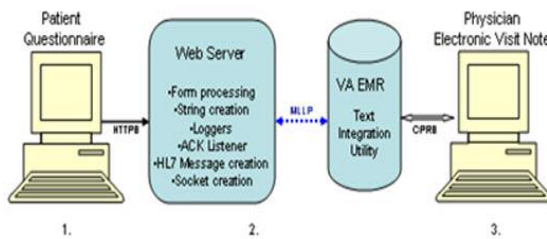
counterbalance them requires innovative solutions in visit management.

Time constraints and patient concerns often leave many basic preventive and chronic disease tasks incomplete at visit end.<sup>2,3</sup> Providers are not always familiar with chronic care guidelines and history gathering and documentation efforts do not always include the salient aspects needed to arrive at proper diagnosis and treatment or to enable the full benefits of clinical decision support systems.<sup>1,4,5</sup> Knowledge support and improvements to visit efficiency through the use of information technology that is designed and deployed with the immutable factors of the clinical care processes in a given organization in mind, and that is tested at the extremes of clinical circumstance where flaws in systems cascade and become most problematic, will be necessary to remedy these problems.

As evidence-based care merges with patient-centered and value-based care that is expected to improve quality, safety and satisfaction while also providing the best return on the dollars invested, requirements for visits and for information access evolve. Despite visit time constraints, the ability to provide patient-centered care requires understanding and honoring the patient's priorities. Studies show that patients are willing to use electronic questionnaires, but these studies often only provide output to paper.<sup>6,7</sup>

Integrating external data, including patient-entered data, into the EMR will help to bring the much-anticipated benefits of EMR use to fruition, e.g. decreasing redundant testing by providing access to test results from other facilities or by highlighting the patient's preferences and priorities. David Brailer noted that isolated, stand-alone EMRs may not improve patient care and represent a lost opportunity.<sup>8</sup> But integrating applications remains challenging because integration solutions are fragmented, continue to evolve over time and are inconsistently provided by EMR vendors in their EMR products.

We developed a proof-of-concept prototype (Figure 1) that will enable patients to become active members of the team by allowing them to enter their data electronically. In order to accomplish this, we chose from among three standards-based integration options available in our organization, balancing each option's trade-offs as no option was problem-free. This paper is a case report of our experience in the design phase. Lessons learned may help others, even in non-VA settings, as they weigh their options for choosing an integration strategy.



**Figure 1.** Overview of the Patient-entered Data Architecture. 1. Browser client for data entry, 2. Middleware, 3. EMR GUI pulls data into a clinician-editable progress note.

## Methods

We decided to use a Java-based solution because we had no MUMPS or Delphi programmers available to us (both languages were used to write the VA legacy system) and we were familiar with Java, a language acceptable for use within the VA. The three integration options we considered were HL7 messaging, web services and Java 2 Enterprise Edition Connector Architecture (JCA).

*N-tier Architecture.* Given our straightforward GUI design, we chose to use multi-tier architecture, as opposed to 2-tier client server architecture, to take advantage of multi-tier benefits, such as easier (hence less costly) maintainability that results from limiting the number of computers that need to have software installation, updates and patches and multi-tier's greater scalability (i.e. users accommodated) compared to client server solutions.

*HL7 Messaging.* We considered using HL7 messaging, a common medical-based standard used by Health Information Systems and healthcare devices. HL7 messaging would allow us to import our patient-entered data directly into

our target package within the VA EMR (i.e. the Text Integration Utility—TIU—EMR package), using the TIU local HL7 interface, which accepts HL7 messages. This HL7 interface would enable us to import data into an unsigned progress note. It primarily is used by transcription services and telemedicine. By including the Java open source HL7 API (HAPI) in our application, we would be able to place patient-gathered data obtained by a web-based JSP form into an HL7 message.

On the negative side, HAPI does not provide a way to encrypt the transmission of the HL7 message between the web-server and the EMR. If encryption is required, one must add additional code, because HAPI uses minimal lower layer protocol, MLLP, not internet protocol, and therefore HTTPS is not available to encrypt the transmission. MLLP also presents hurdles if your application needs to traverse firewalls.

Because VA's electronic medical record is an integrated system of many software packages, communicating with it is more complex than with EMR systems that allow easy addition to the database by way of observation terms. The VA EMR is constrained by its strict data modeling and multiple packages over which the components of the EMR are distributed (Vital signs, consultation, oncology, medicine, surgery, etc.), only a few of which currently have HL7 interfaces. Furthermore, the VA TIU HL7 interface was not written to send outbound messages, other than acknowledgements, back to applications. Although this is not a problem for our prototype, which is designed to allow manual entry of patient and provider data required to create the HL7 MDM T02 message, it would be problematic for a production system, which would need functionality to look up on the fly patient visit data and provider name and number. By adding additional MUMPS code, the local HL7 interface could be modified to access other EMR packages to obtain necessary data and to send outbound messages from the EMR to our application. Given that MUMPS programming was not available to us, an alternative to the patient and physician look-up conundrum would be to make SQL calls from our web application to the National Data Repository, which is a relational database (unlike the MUMPS-based hierarchical legacy EMR) using an appropriate database driver and the Java SQL package or JSTL SQL library. These SQL calls would occur prior to the creation of the HL7 inbound message thus creating a hybrid application. Another

solution, also a hybrid, would rely on VA web services to acquire patient visit data and provider name and number prior to creating the inbound HL7 message. Finally, one could use a JCA adapter to look up and return necessary data prior to HL7 message creation.

*Web Services.* Another solution would be to use a purely web service-based integration strategy for both data lookup and for importing patient-entered data, instead of creating a hybrid solution. VA web services, a.k.a. VistaWeb Services, would enable straightforward coding to create the web service client application, because the service interface handles details such as field names, data types and business rules for the client-side developer who relies on details provided by the web service description language (WSDL) document.<sup>9</sup> VA web services use plain old java objects, rather than the more complex Enterprise Java Beans used by JCA. Web services at VA allow applications to access data as proxy users (provided that the remote procedure call involved is written to allow proxy users, as VistaWeb services use the legacy EMR's RPCs). VA web services can provide access to data in different EMR packages and can provide 2-way encryption using the VA EMR encryption algorithm. Because web services use internet protocols, web services are firewall-friendly, making firewall traversal a straightforward process.

Against using web services was the fact that none of the existing VA web service interfaces at the time could pull the necessary patient visit data along with the provider name and number (and co-signer data, if needed), let alone submit data into an unsigned EMR progress note. Furthermore, none of the VistaWeb service interfaces at the time had "write" functionality. Thus the benefits of web service client coding would be negated by the complexity of writing the service interface, which would require the developer to determine field names, data types, appropriate RPCs and necessary business rules. Also, with web services, you must parse the contents of the web service SOAP message, which is XML-based and thus can be a relatively slow process. All three integration strategies that we considered require optimizing network, server, application and database performance, while web services also requires attention to optimization of the XML parser in order to minimize user frustration from slow response time.

*J2EE Connector Architecture (JCA).* Finally, we considered the use of a purely JCA solution (a.k.a. VistaLink).<sup>10</sup> This is a Java standards-based integration solution that enables scalable, secure, transactional means to communicate between J2EE applications and legacy systems of various types (main frame, database, etc.) in order to work with legacy data. The VistaLink adapter enables Java applications to communicate with the VA legacy EMR. As with VistaWeb services, VistaLink uses the VA EMR's remote procedure calls to access data in the underlying MUMPS database. VistaLink has both "read" and "write" functionality. Similar to web services, VistaLink uses the VA EMR 2-way encryption algorithm and can pull data from different EMR packages. Against the use of VistaLink is the fact that coding is more complex due to its use of Enterprise Java Beans and the developer's need to locate field names, data types, appropriate RPCs and to assure compliance with VA business rules. JCA uses TCP/IP sockets rather than HTTP, so, similar to HL7 messaging, JCA is less firewall-friendly than web services.

Ultimately, we chose the HL7 messaging solution for this proof-of-concept prototype because of its relative ease of coding and the ready availability of the TIU HL7 interface. We realize that a production grade system would require additional MUMPS coding or a hybrid version in order to include real-time data look-up and patient authentication.

## Results

A web-based patient questionnaire was developed with Java Server Pages (JSP), Servlets and the Java Standard Tag Library (JSTL) using NetBeans IDE then deployed to a Tomcat web server. The application submits form data to the Tomcat web server via post method after successful JavaScript form validation. Next, a servlet on the web server processes the patient-entered form data to create a string that is passed to HAPI. Additional data required to create the HL7 message is also processed by the servlet from the JSP form data, including patient social security number, gender, date of birth, physician author information and, if necessary, physician cosigner information. Then HAPI creates a direct MLLP socket connection from Tomcat to the VA EMR HL7 interface. Assuming no problems occur, the data is immediately available in the medical record's unsigned progress note. The

clinician can corroborate the information with the patient and make appropriate edits during the visit.

The servlet creates an acknowledgement listener that awaits the return message from the VA HL7 interface indicating that the progress note creation was either a success or a failure.

Our first iteration application created an electronic version of the 12-years-and-older Asthma Control Test (ACT), a 5-item asthma survey (Glaxo Smith Kline). We placed a summary statement into an unsigned progress note that included ACT score and the response to every item that made up that score. That brief instrument provided a means to easily test and debug the steps required to collect, process and submit patient-entered data into an unsigned progress note. The next questionnaire developed gathers data from COPD patients with the objective of helping clinicians deliver patient-centered, COPD guideline-concordant care (questions include MMRC dyspnea score assessment, indicators of early exacerbation, indicators of hypoxemia, respiratory-related mealtime and sleep difficulties, GERD, etc.).

### Discussion

*Integration, Patient-entered Data and the EMR .* Some EMR vendors (e.g. eClinicalWorks and NextGen) and health information exchange platform and portal vendors (e.g. iHistory, MedSeek) enable integration of patient-supplied data, either via add-on software purchased from independent software vendors (Instant Medical History and HealthQuiz) or via portals.<sup>11,12</sup> EMR vendors do not include integration of patient-entered data out of the box in their products. Moreover, when EMR products use proprietary non-standards-based integration interfaces they increase the difficulty of communicating with software from other vendors. Selling out of the box EMR products that include standards-based, integration-ready interfaces would help to overcome the barriers to integration, including barriers to integrating patient-entered data.

Computerized patient-entered histories have been used in oncology clinics to check for chemotherapy toxicity and gather quality of life data, in rheumatology clinics for quality of life and functional assessments, for medication reconciliation and to update past medical history and problem and allergy lists, for pre-anesthesia assessments, to collect family history and to

perform US Preventive Services Task Force screening.<sup>11,13,14,15,16</sup> Branching logic enables computerized patient interviews to tailor questions specific to the patient as that interview unfolds. These reports indicate that the majority of patients are willing and able to supply history via computer.

*Patient Authentication.* Patient authentication would be required in a production environment to assure the validity of the data originator. Currently, EMRs do not include patients in their user authentication systems. Patient portals tend to have patients register for portal use during a face-to-face encounter and thereafter rely on two-factor authentication for patient login to the portal. Data regarding the patient's authentication credentials would need to be stored and accessed in real-time in a production system. This would require outbound messaging capability, raising the same limitations within VA as discussed in the Methods section.

One workaround should patient authentication and credentialing not be available is to create a program to assign a single-use, globally unique PIN as a session identifier dispensed by a clinic secretary that would point to the correct patient record and visit in the database for a particular clinic appointment. The limited-lifespan session identifier would need to be available, e.g. for 30 minutes after its creation, and would likely use pointers to the patient visit data of interest within the EMR database. Two potential problems with this workaround are that some facilities may not have sufficient staff to add tasks to their clinic staff's responsibilities and the solution still needs to store the single-use PIN.

*Dynamic Routing.* Similar to the VA, other organizations may use dynamic IP addressing to adjust to changes in network and client configuration. In such organizations, when gathering patient data using an HL7 interface or an interface engine that was not designed to deal with dynamic routing, it's necessary to communicate between two computers with fixed IP addresses (servers) in order to guarantee message delivery. Our first iteration of the application was a standalone Java Swing application (client-server) with the client on a laptop, while our next iteration was web-based and ran from a Tomcat server deployed locally on a laptop. When using a laptop or tablet to gather data from patients without an intervening server as sender, dynamic routing of network

traffic means that message delivery can't be guaranteed.

The TIU package's local HL7 interface does not have capability to handle dynamic routing, thus the web server needs a fixed IP address.

*Combining Web Services with JCA or HL7 messaging.* If traversing a firewall is necessary, as when obtaining radiology or laboratory results from an outside organization, you may need to combine web services with HL7 messaging or with JCA, especially when using non-standards-based or complex data types.

*Portal Migration.* Creating the solution as a Java web-based application enables migration to a patient portal, adding value. Pre-visit check-in and data-gathering could occur in the privacy and leisure of one's home prior to the visit. This remote solution may help patients be more accurate partners in the medication reconciliation process and more involved in other aspects of their care.

*Next Steps.* Next, we will evaluate application feasibility, usability and acceptability from the patient's and provider's perspective, as well as gather preliminary data to help estimate effect size for future studies on the application's impact on patient care.

### Conclusion

Any healthcare organization attempting to send patient self-reported data into an EMR may struggle with similar issues. Also, the existence of an HL7 interface does not guarantee the presence of all necessary functionality in that interface. Use of complex or non-standard data types may limit one's integration options.

### Acknowledgement

This work was supported by the VA Special Fellowship in Medical Informatics.

### References

1. Parchman ML, Pugh JA, Romero RL, Bowers KA. Competing demands or clinical inertia: the case of elevated glycosylated hemoglobin. *Ann Fam Med* (2007) 5: 196-201.
2. Yarnall K, Pollak KI, Ostby T, Krause KM, Michener JL. Primary care: is there enough time for prevention? *Am J Public Health* 2003; 93: 635-641.
3. Otsby T et al. Is there time for management of patients with chronic disease in primary care? *Ann Fam Med* 2005;3:209-214.
4. Ramsey PG, Curtis JR, Paauw DS, Carline JD, Wenrich MD. History-taking and preventive medicine skills among primary care physicians: an assessment using standardized patients. *Am J Med* 1998; 104: 152-158
5. Berner ES, Kasiraman RK, Yu F, Ray MN, Houston TK. Data quality in the outpatient setting: impact on clinical decisions support systems. <http://www.amia.org/meetings/f08/docs/paperexample.pdf> Access on March 11, 2008
6. Smith PD and Grasmick M. Computer interviewing in primary care: the patients are ready. *Medinfo* (2004) 11:1162-5.
7. Mark TL, Fortner B, Johnson G. Evaluation of a tablet PC technology to screen and educate oncology patients. *Support Care Cancer*. 2007 Aug 18 (Epub ahead of print).
8. Brailer DJ. Interoperability: the key to the future of the health care system. *Health Aff* (2005) Jan-Jun Suppl w5-19 – w5-21.
9. Gillon J. VistaWeb Services. <http://trac.medora.va.gov/wiki/AboutMedora/WebServices>. Accessed on March 11, 2008.
10. VistaLink 1.5 developer guide. May 2006. VHA Software Document Library.
11. Bachman J. Improving care with automated patient history. *Family Practice Management*. July/August 2007: 39-43.
12. Benoit A et al. Using electronic questionnaires to collect patient reported history. *AMIA 2007 Symposium Proceedings* p. 871.
13. Velikova G et al. Automated collection of Life Data: a comparison of paper and computer touch screen questionnaires. *J Clin Oncol*. 1999; 17:998-1007.
14. Basch E et al. *J Clin Oncol*. 2005; 23:3552-3561.
15. Kantor G and Gage J. Computer-patient interviewing: can it assist in pre-operative evaluation? *ASA Newsletter* Nov. 2004; accessed at [http://www.asahq.org/Newsletters/2004/11\\_04/kantor.html](http://www.asahq.org/Newsletters/2004/11_04/kantor.html), on 6/30/2008.
16. Williams CA et al. Usability of a computer-assisted interview system for unaided self-entry of patient data in an urban Rheumatology clinic. *J Am Inform Assoc* 2004; 11:249-259.