

# HL7 NLM EHR Phase I Final Report

Background .....	2
Existing message sets reviewed.....	3
The requirements .....	3
Subcontract approach .....	4
Rational for choice of final message set .....	5
Functionality excluded for expediency but which should be included in future implementations .....	5
Development tools used .....	6
Description of all artifact files .....	6
Test data set used.....	6
Method of populating and parsing message .....	6
Overall design:.....	6
Primary components: (See Detailed System Recommendations, below) .....	6
System functional description .....	7
Detailed system Requirements .....	7
HL7 V2 and V3 CDA Transformation Issues .....	8
Possible Local Configuration Issues.....	8

## Background

This application, produced as part of the Health Level Seven (HL7) National Library of Medicine (NLM) contract, is a simple demonstration of the application of HL7 Version 3 messages and documents to the problem of information exchange in a distributed, inter-enterprise context with minimal pre-negotiation. It is not intended as a production-ready application, but utilizes security, messaging and document standards that could form the basis for such an implementation. It is the intent of HL7 to extend this work under the NLM contract. Please see the HL7 NLM web page (<http://www.hl7.org/nlmcontract>) for an update on current activities and links to executable files and sample data.

The goal of this 2 year contract is to provide Government Agencies and the private sector with an implementation guide for the transmission of patient information between disparate electronic health record (EHR) systems.

This will be done in stages by leveraging existing work within HL7 to increase the usability of messaging standards. To make solutions available earlier, this project will take an iterative approach where each phase produces a usable working product. Successive phases will refine what is learned in the previous phase.

The EHR project will have at least 2 phases. Phase I will produce a solution that will allow implementation of information flow between EHR systems using existing formats in a standardized query-response message set. It will include an implementation of a secure transmission mechanism encompassing encryption, authentication, and transport verification. Phase II will be an expansion of Phase I with messages that enable more semantic interoperability and a richer query set.

**Phase I** will implement a simple query-response message set executed between two EHR systems in the following conversational format:

1. A query from an authorized requestor to a record holder asks what information could be provided on a specific patient within supplied parameters.
2. The response provides a high level description or index of the information on that specific patient that could be transmitted in response to the request.
3. A follow-up query requests all or a specified subset (by type and date range, for example) of the specified patient's information.
4. The response contains a simplified but standardized information retrieval message that has one or more CDA documents as payload. The CDA documents, in turn, contain human-readable clinical records plus optional source data encoded in HL7 formats.

This conversation takes place within a secure transmission mechanism that handles encryption, segmentation, verification, and transmission over the internet. Mechanisms to move source data into and out of these messages will also be provided.

### Requirements

The system components will be assembled from existing open source implementations, the initial components of which are:

- Transport/encryption/data integrity - candidate CDC PHIN-MS
- Phase I EHR Query-Response Message Set- developed within NLM project by modifying existing messages

Note that this Phase I contract fits within the context of the overall project described in the HL7 EHR Interchange Standard-Overview.doc.

## Existing message sets reviewed

The Phase I subcontracts involved putting together and testing a set of standard HL7 messages to enable one EHR system to transfer existing information on a specific patient to another EHR system with variable semantic interoperability. The payload was to contain, at a minimum, a human readable version of the information provided by the sender. Where available, sender may provide HL7-defined structures (e.g., V2 or V3 messages, CDA Release 1.0 with CDA entries, or simple CDA Release 2.0 documents – equivalent in function and complexity to Release 1.0, but updated to the new release).

## The requirements

The defined transactions must:

- 1) be immediately usable.
- 2) allow the movement of existing EHR data between two systems using a:
  - a) Phase I: “meta” transaction containing a set of whatever HL7 transactions (messages, documents) the sending system can provide.
  - b) Phase II: HL7 V3.x transaction based on a standard information model for an important subset of EHR information.
- 3) facilitate progress toward complete semantic interoperability, e.g. by supporting unstructured data transfers while providing a framework for addition of and evolution towards fully structured data transfer (must fit into larger and ongoing solutions).
- 4) demonstrate its feasibility by at least one implementation for each phase.
- 5) include, at a minimum, allowable requests, responses, and formats for each phase.

The initial stage of the Electronic Health Record Pilot project involves a working prototype to send and receive messages containing individual patient information without pre-negotiation on the part of participant systems. Subsequent stages will expand the working prototype both in depth and in scope.

## V3 messages to be considered:

Transport Specifications:

ebXML, Release 1 DSTU - Pending Board Approval

Webservices SOAP/WSDL Profile, Release 1 DSTU - Pending Board Approval

MLLP, Release 1 ( Membership #1 )

Common Domains: Shared Messages:

Act Status Topic

Act Reference Topic

Infrastructure Management: Transmission Infrastructure:

Generic Message Transmission

Polling Message Transmission

Infrastructure Management: Query Infrastructure:

Query Control Act Topic

Infrastructure Management: Master File/Registry Infrastructure:

Master File Registry Topic

Health and Clinical Management:

Clinical Document Architecture

Laboratory: Result Topic

Medical Records: Document Topic

Public Health Reporting: ICSR Topic

other, as desired (eg, Pharmacy, Blood Bank and so on)

The HL7 NLM contract leadership awarded one subcontract to achieve two Request for Quotes (RFQs) in order to reduce resources needed overall to reach a satisfactory point of conclusion for Phase I. These RFQs are a) "Install Message Transport Layer Contract" and b). "Develop Phase I EHR Query-Response Message Set"

### **Subcontract approach**

1. Develop set of heterogeneous sample source data
2. Configure "Application Data Component" (see PHIN [2], p.2) housing sample data.
3. Write "Message Transformation Component" (see [2], p.2) required to transform sample data into minimal CDA, per 2).
4. Install PHIN send/receive components and interface with 2), 3), above.
5. Receive set of 4 Version 3 query/response messages described in from project sub-contractor.
6. Test various query/retrieval scripts exercising message design, PHIN implementation, database retrieval and transformation.
7. Report conclusions, per statement of work.

The reason for this approach is that the functions of repository configuration, interface and transformation are closely bound with the PHIN installation. In fact, it would have been difficult to coordinate and to accomplish this if the PHIN component were not managed jointly with the repository. Regarding the transforms, they are a routine function of implementation and add nothing to design of the query/response specifications.

This implementation allows a general case of any file with a known MIME type to be transmitted using a message with a Release 1 CDA containing the metadata about the file (entered separately, not automatically extracted from the file) and a link to the file that allows the file to be opened and processed by the receiver, assuming that the receiver has the appropriate software application installed. In a previous release of this application, when transmitting an HL7 Version 2 or Version 3 encoding of clinical data, the original data file was converted (by specific conversion processes) to a CDA for transmission but the original HL7 message file was not transmitted. This would have required the receiver to process the CDA to get the coded data. Although this is logical and the direction we are moving with Phase II, one of the points of the Phase I approach was that the sender and receiver had to change minimally. To support that principle, the original HL7 message file is now being passed, and linked to, just like the other source files

so that the receiver could use their existing V2 or V3 processing mechanisms to use the raw data, if they so desired, rather than forcing them to write a CDA processing module. It is noted, however, that the CDAs produced this way may not meet the letter of the definition of CDA.

## **Rational for choice of final message set**

Rene Spronk and Mike Henderson (HL7 Control Query co-chairs), Wayne Tracy (HL7 Medical Records co-chair), and HL7 volunteers were engaged to assist with the messaging standards and sample message instances. They explored the query-response capabilities of existing HL7 messages and interactions with the chairs of the appropriate HL7 Technical Committees (TCs) and determined the best set of existing HL7 message on which to base the messages needed for this project.

The dataflow requires a four message process to retrieve a document:

- 1) Client sends query to the Server for a list of documents matching a specified criterion. (QUMT\_IN020010)
- 2) Server responds with a list of documents that satisfy the query parameters. (QUMT\_IN020020)
- 3) Client selects document from the list and sends retrieve to the Server. (RCMR\_IN000030T<sup>1</sup>)
- 4) Server returns the specified document. (RCMR\_IN000031T<sup>2</sup>)

Message 1 and 2 are balloted in the HL7 V3 Query Infrastructure. Message 3 and 4 are balloted in the Clinical Domains section of V3, Medical Records. The “T” at the end of the message designation indicates that these are Test/Temporary messages that are not a standard yet. They are a domain specific Medical Records query and response that conform to a proposal made to Medical Records by HL7 Netherlands to extend the domain to manage this use case. See

[http://www.hl7.org/Library/Committees/medrec/20040823\\_prop\\_910.zip](http://www.hl7.org/Library/Committees/medrec/20040823_prop_910.zip)

The prototype form of each message is stored in the database and can be modified by the database administrator if needed.

## **Functionality excluded for expediency but which should be included in future implementations**

Appropriate security, confidentiality, and access control is subject to the rules governing the particular organizations and the agreements between them, and is outside the scope of this specification.

Also outside the scope of this proof-of-concept is the creation and/or maintenance of any type of Master Patient Index.

Unambiguous patient identification, document identification including origin of information will be addressed in Phase II.

This implementation includes viewing the HL7 V2 and V3 sources but the inclusion of those messages into existing HL7 integration environment is left as an implementation issue.

---

<sup>1</sup> As of the time of this publication, the Medical Records Committee has not approved this interaction but has taken on a work item to address this issue

<sup>2</sup> As of the time of this publication, the Medical Records Committee has not approved this interaction but has taken on a work item to address this issue

## Development tools used

- Jakarta Tomcat v4.04
- Java JRE v1.4 or better
- MS SQL Server Database with a JDBC connector

## Description of all artifact files

See Appendix 1 of the Installation and Implementation Guide<sup>3</sup>

## Test data set used

See Appendix 2 of the Installation and Implementation Guide<sup>3</sup>

## Method of populating and parsing message

### Overall design:

- Client: requestor, low-end, small office, low-cost, simple, web-enabled
- Server: data source, high-end, large facility, more sophisticated technically
- Client can request, receive documents; Server can respond to query, supply documents
- Server backend database has mix of data sources including non-CDA reports, CDA, V2 and V3 lab results

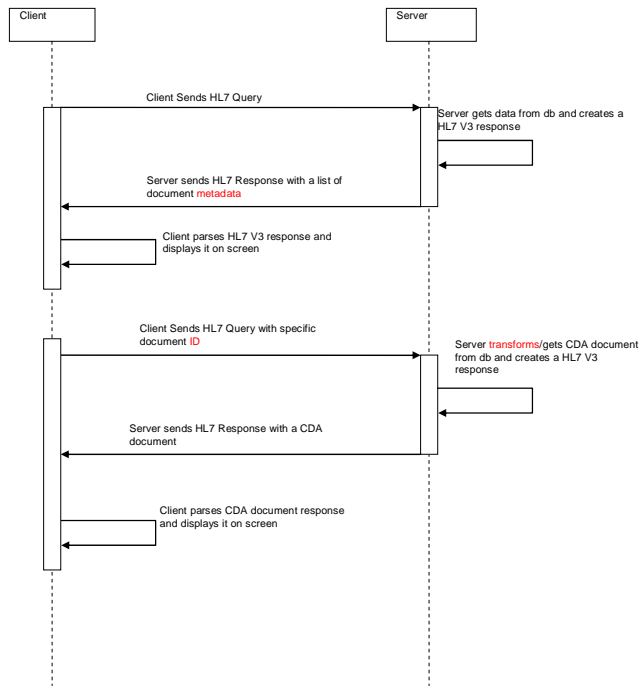
### Primary components: (See Detailed System Recommendations, below)

1. CDC's Version 2.1 messaging system
2. MS SQL Server database development license (server must supply own license)
3. Version 3 query/response messages, to be developed using latest ballot (#8, if available)
4. Utility to wrap unstructured documents into CDA R1.
5. HTML web pages for query definition and viewing of retrieved documents
6. Style sheet for displaying CDA R1 documents.

---

<sup>3</sup> [http://www.hl7.org/documentcenter/public/nlmcontract/ehrfiles/PhaseI/HL7\\_EHR\\_Phase\\_I\\_Guide.pdf](http://www.hl7.org/documentcenter/public/nlmcontract/ehrfiles/PhaseI/HL7_EHR_Phase_I_Guide.pdf)

## System functional description



1. User (person) on the Client (requesting system): Invokes Send Query GUI, enters query parameters:
  - Patient name
  - Patient ID
  - Patient date of birth
  - Report type
  - Date range
2. Client completes the query message, packages it in PHINMS to be sent to Server (responding system).
3. Server unpacks query, searches for corresponding artifacts, formats response per message query parameters, and sends response to Client listing matching data.
4. Client receives response, unpacks message, and renders response as HTML list in browser.
5. User selects record to retrieve, sends request to Server.
6. Server unpacks request, identifies record to be retrieved, renders them in CDA R1, packages in message and sends back to Client.
7. One or more documents received, can be displayed by Client using CDA R1 style sheet on Client java enabled web browser.

## Detailed system Requirements

The Client system requires:

- Windows 2000/NT (or better) or Linux with 256MB.
- Java JDK 1.4 or better

The Server system requires:

- Jakarta Tomcat v4.04
- Java JRE v1.4 or better
- MS SQL Server Database with a JDBC connector

We suggest using a Windows XP / SP2 + java system for the client since this is representative of a typical client configuration. The Server will use Windows 2003 Server OS.

If secure (SSL) transmission will be used, the server will need to have a Certificate from a Certifying Agent that is known to the clients. If this is just for internal-only use, it can be self-certified.

### **HL7 V2 and V3 CDA Transformation Issues**

1. Document ID:
  - a. for every transform?
  - b. every source document?
2. Timestamp:
  - a. time of source report creation?
  - b. transmission?
  - c. time of transform?
3. Document type code: no LOINC scale=doc
4. Provider:
  - a. CDA assumes role in encounter
  - b. relationship to lab unclear
5. Referring physician: anticipates an encounter
6. Lab tech: no such role
7. Order status: no such status
8. Authentication: who is the authenticator?

### **Possible Local Configuration Issues**

Some of the issues that may complicate development and deployment on your systems, for which neither HL7 nor their subcontracts are responsible, may require local support or a time & materials arrangement (possibly with the subcontractor), include:

1. Integration with account management system. Whatever system your system uses for account management (usernames and passwords) has to be integrated both into the authorization to the machine and to the web server. This proof-of-concept does not authenticate users.
2. Integration with firewalls and other network configuration. All of these have to be seamlessly integrated with your environment.
3. Anti-virus software. This will have to be enabled on any MS Windows machine at least.
4. Backup system. The system must be integrated with backup systems and thoroughly tested
5. Redundant server. If you have redundant servers, then the database has to implement this according to your policies.