

Conformance Checking for HL7

Ensuring Messages are Understood by Healthcare Vendors Before and During Processing

Overview

Although the HL7 2.X messaging standard is the most widely used standard in the United States for the exchange of clinical patient data, it varies greatly in how it is implemented by each medical device and application. Consequently, it is often called the “non-standard standard”. The purpose of HL7 2.X is to provide a framework for negotiation so that each healthcare interface is closer to 20% custom rather than 100%.

The variances in implementation lead to different message formats among healthcare vendors and external providers. In order to communicate effectively between systems with different message formats, you must determine where the messages are incompatible and make changes to at least one, and potentially both, of the interfaces that are accepting or sending messages.

Gap analysis or conformance checking for HL7 messages, is a logical process used to determine whether a message from one particular medical device or application is compatible to the standard HL7 messaging format, or a custom format, adopted by another device or application.

Why Does Nonconformance Occur?

You may wonder why you need to perform a gap analysis or check for conformance between devices or applications that say they are HL7 compliant.

Nonconformance occurs for two main reasons:

1. An application team utilizes the flexibility of the HL7 2.X message standard to meet their system’s unique requirements. This occurs primarily in the areas of cardinality and the HL7 version that is used.
2. The HL7 messaging standard can be complex and sometimes is easily misinterpreted.

Cardinality

Cardinality represents the minimum and maximum number of values that can exist inside a given element of a message. The HL7 messaging standard allows cardinality, where elements can be either optional or required, and either singleton (non-repeating) or repeating.

Examples of HL7 message cardinality are shown in the following table.

Field or Segment	Rule	Cardinality
Patient address field (PID-11)	optional and repeating	0 to n
Patient primary language field (PID-15)	optional and non-repeating	0 to 1
Patient class field (PV1-2)	required and non-repeating	1 to 1
The Patient Identification (PID) segment in the Patient Admission message (ADT-A01)	required and non-repeating	1 to 1

How Does Cardinality Cause Nonconformance?

When looking at cardinality, it would seem to provide enough flexibility to be followed by all systems. However, this is an area that frequently causes nonconformance. The following example shows one way this can happen.

One development team (team 1) may implement the cardinality of the Patient Address Field (PID-11) exactly as the standard says. Therefore, they would allow zero to infinite patient addresses (cardinality of 0 to n).

Another development team (team 2), may *require* one patient address but *only allow one* (cardinality 1 to 1). Team 2, in this example, is not HL7 compliant.

If the application developed by team 1 sends a message to team 2's application without a patient address, the application from team 2 would not accept it, even though the message is HL7 compliant.

There are clinical healthcare systems that have misapplied cardinality rules and are not technically HL7 compliant. However, these systems are in use and actively trading HL7 messages with other systems.

HL7 Versions

HL7 2.X is designed to be backward compatible. As new segments and fields are added they are marked as optional, so that older systems communicating with newer systems do not need to contain those elements. In theory, the version of HL7 2.X does not matter. There are a few instances of where nonconformance occurs due to version differences. A few of these are shown in the following table:

Difference	Result
Version 2.1 and 2.2 contain an ORU-R03 message. Version 2.3 and beyond do not contain this message.	If a system using HL7 2.2 sends an ORU-R03 message to a system that uses HL7 2.3, the receiving system may not know what to do with the message and, accordingly, ignores it.
Version 2.4 contains an ADT-A60 message to update allergy information, but previous versions do not contain this message.	If a system using HL7 2.4 sends an ADT-A60 message to update allergy information to a system that uses an earlier HL7 version, the allergy information will not get updated.
Version 2.5 has the new ability to define cardinality for datatype components. For example, one of the components of a datatype can be required or can repeat.	If a system using HL7 2.4 sends a message to a 2.5 system that does not contain the 2.5 required datatype component, the HL7 2.5 system will not accept the message.

Nonconformance due to differing versions is most often caused by one system requiring a message element that did not exist in a previous version, even though HL7 defines it as optional.

Misinterpretation of Standard

Development teams are experts in their specialized area, such as a lab or pharmacy, but not necessarily in HL7 messaging. The HL7 messaging standard, while extremely flexible, is also complex. Interfacing with other applications may not be the highest priority on the list when creating a clinical application. By the time interfacing is approached, decisions regarding how the application will work may have already been made. This could result in any number of messages from the application to lack HL7 compliance.

How Do You Check for HL7 Conformance?

As an analyst trying to ensure correct communication between medical devices and applications, you will have to look at all the possible predictable and unpredictable causes for nonconformance. Where do you start?

First, determine how closely the incoming and outgoing message formats matches your own.

Second, teach your system how to understand the different message format in order to receive and send messages.

Finally, each message should be checked as it is coming into your application during run-time in order to verify that it provides your application with the fields you require. If it does not, make sure it is put aside to be evaluated for possible noncompliance.

Determining Message Format Differences

Determining format differences begins with analyzing the external healthcare vendor's message specifications and sample messages. These specifications are rarely up-to-date; therefore, sample messages will provide you with the most accurate view of the message formats you will receive. This sample of messages should be large enough to represent each type of message you may receive from the external healthcare vendor.

Compare specifications with your message format to determine the areas of nonconformance.

Next, walk through the message samples to confirm that the sample messages match the specification. If you find additional areas of nonconformance in the sample messages, record those issues.

The areas of nonconformance found within the specifications and message samples provide critical information for you to use when negotiating with the external healthcare vendors or providers.

Teaching Your System to Understand the Message Differences

There are many reasons systems that claim to be HL7 compliant have message formats that are not quite HL7 compliant. Additionally, all the flexibility within the HL7 standard creates differences in the messages. However, the fact remains that your system is communicating with these applications, and so it needs to be able to accept and receive messages that are not conformant with your own format standard.

This is commonly accomplished by using an HL7 interface or communications engine that sits between the two separate systems and parses and encodes the messages coming into or leaving the system.

There are a number of different tools available that provide assistance in the creation of the interfaces. Most interfacing experts agree that it is much easier to use an existing tool to parse and encode HL7 messages than to create one individually. The best tools have extensive HL7 knowledge built in and provide maximum flexibility to adjust to varying message formats.

Conformance Checking During Run-Time

You must weigh the advantages and disadvantages of how strict your conformance checking treats incoming messages. The stricter your conformance checking at run-time, the more messages will error and require manual intervention to process.

Typically, you should ignore the unexpected segments when you are in run-time mode, which complies with HL7's recommendation to ignore unexpected elements. You may, however, want to require the existence of required elements in your message to avoid having incomplete data.

Any quality interfacing tool will have configurable conformance checking available at run-time.

Summary

Although complex, engaging in a conformance checking process will greatly improve the integrity of the exchange of clinical data via the HL7 standard. Whether you utilize an interfacing tool or engage in a manual process, several items will need to be checked for conformance to the HL7 standard used:

- Cardinality within fields or segments
- HL7 version features used by both the sending and receiving entity
- Customization and/or misinterpretation of the HL7 standard
- Differences in message formats

About NeoTool

NeoTool is a leading provider of healthcare integration solutions that empower organizations to develop, test, deploy, and manage data exchanges between healthcare applications and providers. Through software, HL7 training, and consulting, NeoTool is dedicated solely to healthcare application interfacing. NeoTool customers include healthcare providers (e.g., hospitals, imaging centers, labs, and clinics), healthcare software application providers, and medical device manufacturers. www.neotool.com

NeoTool
6509 Windcrest Drive
Suite 160B
Plano, Texas 75024
469-229-5000
Sales@neotool.com

