Principles for Profiling Healthcare Data Communication Standards

R. Snelick[1] and F. Oemig[2]
[1]National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA
[2]Agfa Healthcare, Bonn, Germany

Abstract - *Healthcare organizations often have many proprietary heterogeneous information systems that must exchange data reliably. Seamlessly sharing information among systems is complex. The widely adopted HL7 version 2 messaging standard has helped the process of systems integration. However, using the HL7 standard alone does not ensure system interoperability. The HL7 standard offers a wide range of options. Trading partners, without prior agreement, are not likely to implement options that are compatible. As a result, interoperability is hindered and organizations are left to employ their own ad hoc solutions. Message profiles provide a solution to this problem. Message profiles define a standard template that provides a precise definition of the data exchanged between applications in a common format. Defining a set of message profiles for controlling message exchanges establishes a well-defined communications interface among organizations and facilitates interoperability. However in order to be effective, message profiles must be designed and applied correctly. Additionally, with efficient design, a family of message profiles can be developed which leverage existing message profile components. Such a strategy is employed in the development the United States EHR certification family of standards for laboratory ordering and results reporting. This paper presents a methodology and best practices for designing a set of related message profiles. Although the methodology is applied to the healthcare messaging standards it has broad applicability for the class of communication standards.*

Keywords: Conformance; Communication Standards; Interoperability; Message Profiles; Messaging Systems.

1 Introduction

A major challenge for the healthcare industry is achieving interoperability among proprietary applications provided by different vendors. For example, each hospital department may use one or more applications to share clinical and administrative information. Each application may support multiple communication interfaces that must be modified and maintained. This is a difficult way to achieve interoperability. Alternatively, interoperability can be achieved through the use of standardized interfaces; the definition of which can remove the cost of building a separate interface for each associated application. Developers can build applications that conform to the standardized interface definition, increasing the likelihood of interoperability and reducing cost. Maintenance cost is also reduced because the number of interfaces to maintain decreases.

The Health Level Seven (HL7) *Application Protocol for Electronic Data Exchange in Healthcare Environments* Version 2.x standard (hereafter HL7) is the de facto standard for moving clinical and administrative information between healthcare applications [1]. The standard is based on the concept of application-to-application message exchange. An HL7 *message* is an atomic unit of data transferred between systems [1]. Typical HL7 messages include admitting a patient to a hospital or requesting a laboratory order for a blood test. HL7 describes an *abstract message definition* for each real world event (*e.g.*, admitting a patient). The abstract message definition is comprised of a collection of segments in a defined sequence. Rules for building an abstract message definition are specified in the HL7 message framework, which is hierarchical in nature and consists of building blocks generically called *elements*. These elements are *segment groups*, *segments*, *fields*, *components*, and *sub-components*. Each element has associated attributes that further defines and constrains the element. These include optionality, cardinality, value set, length, and data type attributes. Segment groups and segments can contain additional elements, fields and components can contain additional elements or be primitive elements; sub-components are strictly primitive elements. Primitive elements are those that can hold a data value and have no descendant structure.

When originally developed, HL7 was designed to accommodate the many diverse business processes that exist in the healthcare industry. This universal design was necessary to gain broad industry support. However, such broad accommodations resulted in a standard with many optional elements, thus aligning interface implementations presented difficulties.

Applications using HL7 are generally connected in two ways, point-to-point or via middleware, typically communication server products. Point-to-point entails connecting each pair of applications independently of other applications. In the communication server approach, all applications are connected to a centrally located message broker. A set of HL7 message definitions specifies the requirements between the communicating applications. Although the message definitions are specific there are many ways to specify a given HL7 transaction. In practice, vendor-provider specifications may not quite match, therefore differences need to be accounted for in each connection. In point-to-point architectures, each new combination will require a separate implementation. With communication servers, a new mapping transformation definition needs to be defined. In both cases, the breadth of the specification leads to cumbersome and *ad hoc* interface implementations. System implementations are prone to error, difficult to maintain, and do not scale easily.

To help alleviate this shortcoming, the HL7 standard introduced the concept of *conformance message profiles* (also commonly referred to as conformance profiles, message profiles, or profiles—hereafter message profile or profile). Message profiles by defining processing rules and which optional elements in the standard a message might include provide an unambiguous description of HL7 messages.

2 HL7 Message Profile Defined

Message profiles[1] constrain HL7 message structure and requirements for a particular interaction. A message profile provides a mechanism for specifying a single message definition. An implementation guide is often created to organize a collection of message profiles for specifying a set of related HL7 V2.x interactions described by a use case or use cases. Implementation guides typically describe broader conformance requirements such as a use case model, a dynamic definition, a static definition, and application functional requirements. IHE integration profiles can be characterized as implementation guides [7].

The use case model provides a description, defines actor responsibilities, and describes a sequence of actions performed by the sending and receiving applications. The dynamic definition describes the interaction between the sender and the receiver in terms of the expected acknowledgments (or other transactions such as query/response). The static model provides a precise definition of the message structure and constraints for a single message; this is the message profile. Functional requirements describe the application (or actor) level requirements. Such requirements may include how a set of messages are to be used to enact certain application functionality. The message profile definition, use, and organization within an implementaiton guide are key issues addressed in this paper.

A message profile can be represented as an XML document, Figure 1 shows an example XML profile snippet. Each element in the message profile is listed along with its associated attributes. For a more detailed description of a message profile refer to the HL7 standard [1]. It is important to note that the attributes and the constraints a profile places on a message provide a clear and unambiguous definition, thereby, facilitating the design, implementation, and testing of interfaces [3,4,5].

Fig. 1. Snippet from a Message Profile

```
…
&lt;Segment Name="PID" LongName="Patient Identification"
        Usage="R" Min="1" Max="1"/>
    <Reference>3.4.2</Reference>
  <Field Name="Set ID - PID" Usage="R" Min="1" Max="1"
        Datatype="SI" MaxLength="4" MinLength="1">
  </Field>
…
  <Field Name="SSN Number - Patient" Usage="X" Min="0"
      Max="0" Datatype="ST" MaxLength="16" MinLength="1" />
  <Field Name="Driver's License Number - Patient" Usage="R"
        Min="0" Max="0" Datatype="DLN" MaxLength="66"
        MinLength="1">
    <Component Name="License Number" Usage="R"
        Datatype="ST" MaxLength="20" MinLength="1" />
    <Component Name="Issuing State, Province, Country"
        Usage="R" Datatype="IS" Table="0333" MaxLength="20"
        MinLength="1" />
    <Component Name="Expiration Date" Usage="O"
        Datatype="DT" MaxLength="24" MinLength="1" />
  </Field>
…
```

The rules for constructing a message are described by the message framework [1]. In addition, for each real world event, for example “Admitting a Patient”, a specific abstract message structure (ADT_A01) is defined. The message structure defines a template or structure in which the message must comply; it explicitly defines the elements and the order the elements must appear in a message instance. For example, in Figure 1, the “PID” segment contains the field “Set ID – PID”, and so on. The usage attribute refers to the circumstances in which an element appears in a message [1]. For example, the “Driver’s License Number” component in the profile snippet is required (Usage=”R”) and must be present in a valid message instance. Cardinality refers to the minimum and maximum number of occurrences an element may have [1]. An example of an element cardinality is [0..1]; the element may not appear in the message instance, but can only have one occurrence if it does. A table of allowable values can be defined and associated with a certain element. For example, see the

[1] Message profiles are not to be confused with the Integrating the Healthcare Enterprises (IHE) integration profiles. Often IHE integration profiles will use HL7 message profiles.

"Issuing State, province, country" component in Figure 1; this element must be populated with a data value that is defined in Table 0333. The length attributes define the minimum and maximum allowable lengths a value can have for a particular element. The data type defines the allowable data values an element can contain. For primitive data types, such as string (ST), interpretation is straightforward and requirements for each data type are specified in the standard [1]. Complex data types, such as the Extended Person Name (XPN), may be composed of primitive types or other complex data types. For example, an XPN contains a family name (FN), which itself is a complex data type that is composed of five primitive elements, all of type string (ST). All complex data types are ultimately composed of primitive data types.

A message profile is distinguished from a specification by application of the conformance rules, the openness permitted by the base standard is ultimately removed to such an extent that the interface specified by the profile may be directly implemented (Figure 2). The HL7 standard allows for numerous ways to define an interface; profiles reduce the number of possibilities to a manageable set, and their use helps to ensure that systems attempting to communicate with each other implement compatible sets of possibilities. It is important to recognize that profiles do not eliminate possibilities allowed by the standard; they select a specific group from the total set of those allowed. In this regard, a profile defines a constraint on the standard, such that the resultant constrained specification may be used to implement the interface. The profile also imposes a discipline upon the interface partners. This ensures harmony in the actual implementation which is necessary to fulfill a certain use case.

A key development for promoting interoperability was the codification of a means to express message profiles in a standardized way. While natural language documentation of a message profile acceptably facilitates interoperability at the message implementation level, the standardization of the message profile documentation itself adds a new dimension to the promotion of interoperability. The standardized conformance profile is an XML document specified in terms of a normative schema. This standardized form aids in many aspects in documentation, implementation, and testing. The NIST EHR certification conformance test tools use the XML message profile as the basis for validation [5].

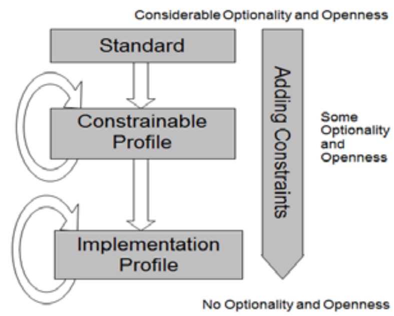
2.1 Message Profile Hierarchy

HL7 V2 message profiles have three levels of specification:

- HL7 Standard Profile Level
- Constrainable Profile Level
- Implementation Profile Level

The HL7 Standard Profile (hereafter standard profile) represents the base standard definitions and constraints for a specific message structure (e.g., ORU_R01 for laboratory results reporting). At this level, the overall structure including the data type definitions are fully defined, however many element attributes are not. The standard profile can be more precisely defined by adding constraints to the elements attributes.

Fig. 2. Message Profile Hierarchy



Other message profile levels are derived from the standard profile. A Constrainable Profile (hereafter constrainable profile) is derived from either the standard profile or another constrainable profile and further constrains the message definition attributes. For example, an element with a usage of "*optional*" may be changed to "*required*", however, the data type structure for that element cannot be changed. In a constrainable profile, analogous to the standard profile, not all element attributes are fully constrained. An Implementation Profile (hereafter implementation profile) defines all elements such that all *optionality* and *openness* is removed. All deployed interfaces are implementation profiles whether they are documented (explicitly) or not (implicitly). It is highly recommended that interfaces are completely documented to the implementation profile level using the profiling mechanisms described in this paper and the HL7 V2.x Conformance Chapter [1]. An implementation profile may also be derived from another implementation profile. In this case all openness has been removed. However, further constraints on attributes can be applied; for example, the usage of "*required, but may be empty*" can be strengthened to "*required*".

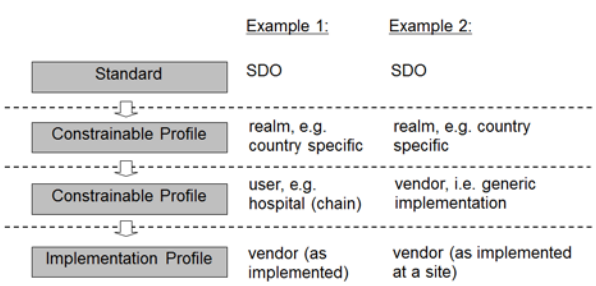
As described, constraints can be added iteratively, thereby forming a hierarchy of messages profiles. As such, a certain set of rules must be followed. A message profile is prohibited to further refining certain requirements defined in the *parent* message profile. For example, if an element (e.g. field) is "*required*" in the parent profile it can't be profiled to "*optional*" in the *child* profile as the requirement is relaxed (The allowable derivations are described in HL7 V2.x Conformance

Chapter [1]). Figure 2 illustrates the concept of profile hierarchy and acceptable derivations.

Two possible real world scenarios for using the profile hierarchy model are presented in Figure 3. In the first case a national level constrainable profile is developed. A hospital (chain) adopts and refines the national level guidance provided in the realm specific constrainable profile. The hospital procures a vendor that has a product that can be configured to satisfy the requirements. The hospital and the vendor finalize the requirements and the software is installed. The resultant interface is documented as an implementation profile. Alternatively, the hospital could have provided the implementation profile directly to the vendor.

In the second case, a vendor refines national level guidance profile and provides a generic implementation based on this constrainable profile. When working with clients in which this profile closely satisfies their requirements a final refinement is made at the specific sites. The vendor will often (or should) provide the documentation of the interface installed in the form of an implementation profile. These examples can be nested and refined to any depth as appropriate (See Figure 2).

Fig. 3. Use of Profile Examples



The concept and use of constrainable profiles is important in practice as this level is often (and should be) what standard organizations (e.g., IHE or the HL7 affiliates [9]) specify. Constrainable profiles can be thought of as a set of harmonized requirements and are useful at a national or any intermediate level down to the local site implementation. Employing implementation profiles at a high-level such as nationally often precludes widespread adoption because of their restrictiveness. Therefore, this practice is not recommended and should be avoided.

Use of the message profile hierarchy is the strategy employed in the United States by the Office of the National Coordinator (ONC) Meaningful Use (MU) electronic health records (EHR) certification program. The named standards in the certification criteria specify "*national level*" requirements—although they are not explicitly named as such. For the HL7 V2 messaging standards these requirements are published in implementation guides and realized as constrainable message profiles—meaning that a selected set of elements are fully specified while others are yet to be determined. This approach guarantees that certified EHR technologies (CEHRT) have a certain level of common capabilities while providing flexibility for local customization. However, these implementation guides are independent, so no harmonization among the profiles is guaranteed. For example, specification of patient demographics does not necessarily coincide in the transmission to immunization registry and laboratory results reporting implementation guides.

Local installations are likely to complete trading-partner agreements. That is, they will further refine the national level requirements to satisfy their local requirements within the framework established by the constrainable profile. It is important that the local trading-partner agreements do not relax or conflict with the national level requirements. The certification of the EHR products seeks to ensure a minimum level of capabilities that will not necessarily meet all local requirements (and often will not). Once local trading-partner agreements are put in place, the EHR technology and partner systems will need to be implemented and configured accordingly. For example, a provider and their state immunization registry will coordinate exchange requirements. The referenced Meaningful Use interoperability standard (i.e., the constrainable profile) provides the basis, but additional requirements may be necessary for this jurisdiction (specified in an implementation profile derived from the constrainable profile). In this case, the system receiving the HL7 V2 messages (i.e., the immunization registry) must be able to consume and understand the state-level information according to CEHRT to achieve the desired interoperability. To ensure accuracy and integrity for this exchange of information, local site testing must be performed. At present, this aspect of testing is not part of the Meaningful Use program; however, using CEHRT provides a shorter pathway to achieving site-specific interoperability.

2.2 Message Profile Component Defined

A message profile component (hereafter profile component) defines a part or a certain aspect of a profile and is used to differentiate requirements from another profile or profile component. A profile component can be applied to any construct or section of a profile. A profile component in a family of profiles can be used to identify different levels of requirements for the same use case or to identify the differences in requirements for different, but closely related, use cases.

In the first case, a specification may want to express different levels of conformance. For example, a profile may be written to require the use of Object Identifiers

(OIDs) for all identifiers. Another profile may be written in which this is a not requirement. An intermediate profile may be written which requires certain identifiers to support the use of OIDs but not all. This specification is describing three levels of conformance. These three levels can be described using a base profile definition and three profile components. The profile components describe the differences in the requirements. A similar scheme as described here is employed in the HL7 V2 2.5.1 Laboratory Results Interface (LRI) implementation guide's laboratory results message profiles (ORU_R01 message structure) [6].

In the second case, a profile component is employed to express requirements for a different, but closely related, use case. Here the profile component is used to leverage the requirements of an existing profile since this profile contains many common requirements. The HL7 V2.5.1 Electronic Laboratory Reporting (ELR) to Public Health Revision 2 implementation guide uses the concept of a profile component in this manner [8].

In the first case, the use case is the same; however, the requirements in which it can be achieved are different. The profile component is expressing a different level of conformance. In the second case, the use case is similar but different, therefore the requirements are different. The profile component concept is used to leverage the common requirements defined by the profile and to express the differences in requirements by defining them in a profile component.

Profile components can express missing requirements for a base profile component, common requirements, additional requirements, or replace requirements in a profile or profile component.

The description of the different conformance levels, profiles, and profile components are expressed in the conformance clause section of a specification. Subsequently an implementer makes a conformance claim as to which level of conformance they support.

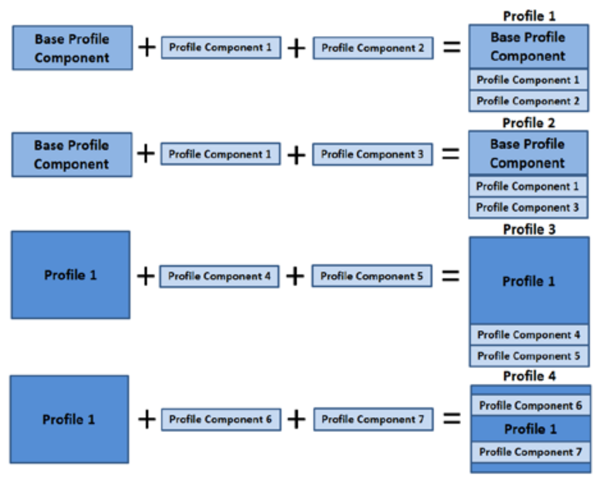
3 Profile Design and Management

This section presents an approach for designing and managing profiles such that profiles and profile components can be leveraged. When writing a set of related profiles (or a family of profiles such as those in IHE or for a particular domain such as laboratory orders and results) it is important to reuse the profile and profile components, to harmonize the requirements and to gain efficiency.

Figure 4 illustrates a sample of possible configurations for composing a family of related profiles. The design principle is to develop a common or base profile component that applies across a family of profiles with the intent of using the profile component concept to specify profiles.

In the first depiction, a base profile component is developed that expresses all of the common requirements for a related set of profiles. Profile component 1 and profile component 2 are also created for aspects that are not defined in the base profile component. Combined the three profile components are used to describe a complete specification, Profile 1. For the second depiction, the base profile component and profile component 1 are reused and combined with profile component 3 to specify Profile 2. In the third depiction, Profile 1 is combined with profile components 4 and 5 to create Profile 3.

Fig. 4. Profiling Design Principles



Profile components can also express requirements that replace requirements established in a base profile component or profile. This may often be the case when different levels of profiles are developed or the profile provides utility outside the original set of related profiles. The fourth depiction illustrates such a case where a subset of requirements for an existing profile is overridden. Here Profile 1 is used. However, certain aspects are redefined according to the rules and documented in profile components 6 and 7 which results in Profile 4. For each of the complete specifications illustrated in Figure 4 the resulting profile can be a constrainable or an implementable profile.

The key design principles for developing a family of related specifications is to leveraged existing profiles or design/create base profiles that are a harmonization of requirements for a related set of use cases. The profile components can be developed at any level of granularity. However, caution should be exercised when creating profile components at the fine grain level. Often creating and managing too many building block artifacts will start to outweigh the benefits. If tooling is available then fine granularity of profile components is attainable. A good practice is to introduce an orthogonal structure of the individual requirements, e.g., data type constraints in one regard and value set definitions in another. This allows for easy integration, combinations, and management.

Unfortunately, often in practice, a related set of profiles are each fully specified that duplicate sizeable sections of the document. These profiles are not harmonized and unnecessarily lead to maintenance issues. It is also important not to confound requirements targeted for different use cases (interactions) within a single profile definition. This also occurs in practice and should be avoided. For each interaction, a separate message profile needs to be defined. The use of profile components as described facilitates this approach.

3.1 Publishing the Specification

An important design principle for publishing the specifications is not to copy entire specifications that express only small variances in requirements. This creates management and maintenance issues when modifications are made in the base profile component. If possible, the profile should be part of the original specification and distinguished as a profile variance through the profile component mechanism. If however, the new profile is created after publishing the profile in which it is derived then only the variations should be published in the new specification. Often this document will be a few short pages. This approach quickly and efficiently alerts the implementers to the modifications from the original (base) profile.

If the specification is developed using authoring tooling then the user is afforded various options for publishing since the tool handles the rendering and maintenance. The National Institute of Standards and Technology (NIST) is developing a tool to enable manipulation of profiles for HL7 V2. This tool builds upon the concepts developed in the Messaging Workbench (MWB) [2,3]. The NIST tool is being designed to allow for the development of profile components. Since all artifacts related to the profile are machine process-able within the tool, the user will have the option to publish a specification that expresses the variance of a profile, the complete profile, or other artifacts such as the XML representation of the profile.

4 Information Mapping

This section describes an approach for system developers for mapping information in their systems to HL7 interface data requirements. This technique provides a flexible and universal methodology to a systematically account for variations in interface requirements of trading partners. Figure 5 illustrates a proposed information mapping approach to support multiple interfaces with varying requirements, which are expressed in a related set of message profiles.

The basic principle is that specifications are not to be influenced by implementation design or by trying to accommodate different use cases (interactions) within a single profile definition. This approach is not recommended because requirements are confounded when they should not be. A profile needs to be written in a manner to express the requirement for a single interaction and nothing more. The profile design principle section describes how one can accommodate similar uses cases (which require different interactions). If this strategy is employed, there can be a gain in efficiency by accommodating the various use cases and without having to rewrite or create entirely new specifications. This approach allows for more choices in implementation design and support. If multiple use cases are comingled into a single specification, then those who choose not to support certain components are forced to deal with the unwanted components. The profile design mechanism proposed in section 3 also provides a clean and flexible avenue for implementers, as they are not tied to implementation choices dictated in the specification.

Fig. 5. Information Mapping to Support Multiple Interfaces

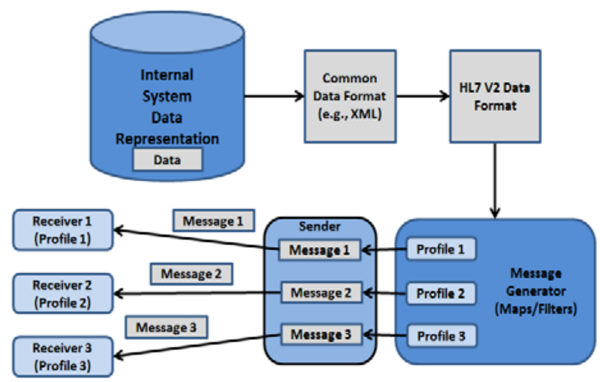


Figure 5 proposes a possible approach that a system might implement to support various profiles to interact with a multitude of systems with similar but different requirements. Internally the system maintains a database containing the information necessary for the application. It is necessary that this application communicate with a number of different trading partners. The trading partner interfaces have slightly different requirements. Various profiles are created that express the requirements for each of the interfaces. The methodology described in section 3 is used to create the profiles. When communicating, the sending system extracts data from the database and transforms it into a common representation (e.g., an XML specification). For each interface, the data is used to populate the message using the message profile as a map (i.e., template).

In the example depicted in Figure 5 there are three HL7 V2 interfaces that have to be supported. Depending on the interface, the application creates a message based on the given requirements expressed in the profile. The profile is represented in XML and acts as a filter of the complete data known to the system. Depending on the trading partner, a different "filter" (profile) is utilized. For each interface, a specific message is created with the necessary requirements. There is no need to disambiguate requirements for different trading partners or interface requirements.

This is a simplification of the process and does not account for all the complications (especially the mapping of the data). However, it does illustrate a simple and straightforward approach to interfacing with multiple trading partners where the requirements are clearly defined in isolation although built upon a common foundation. That is, the requirements of an interface are separated from the implementation and operational aspects of the system. This design is scalable since many more trading partners could be added that have different interface requirements and the only additional artifact needed is a profile. Of course, the above illustration describes an interaction with nearly the same set of requirements, for example, reporting immunization records to an Immunization Information System (IIS). The vendor product will often need to support all or many of the state's IISs each with slightly different reporting requirements.

5 Summary

The ability to share relevant information among diverse healthcare systems and provide consistent data across applications will help improve the quality of care. It will also improve patient safety and reduce the cost of healthcare. HL7 defines the specification for interfaces that allow both centrally located and distributed information systems to communicate. The standard establishes rules for building interfaces and provides many optional features to accommodate the disparate needs of the healthcare industry. However, for interfaces to be reliably implemented, a precise and unambiguous specification must be defined. HL7 introduced the concept of message profiles that precisely declare the structure and constraints of a message. The use of message profiles promotes interoperability by providing trading partners a common format for documenting interface specifications.

There are three levels of profiles that form a hierarchy including the standard level, the constrainable level, and implementation level. A message profile component defines a part or a particular aspect of a profile and is used to differentiate requirements from another profile or profile component. A profile component can be applied to any construct or section of a profile. Combining the concepts of profile levels and profile components provide implementation guide authors with the tools to effectively create and manage a set of related profiles. A profile can be represented in a standardized XML form that enables automatic processing of many facets including publishing and message validation. System developers can take advantage of message profiles to simplify implementations that support many similar or disparate interface requirements.

To ensure interoperability among healthcare systems, installations must be implemented correctly—conformance testing is essential [3,4,5]. Using and specifying well defined message profiles facilitates and promotes more rigorous testing. Employing an implementation and testing strategy based on message profiles and the tools to support them will improve interoperability among healthcare systems. This ultimately leads to more reliable systems and reduced costs.

6 References

[1] Health Level 7 (HL7) Standard Version 2.7, ANSI/HL7, January, 2011, http://www.hl7.org.

[2] Messaging Workbench (MWB). Developed by Peter Rontey. http://www.hl7.org.

[3] *Towards Interoperable Healthcare Information Systems: The HL7 Conformance Profile Approach*. R. Snelick, P. Rontey, L. Gebase, L. Carnahan. Enterprise Interoperability II: New Challenges and Approaches. Springer-Verlag, *London Limited 2007 pp*. 659-670.

[4] *A Framework for testing Distributed Healthcare Applications*. R. Snelick, L. Gebase, G. O'Brien. 2009 Software Engineering Research and Practice (SERP09), WORLDCOMP'09 July 13-16, 2009, Las Vegas, NV.

[5] *NIST Laboratory Results Interface (LRI) EHR Tool.* http://hl7v2-lab-testing.nist.gov/mu-lab/

[6] *HL7 Version 2.5.1 Laboratory Results Interface (LRI) Implementation Guide.* Draft Standard for Trail Use. July 2012. http://www.hl7.org.

[7] *Integrating the Healthcare Enterprises (IHE) Technical Framework.* http://www.ihe.net

[8] *HL7 Version 2.5.1 Electronic Laboratory Reporting (ELR) to Public Health Implementation Guide, Release 2.* Work in Progress. http://www.hl7.org.

[9] *German Message Profile Architecture*. 2004-2007. http://www.hl7.de/download/documents/Profile_2.1.zip.