FHIR Profiles and Consolidated CDA Templates: Data-Modeling Issues With Implications for Patient Safety

Current State

April 6, 2018

DRAFT

Contents

1 Introduction

Interoperability among clinical information systems remains a significant challenge for U.S. health care organizations, including the Veterans Health Administration and the Military Health System. Recently, the specification and growing adoption of two interoperability standards has raised the prospect of improved interoperability: The U.S. Core Implementation Guide for Fast Healthcare Interoperability Resources (US-FHIR) and the Consolidated CDA Implementation Guide for the Clinical Document Architecture (C-CDA). These implementation guides have become required or suggested standards for clinical data exchange under the CMS Meaningful Use Program for EHRs, and thereby have garnered considerable attention (and implementations) on the part of commercial EHR vendors and large healthcare organizations.

As adoption and use of the US-FHIR and C-CDA implementation guides grow, healthcare organizations will exchange more real-world patient data encoded in the underlying data models of these standards. However, to date, no formal examination of these data models has been done with respect to their patient-safety implications. Specifically, to the degree that loss or misinterpretation of clinical data can adversely impact clinical care as patients move between health care organizations, the ability of these standard data models to fully, clearly, and unambiguously represent the clinical meaning of patient data has become important for ensuring patient safety and avoiding medical errors.

This whitepaper examines several data-modeling characteristics of the US-FHIR and C-CDA standards that bear consideration from a patient-safety perspective. The issues include the standards' inability to represent certain important information in a structured way, their propensity to represent certain important clinical information in unexpected and unrecognizable ways, and their manifestation of an excess complexity that increases the risks of misinterpreting important clinical data. Section 2 examines these issues with respect to the US-FHIR implementation guide, and Section 3 examines them with respect to the C-CDA implementation guide.

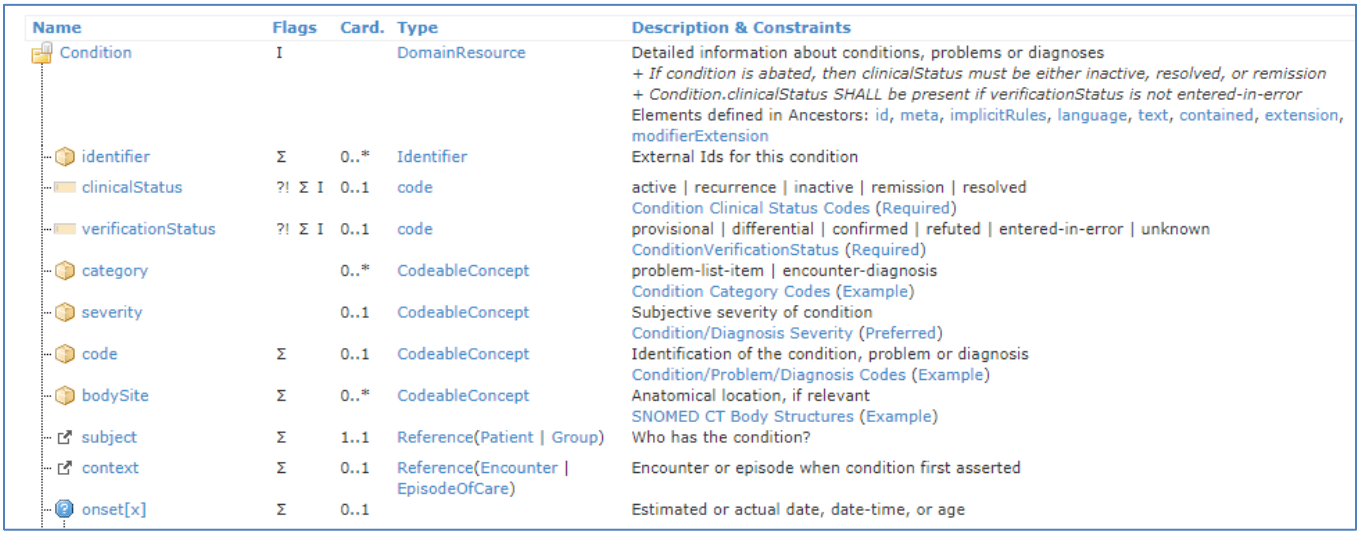
2 FHIR Resources and FHIR Profiles

Fast Healthcare Interoperability Resources (FHIR)[1] is a formalism developed by HL7 to specify "RESTful" web-service interfaces[2] for healthcare applications. The FHIR specification includes a set of roughly 90 data models, called "resources," to represent the various types of clinical data objects that may be retrieved via web-services requests. Examples of resources include Patient Demographics, Problems, Medication Orders, and Test Results.

Resources specify the allowed sub-parts of each such clinical data object, which sub-parts are required versus optional, and the data types or enumerated value sets that may be used to populate the sub-parts. Figure 1 shows part of the specifications of the FHIR resource for representing patients' problems (conditions). Formal resource definitions are critical to interoperability because they specify how the sender of FHIR-encoded information must represent the clinical data it is transmitting and how the recipient of FHIR-encoded information can expect the clinical data it receives to be represented.

The "base" FHIR specification includes relatively few constraints on the representation of clinical data (for example, note that only one of the data elements shown in Figure 1 is required, and no required coding system is specified for the coded value in the "code" data element, which identifies the reported condition). However, the FHIR formalism allows the specification of "profiles" with respect to any resource. Profiles further constrain the allowed or required sub-parts of a resource, as well as the values that may be used to populate the sub-parts. Depending on their design, such enhanced constraints can achieve greater “plug-and-play” interoperability among the organizations that agree to them.

Figure 1. Portion of the specification for the FHIR *Condition* resource.



Name	Flags	Card.	Type	Description & Constraints
Condition	I		DomainResource	Detailed information about conditions, problems or diagnoses *+ If condition is abated, then clinicalStatus must be either inactive, resolved, or remission* *+ Condition.clinicalStatus SHALL be present if verificationStatus is not entered-in-error* Elements defined in Ancestors: id, meta, implicitRules, language, text, contained, extension, modifierExtension
identifier	Σ	0..*	Identifier	External Ids for this condition
clinicalStatus	?! Σ I	0..1	code	active | recurrence | inactive | remission | resolved Condition Clinical Status Codes (Required)
verificationStatus	?! Σ I	0..1	code	provisional | differential | confirmed | refuted | entered-in-error | unknown ConditionVerificationStatus (Required)
category		0..*	CodeableConcept	problem-list-item | encounter-diagnosis Condition Category Codes (Example)
severity		0..1	CodeableConcept	Subjective severity of condition Condition/Diagnosis Severity (Preferred)
code	Σ	0..1	CodeableConcept	Identification of the condition, problem or diagnosis Condition/Problem/Diagnosis Codes (Example)
bodySite	Σ	0..*	CodeableConcept	Anatomical location, if relevant SNOMED CT Body Structures (Example)
subject	Σ	1..1	Reference(Patient | Group)	Who has the condition?
context	Σ	0..1	Reference(Encounter | EpisodeOfCare)	Encounter or episode when condition first asserted
onset[x]	Σ	0..1		Estimated or actual date, date-time, or age

A set of mutually consistent resource profiles that have been specified by a particular group of organizations or for a particular purpose comprises a FHIR “Implementation Guide”. Organizations that wish to interoperate using an implementation guide typically must conform to the entirety of the resource profiles within it.

2.1 U.S. Core Implementation Guide

The FHIR “U.S. Core” implementation guide (US-FHIR) was developed by HL7, in collaboration with the U.S. government and the private sector, specifically to support the interoperability requirements of the CMS meaningful use program. A total of 18 resource profiles are defined within the US-FHIR implementation guide, including *Condition* (i.e., problem), *Results* (i.e., lab results), *Medication*, *Immunization*, and *AllergyIntolerance*.

Although the additional constraints specified in these profiles (relative to the “base” FHIR resource definitions) greatly improve their effectiveness for interoperability, certain contents of the profiles, as well as of the HL7 FHIR profiling mechanism itself, raise potential patient-safety issues. These issues are described in detail in the following sections.

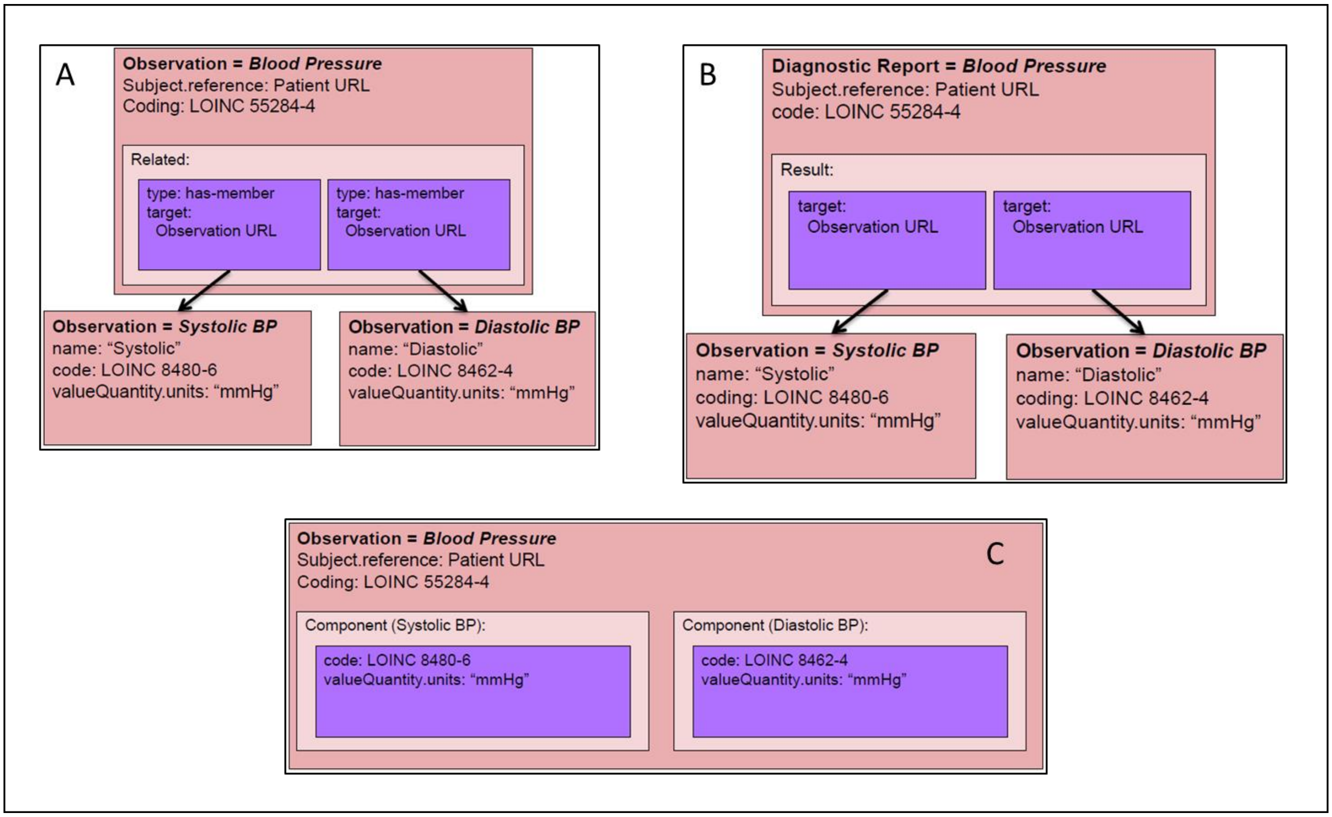
2.2 Remaining Structural Under-Specification

Although the US-FHIR resource profiles add constraints to the base FHIR resources definitions, these constraints still allow important clinical data to be represented using multiple incompatible structures. In other words, two organizations that are fully compliant with the US-FHIR implementation guide could have different interpretations about how certain clinical data are to be represented. When the recipient of information is expecting a different structure than the sender provides, the recipient may not correctly process or store the information received, leading to potential patient-safety risks. Such structural under-specification can occur with respect to pre-defined data elements or with respect to so-called “extension” data elements. Examples of both appear below.

2.2.1 Pre-defined Data Elements

Using the “Observation” and/or “Diagnostic Report” resource profiles of US-FHIR, a simple blood-pressure reading could be represented in at least three different ways using the pre-defined data elements of the profiles, as shown in Figure 2. Specifically, the systolic and diastolic components of the reading could be represented as either references to additional *Observation* instances from a parent Observation instance (A) or from a parent *Diagnostic Report* instance (B). Further, the systolic and diastolic components could alternatively be represented “in-line” within a single *Observation* instance (C).

Figure 2. Three different ways to represent a blood pressure reading using US-FHIR profiles (graphic courtesy of courtesy of Claude Nanjo, Cognitive Medicine, AMIA 2017).

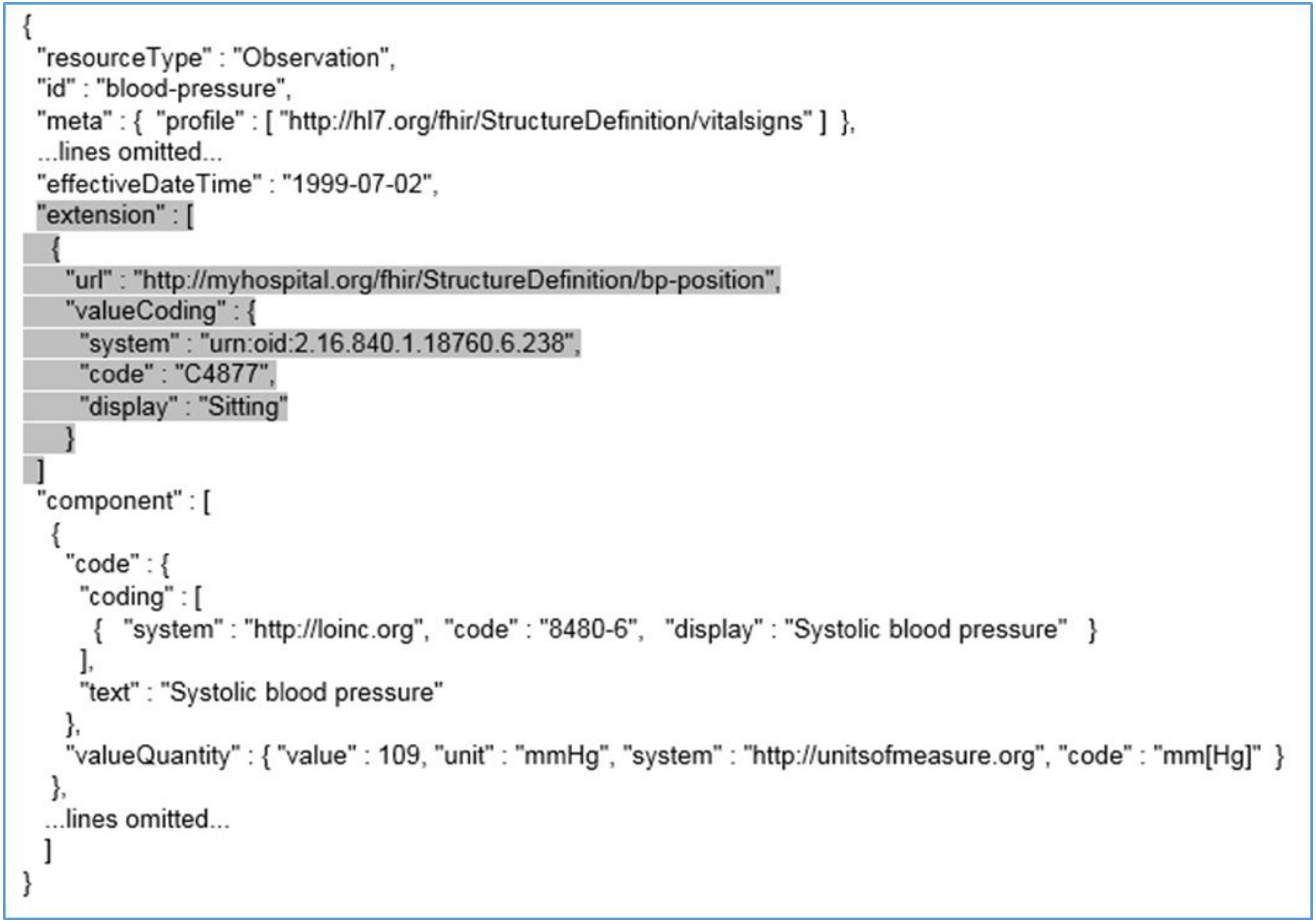

A
Observation = Blood Pressure
Subject.reference: Patient URL
Coding: LOINC 55284-4
Related:
type: has-member
target:
Observation URL
type: has-member
target:
Observation URL
Observation = Systolic BP
name: “Systolic”
code: LOINC 8480-6
valueQuantity.units: “mmHg”
Observation = Diastolic BP
name: “Diastolic”
code: LOINC 8462-4
valueQuantity.units: “mmHg”
B
Diagnostic Report = Blood Pressure
Subject.reference: Patient URL
code: LOINC 55284-4
Result:
target:
Observation URL
target:
Observation URL
Observation = Systolic BP
name: “Systolic”
coding: LOINC 8480-6
valueQuantity.units: “mmHg”
Observation = Diastolic BP
name: “Diastolic”
coding: LOINC 8462-4
valueQuantity.units: “mmHg”
Observation = Blood Pressure
Subject.reference: Patient URL
Coding: LOINC 55284-4
C
Component (Systolic BP):
code: LOINC 8480-6
valueQuantity.units: “mmHg”
Component (Diastolic BP):
code: LOINC 8462-4
valueQuantity.units: “mmHg”

2.2.2 Required “Extension” Data Elements

Although the US-FHIR profiles are further constrained with respect to HL7 “base” resource definitions, they are still quite generic and designed to accommodate a large variety of clinical statements. As a result, data elements that are specific to only certain types of clinical information and are needed to express the clinically important details of that information are often not part of the US-FHIR profiles. For example, the US-FHIR *Observation* profile has no pre-defined attributes to represent the body position of a patient (upright vs. recumbent) when a blood-pressure reading was taken, a potentially significant factor in the clinical interpretation of the reading.

In these cases, representing such details requires the addition of “extension” data elements to individual instances of resources, which the FHIR standard allows[3]. Such locally-added extensions, however, may have arbitrary names and structures, and their values may be represented using arbitrary data types or value sets, all at the discretion of the sending organization. Figure 3 shows an example of such an extension data element added to an *Observation* profile instance. If the receiving organization is unaware of these locally-defined modeling constructs, it may not correctly process or even recognize the additional information that has been sent, resulting in a loss of that data between sender and recipient.

Figure 3. Instance of an *Observation* profile including a locally-added extension to represent the patient's position at the time of measurement.



```
{
  "resourceType" : "Observation",
  "id" : "blood-pressure",
  "meta" : {  "profile" : [ "http://hl7.org/fhir/StructureDefinition/vitalsigns" ] },
  ...lines omitted...
  "effectiveDateTime" : "1999-07-02",
  "extension" : [
    {
      "url" : "http://myhospital.org/fhir/StructureDefinition/bp-position",
      "valueCoding" : {
        "system" : "urn:oid:2.16.840.1.18760.6.238",
        "code" : "C4877",
        "display" : "Sitting"
      }
  ]
  "component" : [
    {
      "code" : {
        "coding" : [
          {   "system" : "http://loinc.org",  "code" : "8480-6",   "display" : "Systolic blood pressure"   }
        ],
        "text" : "Systolic blood pressure"
      },
      "valueQuantity" : { "value" : 109, "unit" : "mmHg", "system" : "http://unitsofmeasure.org", "code" : "mm[Hg]"  }
    },
   ...lines omitted...
  ]
}
```

2.3 Negation Issues

In certain clinical situations, it is important to document and communicate that a patient does *not* have a specific condition, medication allergy, past surgical procedure, or family history of a disorder. For decision-support and reporting purposes, it is important to represent such information in a structured and formal manner amenable to automated processing. However, the constructs for negation provided by the FHIR standard, in general, and by the US-FHIR implementation guide, in particular, are still immature and inconsistent. Specifically, there exist multiple, inconsistent ways of negating the same information, the precise scope of negation is poorly defined, and certain types of information lack a structured negation mechanism. All of these issues create the potential for negated information that is communicated in fully-compliant US-FHIR resources to be improperly stored or processed by recipients, with possible patient-safety implications. The sections below discuss each of these types of issues.

2.3.1 Negation Underspecification

Where negation is explicitly supported within US-FHIR resources, there often exist multiple inconsistent methods for negating the same data. For example, the US-FHIR *Condition* profile provides at least three distinct methods to explicitly indicate that a patient does not have a specific condition:

1. The data element “verificationStatus” may be populated with the coded value “refuted”.
2. The data element “clinicalStatus” may be populated with the coded value “inactive” or “resolved”
3. The data element “code” may be populated with a SNOMED-CT code of the type “Situation With Explicit Context”, which itself can denote the absence or negation of a clinical condition (for example, “No cardiovascular symptom” [SCT 162001003]).
4. The data element “code” may be populated with the specific SNOMED-CT code 160245001, which denotes “No current problems or disability.”

Further, no rules or constraints are specified for the *Condition* profile with respect to mutual dependencies among these attributes. For example, if the SNOMED-CT code for “No cardiovascular symptom” is populated as the value of the data element “code,” can the resource instance also contain the value “resolved” in the clinicalStatus data element and/or a value of “refuted” in the verificationStatus data element? What would be the correct meaning of such an instance if all three attributes denoted negation (e.g., a triple negative)? The *Condition* resource specifications does not address these situations.

When multiple methods exist to represent the same semantic meaning with limited constraints on their usage, it creates the potential that a receiving system will not recognize one or more of the methods or will misinterpret some combination of the methods when they are used together. In this case, it could result in a negated condition or allergy being interpreted as being present.

2.3.2 Undefined Scope of Negation

Where resource profiles include explicit attributes for negating information (such as the verificationStatus and clinicalStatus data elements described above), the FHIR standard does not define which other data element values the negation applies to and, therefore, the precise meaning of the resource instance. For example, the *Condition* resource profile includes various qualifying data elements for any condition that may be specified, such as “onsetDate”, “severity”, “bodySite”, etc. When such data elements have been populated and the verificationStatus or clinicalStatus elements denote negation, the FHIR resource definition does not specify whether the condition is negated in its entirety, or only in the specific context of the qualifier values.

For example, if a verificationStatus of “refuted” is specified for the condition “joint pain” with location “left knee”, severity “moderate”, and onsetDate “January 2018”, does this representation mean that the patient had no joint pain anywhere and of any severity and any duration at the time the condition was recorded, or only that the patient had no joint pain with the specific severity, location, and onsetDate indicated (although pain in a different joint/and or of a different severity may have been present)? The specific interpretation made by a system receiving such a FHIR resource instance could influence decision-support rules or even the display of the data to users, thereby potentially impacting patient care and patient safety if the interpretation were incorrect.

2.3.3 Absence of Negation for Certain Resource Profiles

In addition to FHIR resources with underspecified mechanisms for denoting negation and an undefined scope of negation, there exist other FHIR resources that include no explicit mechanisms for denoting negation. These include the resources Procedure and FamilyMemberHistory. In these cases, it is not possible to communicate, for example, that a patient has never had cardiac catheterization, or that neither the patient's father nor mother have a history of cancer[*]. To the extent that such clinical statements recorded in the patient's record are important for guiding appropriate clinical care, the absence of this negating capability could affect patient safety.

2.4 Modifying Elements

The FHIR data model includes the notion of "Modifying Elements"[4], which are defined as those data elements of a FHIR resource whose values could substantively modify the meaning of the entire resource instance. Negating data elements, such as clinicalStatus and verificationStatus, are examples of such modifying elements, but others also exist, such as the "notGiven" element of the Immunization resource and the "prohibited" element of the Care Plan resource.

The importance of modifying elements is that they have the ability not only to *add* detail about the primary clinical information communicated by the resource (such as adding the date of a patient's immunization or the severity of a patient's diagnosis), but they can fundamentally *change the meaning* of the primary clinical statement. Operationally, this means that receiving systems cannot choose to ignore modifying elements when processing or storing the data in a FHIR resource instance (such as whether the primary clinical statement is, in fact, absent), whereas they could otherwise ignore certain details of a resource instance that are not of interest (such as the specific person who recorded a clinical observation or the specific reason a procedure was performed, which do not change the fact that the observation was made or the procedure was performed). The following sections discuss the patient-safety implications of modifying elements.

2.4.1 Potential for Misinterpretation by Senders or Recipients

The patient-safety implications of modifying elements is that they may not be handled or interpreted correctly by receiving systems. As discussed in Sections 2.3.1 and 2.3.2, the semantics of modifying elements that explicitly negate resource instances are not well defined, and the meanings of their values may be interpreted differently by a sending and receiving system or used incorrectly by one or both. For example, the verificationStatus value of "refuted" is formally defined by HL7 as "Has been ruled out by diagnostic and clinical evidence," but could be easily misconstrued by a sending system to mean "has been characterized by the patient as not present" (although patients are sometimes mistaken or do not remember everything). If such an error is made, a receiving system hewing to the formal HL7 definition would conclude that some clinical condition is definitively absent and proceed with decision-support advice under that assumption, whereas the sending system would not have intended to communicate that level of certainly. Obvious patient-safety issues may result from such a misunderstanding between systems.

[*] With the exception of those procedure and family history observations where a discrete code within the designated value set exists that, itself, denotes the negation of the observation, such as a specific code for "no prior cardiac catheterization" or "no family history of cancer". Only a limited set of observations, however, have such explicitly negated codes defined for them.

2.4.2 Modifying Extension Elements

As mentioned in Section 2.2.2, the sender of a FHIR resource instance may add an “extension” element to any resource or any element within a resource. Such extension elements serve to add additional information to the resource data structure that has not been defined *a priori* in the base definition or the profile definition of the resource. Usually, if a receiving system does not recognize an extension element within a resource instance, it can choose to simply ignore it and only process those data elements it recognizes without risk of misinterpreting them.

In certain cases, however, such extension elements may, themselves, be Modifying Elements (as described in Section 2.4), such that they can change the fundamental meaning of the entire resource instance). These elements are referred to as “Modifier Extensions”. For example, a sending application could theoretically add a modifier extension element to a *MedicationRequest* resource instance that indicates the patient should NOT take the medication as prescribed for the next 3 days.

Modifier extensions must be recognized and correctly processed by receiving systems, i.e., such systems cannot safely interpret the meaning of the transmitted data without appropriately considering and applying the value of any modifier extensions. However, modifier extensions that are not part of the US-FHIR implementation guide could be entirely unexpected and unknown to a receiving system, although a resource instance containing such extensions would be entirely compliant with the US-FHIR implementation guide (such as the example above). In these cases, the HL7 FHIR standard specifies that the receiving system must do one of two things[5]:

1. Not process (i.e., reject) the resource instance in its entirety
2. Display an exception to a human user and ask her to appropriately instruct the system as to how to store or otherwise process the resource instance

Neither of these options is satisfactory with respect to patient safety. With the first option, important clinical information may be lost from a transmitted patient record. With the second option, only real-time user-facing applications could process the resource (not, for example, a “headless” application that automatically retrieves patient data prior to a patient’s appointment), and even then, the user may not be knowledgeable enough to correctly interpret the modifier extension and specify the appropriate processing.

Hence, the ability provided by the FHIR standard for sending systems to include arbitrary modifier extensions creates the potential for patient-safety risks, even if such extensions are infrequently used.

2.5 Must-Support Elements

An important specific feature of FHIR profiles is that they may designate certain data elements within a resource profile as “Must Support,” meaning that every implementation that produces or consumes instances of the resource profile mustbe aware of and process the data element in some meaningful way[6]. However, the FHIR standard does not explicitly define “meaningful support” in this context. Rather, the standard only requires that every resource profile that includes one or more “Must Support” data elements clearly specify the degree of support that is required for any such data[7]. Such specified support for receiving systems, for example, could include:

- The system must be able to store and retrieve the element.
- The system must display the element to the user and/or allow the user to capture the element via the UI.
- The element must appear in an output report.

- The element must be taken into account when performing decision support, calculations or other processing.

The Must-Support designation is obviously important for any Modifying Elements that appear in resource profiles (including Modifier Extensions), for the reasons described in Section 2.4. The US-FHIR profile, therefore, includes Must-Support designations for all such elements. However, the profile does not clearly specify the degree or nature of support that is required for most such data elements (contrary to the profiling requirements of the base FHIR standard), leaving it to the implementers of sending and receiving systems to interpret the meaning of “Must Support” for each one. Such latitude may result in different, and sometimes incorrect, interpretations across varying implementers, with the result that Must-Support elements are not correctly processed by at least some recipients of US-FHIR resource instances. To the degree that these elements denote clinically relevant information related to negation and other important semantics, the absence of clear instructions in the US-FHIR implementation guide regarding what support is required for these data elements may pose potential patient-safety risks.

2.6 Terminology Issues

The resource profiles of the US-FHIR implementation guide constrain the coding systems and value sets that may be used to populate patient data much more than the base FHIR resource specifications do. For example, the base FHIR specification for the *Condition* resource specifies no constraints on the codes used to identify a reported condition[8], whereas the US-FHIR profile for *Condition* requires this code to be drawn from a specific enumerated value set based on SNOMED-CT[9].

However, the terminology constraints specified by the US-FHIR implementation guide still leave substantial latitude in the way that certain clinical concepts are coded, creating the potential for receiving systems to not recognize or to misinterpret important patient data. When complete and accurate data are needed for clinical decision support, such gaps can lead to patient-safety problems. The terminology issues are of several types, as described below.

2.6.1 Overlapping Coding Systems/Value Sets

For certain data elements, the code constraint specifies multiple allowed coding systems or value sets, with overlapping content. For example, the *AllergyIntolerance* resource profile allows drugs to be encoded using either RxNorm, SNOMED-CT, or NDF-RT (with prioritization of NDF-RT for drug classes, and prioritization of RxNorm when an RxNorm code is applicable). However, if a sending system were to incorrectly use SNOMED-CT to transmit a drug class or a specific drug name, a receiving system that expects senders to scrupulously apply the prioritization rules might not recognize the code, resulting in a missed patient drug allergy.

Such coding latitude granted to the senders of FHIR resource instances requires recipients to recognize and process multiple coding systems, which not all recipients may be prepared to do correctly or at all. Although a prioritization of code-system usage is often specified by the resource profile, depending on the specific data to be transmitted, sending systems may not always be aware of this “fine print” in the specifications and may transmit values from improper coding systems. Most validation engines only recognize when a *disallowed* code has been used, rather than when an allowed code has been used incorrectly due a sender’s incorrect application of the prioritization rules.

As another example, the *Condition* resource profile allows patient problems to be represented using codes from either the SNOMED-CT “Clinical Finding” hierarchy or the SNOMED-CT “Situation-With-Explicit-Context” hierarchy (i.e., both hierarchies are included in the specified value set). For certain clinical concepts, however, these two encoding methods use different SNOMED-CT terminology models to encode essentially the same concepts. Specifically, the Clinical Finding hierarchy encodes the problem as a single finding concept (such as "Dizziness (finding)" [SCTID: 404640003]) whereas the Situation-With-Explicit-Context hierarchy encodes it as a combination of a finding and associated presence/absence attribute (such as "Dizziness present (situation)" [SCTID: 162260006]). In this example, both codes indicate that the patient experienced dizziness, but a receiving system may not expect the more complex "Situation-With-Explicit-Context" code, nor be able to correctly classify it in a disease hierarchy or map it to its local coding system. Again, if a previously documented problem of dizziness is not recognized and appropriately stored by the receiving system, a missed diagnosis or other risk to patient safety could ensue.

2.6.2 Optional Coding Systems/Value Sets

Other data elements within US-FHIR resource profiles may specify a single coding system or value set, but also allow codes outside of that enumerated list to be used when a resource instance is created by a sending system.

For example, the *Observation* resource profile specifies that implementers "SHOULD" use only codes from SNOMED-CT for coded results when populating the "value" data element. However, the HL7 FHIR conformance rules define the "SHOULD" constraint as "a best practice or recommendation to be considered by implementers within the context of their particular implementation."[10] This is in distinction to the "SHALL" constraint, which the conformance rules define as "an absolute requirement for all implementations." The result of a "SHOULD" rather than a "SHALL" constraint, therefore, is that senders of *Observation* resource instances are free to use any coding system whatsoever for the "value" field and still remain fully conformant to the US-FHIR implementation guide (i.e., the use of a non-standard code should not be flagged as an error by US-FHIR validation engines).

This optionality in coding could lead to patient-safety issues because the "value" data element in an Observation resource instance can contain clinically important coded data, such as the identity of cultured organisms for patients with serious infectious diseases. To the extent that receiving systems may not recognize the cultured organisms if they are not coded per the suggested (but not required) SNOMED-CT coding system, any decision-support or automated disease-surveillance systems containing logic that depends on coded culture results may fail, with potentially adverse patient-safety implications.

As another example, the *Condition* resource profile specifies that implementers must use codes from a designated "Problem" value set when populating the "code" data element, but this terminology constraint is designated as "extensible". The HL7 FHIR conformance rules define an "extensible" terminology constraint as follows[11]:

> *The code populating this data element SHALL be from the specified value set if any of the codes within the value set can apply to the concept being communicated. If the value set does not cover the concept (based on human review), alternate codes (or text) may be included instead.*

The "extensible" designation allows a sending system to transmit an arbitrary code if it determines, at its own discretion, that no code from the Problem value set exactly corresponds to the condition being reported. For example, if the sending system stores a patient problem as a highly specific ICD-10 code for which no exactly matching SNOMED-CT code exists (such as "Nodular lymphocyte predominant Hodgkin lymphoma, lymph nodes of inguinal region and lower limb" [ICD-10 C81.05]), then the sending system is allowed to send the ICD-10 code or even free text in the "code" element of the *Condition* resource instance. In such a case, it is again unlikely that an automated validation engine would detect this code substitution as an error even if a very similar SNOMED-CT code existed (for example, representing a slightly more general concept, such as "Hodgkin lymphoma, nodular lymphocyte predominance" [SCTID 70600005]).

A similar issue exists for the "code" data element in the *Observation* resource profile. In this case, the coding constraint specifies that LOINC codes must be used if an applicable LOINC code exists, but leaves it to the sending system to make that determination and to use an alternative coding system if needed.

3 Clinical Document Architecture and CDA Templates

The Clinical Document Architecture (CDA) is another formalism developed by HL7 to standardize clinical data representation for purposes of interoperability. The CDA is analogous to FHIR, although the underlying technologies and representation models are different. The CDA standardizes the contents and formatting of *XML documents* that may be used to store and/or transmit patient data. The current version of the CDA (Release 2.0) was published in 2005[12], and its data model is based on the HL7 Reference Information Model (RIM) and related artifacts (such as HL7 Version 3 data types and vocabulary domains)[13].

In general, the CDA may be used to define the structure of (1) entire *documents* (such as discharge summaries, referral notes, etc.), (2) the *sections* within document types (such as problem-list sections, medication-list sections, etc.), and (3) *entries* that may appear within specific sections (such as individual diagnoses or medications).

Like the base FHIR specification, the base CDA specification is also quite abstract and under-constrained, allowing conformant systems to generate highly variable representations of the same information, which undermines interoperability. To address this shortcoming, various initiatives have used the CDA as the basis for deriving XML document specifications that further constrain the CDA data model to try to bring document standardization closer to "plug-and-play." Notably, the Health Information Technology Standards Panel (HITSP) developed a family of such specifications for various document types approximately 10 years ago, including the continuity of care document (CCD)[14].

The primary mechanism for further constraining the CDA is through the definition of CDA "templates"[15], which specify additional constraints on the XML structure and coding of data values. In this sense, CDA templates are analogous to FHIR resource profiles, although templates allow further constraints on CDA documents, sections, or entries.

A set of mutually consistent CDA templates that have been specified by a particular group of organizations or for a particular purpose comprises a CDA "Implementation Guide". Organizations that wish to interoperate using such an implementation guide typically must conform to the entirety of the CDA document, section, and entry templates within it.

3.1 The Consolidated CDA Implementation Guide

In 2012, a consortium of entities including HL7, IHE, and the U.S. federal government developed a CDA implementation guide called the "Consolidated CDA" (C-CDA Release 1.1)[16]. C-CDA R1.1 integrated the templates developed earlier by HITSP, added modifications intended specifically to support the interoperability needs of the EHR Meaningful Use incentive program, and published the resulting implementation guide as a single consolidated standards-specification document, which facilitated review and implementation.

In 2015, the same consortium published Release 2.1 of the C-CDA specification to accommodate changes needed for the next iteration of the meaningful use program[17]. Today, conformance to C-CDA R1.1 and R2.1 are included among the EHR certification requirements under stage 1 and stage 2 (respectively) of the Meaningful Use incentive program. Therefore, many EHRs and other health information technologies have implemented these CDA implementation guides and are using them widely for the real-world exchange of patient data.

The additional constraints specified in the C-CDA R1.1 and R2.1 implementation guides (relative to the "base" CDA specification) greatly improve their effectiveness for interoperability. However, as with the US Core FHIR profiles, certain contents of the templates in these implementation guides also raise potential patient-safety issues. For purposes of reviewing these content issues, this whitepaper focusses on the C-CDA R2.1 implementation guide in the sections below. However, many of the same issues are present in the R1.1 version of the C-CDA.

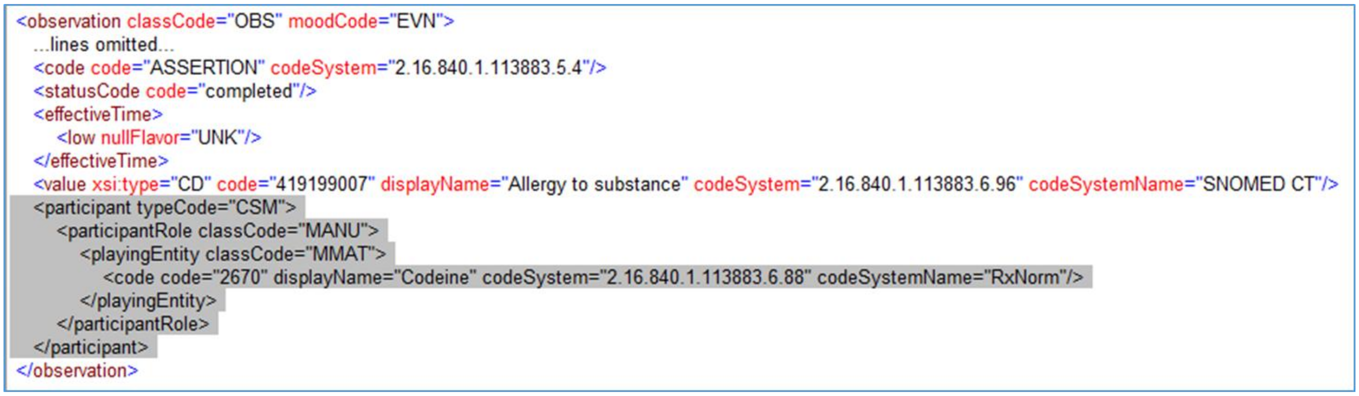
3.2 Unnecessary complexity

In certain cases, the specified representation of important information is unnecessarily complex, which could result in the erroneous population of this information by sending systems or misinterpretation of the information by receiving systems. Several such examples exist, as described below.

3.2.1 Allergenic Substance in "Allergies and Intolerances" Section

The C-CDA template *Allergy Intolerance Observation* specifies the required data structure for representing medications and other substances to which a patient is allergic. The identity of the allergenic substance, however, appears within the "participant" data element of this data structure (rather than the "value" or "code" data element), and the actual substance code therein is nested three layers deep. Figure 4 shows an example *Allergy Intolerance Observation* instance denoting that a patient is allergic to codeine.

Figure 4. An instance of the *Allergy Intolerance Observation* template showing deep nesting of the allergenic substance code.



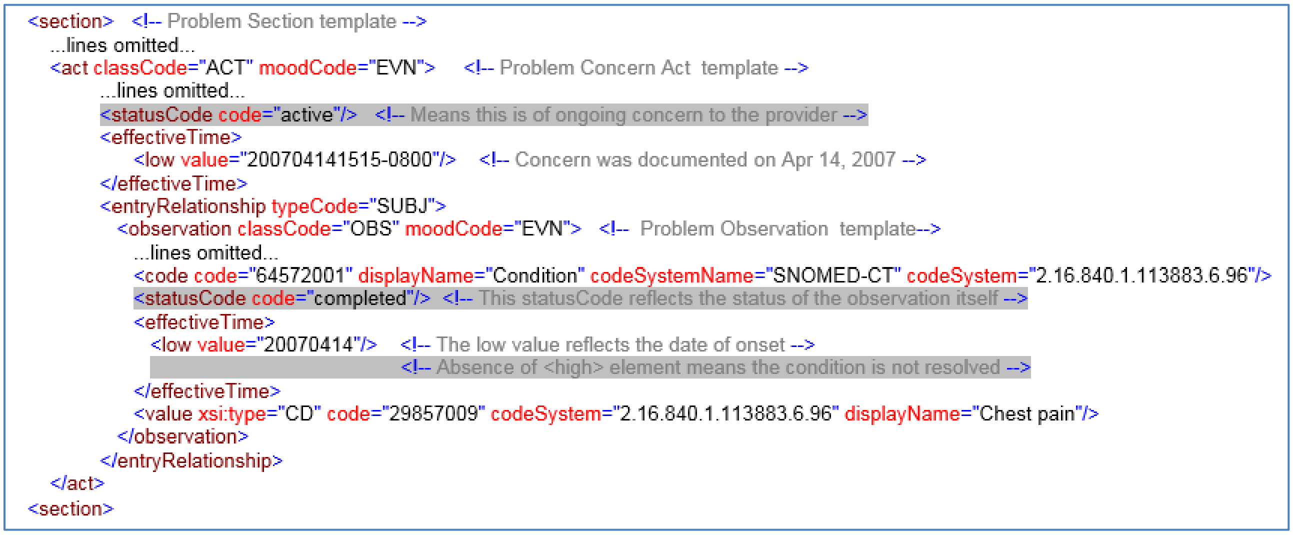
```
<observation classCode="OBS" moodCode="EVN">
  ...lines omitted...
  <code code="ASSERTION" codeSystem="2.16.840.1.113883.5.4"/>
  <statusCode code="completed"/>
  <effectiveTime>
    <low nullFlavor="UNK"/>
  </effectiveTime>
  <value xsi:type="CD" code="419199007" displayName="Allergy to substance" codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMED CT"/>
  <participant typeCode="CSM">
    <participantRole classCode="MANU">
      <playingEntity classCode="MMAT">
        <code code="2670" displayName="Codeine" codeSystem="2.16.840.1.113883.6.88" codeSystemName="RxNorm"/>
      </playingEntity>
    </participantRole>
  </participant>
</observation>
```

Further, the required typeCode value of "CSM" indicates that the allergenic substance is consumable (although it need not be, as allergens could be non-consumable substances, such as animal dander or pollen); the required typeCode value of "MANU" indicates that the substance is manufactured (again, although it could be naturally occurring); and the classCode value of "MMAT" indicates that it is a manufactured material (also not necessarily the case). The result is a convoluted data structure in which the template constraints may require instances to contain incorrect typeCode values. Such unnecessarily complex structures increase the potential that a patient's drug allergies will be improperly recorded by a sending system or misinterpreted by a receiving system, with obvious adverse consequences for patient safety.

3.2.2 statusCode Values

A second source of unnecessary complexity is the manner in which the status of clinical observations (such as diagnoses) is represented within the C-CDA *Problem Section* template. Figure 5 shows an example problem list recorded using this template, in this case containing a single problem of "Chest Pain". Note that a C-CDA problem list consists of three nested templates: *Problem Section* template (the entire problem list), Problem Concern Act template (a container for problems that are of current concern), and *Problem Observation* template (each actual problem of concern).

Figure 5. An instance of the *Problem Section* template showing how the status of a clinical problem is represented.



```
<section>  <!-- Problem Section template -->
  ...lines omitted...
  <act classCode="ACT" moodCode="EVN">   <!-- Problem Concern Act  template -->
      ...lines omitted...
      <statusCode code="active"/>  <!-- Means this is of ongoing concern to the provider -->
      <effectiveTime>
         <low value="200704141515-0800"/>   <!-- Concern was documented on Apr 14, 2007 -->
      </effectiveTime>
      <entryRelationship typeCode="SUBJ">
        <observation classCode="OBS" moodCode="EVN">   <!--  Problem Observation  template-->
          ...lines omitted...
          <code code="64572001" displayName="Condition" codeSystemName="SNOMED-CT" codeSystem="2.16.840.1.113883.6.96"/>
          <statusCode code="completed"/>  <!-- This statusCode reflects the status of the observation itself -->
          <effectiveTime>
            <low value="20070414"/>    <!-- The low value reflects the date of onset -->
                                       <!-- Absence of <high> element means the condition is not resolved -->
          </effectiveTime>
          <value xsi:type="CD" code="29857009" codeSystem="2.16.840.1.113883.6.96" displayName="Chest pain"/>
        </observation>
      </entryRelationship>
  </act>
<section>
```

Intuitively, one would expect the "statusCode" data element within the nested *Problem Observation* template to denote whether the recorded condition is active or resolved, but this is not the case. In fact, the C-CDA *Problem Observation* template requires that this statusCode value be hard-coded to "completed" (see Figure) indicating that the process of observing the problem has been completed. The status of the problem, itself, is denoted by the value of the "effectiveTime" data element, specifically by the value of the "high" sub-element. If no "high" element exists to denote the date at which the problem resolved, then the problem should be deemed active. If a date is recorded within the "high" element, then the problem should be deemed resolved.

Again, this representational rubric is convoluted and subject to error on the part of both the sending system and the receiving system. For example, a recipient of the data shown in Figure 5 could misinterpret the value "completed" within the "statusCode" data element to indicate that the condition (chest pain, in this case) had resolved when it had not (as indicated by the absence of a "high" data element within the "effectiveTime" element). Such an error could have potentially serious patient-safety implications.

Conversely, the appearance of another "statusCode" data element within the *Problem Concern Act* template could be misinterpreted by a receiving system to indicate that a problem was active when it no longer was. Specifically, per the *Problem Concern Act* template specifications, a value of "active" in this data element indicates that the contained problem is of active concern to the clinician, not necessarily that the problem itself is still active (versus being resolved). In certain cases, past incidents of a condition or symptom (such as chest pain) may remain of concern to a clinician even if the condition or symptom is no longer present at the time the problem list is transmitted. In the example of Figure 5, even if the value of the "high" data element within the *Problem Observation* template were populated rather than omitted (indicating that the chest pain had resolved), a receiving system could misinterpret the "active" value of the "statusCode" data element in the *Problem Concern Act* template as indicating that the chest pain was still active. Again, such a misunderstanding could result in diagnostic or treatment errors to the patient, owing to the excessive complexity of the *Problem Section* template and its nested sub-templates.

Further complicating the situation, the specification of the *Problem Observation* template states "If the problem is known to be resolved, but the date of resolution is not known, then the "high" element SHALL be present, and the nullFlavor attribute SHALL be set to 'UNK'. Therefore, the existence of a "high" element within a problem does indicate that the problem has been resolved." Again, a receiving system could misunderstand the subtle distinction between a "high" element being absent (indicating that the problem is still active) and a "high" element being present and containing no actual date, but having its nullFlavor attribute set to "unknown" (indicating that the problem is resolved).

3.3 Potentially Missing "Required" Values

Templates within the C-CDA implementation guide specify various data elements as required because these elements communicate important clinical information about patients. Such data elements include the following (with the template in which they appear and their HL7 v3 data type also shown):

Template Name	Data Element	DataType	Description
Vital Sign Observation	value	PQ	Value and unit of measure for the vital sign
Immunization Activity	effectiveTime	TS	Date/time at which immunization was given
Problem Observation	effectiveTime	TS	Date/time of problem onset and resolution
Medication Activity	doseQuantity	PQ	Dose of medication prescribed/administered
Medication Activity	effectiveTime	TS	Date/time when medication started and stopped

The manner in which templates specify that a certain data element must be populated is by declaring that the element "SHALL contain" a value. For example, the template specification for *Vital Sign Observation* declares the following for the value/unit data element: "SHALL contain exactly one [1..1] value with data type="PQ"."

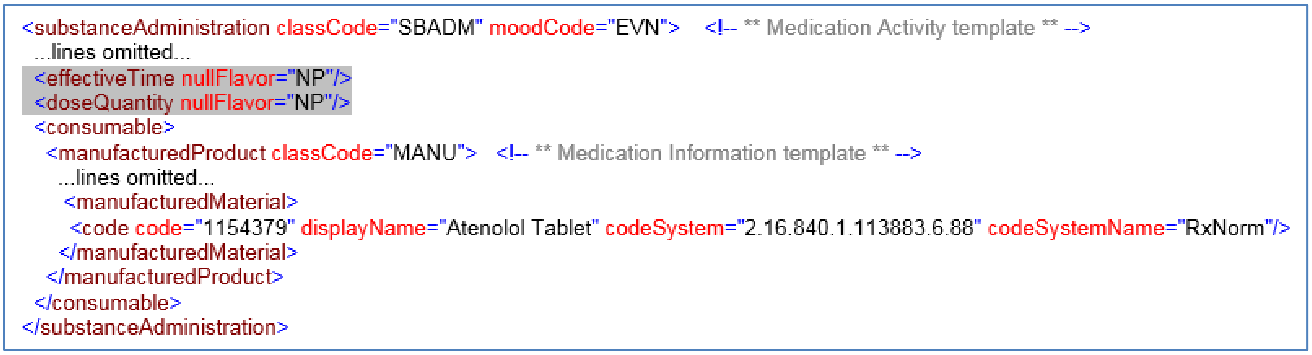
However, the C-CDA implementation guide allows many required data elements to contain no value, but instead a "nullFlavor" annotation that indicates why the value is not present. Specifically, the C-CDA implementation guide states

> *Any SHALL, SHOULD or MAY conformance statement may use nullFlavor, unless the nullFlavor is explicitly disallowed (e.g., through another conformance statement which includes a SHALL conformance for a vocabulary binding to the @code attribute, or through an explicit SHALL NOT allow use of nullFlavor conformance)."*[18]

Note that the CDA Release 2 specification on which the C-CDA is based allows such nullFlavor annotations to be applied to many data types, including all of those in the examples above[19].

Hence, the data shown in Figure 6 regarding a patient's prescription for Atenolol would be fully conformant with the C-CDA implementation guide, although it omits the timing and dose of the medication (note that the nullFlavor value of "NP" signifies that the value is "not present" in the transmitted data, with no reason specified).

Figure 6. An instance of the *Medication Activity* template showing how the values of even required data elements may be substituted with "nullFlavor" placeholders.



```
<substanceAdministration classCode="SBADM" moodCode="EVN">   <!-- ** Medication Activity template ** -->
 ...lines omitted...
 <effectiveTime nullFlavor="NP"/>
 <doseQuantity nullFlavor="NP"/>
 <consumable>
   <manufacturedProduct classCode="MANU">   <!-- ** Medication Information template ** -->
     ...lines omitted...
      <manufacturedMaterial>
        <code code="1154379" displayName="Atenolol Tablet" codeSystem="2.16.840.1.113883.6.88" codeSystemName="RxNorm"/>
     </manufacturedMaterial>
   </manufacturedProduct>
 </consumable>
</substanceAdministration>
```

Again, the latitude for sending systems to omit important clinical data through the "nullFlavor" mechanism is allowed by the C-CDA specification, although it creates the potential for adverse patient-safety consequences. Further, such documents will pass C-CDA validation testing in these cases. Note that such situations may arise even when a sending system locally stores the omitted data, but just cannot format them correctly per the C-CDA implementation guide (e.g., if the data are stored as free-text strings).

3.4 Negation Issues

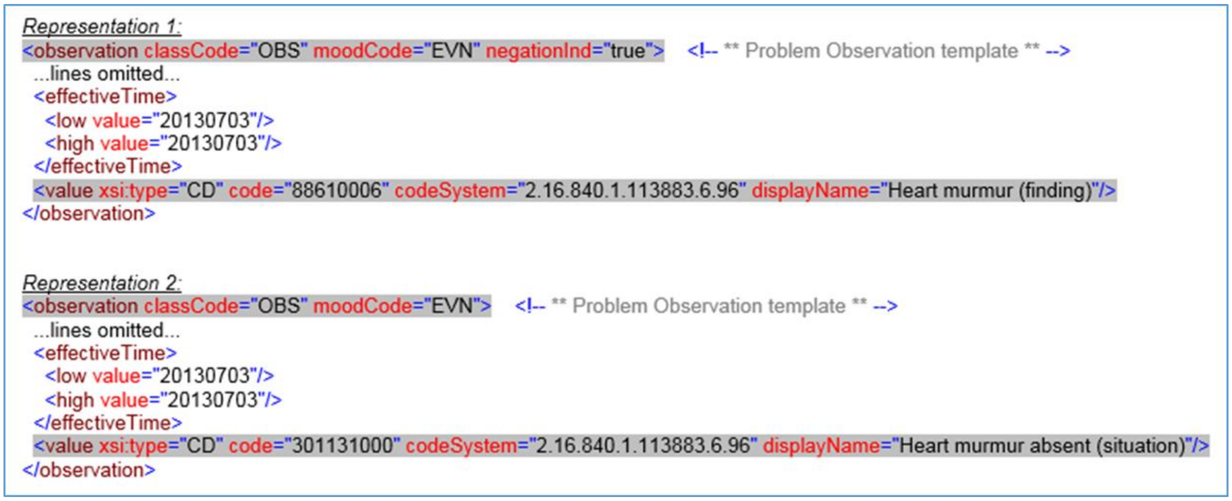
As with FHIR resource profiles, C-CDA templates also have insufficiently defined mechanisms for negating clinical statements. These shortcoming include multiple mechanisms for negating the same information (underspecification), undefined scope of negation, and absence of mechanisms to negate certain clinical observations.

3.4.1 Underspecification

Certain C-CDA templates provide multiple methods to negate the same information. For example, the *Problem Observation* template includes the data element "negationInd" (negation indicator), which may optionally be used to negate the condition denoted by the code that appears in the "value" data element.

However, the value set for codes in the "value" data element also allows members of the SNOMED-CT "Situation-With-Explicit-Context" hierarchy. Concepts in this hierarchy can, themselves, include negation semantics, such as "Heart murmur absent (situation)" [SCTID: 301131000]. Figure 7 shows these two alternative methods for negating a condition that are allowed by the C-CDA *Problem Observation* template. This redundancy of representation methods increases the potential that a receiving system may not recognize them both, with adverse patient-safety results.

Figure 7. Alternative methods for negating a clinical statement in the *Problem Observation* template.



```
Representation 1:
<observation classCode="OBS" moodCode="EVN" negationInd="true">   <!-- ** Problem Observation template ** -->
  ...lines omitted...
  <effectiveTime>
    <low value="20130703"/>
    <high value="20130703"/>
  </effectiveTime>
  <value xsi:type="CD" code="88610006" codeSystem="2.16.840.1.113883.6.96" displayName="Heart murmur (finding)"/>
</observation>


Representation 2:
<observation classCode="OBS" moodCode="EVN">   <!-- ** Problem Observation template ** -->
  ...lines omitted...
  <effectiveTime>
    <low value="20130703"/>
    <high value="20130703"/>
  </effectiveTime>
  <value xsi:type="CD" code="301131000" codeSystem="2.16.840.1.113883.6.96" displayName="Heart murmur absent (situation)"/>
</observation>
```

Another type of representational redundancy occurs in the *Smoking Status – Meaningful Use* template, where an unknown smoking status may be represented in two separate ways per the C-CDA implementation guide:

1. By populating the "value" data element with the SNOMED-CT code 266927001 ("Unknown if ever smoked"):
   ```
   <value code="266927001 " codeSystem="2.16.840.1.113883.6.96"
          displayName="Unknown if ever smoked"/>
   ```

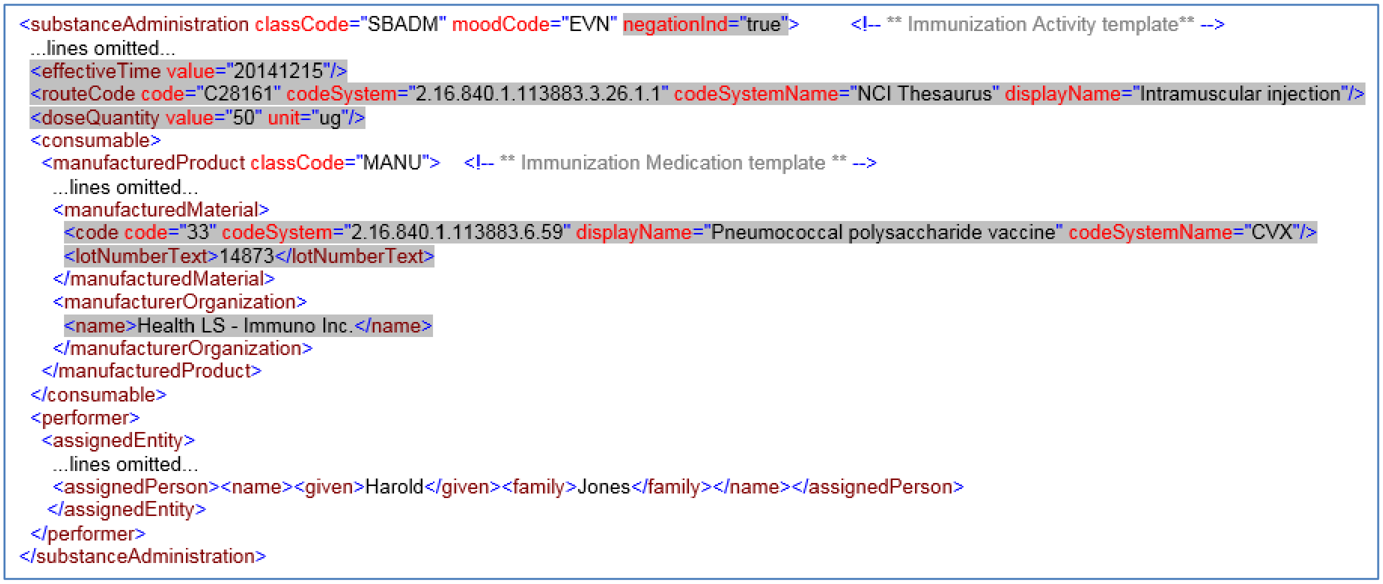
2. By including a "value" data element with no coded value, but rather the nullFlavor annotation "UNK" (unknown):
   ```
   <value nullFlavor="UNK"/>
   ```

Again, a receiving system may not recognize both of these allowed representation methods.

3.4.2 Unclear Scope of Negation

In certain cases, when the clinical statement within a C-CDA template is negated via the "negationInd" attribute, it may not be clear which specific aspects of the clinical statement are necessarily untrue and which may remain true. For example, the *Immunization Activity* template instance in Figure 8 suggests that no immunization was given for pneumococcus.

Figure 8. A negated instance of the *Immunization Activity* template.



```
<substanceAdministration classCode="SBADM" moodCode="EVN" negationInd="true">       <!-- ** Immunization Activity template** -->
  ...lines omitted...
  <effectiveTime value="20141215"/>
  <routeCode code="C28161" codeSystem="2.16.840.1.113883.3.26.1.1" codeSystemName="NCI Thesaurus" displayName="Intramuscular injection"/>
  <doseQuantity value="50" unit="ug"/>
  <consumable>
    <manufacturedProduct classCode="MANU">   <!-- ** Immunization Medication template ** -->
      ...lines omitted...
      <manufacturedMaterial>
        <code code="33" codeSystem="2.16.840.1.113883.6.59" displayName="Pneumococcal polysaccharide vaccine" codeSystemName="CVX"/>
        <lotNumberText>14873</lotNumberText>
      </manufacturedMaterial>
      <manufacturerOrganization>
        <name>Health LS - Immuno Inc.</name>
      </manufacturerOrganization>
    </manufacturedProduct>
  </consumable>
  <performer>
    <assignedEntity>
      ...lines omitted...
      <assignedPerson><name><given>Harold</given><family>Jones</family></name></assignedPerson>
     </assignedEntity>
  </performer>
</substanceAdministration>
```

Intuitively, a receiving system might interpret the negated statement in Figure 8 to mean that no dose of pneumococcal vaccine at all was given to the patient on the indicated date. However, the formal definition of the “negationInd” attribute, per C-CDA standard specification, suggests something slightly different: The statement means only that no pneumococcal vaccine was given *on the stated date via the stated route in the stated dose quantity from the stated lot number provided by the stated manufacturer.* In fact, a pneumococcal vaccine may have been given to the patient *from a different lot number or in a different dose quantity* on the stated date. At the same time, other data-element values in the *Immunization Activity* template may be intended to remain un-negated (such as the performer of the vaccination). The exact scope of the negation is, therefore, unclear. The lack of clarity is underscored by the following language in the specification of the HL7 v3 reference information model on which the CDA specification and derived templates are based[20]:

> *The negationInd negates the Act as described by the descriptive properties (including Act.code, Act.effectiveTime, Observation.value, Act.doseQty, etc.) and any of its components…For example, a highly confidential order written by Dr. Jones, to explicitly not give "succinyl choline" for the "reason" (ActRelationship) of a history of malignant hyperthermia (Observation) negates the descriptive properties "give succinyl choline" (Act.code), but it is still positively an order and written by Dr. Jones and for patient John Smith, and the reason for this order is the patient’s history of malignant hyperthermia. However, additional detail in descriptive attributes will limit the effective scope of the negation. For example, had the order not to give a substance included a doseQuantity, it would mean that the substance should not be given at that particular dose, but does not prohibit medication at any other dose.*

Given the complexity and inscrutability of the intended semantics of clinical statements when they are negated via the “negationInd” attribute, it’s not unlikely that receiving systems could misinterpret the meaning of such statements, and decision-support rules that depend on correct interpretations of these data could fail, with adverse patient-safety consequences.

3.4.3 Absence of Explicit Negation for Certain C-CDA Templates

At the same time, other C-CDA templates provide no obvious mechanism to negate certain clinical observations, even when such negation may be needed. For example, the *Result Observation* template lacks an explicit negation mechanism, although one may be needed to report the negative qualitative results of diagnostic procedures (e.g., an imaging exam that reveals "no pleural effusion").

Specifically, the *Result Observation* template, itself, includes no "negationInd" attribute. Sending systems attempting to encode such negative results will need to resort to other, less precisely specified, mechanisms, such as representing the observation as a coded value in the "code" data element (e.g., "Pleural Effusion" [SCTID 60046008]) and the negation of the observation as a Boolean-typed value in the "value" data element (e.g., "false"). Alternatively, sending systems might resort to using local codes to represent negated results, believing that no standard codes or other standardized methods exist to represent them. This approach would be entirely consistent with the C-CDA implementation guide because the "value" data element in the *Result Observation* template allows the use of codes from coding systems other than LOINC if LOINC does not include an applicable code.

In these cases, receiving systems may not recognize the use of a Boolean-type value or a local code used to represent the negated observation and may therefore misinterpret the meaning of the transmitted data. Again, these situations could result in adverse patient-safety outcomes.

3.5 Terminology Issues

As with the FHIR resource profiles, the terminology constraints specified by the C-CDA templates still leave substantial latitude in the way that certain clinical concepts are coded, creating the potential for receiving systems to not recognize or to not correctly interpret important patient data.

3.5.1 Overlapping Coding Systems/Value Sets

In certain cases, C-CDA templates allow sending systems to code certain data elements using multiple coding systems or value sets with overlapping content, including local, non-standard coding systems. This under-constraint of coded terminologies may require receiving systems to recognize and correctly process a variety of different terminologies, increasing the chances that an important clinical datum will not be correctly processed.

For example, the *Result Organizer* template specifies that the identity of a reported test panel "SHOULD be selected from LOINC OR SNOMED CT, and MAY be selected from CPT-4; Laboratory results SHOULD be from LOINC or other constrained terminology named by the US Department of Health and Human Services Office of National Coordinator or other federal agency." Although the specification suggests a prioritization of LOINC over other coding systems for lab results, it does not require even that, allowing SNOMED-CT and CPT-4 codes to be substituted instead, at the discretion of the sending system.

As another example, the *Allergy-Intolerance Observation* template specifies that the substances to which a patient is allergic must be represented using a code from a specified value set, but this value set includes codes from four overlapping coding systems: NDFRT drug class codes, RxNORM ingredient codes, UNII ingredient codes, and SNOMED CT substance codes. The specification clarifies the process by which a sending system should select the correct code as follows: "The expectation for use is that the chosen concept identifier for a substance should be appropriately specific and drawn from the available code systems in the following priority order: NDFRT, then RXNORM, then UNII, then SNOMED CT." However, any of the four coding systems are allowed and would be accepted by a C-CDA validation check.

Finally, in the *Family History Observation* template, the condition reported for a designated family member must be selected from a value set that includes both SNOMED-CT finding codes (such as "Blood coagulation disorder (disorder)" [SCTID 64779008]) as well as SNOMED-CT situation-with-explicit-context codes (such as "Family history of blood coagulation disorder (situation)" [SCTID 108801000119109]). In the context of a family history, these two codes have the exact same semantics, and either may be selected by the sending system, although certain receiving systems may not be prepared to process a situation-with-explicit-context codes correctly (for example, not being able to identify the actual disorder within the pre-coordinated concept).

3.5.2 Optional Coding Systems/Value Sets

Beyond the complexity arising from data-element values that may be coded using different, overlapping coding systems, the C-CDA implementation guide also introduces complexity and uncertainty by under-constraining whether values must be coded using a standard coding system at all. Specifically, within numerous C-CDA templates, the value set specified for an important code has a "SHOULD" designation, instead of a "SHALL" designation, so senders can choose to use other than the designated terminologies and still be compliant with the CCD standard (although such data may be useless to the recipient from a semantic interoperability perspective – e.g., for automated decision support).

Specifically, the "SHALL" and "SHOULD" designations are defined as follows in the C-CDA specification[21]:

- <u>SHALL</u>: An absolute requirement.
- <u>SHOULD</u>: Best practice or recommendation. There may be valid reasons to ignore an item, but the full implications must be understood and carefully weighed before choosing a different course

Such under-constraining of terminology via a "SHOULD" designation occurs in at least the following C-CDA templates:

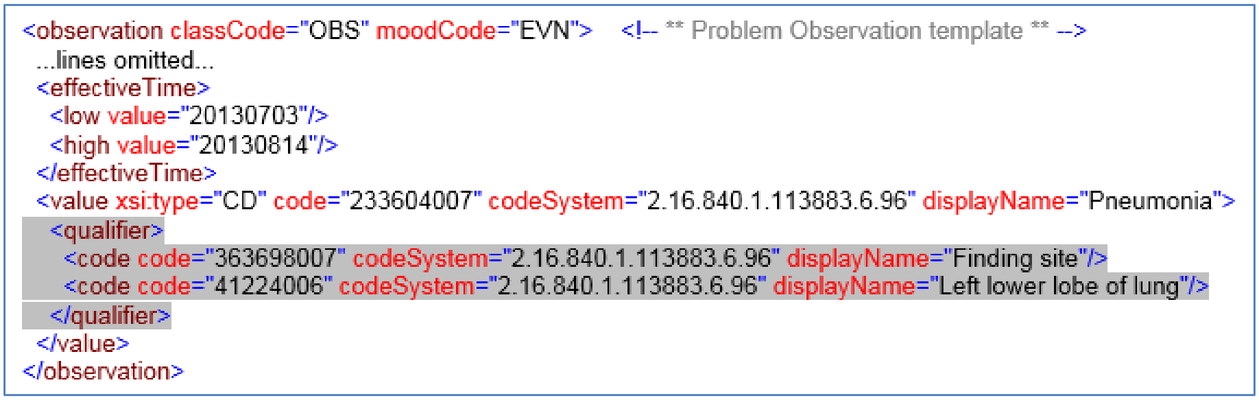
Template	Data Element	Description
Vital Sign Observation	code	The type of reported vital sign (BP, HR, etc.)
Result Observation	code	The reported test (Serum sodium, X-Ray, etc.)
Result Observation	value/@code	The reported value of a non-numeric test result (e.g., Pneumococcus culture result)
Problem Observation	value/@code	The reported problem (Diabetes, CHF, etc.)
Procedure Activity	code	The procedure performed (Stent placement, Polypectomy, etc.)
Plan of Treatment	code	The planned action (Colonoscopy, Post-op visit, etc.)
Social History Observation	code	The type of reported attribute (Alcohol intake, Tobacco use, etc.)
Family History Organizer	relatedSubject/code	The family member whose history is reported (Aunt, Grandparent, etc.)

3.5.3 Underspecification of Post-Coordinated Expressions in Problem Observations

The post-coordination of clinical concepts using a well-defined terminology model and syntax provides the opportunity to formally code a much greater variety of clinical statements than possible using only the pre-coordinated contents of even very large coding systems. Such post-coordination has the benefits of preserving more of the information from the originally recorded patient data in a sending system without having to resort to the transmission of local proprietary codes or free text.

The C-CDA *Problem Observation* template ostensibly supports the inclusion of post-coordinated coding expressions in problem list entries via the “qualifier” data element. Specifically, this data element can be used to further detail in a structured fashion the condition coded in the “value” data element of the template. For example, Figure 9 shows a *Problem Observation* template in which the “qualifier” data element is used to further specify the location of the reported “pneumonia” using additional SNOMED-CT codes (i.e., a post-coordinated expression).

Figure 9. An instance of the *Problem Observation* template containing a post-coordinated expression.



```
<observation classCode="OBS" moodCode="EVN">    <!-- ** Problem Observation template ** -->
  ...lines omitted...
  <effectiveTime>
    <low value="20130703"/>
    <high value="20130814"/>
  </effectiveTime>
  <value xsi:type="CD" code="233604007" codeSystem="2.16.840.1.113883.6.96" displayName="Pneumonia">
    <qualifier>
      <code code="363698007" codeSystem="2.16.840.1.113883.6.96" displayName="Finding site"/>
      <code code="41224006" codeSystem="2.16.840.1.113883.6.96" displayName="Left lower lobe of lung"/>
    </qualifier>
  </value>
</observation>
```

Although this mechanism for representing post-coordinated expressions in the *Problem Observation* template is potentially useful, the C-CDA template specification does not, in fact, specify or constrain how valid post-coordinated expressions can be constructed. In fact, the template description states only the following:

> *The observation/value and all the qualifiers together (often referred to as a post-coordinated expression) make up one concept. Qualifiers constrain the meaning of the primary code, and cannot negate it or change its meaning. Qualifiers can only be used according to well-defined rules of post-coordination and only if the underlying code system defines the use of such qualifiers or if there is a third code system that specifies how other code systems may be combined.*

Although SNOMED-CT (the recommended coding system for representing problems in the *Problem Observation* template) does specify a terminology model and compositional grammar (syntax) for representing post-coordinated expressions, the C-CDA template specification does not indicate how expressions constructed using this grammar should be specifically represented within template instances (Note that the example in Figure 9 provides just one possible representation, developed by the author).

Because SNOMED-CT post-coordinated concepts can constitute complex, multiply-nested expressions, the absence of a formal syntax within the C-CDA template for representing such expressions leaves latitude for different sending systems to represent the same expressions differently, which complicates or even obviates the ability for receiving systems to recognize and process such expressions correctly. When the expressions contain important information relevant to decision-support processes, the result of this underspecification may be an adverse patient-safety outcome.

4 Summary and Next Steps

This whitepaper presents a number of data-modeling issues within the US-FHIR and C-CDA implementation guides that create the potential for patient-safety problems. Although certain of these issues may be fundamental to the underlying formalisms of FHIR and CDA, others are a result of specific choices made in defining US-FHIR resource profiles and C-CDA templates. A forthcoming whitepaper on this topic will suggest ways that these profiles and templates could be modified to reduce or eliminate the potential patient-safety problems raised by their current designs.

5 References

[1] http://www.hl7.org/fhir/ (accessed 4/6/2018)

[2] Richardson L., Ruby S. RESTful Web Services. O'Reilly Media, December, 2008.

[3] http://hl7.org/fhir/extensibility.html#Extension (accessed 4/6/2018)

[4] https://hl7.org/implement/standards/fhir/conformance-rules.html#isModifier (accessed 4/6/2018)

[5] http://hl7.org/fhir/extensibility.html#modifierExtension (accessed 4/6/2018)

[6] https://hl7.org/implement/standards/fhir/conformance-rules.html#mustSupport (accessed 4/6/2018)

[7] https://www.hl7.org/fhir/profiling.html#mustsupport (accessed 4/6/2018)

[8] http://hl7.org/fhir/condition.html (accessed 4/6/2018)

[9] http://www.hl7.org/fhir/us/core/StructureDefinition-us-core-condition.html (accessed 4/6/2018)

[10] https://www.hl7.org/fhir/conformance-rules.html (accessed 4/6/2018)

[11] http://hl7.org/fhir/terminologies.html#extensible (accessed 4/6/2018)

[12] http://hl7.ihelse.net/hl7v3/infrastructure/cda/cda.html (accessed 4/6/2018)

[13] http://hl7.ihelse.net/hl7v3/infrastructure/rim/rim.html (accessed 4/6/2013)

[14] https://ushik.ahrq.gov/portals/hitsp/reference_documents/HITSP_V1.0_2008_TN901_-_Technical_Note_for_Clinical_Documents.pdf (accessed 4/6/2018)

[15] Kernan R. Clinical Document Architecture (CDA), Consolidated-CDA (C-CDA) and their Role in Meaningful Use (MU). ONC Tutorial, 8/22/2012 (https://www.healthit.gov/sites/default/files/cda_c-cda_theirrole_in_mu.pdf accessed 4/6/2018)

[16] HL7 Implementation Guide for CDA® Release 2: IHE Health Story Consolidation, DSTU Release 1.1 (US Realm). HL7 International, July 2012. (http://www.hl7.org/implement/standards/product_brief.cfm?product_id=258 accessed 4/6/2018)

[17] HL7 Implementation Guide for CDA® Release 2: Consolidated CDA Templates for Clinical Notes (US Realm) Draft Standard for Trial Use Release 2.1. HL7 International, August 2015. (http://www.hl7.org/implement/standards/product_brief.cfm?product_id=379 accessed 4/6/2018)

[18] HL7 Implementation Guide for CDA® Release 2: Consolidated CDA Templates for Clinical Notes (US Realm) Draft Standard for Trial Use Release 2.1 – Volume 1 – Introductory Material, p. 26. HL7 International, August 2015 (http://www.hl7.org/implement/standards/product_brief.cfm?product_id=379 accessed 4/6/2018)

[19] https://www.hl7.org/documentcenter/public_temp_25D88232-1C23-BA17-0CD7E8EBF33BEBB2/wg/inm/datatypes-its-xml20050714.htm (accessed 4/6/2018)

[20] http://hl7.ihelse.net/hl7v3/infrastructure/rim/rim.html Sec. 6.5.6 (accessed 4/6/2018)

[21] HL7 Implementation Guide for CDA® Release 2: Consolidated CDA Templates for Clinical Notes (US Realm) Draft Standard for Trial Use Release 2.1 – Volume 1 – Introductory Material, p. 33. HL7 International, August 2015 (http://www.hl7.org/implement/standards/product_brief.cfm?product_id=379 accessed 4/6/2018)