July 8, 2009
Version 1.0

HITSP Data Architecture Technical Note

HITSP/TN903

Submitted to:

Healthcare Information Technology Standards Panel

Submitted by:

Data Architecture (Elements/Templates/Value Sets) Tiger Team

DOCUMENT CHANGE HISTORY

Version Number	Description of Change	Name of Author	Date Published
0.0.1	Review Copy	Data Architecture (Elements/Templates/Value Sets) Tiger Team	June 30, 2009
1.0	Released for Implementation	Data Architecture (Elements/Templates/Value Sets) Tiger Team	July 8, 2009

TABLE OF CONTENTS

FIGURES AND TABLES

1.0 INTRODUCTION

1.1 PURPOSE AND SCOPE

The purpose of this document is to describe the Healthcare Information Technology Standards Panel (HITSP) Data Architecture, and the related processes and tools that HITSP uses to identify the data elements, templates and value sets used in Information Exchanges. It explains how within HITSP Specifications:

- base and composite standards are related to the data architecture
- data elements are harmonized across various standards
- constraints are applied within HITSP Specifications
- metadata registries support development and implementation

1.2 INTENDED AUDIENCE

The audience for this document includes:

- Users of Electronic Health Record (EHR) Systems wanting to understand the interoperable data elements being exchanged by those systems
- Experts in clinical practice developing guidelines for care or clinical decision support or creating quality measures
- Policy-makers and influencers establishing national and regional policies with respect to interoperable Information Exchange
- Implementers of and integrators with EHR systems needing to understand the data elements being exchanged using HITSP selected standards

1.3 CONVENTIONS USED IN THIS DOCUMENT

The key words **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT** and **MAY** are to be interpreted as described in RFC 2119 and will appear when used in that fashion in this **TYPEFACE**.

Terms defined in the definition section will appear in this **Typeface** and are linked to their definitions the first time they appear within a section.

CC-1-[1] Constraints appear using this style within the document and SHALL have a unique identifier within the document.

Material that has been adopted without change from other publications will be introduced with the phrase [Adopted from Publication Name].

Material that has been adapted from other publications, but which may have been revised from the originally published content will be introduced with the phrase *[Adapted from Publication Name]*.

1.4 DOCUMENT ORGANIZATION AND STRUCTURE

1. Introduction – This section
2. Executive Summary – Summarizes this document
3. Background – Provides an overview of healthcare related standards and how they relate to the data architecture, and explains how these are used in HITSP Specifications
4. HITSP Data Architecture – Defines key concepts used in this document, and describes the HITSP Harmonization Framework
5. Use of Metadata Registries – Describes a metadata registry, and how these are used to support navigation of the selected standards and HITSP Specifications

Users of EHR systems will be most interested in Sections 2.0 through 4.0, and may want to review Sections 5.0 to understand how to inspect the data elements used in HITSP Specifications

Experts in clinical practice will be most interested in Sections 4.0 and 5.0, and may want to investigate Section 3.0 to understand how the data elements of different standards that they may not be familiar with are organized within the standard.

Policy makers and influences will be most interested in Section 2.0 and 4.0. They may want to read Section 3.0 to understand how data elements are organized by the different standards.

Implementers and integrators will want to review Section 4.0 and 5.0 to determine how the HITSP Data Architecture framework impacts implementations of Information Exchanges. They may find Section 3.0 instructive for standards that they may not be familiar with.

1.5 COPYRIGHT INFORMATION

This document utilizes material from several other publications. When material from another publication is used, it will be introduced as: *[<Adopted/Adapted> from <Publication Name>]*

Material that is simply copied from the source publication will be marked as *Adopted.* Material that has been altered from the source publication will be marked as *Adapted.* Where necessary, the end of the quoted material will conclude with *[End of <adopted/adapted> <Source> Material].*

Material credited to the *OASIS EbXML Glossary* comes from the ebXML Glossary Version 0.99, 2001, found at http://www.ebxml.org/specs/ebGLOSS.pdf, and is Copyright © UN/CEFACT and OASIS, 2001. All Rights Reserved. This material is used with permission. The following copyright notice applies to this material.

> This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the ebXML organizations, except as needed for the purpose of developing standards in which case the procedures for copyrights defined in the ebXML Standards process must be followed, or as required to translate it into languages other than English.

Material credited to HL7 is adopted or adapted from one of the following publications and is used with permission:

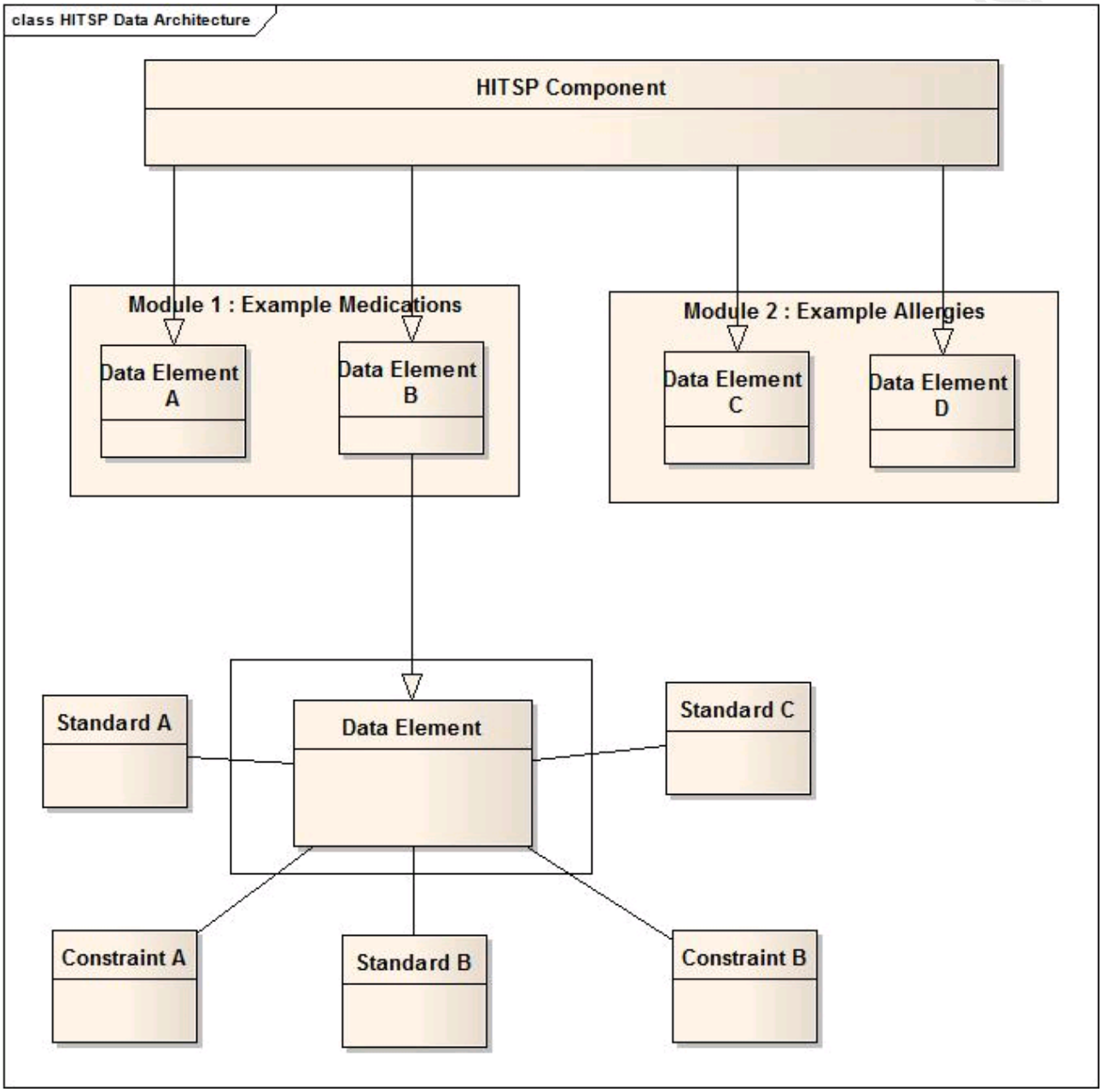
HL7 Version 3 Standard: Core Principles and Properties of Version 3 Models, Release 1 For Comment Only Ballot 1 - May 2009, found at www.hl7.org/v3ballot/html/infrastructure/coreprinciples/v3modelcoreprinciples.htm and is © 2008 Health Level Seven™, Inc. All Rights Reserved.

HL7 Version 3 Standard: Specification and Use of Reusable Constraint Templates, Release 2, February 2008 found at www.hl7.org/v3ballot/html/infrastructure/templates/templates.htm and is © 2008 Health Level Seven™, Inc. All Rights Reserved.

2.0 EXECUTIVE SUMMARY

The HITSP Data Architecture provides a framework that allows HITSP to identify similar data elements used in a wide variety of healthcare standards and to constrain its expression across those standards in a consistent fashion to ensure maximum interoperability. A high level overview of the architecture is shown below in Figure 2-1.

Figure 2-1 HITSP Data Architecture



The data architecture is hierarchical. The top level identifies the basic unit of data used in a single Information Exchange as a HITSP Component. Components are made up of data elements. The data

elements are organized categorically into modules in the HITSP/C83 CDA AND CCD Content Modules[1] specification by their clinical purpose (e.g., problems, medications, allergies, immunizations, et cetera). Each HITSP Data Element is defined in a way that provides for consistent use of the data elements across different Information Exchanges. Constraints are applied where necessary to each data element to facilitate the use of it across a wide variety of standards. When appropriate, additional constraints may be applied within Interoperability Specifications (ISs).

Constraints can be applied by making certain data elements required, or by refining the information that may be provided in that data element when it is used in an exchange. Some constraints are expressed by requiring the use of specific value sets for the data element. These value sets in turn reference standard terminologies. Groups of constraints can also be uniquely identified and applied using templates based on the HL7 Version 3 Draft Standard for Trial Use (DSTU): Specification and Use of Reusable Constraint Templates.

While there is no overarching HITSP reference model for data elements, the HITSP Data Architecture is structured in a way that allows it to take advantage of the reference models of a variety of different standards. HITSP Data Elements are derived from the data elements of one or more exchange standards, and mapped back to those standards. HITSP Data Element constraints allow for consistent meaning to be established across the various harmonized standards.

HITSP is using automated information systems known as metadata registries to support the publication, use and implementation of HITSP Specifications. These automated information systems are designed to provide for quick and easy access to critical definitions and knowledge necessary to implement HITSP Specifications.

HITSP is continuing to work with its member Standards Development Organizations to ensure that the data in these metadata registries is up to date. HITSP are also collaborating with organizations such as the Agency for Healthcare Research and Quality (AHRQ) to support the development and population of the United States Health Information Knowledgebase as a metadata registry in support of the HITSP process. Other Federal agencies are providing support as well, including the Centers for Disease Control through the Public Health Information Network (PHIN) and their Vocabulary Access and Distribution System (PHIN-VADS), and the Veterans Health Administration.

[1] HITSP anticipates that HITSP Data Elements will eventually become a separate HITSP publication maintained jointly by the respective HITSP Domain Technical Committees.

3.0 BACKGROUND

This section provides background for the reader on terms used in this technical note, on the healthcare related standards and the organization of HITSP Specifications. If you are already familiar with this material, you can skip to Section 4.0 for a description of the HITSP Data Architecture.

3.1 DEFINITIONS

The following key concepts used within this technical note are briefly defined here.

Capability — A HITSP Capability is an implementable business service that specifies interoperable Information Exchanges using HITSP constructs. A Capability supports stakeholder requirements and business processes and includes workflow, information content, infrastructure, security and privacy.

Code — *[Adapted from HL7 Version 3 Core Principals]* A code is a **Concept** representation published by the author of a **Code System** as part of a code system, and it is an entity of that code system. It is the preferred unique identifier for that concept in that code system and used in the code property of a coded data type. The meaning of a code within a particular code system entity is valid only within that code system.

Code System — *[Adapted from HL7 Version 3 Core Principals]* A code system is a managed collection of **Concept** identifiers, usually **Codes**, but sometimes more complex sets of rules and references. They are often described as collections of uniquely identifiable concepts with associated representations, designations, associations, and meanings. Examples of code systems include ICD-9 CM, SNOMED CT, LOINC, CPT and UCUM. To meet the requirements of a code system, a given concept representation must resolve to one and only one meaning within the code system. Code systems are often referred to as terminologies, vocabularies, or coding schemes.

Component — A HITSP Component (C) is a construct that defines the set of data elements, structures, relationships, **Constraints** and terminology needed to support specific reusable information content. A component may also express constraints on base or composite standards. Examples are the lab result message and lab result context. Components describe the payload of an Information Exchange.

Concept — *[Adapted from HL7 Version 3 Core Principals]* A concept is a unitary mental representation of a real or abstract thing; an atomic unit of thought. It should be unique in a given **Code System**. A concept may have synonyms in terms of representation and it may be a primitive term or composed of other terms in the code system.

Constraint — A constraint is an expression of a business rule applied to an Information Exchange. It can restrict the values that appear within the exchange in a variety of different ways, and appear in both HITSP Specifications and in the standards those specifications select.

Data Element — A data element is the smallest unit of data pertinent to an Information Exchange. A data element may contain several discrete values (e.g., month, day and year to convey a date, or code and code system to convey a concept, or number and unit to convey a measure of a physical quantity). The selected standards use the terms attribute, component, data element or field to describe what HITSP calls a Data Element.

Extensional Value Set	*[Adapted from HL7 Version 3 Core Principals]* An extensional value set definition is an enumeration of all of the concepts within the value set. Value sets defined by extension are composed of explicitly enumerated sets of concept representations (with the code system in which they are valid). The simplest case is when the value set consists of only one code. See also **Value Set**.
Intensional Value Set	*[Adopted from HL7 Version 3 Core Principals]* An intensional value set definition is a set of rules that can be resolved (ideally computationally) to an exact list of concept representations at a particular point in time. See also **Value Set**.
Metadata	Metadata is data about data. HITSP specifies certain metadata used to express information about the **Data Elements**, **Value Sets** and **Templates** used within its specifications.
Metadata Registry	A Metadata Registry is an information system that stores metadata about an entity to enable the retrieval of it by humans and software applications. HITSP uses metadata registries for testing and implementation of conforming systems. Metadata registries can store metadata about **Data Elements**, **Value Sets** and **Templates**. See also **Metadata** and **Registry**.
Module	A module is a group of related **Data Elements**.
Registry	*[Adopted from the OASIS EbXML Glossary]* A mechanism whereby relevant **Repository** items and **Metadata** about them can be registered such that a pointer to their location, and all their metadata, can be retrieved as a result of a query.
Repository	*[Adopted from the OASIS EbXML Glossary]* A location or set of distributed locations where Repository Items, pointed at by the **Registry Metadata**, reside and from which they can be retrieved.
Service Collaboration	*[Adapted from HITSP Harmonization Framework and Exchange Architecture TN904]* A HITSP Service Collaboration (SC) is the composition of HITSP constructs into a reusable workflow.
Template	In normal use, a template is a pattern that must be followed in the construction of something. Within HL7 version 3 based standards a template is set of business rules (constraints) used to create an artifact used in an Information Exchange. Templates are formally defined in the HL7 Version 3. Templates are used by HITSP, HL7 and IHE as a way to express the business rules applied to HL7 Version 3 based documents and messages.
Transaction	A HITSP Transaction (**T**) is a logical grouping of data exchanges and transport methods that must all succeed or fail as a group. Examples are the query lab result or send lab result. A transaction is a HITSP Specification that fulfills the actions between two or more systems needed to meet one or more interoperability requirements.
Transaction Package	A HITSP Transaction Package (**TP**) is a logical grouping of two or more Transactions, Transaction Packages, and/or composite standards used to fulfill Information Exchange requirements (IERs). A Transaction Package is not required to succeed or fail as a whole. Examples include the Record Locator Service and Entity Identification Service
Vocabulary	*[adapted from HL7 Version 3 Core Principals]* A vocabulary is a set of **Concepts** designated by terms belonging to a special domain of knowledge, or subject field.

Value Set *[Adapted from HL7 Version 3 Core Principals]* A Value Set represents a uniquely identifiable set of valid **Concept** representations, where any concept representation can be tested to determine whether or not it is a member of the value set. See also **Intensional Value Set** and **Extensional Value Set**. *[End HL7 Adapted material]*
Note: A value set is intended to be a set in the formal sense, and so should contain only one code for each uniquely identifiable concept that it contains.

3.2 OVERVIEW OF HEALTHCARE RELATED STANDARDS

This section provides a brief overview the data architecture of healthcare related standards that have been adopted by HITSP. The purpose of this section is to familiarize the reader with the data architectures of these standards and how they relate to the HITSP Data Architecture. Each of the selected standards uses different models and terms to describe data.

Table 3-1 below shows the relationships between the terms used in the standards and the terms used in this specification. Sections following that table provide a brief overview of each standard model and its terms, and the relationships of those terms to the HITSP Specifications. This section is not intended to be a complete description of the standards. For detailed descriptions of the standards and the SDO modeling, readers are advised to review the SDO publications.

Table 3-1 Relationship between HITSP and SDO Concepts

Element	Component	Module	Data Element	Value Set	Template
ASC X12	Transaction or Transaction Set	Table, Loop, Segment or Composite Data Element	Simple Data Element	Internal and External Code Sets	Implementation Guides, or a type 3 Technical Report.
DICOM	Service Object Pair	Information Entity	Attribute	Context ID	Information Object Definition
HL7 Version 2	Message	Segment or Field	Field or Component	Table	Conformance Profile
HL7 Version 3 including CDA	Message or Document	RIM Class, Model, R-MIM, HMD or C-MET	(RIM) Attribute	Concept Domain, Code System or Value Set	Template
NCPDP Formulary & Benefit	Message	Record	Field	External Code List (ECL)	Implementation Guide
NCPDP SCRIPT	Transaction	Segment	Field	External Code List (ECL)	Implementation Guide
NCPDP Telecommunications	Transmission	Segment	Field	External Code List (ECL)	Implementation Guide
OASIS ebXML	Message	Class	(ebRIM) Attribute	ClassificationScheme	*Not Available*
OASIS EDXL	Message	Container	Element	enumeration	*Not Available*

3.2.1 ASC X12

The basic unit of transmission in Accredited Standards Committee (ASC) X12 is a transaction set which is composed of loops and segments. Transaction set components are delineated by control segments. An individual segment is composed of one or more composite data structures or simple data elements. Sections are repeatable. Information may be nested.

Figure 3-1 ASC X12 Message Example

```
ISA*00*          *00*          *ZZ*4137147        *ZZ*PLANA
*060119*1545*U*00401*999999999*0*P*:~
GS*HS*4137147*PLANA*20060119*1545*999999999*X*004010X092A1~
ST*270*0001~
BHT*0022*13*999999999*20060119*1545~
HL*1**20*1~
NM1*PR*2******PI*00999~
HL*2*1*21*1~
NM1*1P*1*ALLISON*JASON*H***SV*ABC123DEF~
HL*3*2*22*0~
NM1*IL*1*ARROJO*ROLANDO*A***MI*5643296~
DMG*D8*19710102~
EQ*30~
SE*11*0001~
GE*1*999999999~
IEA*1*999999999~
```

Transaction sets are identified by a number. Figure 3-1 above shows an example of the 270 transaction set otherwise known as the Eligibility Benefit Inquiry transaction. This example shows the message with its various segments are identified by a unique two- or three-character identifier that serves as a label for a component data structure, and terminated by a single tilde (~) character. The simple or composite data elements appear in the message separated by asterisks (*). These data elements are identified within the segment by their ordinal position within it. The ASC X12 data elements appearing within a message are assigned unique reference numbers in the ASC X12 standard. These references are included in the X12N implementation guides along with associated information such as the data element name, description, type, length, ASC X12 code values, and the restrictions on their use, etc.

Table 3-2 below shows how the parts of an ASC X12 message will be identified in a mapping from a HITSP Data Element to the ASC X12 data element:

Table 3-2 Addressing X12 Data Elements

Data Element Concepts	Transaction Set ID	Loop ID	Segment & Data Element Position	Data Element Number
Concept Name	Eligibility and Benefit Verification Transaction Set	Subscriber Loop	Segment Name	Last Name or Organization Name
Data Element Identifier 270_2100C_NM103 1035	270	2100C	NM103	1035

3.2.2 DICOM

The basic unit of transmission in DICOM is a DICOM message. A DICOM message is composed of two parts specified in a Service Object Pair (SOP) class. The first part gives a sequence of commands to execute upon the Information Object in the second part. In its simplest form, the Information Object defines the data being managed, such as an image or a structured report.

The Information Object Definition (IOD) specifies the relevant attributes that need to be captured about the real-world objects they represent. These include items such as an X-Ray Image or a collection of key images. Each DICOM IOD is composed of a collection of attributes and modules. Attributes are the basic data elements, and can include information such as the patient's name, a patient identifier, a study instance identifier, or pixel data. A Module is a collection of attributes that are logically related to each other. When transmitting a DICOM message the information is conveyed as a list of identifying tags, each followed by the data for that attribute.

The attribute tag is represented as a group number and a element number in a hexadecimal notation (gggg,eeee). There are no semantics placed on the tag, it is only a number to identify the attribute. HITSP

will reference DICOM data elements using the notation used by DICOM, followed by the tag name. Table 3-3 shows the components that HITSP uses to address DICOM Data Elements. This table uses the Patient Name as an example.

Table 3-3 Addressing DICOM Data Elements

Data Element Concepts	Tag Identifier	Tag Name
Concept Name	(0010,0010)	Patient Name

DICOM Data is typically exchanged as raw data bytes. Figure 3-3 below shows a hex dump of part of a DICOM data stream. The second line shows the patient name in this stream.

Figure 3-2 Example DICOM Data Object

```
0400  30 30 30 20 08 00 11 11-53 51 00 00 00 00 00 00   000 ....SQ......
0410  10 00 10 00 50 4E 0E 00-42 72 75 63 65 5E 53 74   ....PN..Bruce^St
0420  65 70 68 61 6E 20 10 00-20 00 4C 4F 1C 00 54 2E   ephan .. .LO..T.
0430  35 39 34 38 32 30 30 38-30 34 30 33 31 35 35 39   5948200804031559
0440  31 30 32 36 35 2E 31 33-2E 38 10 00 30 00 44 41   10265.13.8..0.DA
0450  08 00 31 39 30 39 30 39-31 35 10 00 40 00 43 53   ..19090915..@.CS
0460  02 00 4D 20 10 00 10 10-41 53 04 00 30 34 38 59   ..M ....AS..048Y
0470  10 00 30 10 44 53 0A 00-38 30 2E 30 30 30 30 30   ..0.DS..80.00000
```

3.2.3 HL7 VERSION 2

The basic unit of transmission in HL7 is a message. An example message is shown in Figure 3-3 below. Messages are made up of segments that are terminated by a carriage return. The first segment of each message describes the message itself. Each segment begins with a three-character segment identifier and is followed by a structured sequence of related fields separated by pipe (|) characters. Each field may be further subdivided into separate components by a caret (^) symbol. Those may have subcomponents separated by ampersands (&), depending upon the data type associated with the field. All fields, components and subcomponents in an HL7 Version 2 message use the data types defined in that standard. HL7 Version 2 standards recommend or require vocabulary used in the communication. These are defined in User or HL7 defined tables in the HL7 Version 2 Messaging standard.

Figure 3-3 HL7 Version 2 Example

```
MSH|^~\&|RADLIS|SendingFac^<OID>^ISO|ReceivingApp^<OID>^ISO|ReceivingFac^<OID>
^ISO|200709101912133||ORU^R01^ORU_R01|2007091018321330035|D|2.5
PID|1||P410010^^^&<OID>&ISO||""||196712|M
OBR|1||XR312739|^^^CXR^Chest X-Ray^L|||||||||||||||||||RAD|F||||||786.51^PRECORDIAL
PAIN^I9C~786.7^ABNORMAL CHEST SOUNDS^I9C~786.05^SHORTNESS OF BREATH^I9C
OBX|1|TX|19005-8^XR IMPRESSION^LN||The film shows disseminated left lower lobe infiltrates
consistent with pneumonia.
```

The different parts of an HL7 message will be identified within HITSP Specifications using the segment name, followed by a hyphen, and then the field number. If necessary, the component and subcomponent numbers are appended to the field number, delimited with a period. The components of the address are shown below in Table 3-4.

Table 3-4 Addressing HL7 Version 2 Data Elements

Data Element Concepts	Segment	Field	Component	Sub-Component
Concept Name	Patient Identification Segment	Patient Identifier List	Components of Assigning Authority	Sub-Components of Assigning Authority – Assigning Authority's Universal ID
Data Element Identifier PID-3.4.2	*PID*	*3*	*4*	*2*

3.2.4 HL7 VERSION 3 INCLUDING THE HL7 CLINICAL DOCUMENT ARCHITECTURE

HL7 Version 3 is a standard that is used to exchange messages (from a variety of HL7 Version 3 standards), and clinical documents using the HL7 Version 3 Clinical Document Architecture Release 2. HITSP refers to these standard messages and clinical documents as HL7 Version 3 artifacts for the remainder of this section.

The basic unit of transmission or storage of HL7 Version 3 artifacts is XML. HL7 describes the components of these artifacts as classes in the HL7 Reference Information Model (HL7 RIM). The core classes in the HL7 RIM are the Act, Act Relationship, Entity, Role and Participation classes. These classes are further constrained by the HL7 RIM in the way that they can be connected to each other. These classes are further composed of attributes that are represented using the HL7 Version 3 data types. HL7 Version 3 standards rely heavily on code systems and value sets to provide additional information in the communication.

The HL7 Version 3 artifacts are composed of classes as defined in the HL7 RIM, and their names, attributes and value sets are further constrained by the relevant HL7 Version 3 standards. HL7 Version 3 artifacts are exchanged using XML as described in the HL7 XML Information Technology Specification (XML ITS). An example of an HL7 CDA Document is shown below in Figure 3-4. This example shows the **`ClinicalDocument`** Act and some of its associated attributes.

Figure 3-4 CDA Example

```
<ClinicalDocument>
        <realmCode code='US'/>
        <typeId extension='POCD_HD000040' root='2.16.840.1.113883.1.3'/>
        <templateId root='1.3.6.1.4.1.19376.1.5.3.1.1.1'/>
        <templateId root='2.16.840.1.113883.10.20.3'/>
        :
```

Because the HL7 Version 3 specifications selected by HITSP use XML, HITSP identify elements within those specifications using the XPath notation described by the W3C. Refer to Section 6.1 XPath Locations in XML in the appendix to understand how XPath is used to locate data elements in XML documents.

3.2.5 NCPDP SCRIPT

The basic unit of information in NCPDP SCRIPT Standard Implementation Guide is a transmission, which contains a message. An example message is shown below in Figure 3-5. Messages are made up of segments identified by a three character identifier. Segments are subdivided into composites or fields using the plus (+) symbol. Each field may have components separated by a colon (:) and those may have subcomponents separated by slashes (/).

Figure 3-5 NCPDP SCRIPT Example

```
UNA:+./*'
UIB+UNOA:Ø++1234567+++77777777:C:PASSWORDA+77Ø163Ø:P+19971ØØ1:Ø81322'
UIH+SCRIPT:ØØ8:ØØ1:NEWRX+11ØØ72+++19971ØØ1:Ø81322'
PVD+P1+77Ø163Ø:D3+++++MAIN STREET PHARMACY++61522Ø5656:TE'
PVD+PC+6666666:ØB+++JONES:MARK++++61522198ØØ:TE'
PTT++19541225+SMITH:MARY+F+333445555:SY'
DRU+P:CALAN SR 24ØMG::::24Ø:ME+EA:6Ø:38+:1 TID -TAKE ONE TABLET TWO TIMES A
DAY UNTIL GONE+85:19971ØØ1:1Ø2*ZDS:3Ø:8Ø4+Ø+R:1'
UIT+11ØØ72+6'
UIZ++1'
```

Parts of an NCPDP SCRIPT message are identified using the segment name, followed by a hyphen, and then the section and field sequence number.

The following is an example of the identifier for the NCPDP SCRIPT Street and Number /P.O. Box field of the Patient Address (PTT-Ø6Ø-Ø1).

Figure 3-6 Example NCPDP SCRIPT

Concepts	Segment	Section	Field Sequence
Concept Name	Patient Segment	Address	Street and Number /P.O. Box
Identifier Component	PTT	Ø6Ø	Ø1

3.2.6 NCPDP TELECOMMUNICATION GUIDE

The basic unit of information in NCPDP Telecommunication Standard Implementation Guide is a transmission, which contains up to four transactions. Transactions are made up of segments. Segments are subdivided into fields that contain the Information Exchanged.

Parts of a Telecommunication message are identified using the transaction name, field name and field number. The field number and name are the only things necessary to identify a data element within a specific communication. Table 3-5 shows how HITSP will identify NCPDP Telecommunication data elements in the mappings to HITSP Data Element. The HITSP Specifications reference data elements from the NCPDP Telecommunications Guide using the Field number and then the Field name. For example: 202-B2 Service Provider ID Qualifier.

Table 3-5 Addressing NCPDP Telecommunications Data Elements

Concepts	Field Number	Field Name
Identifier Components	202-B2	Service Provider ID Qualifier

3.2.7 NCPDP FORMULARY AND BENEFIT

The basic unit of information in NCPDP Formulary and Benefit Standard Implementation Guide is a batch file transmission. The transmission contains multiple record types. Records contain fields.

Data elements found in a Formulary and Benefits transmission are identified using the record type, field number and field name. HITSP Specifications will reference data elements from the Formulary and Benefit Guide using the record type, a hyphen, and then the field number, followed by the field name. For example: PAD-661-T9 Prior Authorization Question Sequence

Example:

Table 3-6 Addressing NCPDP Formulary and Benefit Data Elements

Concepts	Record Type	Field Number	Field Name
Component Identifier	PAD	661-T9	Prior Authorization Question Sequence

3.2.8 OASIS EBXML

The OASIS ebXML ebRIM and ebRS standards are the base standards used for document exchange in HITSP Specifications. This ebRIM standard defines the Registry Information Model. This model is described in terms of classes and attributes. The information model describes how these classes are related, and which attributes may appear in each class. The core class in the ebRIM is the **`RegistryObject`** from which almost everything else is derived. Other key classes are **`Associations`**, **`Classifications`**, **`ExternalIdentifiers`** and **`RegistryPackages`**. **`RegistryPackages`** are composed of **`RegistryObjects`**. **`RegistryObjects`** can be linked together using **`Associations`**, coded using **`Classifications`**, and identified using **`ExternalIdentifiers`**. Additional data is stored in **`Slots`** contained within the **`RegistryObject`**.

The ebRS standard defines the services that are supplied by a registry (e.g., registration or query), and the binding of these services to the SOAP protocol.

Classes and attributes of the ebRIM are exchanged using SOAP messages. The classes and attributes are expressed using XML elements. Figure 3-7 below shows a partial example of an ebXML message being sent to register a document. This example shows part of the **`RegistryPackage`** in the list of objects to be registered. That package contains data elements describing what is being registered in the **`Slot`**, **`Classification`** and **`ExternalIdentifier`** elements found in the message.

Figure 3-7 ebXML Example

```
<?xml version="1.0" encoding="UTF-8"?>
<SubmitObjectsRequest xmlns="urn:oasis:names:tc:ebxml-regrep:registry:xsd:2.1">
  <LeafRegistryObjectList xmlns="urn:oasis:names:tc:ebxml-regrep:rim:xsd:2.1">
    <ObjectRef id="urn:uuid:1ba97051-7806-41a8-a48b-8fce7af683c5"/>
    <ObjectRef id="urn:uuid:2c6b8cb7-8b2a-4051-b291-b1ae6a575ef4"/>
                                    ⋮
    <ObjectRef id="urn:uuid:f64ffdf0-4b97-4e06-b79f-a52b38ec2f8a"/>
    <RegistryPackage objectType="urn:uuid:a54d6aa5-d40d-43f9-88c5-b4633d873bdd"
id="SubmissionSet">
      <Name>
        <LocalizedString value="Document Submission"/>
      </Name>
      <Description>
        <LocalizedString value="Consultation"/>
      </Description>
      <Slot name="submissionTime">
        <ValueList>
          <Value>20080825081753</Value>
        </ValueList>
      </Slot>
    <Classification nodeRepresentation="Summarization of episode"
     classifiedObject="SubmissionSet"
     classificationScheme="urn:uuid:aa543740-bdda-424e-8c96-df4873be8500">
     <Name>
      <LocalizedString value="Summarization of episode" charset="UTF-8"/>
     </Name>
     <Slot name="codingScheme">
      <ValueList>
       <Value>Connect-a-thon contentTypeCodes</Value>
      </ValueList>
     </Slot>
    </Classification>
    <ExternalIdentifier
     value="82-TEST011^^^&amp;1.3.6.1.4.1.21367.2005.3.7&amp;ISO"
     identificationScheme="urn:uuid:6b5aea1a-874d-4603-a4bc-96a0a7b38446">
     <Name>
      <LocalizedString value="XDSSubmissionSet.patientId"/>
     </Name>
    </ExternalIdentifier>
```

Because these specifications use XML, HITSP identify elements within these specifications using the XPath notation described by the W3C.

Refer to Section 6.1 XPath Locations in XML in the Appendix to understand how XPath is used to locate data elements in XML documents.

3.2.9 OASIS EDXL

The Emergency Data Exchange Language (EDXL) family of standards are built for XML, and make use of XML elements to convey data. Each of these specifications includes a W3C Schema that describes the

message. Thus, the fundamental component of each of these is an XML element, which can be composed of data or other XML elements. The data types of these standards are drawn from the data types defined by the XML Schema standard.

HITSP has selected several standards from the OASIS EDXL family of standards, including the Common Alerting Protocol (CAP), the Emergency Data Exchange Language Distribution Element (EDXL-DE), and the Emergency Data Exchange Language Hospital Availability Exchange (EDXL-HAVE) standards.

The Common Alerting Protocol and the Distribution Element specification primarily contain information pertinent to the routing and transport of information. That content does not typically make use of the HITSP Data Architecture since it is usually utilized at a lower level in Information Exchanges.

The HAVE specification communicates information about resource availability as hospitals and contains numerous data elements that can be used by healthcare facilities to communicate their current status and availability of resources. These data elements are stored as elements in the XML exchange, and are organized into higher level container elements, finally rolling up into the highest level container which comprises all of the information. Figure 3-8 shows a partial example of a communication using the HAVE specification.[2]

Figure 3-8 HAVE Example

```
<have:HospitalStatus xmlns:have="urn:oasis:names:tc:emergency:EDXL:HAVE:1.0"
        xmlns:xal="urn:oasis:names:tc:ciq:xal:3" xmlns:xnl="urn:oasis:names:tc:ciq:xnl:3"
>
        <have:Hospital>
                <have:Organization>
                        <have:OrganizationInformation>
                                <xnl:OrganisationName>
                                        <xnl:NameElement>XYZ
Hospital</xnl:NameElement>
                                </xnl:OrganisationName>
                                                ⋮
```

Because the EDXL specifications use XML, HITSP Specifications will identify elements within those standards using the XPath notation described by the W3C.

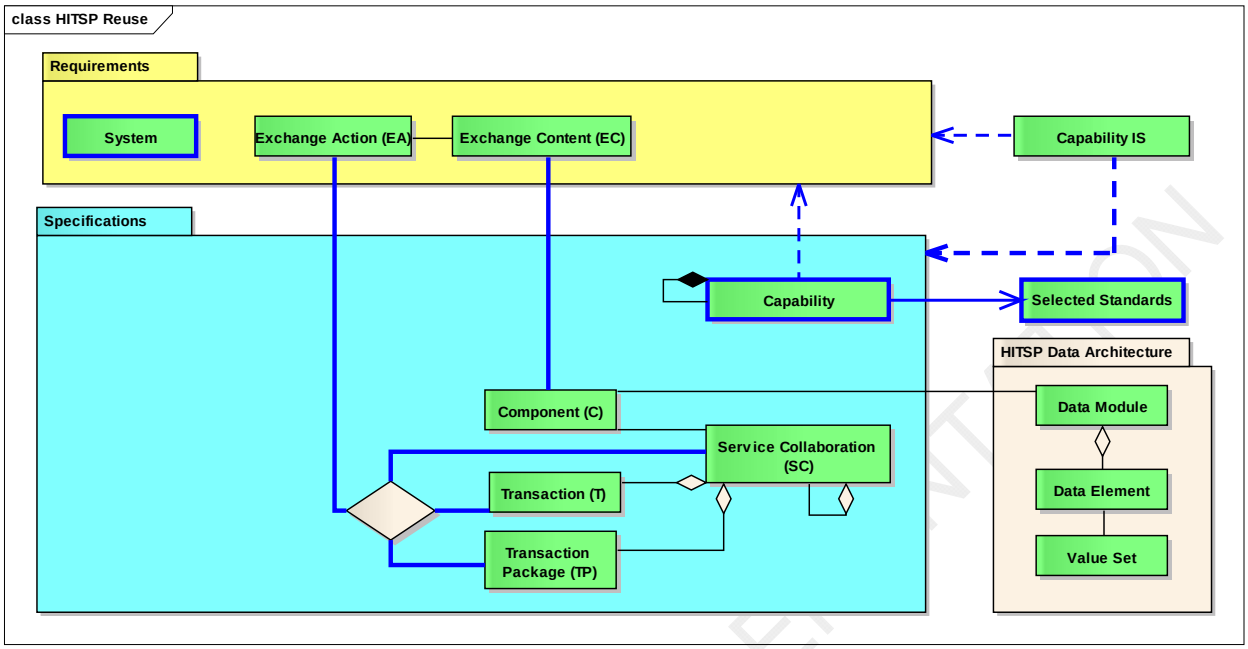
Refer to Section 6.1 XPath Locations in XML in the appendix to understand how XPath is used to locate data elements in XML documents.

3.3 HITSP DATA ARCHITECTURE IN THE CONTEXT OF THE HITSP FRAMEWORK

The HITSP Data Architecture described by this technical note is a portion of the entire HITSP Harmonization Framework shown in Figure 3-9. The package labeled data architecture in this diagram is described in greater detail in Section 4.0 below.

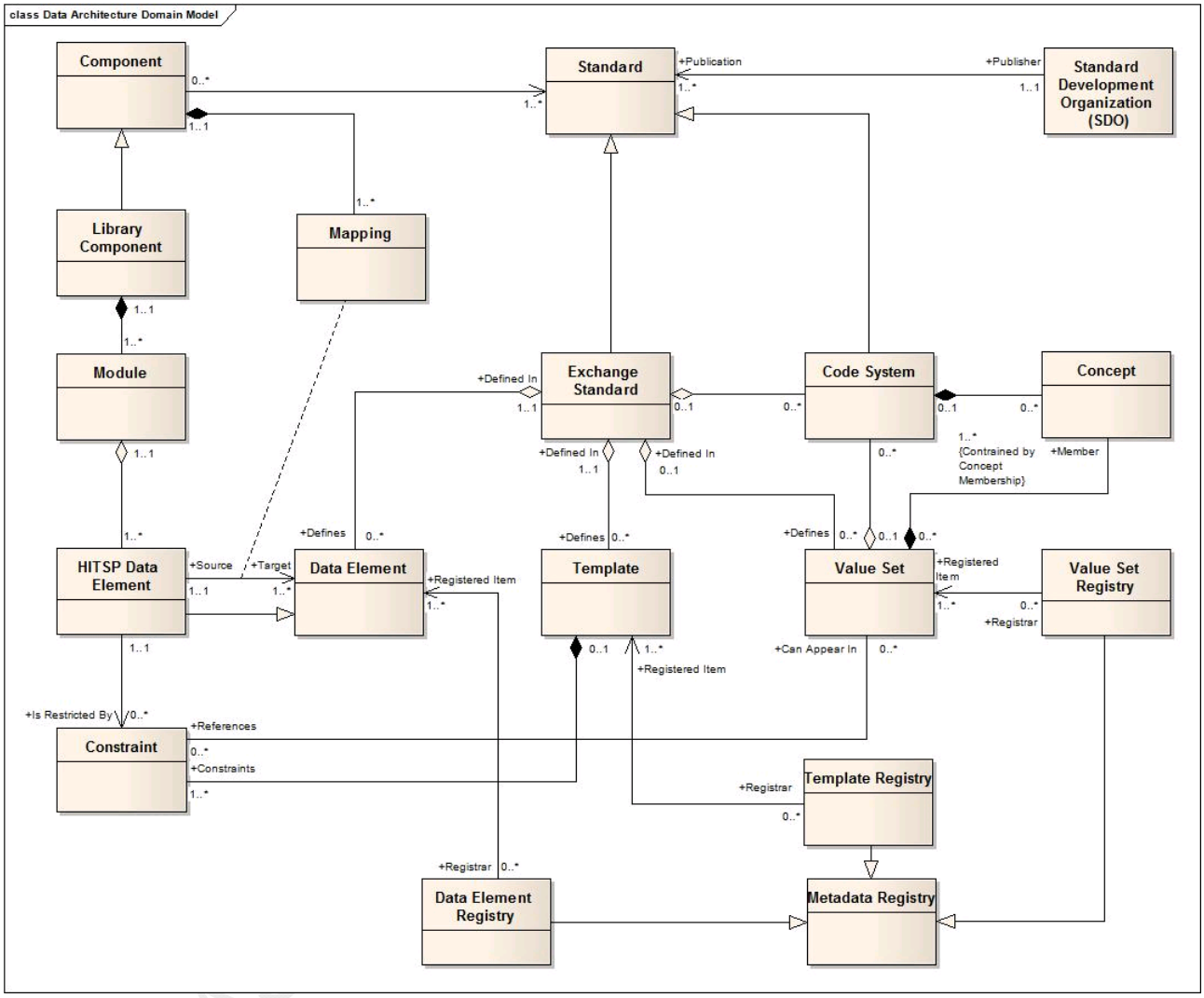
[2] Note that the spelling of OrganisationName in this example is based upon exactly what is written in the standard.

Figure 3-9 Data Architecture in the HITSP Framework



4.0 HITSP DATA ARCHITECTURE

Figure 4-1 HITSP Data Architecture



The HITSP Data Architecture is a component of the broader HITSP Harmonization Framework shown briefly in Section 3.3 and described in detail in HITSP TN904 HITSP Harmonization Framework. This architecture is based upon concepts currently used within standards, and relies critically upon the concepts of data elements, value sets and templates described briefly in the definitions section above, and expanded upon below. The SDOs produce standards that define the data elements, templates and value sets used in the exchange of information. The value sets are composed of concepts published by an SDO within a code system.

SDOs have their own terminology for the various hierarchies and groupings of data. The SDOs also define the data types permitted within a data element. HITSP Specifications assume that information will be represented in the data types required by the selected standard. A brief overview of the relationship of these standards and their terms to the HITSP Data Architecture is provided in the Overview of Healthcare Related Standards Section above.

HITSP Components reference these SDO publications and may further constrain the data elements, templates and value sets. Data elements of the published standards are constrained first by mapping that data element to a HITSP Data Element. The HITSP Data Element may already exist or the HITSP Domain Committee may create it where necessary. The mapping of data elements across the various standards is a key step in the harmonization of standards. The HITSP Data Elements are further constrained where necessary to ensure consistent data exchanges across the various standards.

Additional constraints may further refine the content or use of components based upon the specific business need identified by a harmonization request. These constraints will appear within the Interoperability Specifications that identify those business needs.

A few HITSP Components are developed explicitly as libraries of components that can be reused by other HITSP Specifications. These include the HITSP/C83 CDA AND CCD Content Modules and the HITSP/C80 Clinical Document and Message Terminology. These components are not intended to be implemented by themselves. They are referenced by other components to ensure consistency of Information Exchange across the HITSP Specifications.

Finally, all data elements, templates and value sets defined within HITSP can be readily accessed through metadata registries. The metadata registries allow navigation through the variety of artifacts described in SDO publications, and point back to the publications that define those artifacts.

The AHRQ-USHIK metadata registry permits navigation of each HITSP Data Element to the supporting or originating harmonization request, interoperability specifications, value sets, code systems and standards. AHRQ-USHIK contains a mapping of each HITSP Data Element to every HITSP Specification in which it appears.

4.1 COMPONENT

The term **Component** was defined in the Definitions Section. A component is the smallest complete unit of information that can be transferred at one time in the payload of exchange. The standards selected by the component define this unit of exchange in terms of documents, transactions, transaction sets, or messages. There is a direct mapping of these units of exchange to a HITSP Component. The standards also define the format and structure of the exchange. A HITSP Component:

- maps the HITSP Data Elements described in this technical note to the data elements of the selected standard
- applies any constraints defined within those data elements
- may provide other constraints on the use of the standard to meet the business requirements established for its use

Exchange content may include one or more components in its payload. For example, a single exchange using the HITSP/TP13 Manage Sharing of Documents may return multiple instances of a HITSP Component as a result of issuing a Retrieve Document Set transaction, or may register multiple HITSP Components in a Register Documents transaction.

4.2 MODULE

The payload of the component is made up of HITSP Data Elements found in one or more modules defined in the HITSP/C83 CDA AND CCD Content Modules . Modules are simply used within the HITSP/C83 CDA AND CCD Content Modules to organize or classify those data elements based on the information they contain. The categories used to organize these modules are described briefly in Table 4-1 below. A more detailed definition of these modules can be found in the HITSP/C83 CDA and CCD Content Modules.

Table 4-1 Module Categories

Module Category	Description
Personal Information	The personal information includes name, address, contact information, personal identification information, ethnic and racial affiliation and marital status of a person
Support	Support includes the patient's sources of support, such as immediate family, relatives and/or guardians. This includes next of kin, caregivers, support organizations, and key contacts relative to healthcare decisions. Support providers may include providers of healthcare related services, such as a personally controlled health record, or registry of emergency contacts
Healthcare Providers	This includes a list of the healthcare providers and organizations that provide or have provided care to the patient
Insurance Providers and Payers	Insurance providers include data about the organizations or individuals who may pay for a patient's healthcare, and the relationships, demographics and identifiers of those individuals with respect to the payer. Such organizations or individuals may be health insurance plans, other payers, guarantors, parties with financial responsibility, some combination of payers or the patient directly
Allergies and Drug Sensitivities	This includes the allergy or intolerance conditions, severity and associated adverse reactions suffered by the patient
Conditions	This includes relevant clinical problems and conditions for which the patient is receiving care, including information about onset, severity, and providers treating the condition. Conditions are broader than, but include diagnoses
Medications	This includes the patient's prescription or non-prescription medications and medication history, and may include prescriptions, fulfillments and medication administration activities
Immunizations	This includes data describing the patient's immunization history
Vital Signs	This includes data about the patient's vital signs
Test Results	This includes data about current and historical test results from laboratory or other diagnostic testing performed on the patient
Encounter	This includes data describing the interactions between the patient and clinicians. Interaction includes both in-person and non-in-person encounters such as telephone and email
Procedures	This includes data describing procedures performed on a patient
Family History	Data defining the patient's genetic relatives in terms of possible or relevant health risk factors that have a potential impact on the patient's health
Social History	Data defining the patient's occupational, personal (e.g. lifestyle), social, and environmental history that have a potential impact on the patient's health
Medical Equipment	Medical Equipment includes implanted and external medical devices and equipment that a patient's health status depends on, as well as any pertinent equipment or device history
Functional Status	Data defining the patient's functional status with respect to, ambulatory ability, mental status or competency, activities of daily living, including bathing, dressing, feeding, grooming, home/living situation having an effect on the health status of the patient, and ability to care for self
Plan of Care	The plan of care contains data defining prospective or intended orders, interventions, encounters, services, and procedures for the patient

4.3 DATA ELEMENTS

HITSP Domain Technical Committees (TC) define HITSP Data Elements in response to the business requirements identified for an Information Exchange. The Domain Technical Committees use existing data elements where feasible, and identify new data elements when existing HITSP Data Elements do not meet the established business requirements.

4.3.1 DATA ELEMENT DEFINITION

The HITSP Data Elements are defined in the HITSP/C83 CDA and CCD Content Modules, as described in Table 4-2 below. While the HITSP/C83 CDA and CCD Content Modules includes only this fixed view within it, this information will also appear in the AHRQ-USHIK Data Element registry. A variety of views are already available in that registry, and customized views can be created. Users of the HITSP Specifications can access the registry and view or download this information to assist in their implementation, evaluation and/or research.

Table 4-2 Data Element Definition

Data Element Identifier	Data Element Name	Data Element Definition	Data Element Constraints
A numeric identification of the data element used to reference it	The name of the data element being defined	A concise definition of the data element	Additional HITSP constraints for this data element

4.3.1.1 DATA ELEMENT IDENTIFIER

Each data element has an identifier that uniquely identifies it. The first part of the identifier is assigned based upon the module where it is found. The second part of the identifier uniquely identifies the data element within the module. As new data elements are created, they are added to the end of the module. The data element identifiers are persistent and will not be changed or reused between versions of HITSP Specifications.

4.3.1.2 DATA ELEMENT NAME

Each data element has a name that briefly describes the content and purpose of the data element. Data element names may be changed between versions of HITSP Specifications to better describe the content and purpose.

4.3.1.3 DATA ELEMENT DEFINITION

Each HITSP Data Element has a definition that is intended to precisely describe the purpose and structure of the data element independent from the standards that it may be mapped to. This independence allows HITSP Data Elements to be mapped to data elements using a variety of standards. The concise definition and mapping to the standards data element also supports harmonization of data across exchanges using different standards. The definition should describe the data element with sufficient enough detail to clearly indicate the purpose and content of the data element.

4.3.1.4 DATA ELEMENT CONSTRAINTS

In some cases, the data element will have additional restrictions limiting the values that can be communicated within it. HITSP may apply restrictions to a data element when it is communicated. These restrictions could be with regard to its precision, the units, and the range of legal values that may be transmitted, or other restrictions as necessary. These will be described in or referred to by this column.

These restrictions are known as **Constraints**, and are defined in more detail in Section 4.6 Constraints later within this document. Constraints are just one of several attributes that can be applied to an exchange within the HITSP framework. These constraints may describe the abstract **Concepts** allowed within the data element. For example, a data element carrying the administrative gender could be limited to the concepts of Male, Female, Undifferentiated or Unknown. The restrictions may also be concrete, identifying specific **Value Sets** that must be used within the data element. For example, the administrative gender could be limited to the HL7 defined value set identified using the 2.16.840.1.113883.1.11.1 OID. It is not always possible to limit Information Exchanges to a concrete value set due to variations in the **Code Systems** supported by the various standards. Value set constraints are concrete where possible.

4.3.2 DATA ELEMENT MAPPING

A HITSP Component maps HITSP Data Elements to the data elements of the standard selected by that component. The mapping of HITSP Data Elements to those of the selected standard is presented in a table in that component. The structure of this mapping table is shown in Table 4-3 below.

Table 4-3 Mapping to HITSP Data Elements

SDO Identifier and Name	HITSP Data Element ID and Name	HITSP Opt*/Repeat	HITSP Additional Constraints for Component
The identity or location of the data element in the exchange	The numeric identifier and name of the data element from the table above	This column indicates whether the data element is required or not, and whether it is repeatable	Specifies additional HITSP constraints for use in the standard where needed

4.3.2.1 SDO IDENTIFIER AND NAME

The various standards have distinctive ways to identify their data elements. When the standard makes use of XML, HITSP uses the XPath expression to identify the XML element of the component that is being mapped to the HITSP Data Element. When the standard does not use XML, HITSP uses the nomenclature found within the standard to identify a data element.

Table 4-4 below shows examples for SDO identifiers and names for the various HITSP selected standards.

Table 4-4 SDO Identifier and Name Examples

Standard	Example Identifier and Name
ASC X12	270_2100C_NM103 1035
DICOM	(0010,0010) Patient Name
HL7 Version 2	PID-3.4.2
HL7 Version 3	/ClinicalDocument/code subject/patient
NCPDP Formulary and Benefit	PAD-661-T9 Prior Authorization Applicable List Detail
NCPDP Script	PTT-Ø6Ø-Ø1
NCPDP Telecommunications	202-B2 Service Provider ID Qualifier
OASIS ebXML	//rim:RegistryPackage
OASIS EDXL	/have:HospitalStatus/have:Hospital/have:Organization

4.3.2.2 HITSP DATA ELEMENT ID AND NAME

This column shows the HITSP Data Element being represented in the component by the HITSP Data Element identifier and name. The HITSP Data Element identifier is given first, followed by a hyphen, then the HITSP Data Element Name. The HITSP Data Elements are defined in the HITSP/C83 CDA and CCD Content Modules. When a mapping from the standard data element to a HITSP Data Element is identified in the component, all constraints on the HITSP Data Element apply to that component. If the HITSP Data Element indicates that the information shall be communicated to a certain precision, or in certain units of measure, or using a particular value set, these constraints are applied. The existence of this mapping can be translated into the following conformance statement:

> The (standard data element in the table) **SHALL** be communicated applying all constraints defined for (the HITSP Data Element in the table).

4.3.2.3 HITSP OPTIONALITY/REPEATABILITY

This column identifies the conditions under which the data element is sent, and whether it may be repeated in the exchange. The column contains two fields separate by a slash (/). The first field indicates when the data element is to be sent and the list of values used in that column is described below in Table 4-5.

Table 4-5 Optionality

Value	Definition
R	REQUIRED - Required data elements must always be sent. Data elements that are required may under exceptional circumstances have an unknown value (e.g., the name of an unconscious patient). In these cases the sending application is required to indicate the reason that the data are not available where the standard permits. Some standards may not permit an unknown value at all
R2	Required if known - If the sending application has data for the data element, it is REQUIRED to populate the data element. If the value is not known, the data element need not be sent
O	OPTIONAL - Data elements that are marked optional may be sent at the choice of the sending application. An optional element need not be sent, but when it is sent, the data module defines the meaning of that data element and a receiver can always be assured of what that data element represents when it is present. Senders should not send an optional data element with an unknown value. If the value is not known, simply do not send the data element
C	Conditional - Data elements that are marked conditional (C) are REQUIRED to be sent when the conditions specified in the HITSP additional specifications column are true. The conditions under which the data element is to be exchanged will be specified as a constraint on the data element in the last column

The second field indicates whether the data element is repeatable and the list of values used is described below in Table 4-6.

Table 4-6 Repeatability

Value	Definition
N	No. The data element SHALL NOT be repeated
Y	Yes. The data element MAY be repeated

Further constraints on repeatability with respect to minimum or maximum number of occurrences will be defined as more specific constraints in the last column.

4.3.2.4 DATA ELEMENT CONSTRAINTS

Finally, the HITSP Specification presents additional constraints regarding optionality, cardinality and value sets to be used for the data element.

4.4 VALUE SETS

Data elements may be constrained to use a specific **Value Set**. The HITSP/C80 Clinical Document and Message Terminology identify the value sets that have been selected or created by HITSP for use in its Specifications.

The HITSP Specifications have attempted to ensure that all data elements used in any Information Exchange make use of consistently defined value sets. As an example, the value set used to describe problems in a HITSP Specification will be the VA/Kaiser Permanente subset of SNOMED CT in almost all components that need to exchange problems.

In some cases, different HITSP Specifications will select standards that require the use of different **Code Systems** in the expression of the same Data Element. In these cases, all attempts will be made to ensure that the value sets required in the different exchanges can be mapped. In these cases, HITSP will define the abstract list of concepts that must be present in the value set used with the data element.

4.4.1 VALUE SET DEFINITION

Table 4-7 below describes how a Value Set is conveyed in HITSP Specifications.

Table 4-7 Value Set Metadata

Element	Description
Identifier	This is the unique identifier of the value set
Name	This is the name of the value set
Source	This is the source of the value set, identifying the originator or publisher of the information
URL	A URL referencing the value set members or its definition at the time of publication
Purpose	Brief description about the general purpose of value set
Definition	A text definition formally describing how concepts in the value set are (intensional) or were (extensional) selected
Version	This row contains a string identifying, where necessary, the specific version of the value set
Type	Extensional (Enumerated) or Intensional (Criteria-based)
Binding	Static or Dynamic
Status	Active (Current) or Inactive (Retired)
Effective Date	The date when the value set is expected to be effective
Expiration Date	The date when the value set is no longer expected to be used
Creation Data	The date of creation of the value set
Revision Date	The date of revision of the value set

Additional metadata describing the value set may be present in metadata registries, showing for example, the relationships between different value sets.

4.4.1.1 VALUE SET IDENTIFIER

This is an OID that uniquely identifies the value set. All OIDs used in value sets used within HITSP Specifications will be registered with the HL7 OID Registry at http://www.hl7.org/oid/index.cfm.

4.4.1.2 VALUE SET NAME

The value set name is a short descriptive name for the value set. The name given is the name used by the originator of the value set.

4.4.1.3 VALUE SET SOURCE

An informative reference to the source of the value set is always included. The originator of a value set and the publisher of it may be different organizations. The source reported in this metadata is an authoritative source, and may contain either of these organizations.

4.4.1.4 VALUE SET URL

Each value set will include a link to an online resource where the content of the value set is available. Implementers should carefully verify on-line content against original specifications.

4.4.1.5 VALUE SET PURPOSE

The purpose of the value set will be included. This should explain the reason for use of the collection of concepts in the value set.

The purpose of the value set should be clinically descriptive, to allow for value sets to be reused in different contexts. For example: This value set may be used to help identify patients that have recently been under anesthesia. If the purpose above had simply indicated that the value set is intended for use with the *Measure Name* quality measure, the value set could not readily be reused. One would first need to understand how and why that value set was used in the measure.

4.4.1.6 VALUE SET DEFINITION

The value set definition gives the algorithm by which an **Intensional Value Set** is or can be constructed, or the mechanism by which an **Extensional Value Set** was created. Presently there is no identified standard for the expression of the contents of a value set. Until one can be identified these definitions will be text descriptions. The definition of an intensional value set should be specific well enough to compute the members of the value set.

An example intensional definition follows: All direct children of “394658006 clinical specialty”, plus direct children of“394814009 general practice” and “394733009 medical specialty”, excluding those concepts whose children are included, and excluding “394802001 general medicine” (which is redundant with “general medical practice”).

An example extensional definition follows: The value set for laboratory results listed below was constructed by enumerating all LOINC codes used in the HEDIS measures for laboratory results and those LOINC codes published by the Council of State and Territorial Epidemiologists (CSTE) to identify laboratory results used for reportable and notifiable conditions.

Some value sets may be defined as being a subset or superset of other value sets. These relationships will be specified in the value set definition. For example, if the HEDIS and CSTE lists of LOINC codes are available as separate values sets, the example given for laboratory results above could have been stated intensionally as: The value set of for laboratory result codes is the union of the value sets for HEDIS Laboratory Results (OID: 2.16…) and the CSTE Value set for reportable and notifiable conditions (OID: 2.16…).

4.4.1.7 VALUE SET VERSION

This attribute will record the version number of the value set using the format specified by the originator of the value set. If no version number is identified by the originator, HITSP will assign a version number using the publication date associated with the value set, in the form YYYYMMDD.

4.4.1.8 VALUE SET TYPE

This attribute will record whether the value set is intensional or enumerated.

4.4.1.9 VALUE SET BINDING

This attribute will record whether the value set is bound to data elements used in HITSP Specifications statically or dynamically. A statically bound value set has its values fixed until a new version of the value set is released. Extensional Value Sets are typically statically bound. A dynamically bound value set has its definitions fixed, but the values in the set may vary as new versions of the code system upon which they are based are released. Intensional values sets are often dynamically bound. When statically bound, an intensional value set must specify the version of the code system being used before the members of the value set can be computed.

4.4.1.10 VALUE SET STATUS

This attribute records the current status of the value set. An active value set is one that can be used within an exchange. An inactive value set should not be used in an exchange, as it has been retired from use. It may no longer be actively maintained, be replaced by another value set, or simply be no longer needed in the exchange.

4.4.1.11 VALUE SET EFFECTIVE DATE AND VALUE SET EXPIRATION DATE

Some value sets are updated on a routine cycle. These value sets will have an effective and expiration date describing when the value set should be used. For example, value sets based on ICD-9 CM could

be updated between July and September to make the implementers ready for the next fiscal year starting on October 1st.

4.4.1.12 VALUE SET CREATION DATE AND VALUE SET REVISION DATE

HITSP managed value sets will record the date of their creation and revision. For HITSP managed value sets, the Creation Date is the date of first publication of the value set in a panel-approved specification, and the . This information need not be recorded in HITSP publications for value sets managed by external organizations, as the information may not be available.

4.4.2 CODE SYSTEM METADATA

A value set may contain codes from one or more code systems. Each code system used by the value set will be described in the dependencies table found within the HITSP/C80 Clinical Document and Message Terminology. That table will contain the metadata identified below in Table 4-8. Additional metadata describing the value set may be present in metadata registries, providing for example, links to places where the code system may be downloaded.

Table 4-8 Code System Metadata

Element	Description
Identifier	This is the identifier for a code system from which the value set is drawn
Name	This row provides the name of the code system associated with the value set
Source	This row identifies the source for the code system
URL	This row identifies the URL for the code system
HL7 Identifier	The identifier used to identify this code system in HL7 Version 2.X messages.
Version	This row contains a string identifying, where necessary, the specific version of the code system used

4.4.2.1 CODE SYSTEM IDENTIFIER

The Code System Identifier is an OID that uniquely identifies the code system. All OIDs used in HITSP Specification will be registered in the HL7 OID Registry at http://www.hl7.org/oid/index.cfm.

4.4.2.2 CODE SYSTEM NAME

This is the official name of the code system as determined by its publisher.

4.4.2.3 CODE SYSTEM SOURCE

An informative reference to the source of the code system is always included. The originator of a code system and the publisher of it may be different organizations (e.g., for SNOMED CT, the originator is the IHTSDO, but in the US, it is also published by the National Library of Medicine). The source reported in this metadata is an authoritative source, and may contain either of these organizations.

4.4.2.4 CODE SYSTEM URL

When the code system is available online, HITSP will include a link to that online resource. When it is not available online, HITSP will include a link to information about where the code system may be obtained.

4.4.2.5 CODE SYSTEM HL7 IDENTIFIER

HL7 Version 2 does not use object identifiers to uniquely identify a code system. It uses the name of the coding system instead. The names of coding systems for HL7 messages have been specified in Table 0396 of the HL7 Version 2 standard. To facilitate use of the various code systems, HITSP will identify the appropriate code system name established in this table.

4.4.2.6 CODE SYSTEM VERSION

This attribute will record the version number of the code system using the format specified by the originator of the code system. If no version number is identified by the originator, HITSP will assign a version number using the publication date associated with the code system, in the form YYYYMMDD.

4.4.3 VALUE SET MEMBER METADATA

As a point of convenience for implementers and where space and policies permit, value sets will be included within HITSP/C80 Clinical Document and Message Terminology. Even when values sets appear within HITSP Specifications, developers should consult an authoritative source for the current version of the value set.

Value sets will be published within HITSP Specifications under the following conditions:

- The value set is small and stable, or cannot be readily obtained from a source other than HITSP
- Permission can be obtained to publish the values in the value set, or HITSP is the originator of it

Value sets that are published within HITSP Specifications will be reproduced in a table following their definition. The attributes of the value set members are illustrated below in Table 4-9. Additional metadata may be associated with values sets in metadata registries, including information such as keywords associated with the value set, or a description of the changes made to it in a given version.

Table 4-9 Value Set Concept Metadata

Concept Code	Concept Name	Code System Identifier	Code System Name	Definition	Usage Note
The code from the code system	The name of the concept from the code system	The OID identifying the code system	The name of the code system	A narrative definition of the concept	Usage notes for the concept

4.4.3.1 CONCEPT CODE

This is the code used to uniquely identify the concept.

4.4.3.2 CONCEPT NAME

The concept name is the preferred name of the concept as published by the originator of the value set. Some code systems provide more than one name for a concept. In these cases, HITSP will identify the best name to use when the standard has not identified a preferred name. HITSP will use the LOINC Short Name as the concept name for codes coming from the LOINC code system, and the fully specified preferred name as the concept name for codes coming from the SNOMED CT code system.

4.4.3.3 CODE SYSTEM IDENTIFIER AND CODE SYSTEM NAME

There may be times when a selected value set will contain concepts from more than one code system. When this occurs, the code system identifier and name will also be listed for each concept, to ensure that users know the source of each concept in the value set. These columns may be omitted when the value set definition clearly identifies that the value set comes from only one code system. The Code System Identifier and Code System Name will be the same values using the Code System Metadata table published by with the HITSP/C80 Clinical Document and Message Terminology.

4.4.3.4 DEFINITION

As an aid to implementers, HITSP will also provide the definition of the concept from the code system when it is available, or clearly indicate that the definition is not available. This will facilitate correct use of the concepts within the value set.

4.4.3.5 USAGE NOTE

Occasionally HITSP may need to describe how each concept is to be used. These will appear in the Usage Note column. This column may be omitted if it is not necessary.

4.4.4 VALUE SET VERSIONING

The process for managing HITSP value sets draws from that adopted by HL7.

Begin HL7 Version Set Versioning Process [adapted from HL7 Version 3 Core Principals]

> The definition of a value set can change over time. New identifiers may be added to or removed from a value set definition, and the rules used to construct the set may change. When a value set definition changes, it should be done in a way that ensures both the old and new versions are available for comparison.
>
> There are multiple strategies for tracking value set versions. Two of the most common are
>
> 1. To increment the version number each time a change is made to the value set
> 2. To track modification dates for each change to the value set.
>
> HITSP managed value set versions are determined by the approval date. This is date of the panel-approved publication in which it is defined is released, and not by available date (the date the value set version was made available within an HITSP) or by a version number. This policy has the following implications:
>
> 1. For enumerated value sets maintained by HITSP, the activation date and deactivation date for individual codes in the value set must be maintained as part of the value set metadata
> 2. For intensionally defined value sets in HITSP managed value sets, the activation date and superseded date must be recorded (tracked) each time the logic of the definition is changed
> 3. For externally maintained terminologies that have named or numbered releases, a table must be maintained that shows the modification dates for the named or numbered releases
>
> For externally maintained terminologies that maintain modification dates for each individual code change, no additional information is needed.

End HL7 Version Set Versioning Process [adapted from HL7 Version 3 Core Principals]

4.5 TEMPLATES

Templates are used by HL7 in Version 3 specifications such as the HL7 Clinical Document Architecture or Version 3 messages like the Care Record DSTU to further constrain their contents. A template is a formal collection of constraints (business rules) that are applied to the content of the exchange. Templates can be applied to components, modules used in a component, or to specific data elements or even parts of a data element. HITSP makes use of a number of base and composite standards that use templates, including those developed by Integrating the Healthcare Enterprise and HL7, and also uses templates of its own to define the HITSP constraints that are applied to HL7 Version 3 artifacts.

[Adapted from HL7 Version 3 Standard: Specification and Use of Reusable Constraint Templates]

> A template is an expression of a set of constraints on a model that is used to apply additional constraints to a portion of an instance of data that is expressed in terms of some other Static Model. Templates are used to further define and refine these existing models to specify a narrower and more focused scope. A template is represented by a formal definition in one or more human readable languages or notations, and may optionally include a formal definition as a static model and one or

more implementation specific representations that can be used to validate instances in a particular context. While the constraints that the template expresses must be based upon a static model, and must be expressible as a static model, templates need not actually express the constraints as a static model, though this is encouraged where possible. If a static model is present, it is considered to be a human readable notation. There is a set of metadata associated with every template to describe the purpose and use of the template.

End of HL7 Adapted material

Templates provide HITSP and SDOs with a mechanism to express the conformance rules. A number of different technical representations of those rules are in use today, and many of these can be translated to different forms. Base and composite standards that HITSP has thus far selected all express these rules in ISO Schematron. Schematron is a programming language that can be used to test artifacts for conformance to these business rules.

A unique property of templates is that an Information Exchange complying with a template asserts conformance to its business rule, allowing for easier validation of conformance.

Table 4-10 below describes the metadata that should be available for templates uses or created by HITSP in a metadata registry. The metadata is this table uses names similar to that used in the other metadata definitions within this technical note. HITSP note however, that the HL7 Version 3 Standard: Specification and Use of Reusable Constraint Templates defines a detailed list of metadata that can be used to describe templates[3]. The HITSP template metadata has been mapped to the metadata described within this specification in the last column of this table.

The table below does not include all mandatory metadata defined in that publication, but that additional metadata can certainly be stored in a template metadata registry. HITSP recommend that template metadata registries review and conform to that specification.

Table 4-10 Template Metadata

Element	Description	HITSP Template Metadata
Identifier	This is the identifier of the template	TemplateId
Name	This is the name of the template	templateName
Source	This row identifies the source of the template, the originator or publisher of it.	originatingAuthorEntityID publisher
URL	A URL pointing to an online resource defining the template	templateRepositoryIdentifier
Purpose	A brief description of the purpose for the template	intention
Definition	Brief description of the template	templateDescription
Inherited Templates	Templates may require the use of other templates for the artifact to which this template is applied. This entry indicates which templates must be used	*Not Available*
Templates Used	Templates may require the use of other templates in artifacts that are subordinate to the artifact to which this template applies. This entry indicates which of these templates are required or optional	*Not Available*
Version	This row contains a string identifying, where necessary, the specific version of the template	version
Effective Date	The date that the template becomes effective	effectiveDate

[3] As of the date of this publication HL7 is presently engaged in the development of a set of business requirements for a template registry, which members of HITSP have also been participating in. There is a great deal of cross-fertilization going on in this project, but as of today, there are no template registries currently in use. AHRQ-USHIK has captured some of the template metadata in its data element registry, and will be working with HITSP and other member SDOs to develop a template registry. PHIN-VADS has indicated that they will be developing a template registry for CDA templates describing population health and biosurveillance.

Element	Description	HITSP Template Metadata
Expiration Date	The date after which the template should no longer be used	supersededDate
Status	Active (Current) or Inactive (Retired)	publicationStatus
Creation Date	The date of the creation of the template when available	revisionHistory
Revision Date	The date of the revision of the template when available	revisionHistory

4.5.1.1 TEMPLATE IDENTIFIER

This identifier is an OID that uniquely identifies the template as described in more details below. All OIDs used in this specification can be found through the HL7 OID Registry at www.hl7.org/oid/index.cfm.

4.5.1.2 TEMPLATE NAME

This is the name of the template as established by its originator.

4.5.1.3 TEMPLATE SOURCE

An informative reference to the source of the template is always included. The originator of a template and the publisher of it may be different organizations. The source reported in this metadata is an authoritative source, and may contain either of these organizations.

4.5.1.4 TEMPLATE URL

When the template is available online, HITSP will include a link to that online resource. When it is not available online, HITSP will include a link to information about where the template definition may be obtained.

4.5.1.5 TEMPLATE PURPOSE

The purpose of the template will be included. This should explain the reason for its use.

4.5.1.6 TEMPLATE DEFINITION

This should be a brief description of what the template defines, and should include any important relationships with other templates.

4.5.1.7 INHERITED TEMPLATES

Templates for an artifact may require that same artifact to conform to other templates. For example, a CDA document conforming to the IHE Exchange of Personal Health Records (XPHR) profile requires that document to also conform to the HL7 Continuity of Care Document implementation guide (CCD). Thus, an XPHR document is a CCD document.

Identifying these relationships within the template metadata allows implementers to navigate all of the relationships and constraints applied by HITSP Specifications and their selected standards. An exchange can conform to more than one template at a time.

4.5.1.8 TEMPLATES USED

This row lists the set of templates that have been identified as being optional or required components of the artifact being constrained the source template. Again, this allows implementers to navigate the various relationships and constraints applied by HITSP Specifications.

4.5.1.9 TEMPLATE VERSION

This attribute will record the version number of the template using the format specified by the originator of the template. If the originator identifies no version number, HITSP will assign a version number using the publication date associated with the template, in the form YYYYMMDD.

4.5.1.10 TEMPLATE EFFECTIVE DATE AND TEMPLATE EXPIRATION DATE

Templates may need to be updated or withdrawn. Templates will have an effective and expiration date describing when the template was valid.

4.5.1.11 TEMPLATE STATUS

This attribute records the current status of the template. An active template is one that can be used within an exchange. An inactive template should not be used in an exchange, as it has been retired from use. It may no longer be actively maintained, be replaced by another template, or simply be no longer needed in the exchange.

4.5.1.12 TEMPLATE CREATION DATE AND TEMPLATE REVISION DATE

This metadata identifies when the template was created and last revised. This information need not be recorded in HITSP publications for Templates managed by external organizations, as the information may not be available.

Templates originating with HITSP will record the date of their creation and revision. For these templates, the creation date is the date of first publication and the revision date is the date of publication of the revision in a panel-approved specification.

4.5.2 OTHER MECHANISMS TO APPLY BUSINESS RULES TO EXCHANGES

Templates can be extended to other Information Exchange standards. Approaches similar to templates are used by other standards. For example, HL7 provides conformance profiles for HL7 Version 2 messages, and maintains a registry of conformance profiles for its members on its web site at www.hl7.org/memonly/conformance/profiles.cfm. The X12N Implementation guides selected by HITSP are similar, in that they specify the constraints on the ASC X12 standard for specific purposes, such as Eligibility and Benefits inquiry.

Conformance profiles are very similar to templates, except that they are applied to an entire message, and are not compositionally constructed.

HITSP will also be investigating the use of these technologies to apply business constraints to other sorts of Information Exchanges.

4.5.3 TEMPLATE VERSIONING

Templates can be versioned just like other artifacts. However, template versioning requires some additional considerations. Use of a template identifier in an interoperable exchange indicates conformance to a contract. The creator of an exchange artifact that asserts a template identifier is declaring that it conforms to the business rules expressed by the template. Thus, changes to a template that are not backwards compatible from the perspective of the creator of the artifact must not be made to the template. Similarly, changes that are not backwards compatible from the perspective of the artifact consumer must not be made either. In either of these cases, the old template should be retired, and a new template identifier assigned.

With regard to “backwards compatibility”, one must look at the exchange occurring between consumers and creators of the artifact that are designed using both the old and the new template definition. If an old creator cannot successfully communicate with a new consumer, or visa versa, then the changes are not considered to be backwards compatible. Changes that are typically indicative of the need to assign a new

template identifier include the strengthening of any constraint from a “should” (recommended) or “may” (optional) to a “shall”, or relaxing a constraint from a “shall” to a “should” or “may”, changing the cardinality of any contained data element, changing the conditions under which a conditional element must be sent, or altering the value set identity[4] or precision of the data being exchanged. This usually limits such changes to clarifications, or the exchange of the terms “should” (recommended) and “may” (optional), or the addition of recommended or optional capabilities that were not previously defined for the template.

HITSP managed templates will adhere to the above rules with respect to template identifiers and backwards compatibility.

4.6 CONSTRAINTS

HITSP may provide constraints in any of its specifications. These constraints may be on data elements, value sets or templates used in the specifications. HITSP documents all constraints that it has added to the selected standards, and does not usually reproduce the constraints already in the chosen standards.

Constraints can restrict the values that may appear, the order, number of occurrences, or other content appearing. Constraints do not violate selected standards. Constraints use the terms SHALL, SHOULD, MAY, SHOULD NOT, and SHALL NOT to indicate required, recommended, optional, not recommended and prohibited content within a component or data element. Constraints may be applied across multiple data elements (e.g., if this data element contains X, that data element must contain Y).

Constraints may be layered. For example, the HL7 Continuity of Care Document (CCD) specification includes a number of constraints on how to use the HL7 CDA Standard. The IHE XPHR profile builds upon the CCD and includes constraints of its own. Finally, HITSP has constraints that it places up the IHE profile. A component conforming to the HITSP Specification must conform to all layered constraints.

Typical examples of HITSP constraints are:

- Require the use of particular data elements within the exchange, even though they are optional in the base or composite standard
- Require the use of a specific value in a data element. For example, in the HITSP T40 Patient Health Plan Eligibility Verification transaction, HITSP require that the 270_2100C_NM108 66, Identification Code Qualifier Description shall be the value MI to indicate “Member Identification”
- Require conformance to one or more templates within an exchange. For example, in C83 CDA Sections and Entries, HITSP requires the History of Past Illness Section of a CDA document to conform to templates from one IHE and two HL7 implementation guides
- Require the use of a specific code system or value set
- Require information to be supplied to a specified degree of precision. For example, HITSP might require birth date to be precise at least to the year
- Require information to be supplied in particular units. For example, HITSP could require age to be specified in years, volume of fluid in milliliters, or velocity in furlongs per fortnight

Constraints on HITSP Data Elements will be provided in the HITSP/C83 CDA and CCD Content Modules. The value sets that are used by these HITSP constraints will appear in the HITSP/C80 Clinical Document and Message Terminology. Other HITSP Specifications may supply constraints where there are business requirements relevant only to that specification.

4.6.1 CONSTRAINT LANGUAGE

The key words **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT** and **MAY** are to be interpreted as described in RFC 2119 and will appear when used in that fashion in this **TYPEFACE** within HITSP Specifications.

[4] If a template uses value set 10 in one definition, and then uses value set 11 in a new definition, this is a change in value set identity.

The key words **REQUIRED** and **OPTIONAL** are also to be interpreted as described in RFC 2119 when they are used to indicate the optionality of components used in an exchange.

Constraints within HITSP Specifications will appear in a distinctive style to differentiate them from other text, and they will be uniquely identified within that specification. The constraint identifier will be based upon the identifier of the artifact being constrained, and the sequential number of the constraint on that artifact. For example, constraints on a HITSP Data Element will use the data element identifier. An example HITSP constraint is shown below:

> [DE-7.04-1] The problem type **SHALL** be coded as specified in HITSP/C80 Section 2.2.1.1.4.1.2 Problem Type.

The constraint identifier is enclosed in brackets. The first two characters indicate that this is a constraint on a data element defined within the HITSP/C83 CDA and CCD Content Modules. The next four characters uniquely identify the data element being constrained. The final number indicates that this is the first constraint on this data element within this document.

A variety of different entities described within HITSP Specification are subject to constraints in addition to data elements. Table 4-11 describes what is used to uniquely identify each of these entities.

Table 4-11 Constraint Identifier Components

Constraint Target	Code	Unique Identifier
Capability	CAP	Capability Number
Component	C	Component Number
Component Subtype[5]	CT	Subtype sequence number within the Component
Data Element	DE	Data Element Identifier from C83
Service Collaboration	SC	Service Collaboration Number
Subset	SUB	Subset Sequence Number
Transaction	T	Transaction Number
Transaction Package	TP	Transaction Package Number

The unique identifier of a HITSP constraint can be determined from the unique identifier of the document that defines it, followed by its unique identifier within the document. The constraint given in the example above is uniquely identified by the string C83-[DE-7.04-1]. This is the form that should be used to reference a constraint defined in a HITSP document from any other external source (e.g., an error report from a testing tool, or from within another HITSP Specification).

4.6.1.1 SUBSETS

HITSP defines a number of subsets of the HITSP/C32 Summary Documents Using HL7 Continuity of Care Document (CCD) in several of its Interoperability Specifications. The creation of these subsets is driven by the data requirements of the exchanges used in the Interoperability Specification. For example, HITSP/IS07 Medication Management defines the *Medication and Allergies Information Coded Subset* on the HITSP/C32 Summary Documents Using HL7 Continuity of Care Document (CCD). This subset was defined to ensure that medication and allergy information could be communicated using the HITSP/C32 Summary Documents Using HL7 Continuity of Care Document (CCD) when it is used in the context of medication reconciliation.

When HITSP defines a subset, it will always conform to the underlying constraints given by the HITSP Specification upon which it is based. Thus, all constraints with respect to vocabulary, data elements, et cetera, must be adhered to within the subset. In the example given above, the *Medications and Allergies*

[5] For example, C48 and C84 describe constraints for two different document subtypes within the component.

Subset conforms to the requirements of the HITSP/C32 Summary Documents Using HL7 Continuity of Care Document (CCD).

A subset is a collection of constraints that the Interoperability Specification (IS) requires upon the use of the HITSP construct within the context of that IS. Each subset will be associated with a template that is based upon the template established or used for the HITSP construct. The template identifiers are created for the purpose of supporting template metadata registries, and need not be used in the exchange of the HITSP construct specified in the IS. These templates will be created during the next maintenance cycle using the OID that HITSP has obtained for its templates[6].

[6] The OIDs for these templates will start with 2.16.840.1.113883.3.88.11, and will be followed with the IS number, and the subset sequence number in the IS where it was defined.

5.0 METADATA REGISTRIES

HITSP Specifications rely on copyrighted material published by a number of different standards development organizations. While HITSP chooses the base and composite standards, the HITSP Specifications do not re-document these copyrighted works. In order to understand the full extent of the HITSP Specifications, implementers and users need easy access to the full set of information. To simplify navigation of this material HITSP will be making use of metadata registries.

A metadata registry is an information system that stores data about other data and ensures consistency of that data. Its primary purpose is to help rapidly locate content by searching the available metadata. A metadata registry has a formal data element submission and publishing approval process. Each data element is accepted by a data stewardship team and reviewed before data elements are published. After publication change control processes are used. This allows collaboration in the registration and use of data elements by implementing a governance process.

The metadata for the **Data Element**s, **Template**s and **Value Set**s that HITSP Specifications makes use of will be stored in metadata registries. This will enable the retrieval of it by humans and software applications. This information may be used to support implementation and testing of systems conforming to HITSP Specifications. Different types of metadata registries may be applicable for each of the three entities: data elements, value sets and templates.

5.1 RELATIONSHIP OF HITSP AND DATA REGISTRIES

The relationship between HITSP and public data registries plays a key role in the success of the understanding and implementation of HITSP Specifications. HITSP has engaged with registry developers at AHRQ-USHIK and CDC-PHIN-VADS in order to support the implementation and testing of its specifications, and is also working with the HL7 Templates Workgroup to help define the business requirements for a Template Registry.

These agencies are working together to provide Metadata Registries to facilitate access to the complete information. They can register information not only in HITSP Specifications, but also data elements from the composite and base standards, and the value sets, and templates that HITSP has selected. Much of the metadata already available from the standards has been loaded into ARHQ-USHIK. CDC-PHIN-VADS has also loaded the value sets from the HITSP Specifications, and supports a number of other value sets useful for public health.

In addition to working with these registry suppliers, HITSP will also be working with its member SDOs to ensure that up to date information about these are available in these registries. The registries support not only implementation and testing of HITSP Specifications, they also provide data mining tools to browse, search, query, compare, and gather information for review and analysis. These data mining tools are currently used by HITSP to identify data elements and value sets that may be appropriate for use in harmonization requests.

In deciding that HITSP need to use registries to support the deployment of HITSP Specifications, HITSP has also identified several key issues with respect to governance and access to registry information. These issues are detailed in the sections below.

5.1.1 RIGHTS TO INFORMATION

Registries are a place to register information from a variety of sources. The organization or agency of origin of the information registered shall remain the authoritative source for the data, and shall retain any rights to such data, whether in paper or digital form. This is true for HITSP Specifications and also for any SDO specifications that may be conveyed in a registry.

The registry needs to have written and publicly available policies and procedures that govern the use and access to content. These policies must indicate how the rights of the organization or agency of origin are

protected. The registry must clearly document the intellectually property rights asserted by the owners of the information they retain, and make that information available to users of that information. The registry must also provide contact information for the originators of information they maintain. The registry must have processes in place to identify authorized representatives of information originators and to make agreements with information originators to maintain data within the registry.

5.1.2 LACK OF BARRIERS AND EASE OF ACCESS

Use of the registry shall be open to all persons who are directly and materially affected by the material that is maintained within the registry. There must be no undue financial barriers to use of the registry. Use of the registry shall not be conditional upon membership in any organization or unreasonably restricted on the basis of technical qualifications or any other such requirements.

Information originators must be able to supply the registry with information in a computable format for upload. The registry shall clearly document the computable formats it supports. Similarly, information consumers must be able to download information in a computable format.

Information exchanges with the registry should be based upon standards (e.g., HITSP/T66 Terminology Service, ISO 11179, and ISO 15000, HL7 Version 3: Specification and Use of Reusable Constraint Templates) and commonly available tools (e.g., spreadsheets and text files) and technologies (e.g., XML). It must be vendor and implementation neutral with respect to the exchange of registry metadata (for example, the registry shall not require a specific operating system or programming language be used by an information consumer or originator to make use of the information).

Registries should be compliant with Section 508 of the US Rehabilitation Act (see www.section508.gov) to support information access by people with disabilities. Thus, information consumers should also be able to access the information in ways that comply with that act.

5.1.3 MAINTENANCE

Metadata registry suppliers have a responsibility to maintain the registry infrastructure. This requires customary maintenance associated with information systems, including such things as backup, technical support, bug reporting and fixes, and predictable update and maintenance schedules.

In addition to the normal maintenance responsibilities, registries act as an intermediary between consumers of registry information and originators of that information. It will not always be possible for consumers to determine whether the source of an issue is the registry, or the information originator. The registry must therefore have processes in place that support the communication of consumer issues to the information originators.

Registries will be key components for information consumers and originators. Registries must therefore take into account the business requirements of those stakeholders to support scheduled updates of registry information. For example, several key value sets are updated on an annual basis, and consumers and originators will have times of peak need with respect to those annual cycles.

Information originators will need a mechanism to develop and verify information before it is made publicly available. Registries must therefore document the mechanisms or processes by which they support such efforts. Such efforts could for example, be supported by having multiple versions of the registry available for staging and production. It might also be supported by workflows that allow draft content only accessible to authorized reviewers. HITSP hasn't identified a preferred process; only that such a process must exist.

5.1.4 VERSIONING

Specifications may be superseded over time; however registries should maintain past versions online for research and reference purposes. Registries shall have a mechanism to uniquely identify each distinct

version of an entity for which they retain metadata. Registries shall also document the process used to remove or change information in the registry once it has been published.

In addition to registry responsibilities, information originators and consumers also have certain responsibilities. Information consumers must agree to use registry metadata only in ways that are described in the registries published policies for use. Information originators must agree to maintain the data that they supply to the registry, and make corrections or updates to it in a timely fashion.

5.2 DATA REGISTRY "AUDIENCE VIEWS"

Data within registries need to be accessed by many different types of audiences. These audiences have different information requirements; therefore registries need to provide capabilities to support different objectives and views suitable to their audience. Many of the audiences for these views are the same as the Intended Audience for this document.

Some necessary capabilities for metadata registries are suggested below. HITSP continues to work with its partners and member organizations to develop a full set of requirements for metadata registries separate from this technical note. A draft of these requirements will be delivered to the HITSP program management team by July 15th, 2009.

Search Types:

Users should be able to base their searches on any of the metadata described previously in this document. Registries should support the following search capabilities: Exact, Full Word, Contains, Begins with, Ends with and Wild cards on key metadata elements.

5.2.1 INFORMATION VIEWS

Registries should support the creation of views based on:

- a single HITSP Specification (e.g., IS 01 EHR Lab Results Reporting)
- a HITSP data module (e.g., Immunizations, Medications)
- A concept domain (e.g., Demographics, Laboratory Results, Procedures)
- Code Systems used (e.g., SNOMED CT, LOINC, ICD-9 CM)
- Data Exchange Standards utilized (HL7 Version 2, HL7 Version 3, X12N, NCPDP)
- Version Specific Data Exchange structures (e.g., HL7 Version 2 messages using VXU, Version 3 using CDA, X12N using the 271)

Users can also browse the data elements and find its associated value sets. Users can use the filters mentioned above as views while searching the value sets downloads.

Registries should support download by information consumers of core metadata for one or more entities (value sets, templates or data elements), or value set concepts. The registry should support output in commonly used formats, e.g., Tab-delimited, spreadsheet or XML formats, or suitable for import into SQL tables.

5.2.2 UPLOADS

Registries should support upload of metadata by information originators of core and extended metadata utilizing a standards based format.

5.2.3 INTERFACE WITH HITSP PRODUCTION TOOLS

As HITSP begins to make use of metadata registries, it will begin developing tools to facilitate the production of HITSP Specifications using those registries. It is important to HITSP that metadata registries are willing to coordinate with HITSP and to facilitate this use.

6.0 APPENDIX

The following sections include relevant materials referenced throughout this document.

6.1 XPATH LOCATIONS IN XML

HITSP has chosen to use XPath notation to identify data elements within standards that use XML to represent their information content. The XML Path Language (XPath) 1.0 is a standard published by the W3C for the express purpose of locating information in an XML document. The formal specification for XPath can be found here: www.w3.org/TR/xpath.html

What follows is a brief primer on how to read XPath expressions used in HITSP Specifications. It assumes that you have some familiarity with XML, and glosses over many of the fine points of XPath, but covers most of what you would find within a HITSP Specification. For more details, see the standard itself at the URL listed above.

An XPath expression can be thought of as a function operating upon an XML document, and a collections of locations in that document, that is called the XPath context. The context can be viewed as the set of starting points that the function starts its search from. The result of the XPath expression is a new, possibly empty collection of locations within the document that meet the criteria it describes. XPath expressions can be "strung together" by feeding the result of one expression in as the context for the next one.

The following XPath expressions are basic primitives that can be used to find just about any location within an XML document.

Table 6-1 XPath Primitives

Expression	Returns
.	Is that point within the document
..	Is the parent element of that point.
/	By itself Is the root of the document
*	Is any child element that can be reached from that location. Not all locations within a document have element children. Only the root of the document or another element in the document can have an element child
name	Locate the child element with the given name
@name	Locate the attribute of the element
ns:name	Locate the child element in the ns Namespace and the given name
Expression/name	Locate any child element of *expression* with the given name
Expression//name	Locate any descendent element of *expression* with the given name
Expression[*condition*]	Match the XPath expression if the condition within [] is true. The result of this is an XPath expression, so it is possible to have multiple brackets. The condition is a Boolean expression which can use the usual set of Boolean operators
Expression[*number*]	This is shorthand for [position()=number], and allows you to treat the result of *Expression* as an array. *Expression*[1] returns the first location found by *Expression, Expression*[2] is the second, and so forth. The "array" is usually in document order, but if you are searching upwards (see ancestor:: below) or backwards (see preceding-sibling:: below), they appear in the reverse order

The default direction to move in an XPath expression is down the hierarchy of children. However, since XPath is working with tree structured information model, you can also look in other "directions" known in XPath as axes by prefixing the expression with one of the following keywords.

Table 6-2 XPath Axis Keywords

Keyword	Direction
ancestor::	Search upwards through the parent and any ancestor (parents of parents and so on).
ancestor-or-self::	Search upwards through the parent and any ancestor, including the context element.
attribute::	Search the element for any of its attributes.
child::	Search through all the children (the default)
descendant::	Search through all of the descendants (children, and all their children and so on)
descendant-or-self::	Like ancestor-or-self, but headed down.
following-sibling::	Search elements with the same parent that are after the context
parent::	Test the parent
preceding-sibling::	Search elements with the same parent that are after the context
self::	Test the context node

Table 6-3 below shows how HITSP uses XPath expressions within its specifications. This example is for teaching purposes and should not be used outside that context. The first column of this table identifies the location within a CDA document that is being mapped to the HITSP Data Element in the second column. Note the indentation in that column because it is important.

Table 6-3 Example Use of XPath in HITSP Specifications

CDA Data Location	HITSP Data Element Identifier and Name	R/O
/cda:ClinicalDocument/cda:effectiveTime	X.01 – Exchange Timestamp	R
/cda:ClinicalDocument/cda:recordTarget/cda:patientRole	Information Entry	R
cda:id	X.02 – Subject ID	R2
cda:addr	X.03 - Subject Address	O
cda:telecom	X.04 - Subject Phone /Email /URL	O
cda:patient	Personal Information	O
cda:name	1.05 - Person Name	R

The first row contains the XPath Expression **`/cda:ClinicalDocument/cda:effectiveTime`**. This expression begins with the / character, so that part of it identifies the root of the document. That becomes the context for the next part **`cda:ClinicalDocument`**, which is the name of element in the CDA document. If the first element in the XML document is a **`ClinicalDocument`** element in the CDA namespace, then the expression matches that element and returns it. The next part of the expression **`/cda:effectiveTime`** will match all **`effectiveTime`** elements in the CDA namespace that are immediate children of the **`ClinicalDocument`** element.

The expression in the second row finds all **`patientRole`** children of all **`recordTarget`** children of the **`ClinicalDocument`** element.

If you had to write all of that out for every single element that needed to located, you'd quickly get tired of typing, and HITSP wouldn't have nearly enough room in our documents. To simplify matters, whenever the context of the current location can be inferred relative to the previously described location, HITSP do not copy that part of the expression, and HITSP indent the next part. For the third row of the table then, the full XPath expression is **`/cda:ClinicalDocument/cda:recordTarget/cda:patientRole/cda:id`**. The fourth XPath expression isn't relative to position of the third one, but it is relative to the position of the second. So it stays at the same level of indentation as the previous one. HITSP merge the location of the XPath expression at the next higher level of indentation with the expression in the fourth row to get **`/cda:ClinicalDocument/cda:recordTarget/cda:patientRole/cda:addr.`** This can occur

multiple times, so when HITSP finally get to the last row in this table, HITSP joined together 3 different XPath expressions to find the location. It may sound complex, but it's not as hard as it seems.

A last note about how these tables work; the information in the table appears in the order that it appears in the CDA document. Look carefully at the **`cda:name`** and **`cda:patient`** data locations. The table shows that **`cda:name`** is required. However, this it lives "inside" **`cda:patient`**, which is optional. This means that when you transmit the **`cda:patient`** element, you must transmit the **`cda:name`** element inside it. It doesn't mean that you always need to transmit the **`cda:name`** element. If you never transmitted **`cda:patient`** the **`cda:name`** element could never appear.

Although the above example has been shown using CDA, it works equally well with any other XML based standard for Information Exchange.

7.0 DOCUMENT UPDATES

The following sections provide the history of all changes made to this document.

7.1 JUNE 30, 2009

No changes. This is the first published version of the document.

7.2 JULY 8, 2009

Upon approval by the HITSP Panel on July 8, 2009, this document is now Released for Implementation.