2.B Control (continued)

Chapter Chair	Frank Oemig Agfa HealthCare GmbH
Chapter Chair	Jason Rock GlobalSubmit
Chapter Chair	Jennifer Puyenbroek SAIC - Science Applications International Corp
Chapter Chair	Jane Gilbert AHML/University of Ballarat
Sponsoring Work Group:	Implementation/Conformance
List Server:	impl@lists.hl7.org

2.B CONFORMANCE USING MESSAGE PROFILES

Previous sections in this chapter define the rules and conventions for constructing and communicating a message including the parts of a message structure. Messages that adhere to those rules of a specific version of a standard are **compliant** to that version of the standard.

Compliance to the HL7 Standard has historically been impossible to define and measure in a meaningful way. To compensate for this shortcoming, vendors and sites have used various methods of specifying boundary conditions such as optionality and cardinality. Frequently, specifications have given little guidance beyond the often-indefinite constraints provided in the HL7 Standard.

This section presents the methodology for producing a precise and unambiguous specification called a ***message profile***. Messages that adhere to the constraints of a message profile are said to be **conformant** to the profile. For conformance to be measurable, the message profile must specify the following types of information:

- What data will be passed in a message.
- The format in which the data will be passed.
- The acknowledgement responsibilities of the sender and receiver.

A conformance statement is a claim that the behavior of an application or application module agrees with the constraints stated in one or more message profiles. This section defines the message profile; however, the conformance statement will not be discussed further in this document.

Definition: An HL7 message profile is an unambiguous specification of one or more standard HL7 messages that have been analyzed for a particular use case. It prescribes a set of precise constraints upon one or more standard HL7 messages.

An HL7 message profile is compliant, in all aspects, with the HL7 defined message(s) used in the profile. It may specify constraints on the standard HL7 message definition.

A message profile fully describes a conversation between two or more systems through the combination of the following:

a) one use case analysis,

b) one or more dynamic definitions,

c) one or more static definitions, and

d) one table (vocabulary) definition.

The *use case analysis* may be documented as a use case diagram (supported with text) or just a textual description (See section *2.B.2*, "*Use case model*".)

The dynamic definition is an interaction specification for a conversation between 2 or more systems (See Section.*2.B.3*, "*Dynamic definition*".)

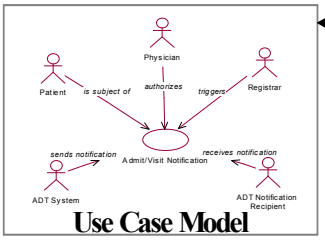
The static definition is an exhaustive specification for a single message structure (see Section *2.B.4*, "*Static definition*"). Normatively expressed as an XML document validated against the normative message profile Schema, it may be registered on the HL7 web site (see Section *2.B.10*, "*Message profile document*").

The table (vocabulary)definition is an exhaustive specification for a single message structure (see Section *2.B.5*, "*Table definition*"). Normatively expressed as an XML document validated against the normative message profile Schema, it may be registered on the HL7 web site (see Section *2.B.10*, "*Message profile document*").

For detailed background information regarding message profiles, the reader is referred to the Implementation/Conformance Work Group balloted informative document, "Message Profiling Specification, Version 2.2", published November 30, 2000, upon which this section is based. This document is available from the HL7 Resources Web site (*http://www.hl7.org*).

A sample message profile is shown on the next page to assist in illustrating the constituents of a message profile and how they work together.

Message Profile Example



Use Case Model

1 Use Case Model

1.1 Use Case: Admit/Visit Notification

2. Dynamic Interaction Model

3 Dynamic Definition: ADT/ACK (Event A01)

3.1 ADT^A01
3.2 ACK^A01

4 Static Definition: - Message Level - ADT/ACK (event A01)

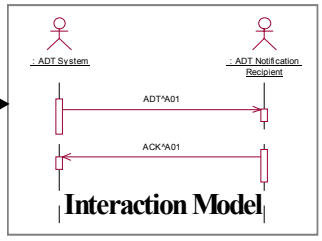
4.1 ADT^A01
4.2 ACK^A01

5 Static Defintiion - Segment Level

5.1 MSH – Message Header Segment Definition
5.2 EVN – Event Type Segment Definition
5.3 PID (Y) – Patient Demographics Segment Definition
5.4 PD1 – Patient Additional Demographic Segment Definition
5.5 NK1 – Next of kin Segment Definition
5.6 PV1 (2) – Admit Visit Info Segment Definition
5.7 AL1 – Allergy Segment Definition
5.8 MSA – Message Acknowledgment Segment Definition
5.9 ERR – Error Segment Definition

6 Static Definition - Field Level

6.1 Table 0001 – Sex
6.2 Table 0002 – Marital Status
6.3 Table 0003 – Event Type Code
6.4 Table 0004 – Patient Class
6.5 Table 0005 – Race
6.6 Table 0006 – Religion
6.7 Table 0007 – Admission Type
6.8 Table 0008 – Acknowledgement Code
6.9 Table 0009 – Ambulatory Status



Interaction Model

Segment	ADT Message	Usage	Cardinality	Chapter
MSH	**Message Header**	**R**	**[1..1]**	**2**
EVN	**Event Type**	**R**	**[1..1]**	**3**
PID	**Patient Identification**	**R**	**[1..1]**	**3**
[PD1]	Additional Demographics	X	[0..0]	3
[{ ROL }]	Role	X	[0..0]	12
[{ NK1 }]	**Next of Kin / Associated Parties**	**RE**	**[0..3]**	**3**
PV1	**Patient Visit**	**R**	**[1..1]**	**3**
[PV2]	**Patient Visit - Additional Info.**	**RE**	**[0..1]**	**3**
[{ ROL }]	Role	X	[0..0]	12
[{ DB1 }]	Disability Information	X	[0..0]	3
[{ OBX }]	Observation/Result	X	[0..0]	7
[{ AL1 }]	**Allergy Information**	**RE**	**[0..*]**	**3**
[{ DG1 }]	Diagnosis Information	X	[0..0]	6
[DRG]	Diagnosis Related Group	X	[0..0]	6
[{		X	[0..0]	
PR1	Procedures	X	[0..0]	6
[{ ROL	Role	X	[0..0]	12
}]				
}]				
[{ GT1 }]	Guarantor	X	[0..0]	6
[{		X	[0..0]	
IN1	Insurance	X	[0..0]	6
[IN2]	Insurance Additional Info.	X	[0..0]	6
[{ IN3	Insurance Additional Info - Cert.	X	[0..0]	6
}]				
[{ ROL	Role	X	[0..0]	12
}]				
}]				
[ACC]	Accident Information	X	[0..0]	6
[UB1]	Universal Bill Information	X	[0..0]	6
[UB2]	Universal Bill 92 Information	X	[0..0]	6
[PDA]	Patient Death and Autopsy	X	[0..0]	3

Static Definition – Message Level

Type	Table	Name	Value	Description	Source	Used
AWO suggested	0007	Admission Type			HL7 2.3, AWO, UB92	PV1-4
	0007		A	Accident	HL7	
	0007		E	Emergency	HL7	
	0007		L	Labor and Delivery	HL7	
	0007		R	Routine	HL7	
	0007		ACC	Accident		
	0007		EMR	Emergency		
	0007		LAD	Labor and Delivery		
	0007		NEB	Newborn		
	0007		ROU	Routine		
	0007		OBS	Observation		
	0007		URG	Urgent		
	0007		ELE	Elective		
	0007		REH	Rehabilitation		
	0007		HEL	Helicopter		
	0007		NEP	Newborn Premature		
	0007		NEP	Newborn Sick Premature		
	0007		OCH	Occupational Health		
	0007		OTI	Outpatient to Inpatient		

Static Definition – Field Level

Vocabulary

1.1 ADT^A01

The dynamic profile of the ADT^A01 event notification requires that an HL7 Accept Acknowledgement be sent but that no HL7 Application Acknowledgement is sent in response to this ADT^A01 message.

Dynamic Profile Identifier:

[awg(1) hl7(1) v2-3(5) dynamic-profile(2) accAL-appNE(5)]

1.2 ACK^A01

The dynamic profile of the ACK^A01 acknowledgement requires that neither an HL7 Accept Acknowledgement or an HL7 Application Acknowledgement is sent in response to this ACK^A01 message.

Dynamic Profile Identifier:

[awg(1) hl7(1) v2-3(5) dynamic-profile(2) accNE-appNE(6)]

Dynamic Definition

SEQ	LEN	DT	Usage	Cardinality	TBL#	ITEM#	ELEMENT NAME
1	4	SI	X			00104	Set ID - PID
2	20	CX	RE	[1..1]		00105	Patient ID
3	20	CX	R	[1..*]		00106	Patient Identifier List
4	20	CX	X			00107	Alternate Patient ID - PID
5	48	XPN	R	[1..*]		00108	Patient Name
6	48	XPN	RE	[1..*]		00109	Mother's Maiden Name
7	26	TS	RE			00110	Date/Time of Birth
8	1	IS	RE		0001	00111	Sex
9	48	XPN	X			00112	Patient Alias
10	80	CE	X		0005	00113	Race
11	106	XAD	RE	[1..3]		00114	Patient Address
12	4	IS	X		0289	00115	County Code
13	40	XTN	RE	[1..3]		00116	Phone Number - Home
14	40	XTN	RE	[1..3]		00117	Phone Number - Business
15	60	CE	X		0296	00118	Primary Language
16	80	CE	X		0002	00119	Marital Status
17	80	CE	X		0006	00120	Religion
18	20	CX	X			00121	Patient Account Number
19	16	ST	RE			00122	SSN Number - Patient
20	25	DLN	X			00123	Driver's License Number - Patient
21	20	CX	X			00124	Mother's Identifier
22	80	CE	X		0189	00125	Ethnic Group
23	60	ST	RE			00126	Birth Place
24	1	ID	X		0136	00127	Multiple Birth Indicator
25	2	NM	X			00128	Birth Order
26	80	CE	X		0171	00129	Citizenship
27	60	CE	X		0172	00130	Veterans Military Status
28	80	CE	X		0212	00739	Nationality
29	26	TS	X			00740	Patient Death Date and Time
30	1	ID	X		0136	00741	Patient Death Indicator

Static Definition – Segment Level

2.B.1 Message profile

Definition: An HL7 message profile is an unambiguous specification of one or more standard HL7 messages that have been analyzed for a particular use case. Each message profile may have a unique identifier as well as publish/subscribe topics.

2.B.1.1 Message profile identifier

Each message profile may have a unique identifier to facilitate reference.

2.B.1.2 Message profile publish/subscribe topics

The message profile publish/subscribe topics is not required to be unique but might be used by publish/subscribe systems to convey aspects of the message profile (see *MSH-21 Message Profile Identifier* in the opening section of chapter 2).

The topics are not a normative constituent of the message profile but, if provided as part of the metadata, should be in the format described below. The topic elements will be separated by the dash (-). Any element that does not have a value should use null. As this information may be used in a message instance; it should not contain any HL7 message delimiters.

Message Profile Publish/Subscribe Topics Elements

Seq	Topic Element Name	Value
1	Implementation/Conformance Work Group ID	confsig
2	An organization identifier	Abbreviated version of the organization name
3	The HL7 version	Refer to *HL7 Table 0104 - Version ID* for valid values
4	Topic Type	profile
5	Accept Acknowledgement	The accept acknowledgement responsibilities.(refer to *HL7 Table 0155 – Accept/application Acknowledgment Conditions* for valid values)
6	Application Acknowledgement	The application acknowledgement responsibilities (refer to *HL7 Table 0155 – Accept/application Acknowledgment Conditions* for valid values)
7	Acknowledgement Mode	Deferred or Immediate

An example of message profile publish/subscribe topics:

```
confSig-MyOrganization-2.4-profile-AL-NE-Immediate
```

2.B.2 Use case model

Definition: A use case model documents the scope and requirements for an HL7 message profile or set of message profiles.

The use case model must:

a) Provide a name that clearly and concisely defines the exchange

b) Document the purpose for each message exchange

c) Define the actors, including the sending and receiving applications

d) Define the flow of events between these actors including, where appropriate, derived events

e) Document the situations in which the exchange of a particular HL7 message profile is required

Refer to the HL7 V3.0 Message Development Framework (MDF 99) for further information on use case models and their uses within HL7.[1]

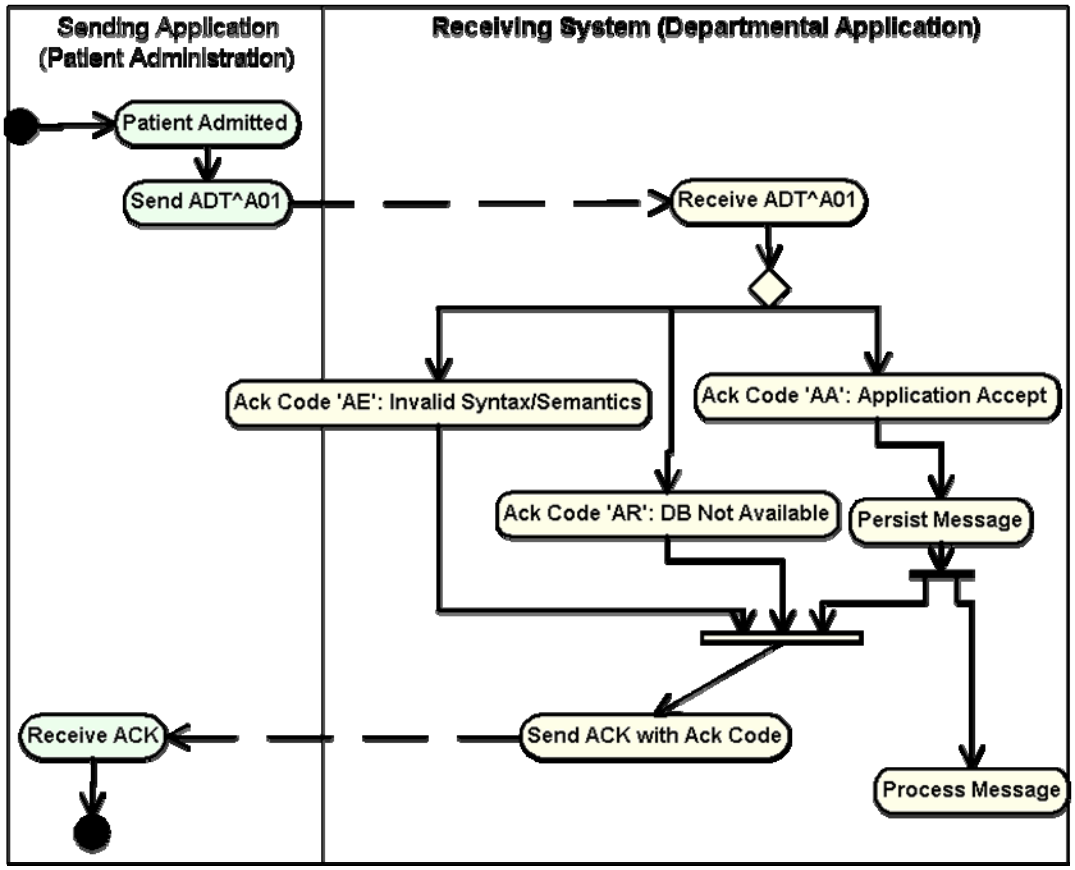
2.B.3 Dynamic definition

Definition: The dynamic definition is an interaction specification for a conversation between 2 or more systems. It may reference one to many static definitions. The dynamic definition may include an interaction model in addition to the acknowledgement responsibilities.

2.B.3.1 Interaction model

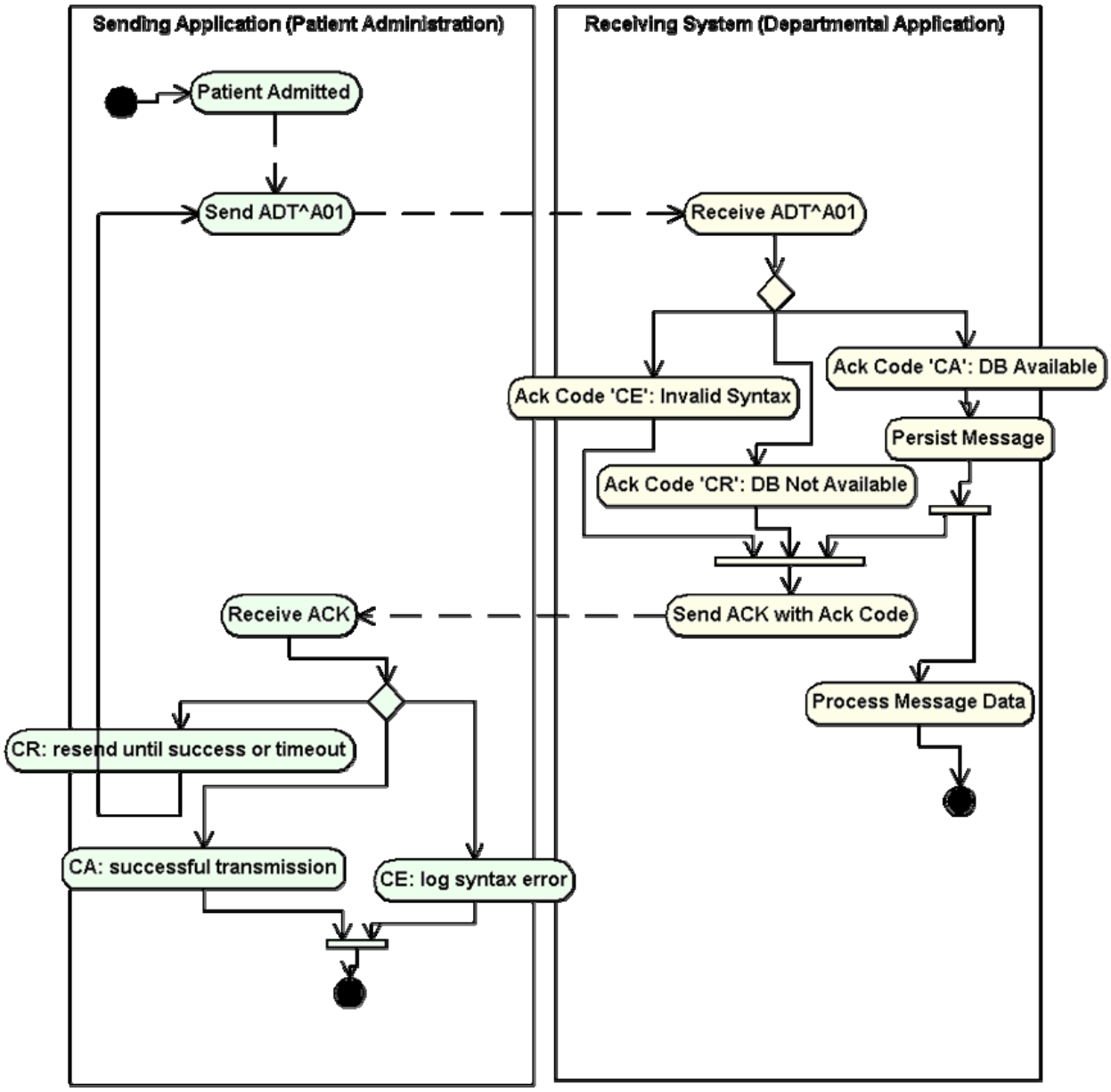
Definition: The Interaction Model illustrates the sequence of trigger events and resulting message flows between 2 or more systems. It may be in literal or graphical form. Graphical form should be a UML activity diagram. Example activity diagrams are shown here for the original and enhanced acknowledgement modes.

Interaction Model Example – ADT^A01/ACK^A01 (Original Acknowledgement Mode)



[1] Even though the MDF is a HL7 v3.0 document, this is general information not bound to v3.0 and, therefore, can be applied to this v2.x document. HL7 is currently working on a HL7 Development Framework (HDF) that will replace the MDF. Once approved by HL7, the HDF will replace the MDF.

Interaction Model Example – ADT^A01/ACK^A01 (Enhanced Acknowledgement Mode)



2.B.3.2 Acknowledgements

The specific HL7 acknowledgements required and/or allowed for use with the specified static definition of the HL7 message profile shall be defined. Specifically, the dynamic definition shall identify whether an **accept** and/or **application** level acknowledgement is allowed or required.

For any one static definition there may be one or more dynamic definition.

The dynamic definition shall define the conditions under which an accept and/or application level acknowledgement is expected.

Allowed conditions include:

a) Always

b) Never

c) Only on success

d) Only on error.

2.B.4 Static definition

Definition: The static definition is an exhaustive specification for a single message. Normatively expressed in XML, it may be registered on the HL7 web site (See Section *2.B.10*, "*Message profile document*"). The static definition is based on a message structure defined in the HL7 Standard. The message code, trigger event, event description, role (Sender or Receiver) and, if applicable, the order control code will be provided. A complete static definition shall be defined at the **message**, **segment**, and **field** levels. A static definition is compliant in **all aspects** with the HL7-defined message it profiles. However, the static definition may define additional constraints on the standard HL7 message.

A static definition identifies only those specific elements of a standard HL7 message that are used in the exchange.

A static definition explicitly defines:

a) Segments, segment groups, fields and components usage rules

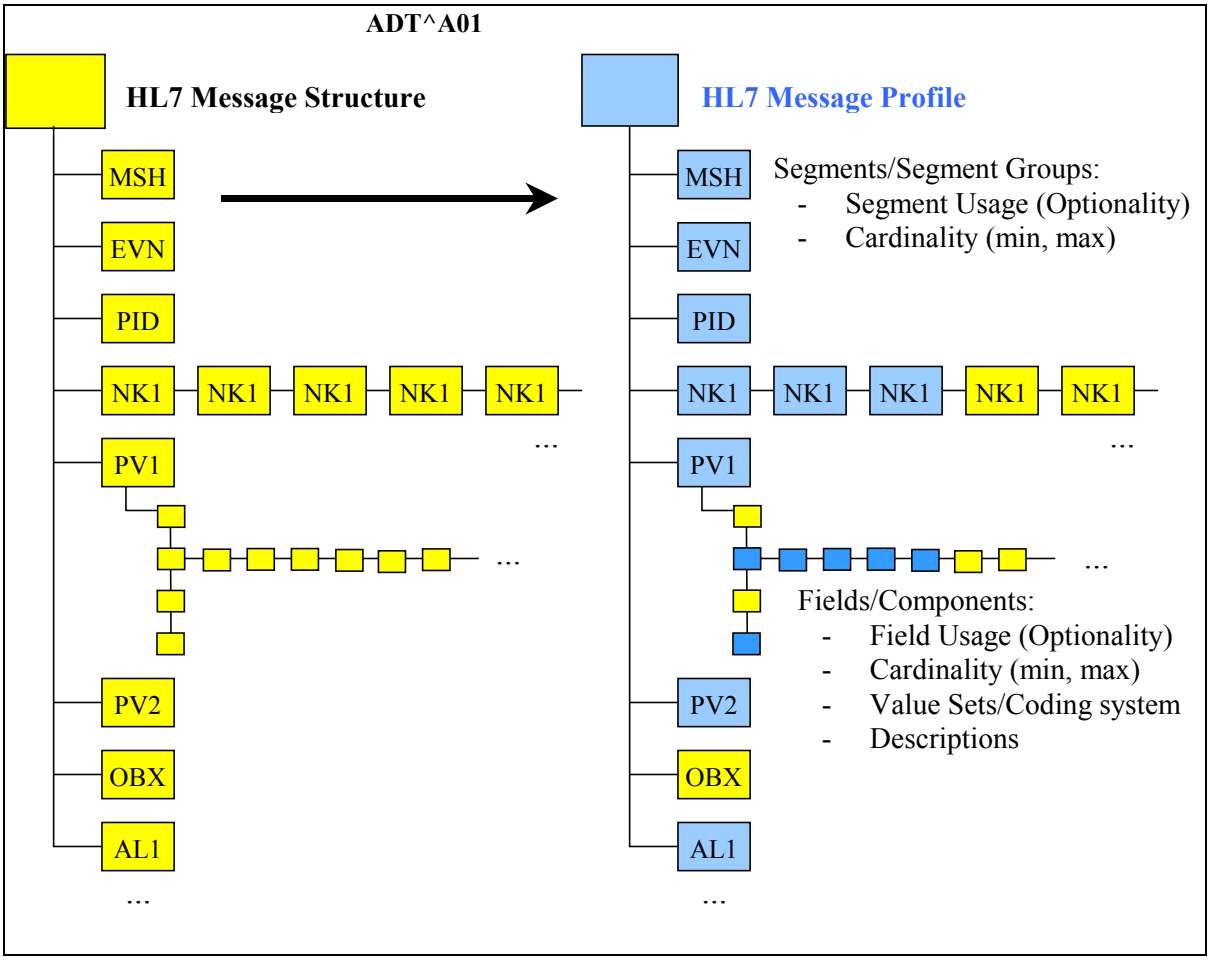
b) Cardinalities

c) Length information

d) Value sets and coding systems.

The following figure depicts, in a graphical way, the concept that the static definition is an overlay of the HL7 message structure further constraining it. For example, where the HL7 message structure shows unlimited number of NK1 Segments, the static definition allows for only three repetitions. Additionally, fields that are optional in the HL7 message structure may be required within the HL7 static definitions.

Static Definition Illustration



2.B.4.1 Static definition identifier

Each static definition must have a unique identifier when registered (See section *2.B.10, "Message profile document").* An authority other than the registry may define this identifier. If, at the time of registration, the static profile does not have an identifier assigned by the submitter's authority, the registry authority will assign one. The static definition identifier would be the identifier used if a system asserts a strict conformance claim (see *MSH-21 Message Profile Identifier* in the first section of chapter 2).

2.B.4.2 Static definition publish/subscribe topics

Static definition publish/subscribe topics convey the static definition aspects of the message profile. These topics may be used by publish/subscribe systems (see *MSH-21 Message Profile Identifier* in the first section of chapter 2).

The topics are not a normative constituent of the message profile but, if provided as part of the metadata (see section *2.B.10, "Message profile document"*), should be in the format described below. The topic elements will be separated by the dash (-). Any element that does not have a value should be null (nothing between the dashes). As this information may be used in a message instance, it should not contain any HL7 message delimiters.

Static Definition Publish/Subscribe Topics Components

Seq	Topic element name	Value(s)
1	Implementation/Conformance	confsig

	Work Group ID	
2	An organization identifier	Abbreviated version of the organization name
3	The HL7 version	Refer to *HL7 Table 0104 – Version ID* for valid values
4	Topic Type	static
5	Message Type Code	Refer to *HL7 Table 0076 - Message type* for valid values
6	Event Type	Refer to *HL7 Table 0003 - Event Type* for valid values (this table may be extended by locally defined Z trigger events)
7	Order Control Code	Refer to HL7 Table 0119 - Order Control Codes for valid values
8	Structure Type	Refer to *HL7 Table 0354 - Message Structure* for valid values (this table may extended by locally defined message structures)
9	Specification Version	Version number of the application, interface, or specification
10	Specification Status	Status of the application, interface, or specification
11	Role	Sender or Receiver

An example of static definition publish/subscribe topics:

```
confsig-MyOrganization-2.4-static-ADT-A04--ADT_A01-v2-draft-Sender
```

2.B.5 Table definition

Table definitions will include statements of table conformance and, if available, the actual table elements supported.

2.B.5.1 Statements of table conformance

Statements of table conformance will consist of the definition of the table and its constituent elements. To the maximum extent practical it should be possible to objectively validate the content of a given message instance against the table definition in the profile.

2.B.5.2 Table values

The table definition can specify tables supported and the usage of values in those tables. The source of the tables will be HL7, User, Local, External, or Imported. For each table, the identifier, description and code system will be supplied. The table identifier and version may also be supplied.

For each element identified in the table, the code, display name, source, and usage (Refer to *2.B.7.5*, "*Usage*") will be supplied. The source of the individual element will be HL7, User, Redefined, or SDO.

2.B.6 Profile type

There are three basic profile types used in documenting standard conformance:

a) HL7 Standard profile (represents a specific HL7 published standard, creation and publication limited to HL7 use),

b) constrainable profile (with “Optional” elements which must be further constrained in order to create implementation profiles), and

c) implementation profile (no “Optional” parts, fully implementable).

This model allows vendors or providers to publish generic profiles from which fully constrained implementation profiles can be created.

In comparison with the HL7 standard, separate constrainable and implementation profiles may exist for the receiving and the sending role.

Both constrainable profiles and implementation profiles focus primarily on the expectations of the sending application, with minimal constraints on the application behavior of the receiver.

Due to the HL7 principle of not specifying application behavior, this message profile section will not address use cases where explicit constraints on the expected behavior of the receiver application (e.g., whether the receiver must process information, ignore it or generate an error) are required.

2.B.6.1 Vendor constrainable profiles

A vendor might develop a message profile to which all their software products must comply but, in itself, is not an implementation profile. The different products serve potentially different domains and might be implemented with products from other vendors. The vendor profile constrains the HL7 Standard by defining agreed-to vocabularies, conditionality rules, supported items, and local extensions that are shared across all products. The profile is not necessarily fully constrained. For example, the vendor profile might allow the usage code of optional as, across different products, an element may be required in some use cases, be optional or conditional in others, and not be supported at all in still others. Furthermore, at this level, providing exact length definition is optional as well.

The vendor's individual software products might themselves have profiles that would build on, and further constrain, their vendor profile. The product profile would specifically define the information model and the elements contained within. The product profile might still be a constrainable profile as elements might result in different HL7 messages based on configuration settings and customizations. Only once all configuration settings and customizations have been taken into account can you have a fully-constrained 'Implementation' profile.

Constrainable profiles can be useful for interface engine applications which must be flexible enough to allow for receipt of messages based on a variety of message profiles. The desire of the application would be to validate message instances against one constrainable profile.

2.B.6.2 Realm constrainable profiles

Realms, national and regional, profiles represent localization and restrictions placed on the appropriate standard, while providing enough optionality for basing the more specific implementation profiles. Some examples of realm constrainable profiles are:

a) AS4700.1-2001 Implementation of HL7 v2.3.1 Part 1:Patient Administration (constrainable profile for Australian Standards, constrains HL7 2.3.1, Chapter 3).

b) AS/NZS 4700.3-1999 Implementation of HL7 v2.3 Part 3: Electronic messages for exchange of information on Drug Prescription (constrainable profile for Australian Standards, constrains HL7 2.3, various Chapters).

2.B.6.3 Implementation profiles

Implementation profiles represent the lowest level of specification required for unambiguous implementation. Examples of some implementation profiles are:

a) Adverse Drug Reaction Implementers Specification, 2001, TGA (implementation profile, constrains Australian Standards and HL7 v2.3.1 constrainable profiles for Therapeutic Goods Administration ADRAC Messaging Implementation Project),

b) Diabetes Reporting Implementers Specification, 2001, UNSW (implementation profile, constrains Australian Standards and HL7 v2.3.1 constrainable profiles for University of NSW Diabetes Messaging Implementation Project),

c) Specific version of a product, as implemented, at a specific provider.

Within an implementation profile the exact length definition must be provided.

2.B.7 Static definition concepts

This section discusses concepts common to each level of the static definition (message, segment and field). It uses the generic term 'element' to refer to segment groups, segments, fields, components and sub-components.

2.B.7.1 Length

An unambiguous definition of length requires a clear understanding of what it applies to and how it is measured. Length is defined to be a constraint on the number of characters that may be present in one occurrence of a message field or element. This definition satisfies both requirements; it applies (strictly) to an element's data value—i.e., the set of characters present in the message representing a value of the element's predefined data type—and it is measured in characters as defined by the rules in chapter 2. The definition is system independent. For example, a system that encodes characters using one byte and a system that uses two-byte encoding would use the same value for length to impose the same length constraint. Length does not count the HL7 characters used to represent the value, only the number of characters in the value itself are counted. If the null character is represented by transmitting "", length conforms to any minimum and maximum length specification.

Length shall be interpreted as a restriction on an element's data value, not on the presence or absence of the element. A length value of zero is appropriate when the null character is transmitted, but this does not imply the element is not present; clearly two characters will be present if null is transmitted. Restricting the applicability of length to data values present in the message is necessary. Not only does it keep the concept simple and eliminate the need to address special cases, it also allows for the transmission of null values for required elements. A required element can have a null value, since this still clearly means there is a data value encoded for the element in the message. If an element is empty or not present in the message, i.e., there is no data value encoded for the element in the message, then length restrictions do not apply since there is nothing to restrict and no length constraint that can be violated.

Length shall be specified using the following syntax: "m..n", where m and n are non-negative integers designating the minimum and maximum number of characters the element may have, respectively, where n ≥ m. When an upper bound for length cannot be determined in advance, the use of the asterisk character, "*", may be used as a place holder for the maximum value, so that, in addition, to the above syntax, where m and n are integer values, a constraint of the form "m..*" may be used to indicate the maximum length constraint is unknown.

Example length constraints are shown in the *Minimum and Maximum Length Examples* table.

Minimum and Maximum Length Examples

Value	Description	Comment
	For constrainable profile: no length defined, i.e. no requirements on the length are given. Leaving this information empty is not allowed for implementable profiles.	
0..0	For withdrawn elements: minimum and maximum set to 0.	
1..1	Element must have exactly one character	
1..n	Element may have up to n characters	
n..n	Element must have exactly "n" characters	
1..*	Element may have any length.	
n..*	Element may have any length which is greater than or equal to "n", where "n" is greater than or equal to 1.	
m..n	Element must have a minimum length of "m" and a maximum length of "n" where "m" is less than or equal to "n" and "m" is greater than or equal to 1.	

Note: Whether or not an element is populated is controlled by cardinality. But if the element is populated with a non null value, the minimum and maximum length definition must hold. The null information representation (two double quotes) is not considered to be a value with applicable length information.

Length should not be specified for composite elements. In these cases, the actual min and max lengths can be very difficult to determine due to the interdependencies on the component content, and the specification of actual lengths is not useful either. For example, if an overall length of 4..20 is assigned to a data element with a type CWE, what does this mean in practice? However if a length is specified, the vertical bar representation of the data must conform to the stated length, allowing for an additional character for each HL7 separator character.

2.B.7.2 Conformance Length

Constrainable specifications may also specify a conformance length. This is the number of characters that any conformant application must be able to correctly handle.For example, a constrainable profile may declare that the min and max lengths of a specific field are 3 and 2500. An implementation profile may further constrain this length to specify what is actually supported by an application. However an application could declare a length of 3..4, which may not be useful within the context of the constrainable profile. A constrainable profile may specify conformance lengths to establish a minimum expectation. In the example case, if the constrainable profile specifies a conformance length of 200, no other profile may assert conformance to the constrainable profile unless its maximum length is 200 or greater.

Conformance length is a redundant concept in implementation profiles that will not be further constrained, and should not be specified.

2.B.7.3 Truncation Flag

As of 2.7, a truncation pattern is defined which can be used to assist applications to manage the existence of data that exceeds the maximum number of characters that can be properly handled. The truncation pattern is described in Chapter 2, section 2.5.5.2, "Truncation Pattern". Note that the actual truncation behavior of the pattern is data type dependent. Applications shall not use truncation if the profile prohibits it. Applications may support truncation if the profile permits it. Message Profiles may specify the truncation behaviour.

The truncation flag is a simple boolean. In a constrainable profile, the value may be true or false. False signifies that the element may not be truncated, while true means that the value may be truncated. If a profile fixes truncation to false, no other further constraining profile may mark this value as true. If the value is fixed to true, other further constraining profiles may mark it as true or false.

In an implementation profile, a value of true for the truncation flag signifies that the application supports the defined truncation behaviour for the appropriate type. A value of false indicates that the application does not support data truncation for this element.

Although the truncation pattern was only defined in v2.7, the bahaviour may be adopted for previous versions of HL7, and the truncation flag may be used with previous version. Note that in these cases, the truncation character cannot be specified in the message, and some other arrangement must exist.

2.B.7.4 Cardinality

In order to separate message content requirements from application behavior requirements, cardinality is used to control message content, and usage is used to define application requirements. Cardinality controls the number of occurrences of an element appearing in a message. Some elements shall always be present (e.g., cardinality [1..1], [1..n]). Others shall never be present (i.e., cardinality [0..0]). Others may be optional with zero or more occurrences (e.g., cardinality [0..n]). Cardinality identifies the minimum and maximum number of occurrences that a message element must have in a conformant message. Cardinalities are expressed as a minimum-maximum pair of non-negative integers. A conformant message must always contain at least the minimum number of occurrences, and shall not contain more than the maximum number of occurrences.

An explicit cardinality range is required for segment group, segment, and field elements. Component and sub-component elements do not explicitly include a cardinality range, but a cardinality range is implicitly

associated with each component and sub-component element. The associated cardinality depends on the element's *usage* code. For components and sub-components with a usage code of *R,* the associated cardinality range is [1..1]; for all elements with a usage code of *RE, C, CE,* or *O,* the associated cardinality is [0..1]; and for all elements with an *X* usage code, the associated cardinality is [0..0].

There are two special values for cardinality. If the minimum number of occurrences is 0, the element may be omitted from a message. In certain circumstances, the maximum number of occurrences may have no specified limit. In this case, it is identified with "*" (e.g., [n..*]).

Valid cardinality values are shown in the *Cardinality* table; combinations not designated in the table are invalid. In particular usage code RE is not allowed with cardinalities [1..1], [1..n], and [1..*]. Cardinality [m..n], where *m* is greater than 1, is allowed with usage codes R and RE. If an element with this cardinality range has a usage code RE, the element may be omitted from the message but if present, it must have at least *m* occurrences and may not have more than *n* occurrences.

Cardinality

Value	Description	Valid Usage Codes	Comment
[0..0]	Element never present	X	
[0..1]	Element may be omitted and it can have at most one occurrence	RE, O, C[2], CE	
[1..1]	Element must have exactly one occurrence	R	
[0..n]	Element may be omitted or may have up to n occurrences	RE, O, C[2], CE	
[1..n]	Element must appear at least once, and may have up to n occurrences	R	
[0..*]	Element may be omitted or may have an unlimited number of occurrences	RE, O, C[2], CE	
[1..*]	Element must appear at least once, and may have an unlimited number of occurrences	R	
[m..n] [3]	Element must have at least "m" occurrences and may have at most "n" occurrences. Except that in the case where the usage code is RE, the element may also be omitted or have zero occurrences	R and RE	
[m..*][3]	Element must have at least "m" occurrences and may have an unlimited number of occurrences. Except that in the case where the usage code is RE, the element may also be omitted or have zero occurrences.	R and RE	

2.B.7.5 Usage

Message content is governed by the cardinality specification associated (explicitly or implicitly) with each element of an HL7 message. Usage rules govern the expected behavior of the sending application and place limited restrictions on the receiving application with respect to the element. These usage codes expand/clarify the optionality codes defined in the HL7 standard.

The standard allows broad flexibility for the message structures that HL7 applications must be able to receive without failing. But while the standard allows that messages may be missing fields or may contain extra fields, it should not be inferred from this requirement that such messages are conformant.

[2] If the usage code is C, then the element must be present if the associated condition predicate evaluates to true.

[3] *m* must be greater than 1 and *n* must be greater than or equal to m; the case where *m* equals 1 is addressed separately.

Usage

Value	Description	Comment
R	Required	A conforming sending application shall populate all "R" elements with a non-empty value. A conforming receiving application should process (save/print/archive/etc.) or ignore the information conveyed by required elements. A conforming receiving application shall not raise an error due to the presence of a required element, but may raise an error due to the absence of a required element. Any element designated as required in a standard HL7 message definition shall also be required in all HL7 message profiles of that standard message.
RE	Required but may be empty	The element may be missing from the message, but should be sent by the sending application if there is relevant data. A conforming sending application should be **capable** of populating all "RE" elements. A conforming sending application that knows the required values for the element, should send that element. A conforming sending application that does not know the required values, shall omit the element. Receiving applications will be expected to process (save/print/archive/etc.) or ignore data contained in the element, but should be able to successfully process the message if the element is omitted (no error message should be generated because the element is missing).
O	Optional	This code indicates that the Usage for this element has not yet been defined. A usage of 'Optional' may **not** be used in 'implementation' profiles (no-optionality profiles). Conformance may not be tested on an Optional field. Narrower profiles may be defined based on this profile, and may assign any usage code to the element
C	Conditional	This usage has an associated condition predicate (See section *2.B.7.9*, "*Condition predicate*"). **If the predicate is satisfied**, follow the rules for R: **If the predicate is NOT satisfied**, follow the rules for X.
CE	Conditional but it may be empty	This usage has an associated condition predicate (See section *2.B.7.9*, "*Condition predicate*"). **If the predicate is satisfied**, follow the rules for the RE usage. **If the predicate is not satisfied**, follow the rules for X.
X	Not supported	For conformant sending applications, the element shall not be sent. Conformant receiving applications may ignore the element if it is sent, or may raise an application error.

2.B.7.6 Relationship between HL7 optionality and conformance usage

Conformance usage codes are more specific than HL7 Optionality codes. Because of the requirement that conformance statements must be compliant with the HL7 message definition it is derived from, there are restrictions on what usage codes may be assigned to an element based on the HL7 Optionality.

HL7 Optionality and Conformance Usage

HL7 Optionality	Allowed Conformance Usage	Comment
R - Required	R	
RE - Required but may be empty[4]	RE, R	
O - Optional	R, RE, O, C, CE, X	O is only permitted for constrainable profiles
C - Conditional	C, CE, R	
X – Not Supported	X	
B – Backward Compatibility	R, RE, O, C, CE, X	B is only permitted for constrainable definitions
W - Withdrawn	X	

[4] This is a optionality added with 2.7. The meaning is slightly different from the message profile usage code of RE.

2.B.7.7 Relationship between usage and cardinality

Cardinality governs the appearance of a field, and usage governs the expected behavior of applications. Nevertheless, a relationship exists between them that must be maintained. The valid combinations of the two are defined in the *Cardinality* table. For purposes of message profiles, selected cardinality and usage combinations are examined here. The constraints on allowed combinations are:

a) If the usage of an element is Required (R), the minimum cardinality for the element shall always be greater than or equal to 1.

b) If the usage of an element is not Required (R) (i.e., any code other than 'R'), the minimum cardinality shall be 0 except in the following condition:

c) If the profile author wishes to express a circumstance where an element will not always be present, but when present must have a minimum number of repetitions greater than one, this may be indicated by specifying the non-required Usage code with the minimum cardinality representing the minimum number of repetitions when the element is present. This is accomplished, as in UML, with an expression of the form as (m..n), indicating that permitted occurrences are either zero, or the range of n through m.

Example combinations:

Example Usage-Cardinality Combinations

Cardinality	Usage	Interpretation
[1..1]	R	There will always be exactly 1 occurrence present.
[1..5]	R	There will be between 1 and 5 occurrences present.
[0..1]	RE	The element must be supported, but may not always be present.
[0..5]	C	If the condition predicate is true, there will be between 1 and 5 occurrences. If the predicate is false, there will be 0 occurrences.
[3..5]	RE	If any values for the element are sent, there must be at least 3 and no more than 5 occurrences. However, the element may be absent (0 occurrences).

2.B.7.8 Usage within hierarchical elements

As part of the conformance framework, there is an additional rule for determining whether a particular 'element' is present. The rule is as follows: For an element to be considered present, it must have content. This means that simple elements (fields, components or sub-components with simple data types such as NM, ST, ID) must have at least one character. Complex elements (those composed of other elements, e.g., Messages, Segment Groups, Segments, Fields with complex data types such as CNE, XPN, etc.), must contain at least one component that is present. Elements that do not meet these conditions are not considered to be present.

For example, if a segment is made up of 10 optional fields, at least one of the fields must be present in order for the segment to be considered present. Thus, if the segment is marked as Required, an instance message would only be conformant if the segment contained at least one field. The reason for this rule is to ensure that the intent of the profile is met. The rule is necessary because the traditional 'vertical bar' encoding allows for a bare segment identifier with no fields (e.g., a line containing just "NTE|" would be considered valid under the standard rules, but would be considered not present as far as testing against a conformance specification). The XML encoding also allows this, as well as fields without their components, components without their sub-components, etc. (e.g., <PID.3/>).

2.B.7.9 Condition predicate

If the usage code of an element is C or CE, then a conditionality predicate must be associated with this element that identifies the conditions under which the element must be or is allowed to be present. The predicate must be testable and based on other values within the message. This predicate may be expressed

as a mathematical expression or in text and may utilize operators such as equivalence, logical AND, logical OR and NOT. The conforming sending and receiving applications shall both evaluate the predicate. When the Usage is not 'C' or 'CE', the conditionality predicate will not be valued.

2.B.7.10 Annotation

Annotations provide further explanations to educate prospective users and/or implementers. These are usually used to enhance the descriptions of the elements of the base specification in order to relate them to a particular Context.

Types of annotations supported:

- Definition: An explanation of the meaning of the element.
- Description: An explanation of the associated element. This may contain formatting markup.
- Design Comment: Internal development note about why particular design decisions were made, outstanding issues and remaining work. They may contain formatting markup. Not intended for external publication.
- Implementation Note: Implementation Notes provide a general description about how the element is intended to be used, as well as hints on using or interpreting the it.
- Other Annotation: Additional content related to the element.
- Example: An example instance
- Added ability to communicate pattern matching and element relationships. These, as well as condition predicate, will allow for text and formal testable constraints.

2.B.8 Static definition - message level

The message level static definition shall be documented using the HL7 **abstract message** syntax, with the addition of specifying cardinality and usage for each of the segments contained within the message structure.

The **usage** column shall be updated to reflect the usage of the segment or group within this particular static definition.

The **cardinality** column shall accurately reflect the minimum and maximum number of repetitions of the field allowed for the segment or group within this particular static definition.

Sample Static Definition - Message Level

ADT^A01^ADT_A01	ADT Message	Status	Usage	Cardinality	Chapter
MSH	Message Header		R	[1..1]	2
[{ SFT }]	Software Segment		X	[0..0]	2
[UAC]	User Authentication Credential		X	[0..0]	2
EVN	Event Type		R	[1..1]	3
PID	Patient Identification		R	[1..1]	3
[PD1]	Additional Demographics		X	[0..0]	3
[ARV]	Access Restrictions		X	[0..0]	3
[{ ROL }]	Role		X	[0..0]	15
[{ NK1 }]	Next of Kin / Associated Parties		RE	[0..3]	3
PV1	Patient Visit		C	[0..1]	3
[ARV]	Access Restrictions		X	[0..0]	3
[PV2]	Patient Visit - Additional Info.		RE	[0..1]	3

ADT^A01^ADT_A01	ADT Message	Status	Usage	Cardinality	Chapter
[{ ROL }]	Role		X	[0..0]	15
[{ DB1 }]	Disability Information		X	[0..0]	3
[{ OBX }]	Observation/Result		X	[0..0]	7
[{ AL1 }]	Allergy Information		RE	[0..10]	3
[{ DG1 }]	Diagnosis Information		X	[0..0]	6
[DRG]	Diagnosis Related Group		X	[0..0]	6
[{	--- PROCEDURE begin		X	[0..0]	
PR1	Procedures		X	[0..0]	6
[{ ROL }]	Role		X	[0..0]	15
}]	--- PROCEDURE end				
[{ GT1 }]	Guarantor		X	[0..0]	6
[{	--- INSURANCE begin		X	[0..0]	
IN1	Insurance		X	[0..0]	6
[IN2]	Insurance Additional Info.		X	[0..0]	6
[{ IN3 }]	Insurance Additional Info - Cert.		X	[0..0]	6
[{ ROL }]	Role		X	[0..0]	15
}]	--- INSURANCE end				
[ACC]	Accident Information		X	[0..0]	6
[UB1]	Universal Bill Information		X	[0..0]	6
[UB2]	Universal Bill 92 Information		X	[0..0]	6
[PDA]	Patient Death and Autopsy		X	[0..0]	3

2.B.8.1 Segment definitions

The set of segments and segment groups included within the message shall be defined. Any segments or segment groups that are required by HL7 shall be included.

2.B.8.2 Segment usage

The usage of the segment or group within a message shall be defined using one of the codes in the previously defined usage table.

2.B.8.3 Segment cardinality

Some segments and segment groups within the HL7 message are allowed to repeat. The cardinality of all the segments and groups within the message shall be defined.

Static definition - segment level

The segment level static definition shall be documented using the HL7 **segment attribute table** format with the addition of specifying length, usage and cardinality for each of the fields contained within the segment.

- The **length column** shall be updated to accurately reflect the maximum allowed length for the field within this segment definition.
- The **usage column** shall accurately reflect the usage of the field within this segment definition.
- The **cardinality column** shall accurately reflect the minimum and maximum number of repetitions of the field allowed for this segment definition.

Sample Segment Level Definition – PID (Patient Identification) Segment

SEQ	LEN	C.LEN	DT	OPT	RP/#	TBL#	ITEM#	ELEMENT NAME
1	1..4		SI	O			00104	Set ID - PID
2				W			00105	Patient ID
3			CX	R	Y		00106	Patient Identifier List
4				W			00107	Alternate Patient ID - PID
5			XPN	R	Y	0200	00108	Patient Name
6			XPN	O	Y		00109	Mother's Maiden Name
7			DTM	O			00110	Date/Time of Birth
8			CWE	O		0001	00111	Administrative Sex
9				W			00112	Patient Alias
10			CWE	O	Y	0005	00113	Race
11			XAD	O	Y		00114	Patient Address
12				W			00115	County Code
13			XTN	B	Y		00116	Phone Number - Home
14			XTN	B	Y		00117	Phone Number - Business
15			CWE	O		0296	00118	Primary Language
16			CWE	O		0002	00119	Marital Status
17			CWE	O		0006	00120	Religion
18			CX	O			00121	Patient Account Number
19				W			00122	SSN Number - Patient
20				W			00123	Driver's License Number - Patient
21			CX	O	Y		00124	Mother's Identifier
22			CWE	O	Y	0189	00125	Ethnic Group
23		250#	ST	O			00126	Birth Place
24	1..1		ID	O		0136	00127	Multiple Birth Indicator
25		2=	NM	O			00128	Birth Order
26			CWE	O	Y	0171	00129	Citizenship
27			CWE	O		0172	00130	Veterans Military Status
28				W			00739	Nationality
29			DTM	O			00740	Patient Death Date and Time
30	1..1		ID	O		0136	00741	Patient Death Indicator
31	1..1		ID	O		0136	01535	Identity Unknown Indicator
32			CWE	O	Y	0445	01536	Identity Reliability Code
33			DTM	O			01537	Last Update Date/Time
34			HD	O			01538	Last Update Facility
35			CWE	C		0446	01539	Species Code
36			CWE	C		0447	01540	Breed Code
37		80=	ST	O			01541	Strain
38			CWE	O	2	0429	01542	Production Class Code
39			CWE	O	Y	0171	01840	Tribal Citizenship

SEQ	LEN	C.LEN	DT	OPT	RP/#	TBL#	ITEM#	ELEMENT NAME
40			XTN	O	Y		02289	Patient Telecommunication Information

2.B.8.5 Field definitions

The set of fields of each segment within the message level definition shall be specified.

If a segment occurs multiple times within a message profile, it may be represented by different segment profiles. This shall be explicitly defined within the message level definition.

2.B.8.6 Field cardinality

Some fields within a segment are allowed to repeat. The cardinality of all the fields within the segment shall be defined.

2.B.8.7 Field usage

The **usage** of the field within a segment shall be defined consistent with the profile type, and using one of codes identified in the previously defined Usage tables.

2.B.8.8 Data type

The data type of the field within a segment shall be updated to accurately reflect the data type for the field within this segment definition.

2.B.8.9 Length

The length of the field within a segment shall be updated to accurately reflect the correct length for the field within this segment definition. Here two information items shall be provided representing the required minimum length a data constituent must have and the maximum length this constituent must not exceed.

2.B.8.10 Conformance Length

This length represents the number of characters that a conformant application must be able to handle. For further information please see Chapter 2, section 2.5.5.3, "Conformance Length".

2.B.8.11 Table reference

The name of the table of the field within a segment shall be updated to accurately reflect the table used for the field within this segment definition.

2.B.9 Static definition - field level

2.B.9.1 Field Definitions

Each individual field within a segment shall be completely defined to eliminate any possible ambiguity.

In cases where HL7 2.x field descriptions are not sufficient, a precise semantic definition shall be specified.

2.B.9.2 User-defined and suggested field values

The allowed code sets (table) for many fields within the HL7 Standard are specified as user-defined (data type IS) or HL7-defined (data type ID) values.

In these cases, the **exact allowed code set** shall be specified. These values shall be defined according to the specified scope of use for the message profile by vendors, provider, or within a realm.

Coded Entry (CE, CF, CWE, and CNE) type fields are specified as being populated based on coding systems. For each of these fields, the specific coding system used shall be identified. Compliant applications are required to use the specified coding system, but may also use an alternate coding system as supported by the data type (See the example within each data type definition).

2.B.9.3 Constant values

If an element will always have a constant value, this shall be specified. Constant values may only be specified for elements that represent primitive data types, i.e., they have no components or sub-components.

2.B.9.4 Data values

A list of example data values for the element may be specified. Data values may only be specified for elements that represent primitive data types i.e. they no components or sub-components.

2.B.9.5 Pattern Matching

Constraints for matching patterns within fields may be specified. In addition to textual description of the constraint, formal expressions may be specified. These formal expressions can be Object Constraint Language (OCL), regular expressions (RegEx)[5], and XML Path Language (XPath)[6]

2.B.9.6 Element Relationships

Element relationships may be may be specified. In addition to textual description of these constraints, formal expressions may be specified. These formal expressions can be Object Constraint Language (OCL), regular expressions (RegEx)[7], and XML Path Language (XPath)[8]

2.B.9.7 Components and subcomponents

Many fields and components in versions of HL7 prior to 2.5 were defined to be **Composite Data (CM)** data types. As of 2.5, all field instances will reference a valid data type other than CM. Addenda for versions 2.3.1 and 2.4 are available that define more precise names for the CM data types. These names allow a more precise data type name for each of the fields using the former CM data type to be more easily used for XML encoding of message instances. Although message profiling is not limited to a specific version of HL7, it is **strongly** encouraged that these new data types be used to increase interoperability between versions.

Each component within composite fields shall be profiled. This requires defining the usage, length, data type, and data values of each of the components. Where there are sub-components of a component, each of the sub-components shall also be profiled using the same method. With the exception of cardinality, the rules for these definitions follow those for fields (See section *2.B.9*, "*Static definition - field level*").

2.B.10 Message profile document

HL7 Headquarters will provide a utility, hereafter called registry, on the Members' Only Web site *(http://www.hl7.org)* where the message profile can be registered.

Messages profiles in the registry are all catalogued with a set of metadata. Those entities submitting message profiles into the registry will need to fill out a form that captures the required metadata information. The registry and the metadata will be documented in an informative document and will not be discussed further in this document.[9]

---

[5] Refer to Part 2 of W3C schema (http://www.w3.org/TR/xmlschema-2/) for details on regular expressions.

[6] Refer to XML Path Language (http://www.w3.org/TR/xpath)

[7] Refer to Part 2 of W3C schema (http://www.w3.org/TR/xmlschema-2/) for details on regular expressions.

[8] Refer to XML Path Language (http://www.w3.org/TR/xpath)

[9] The message profile metadata has changed between V2.5 and V2.6. An attribute to reflect the version of the metadata will be included in the metadata. The registries and tools will need to allow for compatibility between message profile versions. At a minimum, a transform will be available.

2.B.10.1 Message profile document format

The Implementation/Conformance Work Group researched the best approach to standardize the format of a message profile to facilitate comparison and measurement. XML (eXtensible Markup Language XML W3C XML 1.0 2nd Ed[10]) documents appeared to be the best tool for this.

This use of XML is not, in any way, related to the HL7 2.xml encoding specification that describes the XML encoding of message instances. The message profile document format provides structure to the documentation of the message profile and does not limit the encoding of an actual message instance.

2.B.10.2 Message profile document definition

A message profile document will be a valid HL7 message profiles if it conforms to the constraints expressed in the message profile document definition (See section *2.B.12*, "*Message profile document definition*"), and the additional rules described in this document.

2.B.11 Tools

The tools used for creation, sharing, re-use, reporting, analyzing, and comparing message profiles are outside the scope of the HL7 standard. Refer to the Implementation/Conformance Work Group web site for useful links that are of widespread interest to, and in support of, message profiles and the Implementation/Conformance Work Group.[11]

2.B.12 Message profile document definition

The Implementation/Conformance WG researched the various ways to express the structure of the message profile document. The Document Type Definition (DTD) allows for declaring constraints on the use of markup. XML Schema Language provides a more rigorous and comprehensive framework for automated processing of XML documents.[12]

The message profile DTD and schema are both included here. The message profile schema is normative in order to express the rules by which the registry will validate (see section *2.B*, "*Conformance Using Message Profiles*").

2.B.12.1 Message profile schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Jennifer Puyenbroek (HL7
Implementation/Conformance Work Group) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="HL7v2xConformanceProfile">
    <xs:annotation>
      <xs:documentation>An unambiguous specification of one or more standard HL7 messages
that have been analyzed for a particular use case. It prescribes a set of precise
constraints upon one or more standard HL7 messages.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
```

---

10 Refer to the World Wide Web Consortium web site for this recommendation (http://www.w3.org/TR/REC-xml).

11 The Messaging Workbench (MWB) is the first freeware tool available for creating message profiles. This tool is neither developed nor supported by HL7 but its use is widespread among HL7 members. The MWB developer is an active member of the HL7 Implementation/Conformance WG and the tool has evolved with input from this WG. Refer to HL7 Implementation/Conformance WG Documents/Downloads.

The Australian Healthcare Messaging Laboratory (AHML) is a not-for-profit venture of the University of Ballarat providing a developer test-bed available to HL7 members as well as validation of the static profile. Contact AHML (http://www.ahml.com.au) for more information.

12 Refer to the World Wide Web Consortium web site for their recommendations for DTD (http://www.w3.org/TR/REC-xml) and schema (http://www.w3.org/TR/xmlschema-0, http://www.w3.org/TR/xmlschema-1, and http://www.w3.org/TR/xmlschema-2).

```
      <xs:element name="MetaData" type="MetaDataType">
       <xs:annotation>
        <xs:documentation>Provides descriptive information about the life-cycle of the
HL7v2xConformanceProfile, as well as authorship and control
information.</xs:documentation>
       </xs:annotation>
      </xs:element>
      <xs:element name="Annotation" type="AnnotationType" minOccurs="0"
maxOccurs="unbounded">
       <xs:annotation>
        <xs:documentation>Annotations provide a general description about how the profile
is intended to be used, as well as hints on using or interpreting the
profile.</xs:documentation>
       </xs:annotation>
      </xs:element>
      <xs:element name="UseCase">
       <xs:annotation>
        <xs:documentation>A use case model documents the scope and requirements for an
HL7 message profile or set of message profiles.</xs:documentation>
       </xs:annotation>
       <xs:complexType>
        <xs:sequence>
         <xs:element name="Purpose" type="NonEmptyStringType" minOccurs="0">
          <xs:annotation>
           <xs:documentation>Identifies the reason and/or objectives for the
usecase</xs:documentation>
          </xs:annotation>
         </xs:element>
         <xs:element name="Description" type="NonEmptyStringType" minOccurs="0">
          <xs:annotation>
           <xs:documentation>Descriptive text for the use-case. In cases where the use-
case is not broken down into component elements, this will include the complete details of
the usecase. Otherwise, it will contain a basic overview.</xs:documentation>
          </xs:annotation>
         </xs:element>
         <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="Actor" type="UseCaseElementType">
           <xs:annotation>
            <xs:documentation>Identifies and defines the entities involved in the use-
case. This includes the sending and receiving applications</xs:documentation>
           </xs:annotation>
          </xs:element>
          <xs:element name="PreCondition" type="UseCaseElementType">
           <xs:annotation>
            <xs:documentation>Identifies a circumstance that must hold true prior to
the use-case being invoked.</xs:documentation>
           </xs:annotation>
          </xs:element>
          <xs:element name="PostCondition" type="UseCaseElementType">
           <xs:annotation>
            <xs:documentation>Identifies a circumstance that will hold true after the
successful completion of the use-case.</xs:documentation>
           </xs:annotation>
          </xs:element>
          <xs:element name="EventFlow" type="UseCaseElementType">
           <xs:annotation>
            <xs:documentation>Identifies a step within the chain of occurrences that
lead to the successful completion of the use-case. This includes the exchange of messages
between applications.</xs:documentation>
           </xs:annotation>
          </xs:element>
          <xs:element name="DerivedEvent" type="UseCaseElementType">
           <xs:annotation>
            <xs:documentation/>
           </xs:annotation>
          </xs:element>
         </xs:choice>
        </xs:sequence>
       </xs:complexType>
       <xs:key name="ActorNamesUniqueInUseCase">
```

```
          <xs:selector xpath="Actor"/>
          <xs:field xpath="@Name"/>
        </xs:key>
        <xs:key name="PreConditionNamesUniqueInUseCase">
          <xs:selector xpath="PreCondition"/>
          <xs:field xpath="@Name"/>
        </xs:key>
        <xs:key name="PostConditionNamesUniqueInUseCase">
          <xs:selector xpath="PostCondition"/>
          <xs:field xpath="@Name"/>
        </xs:key>
        <xs:key name="EventFlowNamesUniqueInUseCase">
          <xs:selector xpath="EventFlow"/>
          <xs:field xpath="@Name"/>
        </xs:key>
        <xs:key name="DerivedEventNamesUniqueInUseCase">
          <xs:selector xpath="DerivedEvent"/>
          <xs:field xpath="@Name"/>
        </xs:key>
      </xs:element>
      <xs:element name="Encodings">
        <xs:annotation>
          <xs:documentation>Identifies all of the message encoding mechanisms supported by
the profile. Non-traditional encoding mechanisms may be identified if
desired.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Encoding" maxOccurs="unbounded">
              <xs:annotation>
                <xs:documentation>Identifies one of the encoding mechanisms supported by the
profile.</xs:documentation>
              </xs:annotation>
              <xs:simpleType>
                <xs:union memberTypes="xs:NMTOKEN">
                  <xs:simpleType>
                    <xs:restriction base="xs:NMTOKEN">
                      <xs:enumeration value="ER7"/>
                      <xs:enumeration value="XML"/>
                    </xs:restriction>
                  </xs:simpleType>
                </xs:union>
              </xs:simpleType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
        <xs:key name="EncodingUniqueInEncodings">
          <xs:selector xpath="Encoding"/>
          <xs:field xpath="."/>
        </xs:key>
      </xs:element>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="DynamicDef">
          <xs:annotation>
            <xs:documentation>The dynamic definition is an interaction specification for a
conversation between 2 or more systems.</xs:documentation>
          </xs:annotation>
          <xs:complexType>
            <xs:attribute name="AccAck" type="AcknowledgementType" default="NE">
              <xs:annotation>
                <xs:documentation>Identifies when and if HL7 'Accept' acknowledgements are
required. Allowed values are: AL (always), NE (never), SU (on success), ER (on error).
Default is 'NE'.</xs:documentation>
              </xs:annotation>
            </xs:attribute>
            <xs:attribute name="AppAck" type="AcknowledgementType" default="AL">
              <xs:annotation>
                <xs:documentation>Identifies when and if HL7 'Application' acknowledgements
are required. Allowed values are: AL (always), NE (never), SU (on success), ER (on error).
Default is 'AL'.</xs:documentation>
```

```
            </xs:annotation>
           </xs:attribute>
           <xs:attribute name="MsgAckMode" default="Deferred">
            <xs:annotation>
             <xs:documentation>Identifies the type of acknowledgement expected by the
sender of a message. Allowed values are: Immediate and Deferred. Default is
Immediate.</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
             <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="Immediate"/>
              <xs:enumeration value="Deferred"/>
             </xs:restriction>
            </xs:simpleType>
           </xs:attribute>
           <xs:attribute name="QueryMessageType" default="NonQuery">
            <xs:annotation>
             <xs:documentation>Identifies whether the message is query-related, and if
so, what type of query message it is. Allowed values are: NonQuery, Query, Response and
Publish. Default is NonQuery.
             </xs:documentation>
            </xs:annotation>
            <xs:simpleType>
             <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="NonQuery"/>
              <xs:enumeration value="Query"/>
              <xs:enumeration value="Response"/>
              <xs:enumeration value="Publish"/>
             </xs:restriction>
            </xs:simpleType>
           </xs:attribute>
           <xs:attribute name="QueryMode" default="RealTime">
            <xs:annotation>
             <xs:documentation>Identifies the type of query being performed. Allowed
values are: Batch, RealTime or Both.</xs:documentation>
            </xs:annotation>
            <xs:simpleType>
             <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="Batch"/>
              <xs:enumeration value="RealTime"/>
              <xs:enumeration value="Both"/>
             </xs:restriction>
            </xs:simpleType>
           </xs:attribute>
          </xs:complexType>
        </xs:element>
        <xs:choice maxOccurs="unbounded">
         <xs:element ref="HL7v2xStaticDef"/>
         <xs:element name="HL7v2xStaticDefRef">
          <xs:annotation>
           <xs:documentation>Provides an identifier reference to the static definition
for one of the messages used by the profile.</xs:documentation>
          </xs:annotation>
          <xs:complexType>
           <xs:attribute name="Identifier" type="IdentifierType" use="required">
            <xs:annotation>
             <xs:documentation>The identifier for the static definition being
referenced.</xs:documentation>
            </xs:annotation>
           </xs:attribute>
          </xs:complexType>
         </xs:element>
        </xs:choice>
       </xs:sequence>
       <xs:choice>
        <xs:element name="HL7v2xTables">
         <xs:complexType>
          <xs:sequence>
           <xs:element name="Metadata" type="MetaDataType" minOccurs="0"/>
           <xs:element ref="HL7v2xTable" minOccurs="0" maxOccurs="unbounded"/>
```

```
          </xs:sequence>
         </xs:complexType>
        </xs:element>
        <xs:element name="HL7v2xTablesRef">
         <xs:complexType>
          <xs:attribute name="Identifier" type="NonEmptyStringType"/>
         </xs:complexType>
        </xs:element>
       </xs:choice>
     </xs:sequence>
     <xs:attribute name="HL7Version" default="2.6">
       <xs:annotation>
        <xs:documentation>Identifies the HL7 2.x version on which the profile is based and
with which it is expected to comply.</xs:documentation>
       </xs:annotation>
       <xs:simpleType>
        <xs:union memberTypes="xs:NMTOKEN">
         <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
           <xs:enumeration value="2.0"/>
           <xs:enumeration value="2.0D"/>
           <xs:enumeration value="2.1"/>
           <xs:enumeration value="2.2"/>
           <xs:enumeration value="2.3"/>
           <xs:enumeration value="2.3.1"/>
           <xs:enumeration value="2.4"/>
           <xs:enumeration value="2.5"/>
           <xs:enumeration value="2.5.1"/>
           <xs:enumeration value="2.6"/>
           <xs:enumeration value="2.7"/>
          </xs:restriction>
         </xs:simpleType>
        </xs:union>
       </xs:simpleType>
     </xs:attribute>
     <xs:attribute name="ProfileType" use="required">
       <xs:annotation>
        <xs:documentation>Categorizes the profile into one of 3 types: HL7 - represents a
specific HL7 published standard (may only be submitted by the HL7 Organization);
Constrainable - May contain "Optional" elements which must be further constrained in order
to create implementation profiles; Implementation - Fully constrained with no optionality
(reflects the behavior of a runtime system)</xs:documentation>
       </xs:annotation>
       <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">

         <xs:enumeration value="Implementation"/>
         <xs:enumeration value="Constrainable"/>
        </xs:restriction>
       </xs:simpleType>
     </xs:attribute>
     <xs:attribute name="Identiifer" type="IdentifierType">
       <xs:annotation>
        <xs:documentation>A unique identifier for this specific version of this dynamic
profile. If not specified, one will be assigned to the profile upon submission to a
registry.</xs:documentation>
       </xs:annotation>
     </xs:attribute>
   </xs:complexType>
  </xs:element>
  <xs:element name="HL7v2xStaticDef">
    <xs:annotation>
     <xs:documentation>This represents a detailed profile of a single message. It provides
a detailed breakdown of exactly what the message may contain, including optionality and
cardinality.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
     <xs:sequence>
       <xs:element name="MetaData" type="MetaDataType" minOccurs="0">
        <xs:annotation>
```

```
        <xs:documentation>Provides descriptive information about the life-cycle of the
HL7 v2x Static Definition, as well as authorship and control
information.</xs:documentation>
       </xs:annotation>
      </xs:element>
      <xs:group ref="MessageGroup"/>
      <xs:element name="Segment" type="SegmentType">
       <xs:annotation>
        <xs:documentation>Documents the characteristics of a single HL7 segment within
the context of a particular message or segment group.</xs:documentation>
       </xs:annotation>
      </xs:element>
      <xs:group ref="SegGroupOrSegmentGrouping" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="MsgType" type="MsgTypeType" use="required">
     <xs:annotation>
      <xs:documentation>The HL7 message type code, as identified in MSH-9.1 (see HL7
Table 0076 - Message type).</xs:documentation>
     </xs:annotation>
    </xs:attribute>
    <xs:attribute name="EventType" type="EventTypeType" use="required">
     <xs:annotation>
      <xs:documentation>The HL7 event type code, as identified in MSH-9.2 (see HL7 Table
0003 - Event type)</xs:documentation>
     </xs:annotation>
    </xs:attribute>
    <xs:attribute name="MsgStructID" type="MsgStructIDType">
     <xs:annotation>
      <xs:documentation>The HL7 message structure code, as identified in MSH-9.3 (see
HL7 Table 0354 - Message Structure Type). </xs:documentation>
     </xs:annotation>
    </xs:attribute>
    <xs:attribute name="OrderControl" type="OrderControlType">
     <xs:annotation>
      <xs:documentation>The HL7 Order control code, as identified in ORC 1 (see HL7
Table 0119 - Order Control Codes).</xs:documentation>
     </xs:annotation>
    </xs:attribute>
    <xs:attribute name="EventDesc" type="NonEmptyStringType" use="required">
     <xs:annotation>
      <xs:documentation>A description of the event carried by this
message.</xs:documentation>
     </xs:annotation>
    </xs:attribute>
    <xs:attribute name="Identifier" type="IdentifierType" use="optional">
     <xs:annotation>
      <xs:documentation>A unique identifier for this specific version of this static
definition. If not specified, one will be assigned to the profile upon submission to a
registry.</xs:documentation>
     </xs:annotation>
    </xs:attribute>
    <xs:attribute name="Role" default="Sender">
     <xs:annotation>
      <xs:documentation>Identifies whether the profile is constructed from the
perspective of the message generator (Sender) or parser (Receiver). Default is
'Sender'.</xs:documentation>
     </xs:annotation>
     <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
       <xs:enumeration value="Sender"/>
       <xs:enumeration value="Receiver"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>
  <xs:complexType name="UseCaseElementType">
   <xs:simpleContent>
    <xs:extension base="NonEmptyStringType">
     <xs:attribute name="Name" type="NonEmptyStringType" use="required">
```

```
        <xs:annotation>
          <xs:documentation>The unique name or number associated with a particular use-case
element.</xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:extension>
   </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="MetaDataType">
    <xs:attribute name="Name" type="NonEmptyStringType" use="required">
      <xs:annotation>
        <xs:documentation>Provides a name that clearly and concisely defines the message
exchange being profiled.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="OrgName" type="NonEmptyStringType" use="required">
      <xs:annotation>
        <xs:documentation>Name of the organization that submitted the
profile.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="Version" type="NonEmptyStringType" use="optional">
      <xs:annotation>
        <xs:documentation>The version identifier assigned to this profile by the author.
There is no prescribed version numbering scheme.  However 'higher' versions should
generally be interpreted to be more resent.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="Status" type="NonEmptyStringType" use="optional">
      <xs:annotation>
        <xs:documentation>Status of this profile, as assigned by the author. There is no
prescribed status scheme at this time. Possible values might include: 'Draft', 'Active',
'Superceded', 'Withdrawn'</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="Topics" type="NonEmptyStringType" use="optional">
      <xs:annotation>
        <xs:documentation>This provides a list of key-words that relate to the profile and
that may be useful in profile searches.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="MetaVersion" default="2.6">
       <xs:annotation>
         <xs:documentation>Identifies the Message Profile version on which the profile is
based and with which it is expected to comply.
        </xs:documentation>
       </xs:annotation>
       <xs:simpleType>
        <xs:union memberTypes="xs:NMTOKEN">
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:enumeration value="2.5"/>
              <xs:enumeration value="2.6"/>
              <xs:enumeration value="2.7"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:union>
       </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="Context" type="NonEmptyStringType" use="optional">
      <xs:annotation>
        <xs:documentation>As defined, in the HL7 Policies and Procedures Manual, Affiliates
will have decision-making authority.  HL7 Affiliates control Realms.  Realms do not have
decision-making authority.  Realms simply represent a partition of the solution space.
Affiliates choose how the solution space is to be partitioned by authorizing the creation
of zero to many (0..*) Realms.</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
  <xs:group name="SegGroupOrSegmentGrouping">
```

```
   <xs:choice>
    <xs:element name="SegGroup">
     <xs:annotation>
      <xs:documentation>Documents the characteristics of a grouping of HL7 segments
within the context of a particular message or segment group.</xs:documentation>
     </xs:annotation>
     <xs:complexType>
      <xs:sequence>
       <xs:group ref="MessageElementsGroup"/>
       <xs:group ref="SegGroupOrSegmentGrouping" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="Name" use="required">
       <xs:annotation>
        <xs:documentation>This is the short, formal name for the group. It appears in
the tag name when using the XML Encoding syntax.</xs:documentation>
       </xs:annotation>
       <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
         <xs:pattern value="([A-Z]|_)+"/>
        </xs:restriction>
       </xs:simpleType>
      </xs:attribute>
      <xs:attribute ref="LongName" use="required"/>
      <xs:attribute ref="Usage" use="required"/>
      <xs:attributeGroup ref="RepeatableElementAttributes"/>
     </xs:complexType>
    </xs:element>
    <xs:element name="Segment" type="SegmentType">
     <xs:annotation>
      <xs:documentation>Documents the characteristics of a single HL7 segment within the
context of a particular message or segment group.</xs:documentation>
     </xs:annotation>
    </xs:element>
   </xs:choice>
  </xs:group>
  <xs:complexType name="SegmentType">
   <xs:sequence>
    <xs:group ref="MessageElementsGroup"/>
    <xs:element name="Field" maxOccurs="unbounded">
     <xs:annotation>
      <xs:documentation>Documents the characteristics of a single HL7 field within the
context of a particular message segment.</xs:documentation>
     </xs:annotation>
     <xs:complexType>
      <xs:sequence>
       <xs:group ref="LeafMessageElementsGroup"/>
       <xs:element name="Component" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
         <xs:documentation>Documents the characteristics of a single component within
the context of a field.</xs:documentation>
        </xs:annotation>
        <xs:complexType>
         <xs:sequence>
          <xs:group ref="LeafMessageElementsGroup"/>
          <xs:element name="SubComponent" minOccurs="0" maxOccurs="unbounded">
           <xs:annotation>
            <xs:documentation>Documents the characteristics of a single sub-component
within the context of a component.</xs:documentation>
           </xs:annotation>
           <xs:complexType>
            <xs:group ref="LeafMessageElementsGroup"/>
            <xs:attributeGroup ref="LeafElementAttributes"/>
           </xs:complexType>
          </xs:element>
         </xs:sequence>
         <xs:attributeGroup ref="LeafElementAttributes"/>
        </xs:complexType>
       </xs:element>
      </xs:sequence>
      <xs:attributeGroup ref="RepeatableElementAttributes"/>
```

```
        <xs:attributeGroup ref="LeafElementAttributes"/>
        <xs:attribute name="ItemNo">
          <xs:annotation>
            <xs:documentation>The HL7-assigned item number corresponding with the semantic
meaning of the field.</xs:documentation>
          </xs:annotation>
          <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
              <xs:pattern value="\d{5}"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:complexType>
    </xs:element>
   </xs:sequence>
   <xs:attribute name="Name" type="SegmentNameType" use="required">
    <xs:annotation>
      <xs:documentation>This is the short, formal name for the segment. It is used to
identify the segment in both ER7 and XML encodings.</xs:documentation>
    </xs:annotation>
   </xs:attribute>
   <xs:attribute ref="LongName"/>
   <xs:attribute ref="Usage" use="required"/>
   <xs:attributeGroup ref="RepeatableElementAttributes"/>
 </xs:complexType>
 <xs:complexType name="AnnotationType">
   <xs:simpleContent>
    <xs:extension base="NonEmptyStringType">
      <xs:attributeGroup ref="AnnotationAttributes"/>
    </xs:extension>
   </xs:simpleContent>
 </xs:complexType>
 <xs:complexType name="TextOrExpressionType">
   <xs:sequence>
       <xs:element name="Text" type="TextType"/>
       <xs:element name="FormalExpression" type="FormalExpressionType"/>
   </xs:sequence>
 </xs:complexType>
 <xs:complexType name="FormalExpressionType">
   <xs:simpleContent>
    <xs:extension base="NonEmptyStringType">
      <xs:attributeGroup ref="FormalExpressionAttributes"/>
    </xs:extension>
   </xs:simpleContent>
 </xs:complexType>
 <xs:complexType name="TableElementType">
   <xs:sequence>
    <xs:element name="Annotation" type="AnnotationType" minOccurs="0"
maxOccurs="unbounded"/>
   </xs:sequence>
   <xs:attribute name="Code" type="NonEmptyStringType" use="required"/>
   <xs:attribute name="DisplayName" type="NonEmptyStringType" use="required"/>
   <xs:attribute name="Source" default="User">
    <xs:simpleType>
      <xs:union memberTypes="xs:NMTOKEN">
       <xs:simpleType>
         <xs:restriction base="xs:NMTOKEN">
           <xs:enumeration value="HL7"/>
           <xs:enumeration value="User"/>
           <xs:enumeration value="Redefined"/>
           <xs:enumeration value="Context"/>
         </xs:restriction>
       </xs:simpleType>
      </xs:union>
    </xs:simpleType>
   </xs:attribute>
   <xs:attribute name="ElementUsage" default="O">
    <xs:simpleType>
      <xs:union memberTypes="xs:NMTOKEN">
       <xs:simpleType>
```

```
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="R"/>
            <xs:enumeration value="O"/>
            <xs:enumeration value="X"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
    </xs:simpleType>
  </xs:attribute>
 </xs:complexType>
 <xs:attributeGroup name="LeafElementAttributes">
  <xs:attribute name="Name" type="NonEmptyStringType" use="required">
    <xs:annotation>
      <xs:documentation>The descriptive name for the field/component/sub-
component</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute ref="Usage" use="required"/>
  <xs:attribute name="Datatype" type="DatatypeType" use="required">
    <xs:annotation>
      <xs:documentation>Identifies the HL7 datatype associated with the
element.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="MinLength" type="xs:positiveInteger">
    <xs:annotation>
      <xs:documentation>Identifies the minimum allowed length for the content of the
element.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="MaxLength" type="xs:positiveInteger">
    <xs:annotation>
      <xs:documentation>Identifies the maximum allowed length for the content of the
element.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="ConformanceLength" type="xs:positiveInteger">
    <xs:annotation>
      <xs:documentation>The minimum length an application must be able to
handle</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="Truncation" type="xs:boolean">
    <xs:annotation>
      <xs:documentation>whether the truncation pattern does/may apply</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="Table" type="TableType">
    <xs:annotation>
      <xs:documentation>Identifies the name of the table associated with the content of
this element.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="ConstantValue" type="NonEmptyStringType">
    <xs:annotation>
      <xs:documentation>Identifies the fixed value associated with this
element</xs:documentation>
    </xs:annotation>
    <!-- Can only have constant values for leaf elements -->
  </xs:attribute>
 </xs:attributeGroup>
 <xs:attributeGroup name="RepeatableElementAttributes">
  <xs:attribute name="Min" type="xs:nonNegativeInteger" use="required">
    <xs:annotation>
      <xs:documentation>This identifies the minimum number of repetitions of the element
that are permitted in a message instance. This attribute should only be specified if the
minimum number of repetitions is greater than 1, as the minimum for other elements is
always '0'.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
```

```
   <xs:attribute name="Max" use="required">
    <xs:annotation>
     <xs:documentation>This identifies the maximum number of repetitions of the element
that are permitted in a message instance. This attribute should only be specified if the
maximum number of repetitions is greater than 1 and differs from the minimum attribute
(i.e. the maximum number of repetitions is greater than the minimum number of
repetitions). The special value '*' may be used to represent 'unlimited'
repetitions.</xs:documentation>
    </xs:annotation>
    <xs:simpleType>
     <xs:union>
      <xs:simpleType>
       <xs:restriction base="xs:positiveInteger">
        <xs:minInclusive value="1"/>
       </xs:restriction>
      </xs:simpleType>
      <xs:simpleType>
       <xs:restriction base="xs:string">
        <xs:enumeration value="*"/>
       </xs:restriction>
      </xs:simpleType>
     </xs:union>
    </xs:simpleType>
   </xs:attribute>
 </xs:attributeGroup>
 <xs:simpleType name="MatchPatternType">
  <xs:restriction base="NonEmptyStringType"/>
 </xs:simpleType>
 <xs:simpleType name="ElementRelationshipType">
  <xs:restriction base="NonEmptyStringType"/>
 </xs:simpleType>
 <xs:simpleType name="TextType">
  <xs:restriction base="NonEmptyStringType"/>
 </xs:simpleType>
 <xs:attributeGroup name="TextAttributes">
      <xs:attribute name="Language">
       <xs:annotation>
        <xs:documentation>
        </xs:documentation>
       </xs:annotation>
       <xs:simpleType name="">
     <xs:restriction base="NonEmptyStringType"/>
    </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="LastTranslated ">
       <xs:annotation>
        <xs:documentation>
        </xs:documentation>
       </xs:annotation>
       <xs:simpleType name="">
     <xs:restriction base="NonEmptyStringType"/>
    </xs:simpleType>
      </xs:attribute>
 </xs:attributeGroup> <xs:attributeGroup name="FormalExpressionAttributes">
      <xs:attribute name="Type" use="required">
       <xs:annotation>
        <xs:documentation>
        </xs:documentation>
       </xs:annotation>
    <xs:simpleType>
       <xs:union memberTypes="xs:NMTOKEN">
        <xs:simpleType>
         <xs:restriction base="xs:NMTOKEN">
          <xs:enumeration value="Text"/>
          <xs:enumeration value="OCL"/>
          <xs:enumeration value="RegEx"/>
          <xs:enumeration value="XPath"/>
         </xs:restriction>
        </xs:simpleType>
       </xs:union>
```

```
    </xs:simpleType>
      </xs:attribute>
  </xs:attributeGroup>
  <xs:attributeGroup name="AnnotationAttributes">
      <xs:attribute name="Type" use="required">
       <xs:annotation>
         <xs:documentation>
         </xs:documentation>
       </xs:annotation>
    <xs:simpleType>
       <xs:union memberTypes="xs:NMTOKEN">
         <xs:simpleType>
          <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="ImpNote">
            <xs:annotation>
              <xs:documentation>Implementation Notes provide a general description about
how the element is intended to be used, as well as hints on using or interpreting the it.
              </xs:documentation>
            </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="Description">
              <xs:annotation>
                <xs:documentation> An explanation of the associated element.
                </xs:documentation>
              </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="Definition">
              <xs:annotation>
                <xs:documentation> An explanation of the meaning of the element.
                </xs:documentation>
              </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="DesignComment">
              <xs:annotation>
                <xs:documentation> Internal development notes about why particular design
decisions were made, outstanding issues and remaining work.  They may contain formatting
markup.  Not intended for external publication.
                </xs:documentation>
              </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="Other">
              <xs:annotation>
                <xs:documentation> Additional content related to the element.
                </xs:documentation>
              </xs:annotation>
            </xs:enumeration>
            <xs:enumeration value="Example">
              <xs:annotation>
                <xs:documentation> An example instance of the element.
                </xs:documentation>
              </xs:annotation>
            </xs:enumeration>
          </xs:restriction>
         </xs:simpleType>
       </xs:union>
    </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="OtherIdentifier" type="NonEmptyStringType">
       <xs:annotation>
         <xs:documentation>
         </xs:documentation>
       </xs:annotation>
      </xs:attribute>
  </xs:attributeGroup>
  <xs:group name="MessageGroup">
    <xs:sequence>
      <xs:group ref="Annotation" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="Description" type="NonEmptyStringType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>Provides an explanation or definition of what the element
```

```
represents.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="Reference" minOccurs="0">
      <xs:annotation>
       <xs:documentation>Identifies external sources or other locations within the
profile where additional information can be found about this item.</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
       <xs:restriction base="NonEmptyStringType"/>
      </xs:simpleType>
    </xs:element>
   </xs:sequence>
 </xs:group>
 <xs:group name="MessageElementsGroup">
   <xs:sequence>
    <xs:group ref="MessageGroup"/>
    <xs:element name="Predicate" minOccurs="0">
      <xs:annotation>
       <xs:documentation>Identifies the conditionality rule for this element, if
applicable</xs:documentation>
      </xs:annotation>
      <xs:simpleType>
       <xs:restriction base="NonEmptyStringType"/>
      </xs:simpleType>
    </xs:element>
   </xs:sequence>
 </xs:group>
 <xs:group name="LeafMessageElementsGroup">
   <xs:sequence>
    <xs:group ref="MessageElementsGroup"/>
    <xs:element name="DataValues" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
       <xs:attribute name="ExValue" type="NonEmptyStringType">
         <xs:annotation>
          <xs:documentation>Identifies an individual example value.</xs:documentation>
         </xs:annotation>
       </xs:attribute>
      </xs:complexType>
    </xs:element>
   </xs:sequence>
 </xs:group>
 <xs:attribute name="Usage">
   <xs:annotation>
    <xs:documentation>Usage identifies the circumstances under which an element appears
in a message. Possible values are:
      R - Required (must always be present);
      RE - Required or Empty (must be present if available);
      O - Optional (no guidance on when the element should appear);
      C - Conditional (the element is required or allowed to be present when the condition
specified in the Predicate element is true);
      CE - Conditional or Empty (the element is required or allowed to be present when the
condition specified in the Predicate element is true and the information is available)
      X - Not supported (the element will not be sent)
    </xs:documentation>
   </xs:annotation>
   <xs:simpleType>
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="R"/>
      <xs:enumeration value="RE"/>
      <xs:enumeration value="O"/>
      <xs:enumeration value="C"/>
      <xs:enumeration value="CE"/>
      <xs:enumeration value="X"/>
    </xs:restriction>
   </xs:simpleType>
 </xs:attribute>
 <xs:attribute name="LongName" type="NonEmptyStringType">
   <xs:annotation>
    <xs:documentation>This is the descriptive name for the element. It does not appear in
```

```
any encodings.</xs:documentation>
   </xs:annotation>
  </xs:attribute>
  <xs:simpleType name="NonEmptyStringType">
    <xs:restriction base="xs:string">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="AcknowledgementType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="AL"/>
      <xs:enumeration value="NE"/>
      <xs:enumeration value="SU"/>
      <xs:enumeration value="ER"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="IdentifierType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:pattern value="(0|[1-9][0-9]*)(\.(0|[1-9][0-9]*))*"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="MsgTypeType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:pattern value="[A-Z0-9]{3}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="EventTypeType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:pattern value="[A-Z0-9]{3}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="MsgStructIDType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:pattern value="[A-Z0-9]{3}(_[A-Z0-9]{3})?"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="OrderControlType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:pattern value="[A-Z]{2}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="DatatypeType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="SegmentNameType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:pattern value="[A-Z][A-Z0-9]{2}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="TableType">
    <xs:restriction base="xs:NMTOKEN">
      <xs:minLength value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

2.B.12.2 Message profile DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.4 U (http://www.xmlspy.com) by Jennifer Puyenbroek (HL7
Implementation/Conformance Work Group) -->
<!ENTITY % Usage "R | RE | O | C | CE | X">
<!ENTITY % HL7Version "2.0 | 2.0D | 2.1 | 2.2 | 2.3 | 2.3.1 | 2.4 | 2.5 | 2.5.1 | 2.6 | 2.7">
<!ENTITY % MetaVersion "2.5 | 2.6 | 2.7">
<!ENTITY % AcknowledgementType "AL | NE | SU | ER">
<!ENTITY % FormalExpressionType "Text | OCL | RegEx | XPath">
<!ENTITY % AnnotationType " ImpNote | Description | Definition | DesignComment | Other |
Example">
```

```
<!ELEMENT HL7v2xConformanceProfile (MetaData, Annotation*, UseCase, Encodings, (DynamicDef,
(HL7v2xStaticDef | HL7v2xStaticDefRef), (HL7v2xTables | HL7v2xTablesRef))+)>
<!ATTLIST HL7v2xConformanceProfile
  HL7Version (%HL7Version;) "2.7"
  ProfileType (Implementation | Constrainable) #REQUIRED
  Identiifer CDATA #IMPLIED
>
<!ELEMENT MetaData EMPTY>
<!ATTLIST MetaData
  Name CDATA #REQUIRED
  OrgName CDATA #REQUIRED
  Version CDATA #IMPLIED
  Status CDATA #IMPLIED
  Topics CDATA #IMPLIED
  MetaVersion (%MetaVersion;) "2.7"
  Context CDATA #IMPLIED
>
<!ELEMENT UseCase (Purpose?, Description?, (Actor | PreCondition | PostCondition | EventFlow |
DerivedEvent)*)>
<!ELEMENT Purpose (#PCDATA)>
<!ELEMENT Actor (#PCDATA)>
<!ATTLIST Actor
  Name CDATA #REQUIRED
>
<!ELEMENT PreCondition (#PCDATA)>
<!ATTLIST PreCondition
  Name CDATA #REQUIRED
>
<!ELEMENT PostCondition (#PCDATA)>
<!ATTLIST PostCondition
  Name CDATA #REQUIRED
>
<!ELEMENT EventFlow (#PCDATA)>
<!ATTLIST EventFlow
  Name CDATA #REQUIRED
>
<!ELEMENT DerivedEvent (#PCDATA)>
<!ATTLIST DerivedEvent
  Name CDATA #REQUIRED
>
<!ELEMENT DynamicDef EMPTY>
<!ATTLIST DynamicDef
  AccAck (%AcknowledgementType;) "NE"
  AppAck (%AcknowledgementType;) "AL"
  MsgAckMode (Immediate | Deferred) "Deferred"
  QueryMessageType (NonQuery | Query | Response | Publish) "NonQuery"
  QueryMode (Batch | RealTime | Both) "RealTime"
>
<!ELEMENT Encoding (#PCDATA)>
<!ELEMENT Encodings (Encoding+)>
<!ELEMENT HL7v2xStaticDefRef EMPTY>
<!ATTLIST HL7v2xStaticDefRef
  Identifier CDATA #REQUIRED
>
<!ELEMENT HL7v2xStaticDef (MetaData?, Annotation*, Reference?, Segment, (SegGroup | Segment)+)>
<!ATTLIST HL7v2xStaticDef
  MsgType CDATA #REQUIRED
  EventType CDATA #REQUIRED
  MsgStructID CDATA #REQUIRED
  OrderControl CDATA #IMPLIED
  EventDesc CDATA #REQUIRED
  Identifier CDATA #IMPLIED
  Role (Sender | Receiver) "Sender"
>
<!ELEMENT SegGroup (Annotation*, Reference?, Predicate?, (SegGroup | Segment)+)>
<!ATTLIST SegGroup
  Name CDATA #REQUIRED
  LongName CDATA #REQUIRED
  Usage (%Usage;) #REQUIRED
  Min CDATA #REQUIRED
```

```
  Max CDATA #REQUIRED
>
<!ELEMENT Segment (Annotation*, Reference?, Predicate?, Field+)>
<!ATTLIST Segment
  Name CDATA #REQUIRED
  LongName CDATA #IMPLIED
  Usage (%Usage;) #REQUIRED
  Min CDATA #REQUIRED
  Max CDATA #REQUIRED
>
<!ELEMENT Field (Annotation*, Reference?, Predicate?, MatchPattern?, ElementRelationship*,
DataValues*, Component*)>
<!ATTLIST Field
  Name CDATA #REQUIRED
  Usage (%Usage;) #REQUIRED
  Min CDATA #REQUIRED
  Max CDATA #REQUIRED
  Datatype CDATA #REQUIRED
  MinLength CDATA #IMPLIED
  MaxLength CDATA #IMPLIED
  ConformanceLength CDATA #IMPLIED
  Truncation CDATA #IMPLIED
  Table CDATA #IMPLIED
  ConstantValue CDATA #IMPLIED
  ItemNo CDATA #IMPLIED
>
<!ELEMENT Component (Annotation*, Reference?, Predicate?, MatchPattern?, ElementRelationship*,
DataValues*, SubComponent*)>
<!ATTLIST Component
  Name CDATA #REQUIRED
  Usage (%Usage;) #REQUIRED
  Datatype CDATA #REQUIRED
  MinLength CDATA #IMPLIED
  MaxLength CDATA #IMPLIED
  ConformanceLength CDATA #IMPLIED
  Truncation CDATA #IMPLIED
  Table CDATA #IMPLIED
  ConstantValue CDATA #IMPLIED
>
<!ELEMENT SubComponent (Annotations*, Reference?, Predicate?, MatchPattern?,
ElementRelationship*, DataValues*)>
<!ATTLIST SubComponent
  Name CDATA #REQUIRED
  Usage (%Usage;) #REQUIRED
  Datatype CDATA #REQUIRED
  MinLength CDATA #IMPLIED
  MaxLength CDATA #IMPLIED
  ConformanceLength CDATA #IMPLIED
  Truncation CDATA #IMPLIED
  Table CDATA #IMPLIED
  ConstantValue CDATA #IMPLIED
>
<!ELEMENT Annotation (#PCDATA)>
<!ATTLIST Annotation
  Type (%AnnotationType;) #REQUIRED
  OtherIdentifier CDATA #IMPLIED
>
<!ELEMENT Description (#PCDATA)>
<!ELEMENT Predicate (Text+, FormalExpression*)>
<!ELEMENT MatchPattern (Text+, FormalExpression*)>
<!ELEMENT ElementRelationship (Text+, FormalExpression*)>
<!ELEMENT Reference (#PCDATA)>
<!ELEMENT Text (#PCDATA)>
<!ATTLIST Text
  Language CDATA #IMPLIED
  LastTranslated CDATA #IMPLIED
>
<!ELEMENT FormalExpression (#PCDATA)>
<!ATTLIST FormalExpression
  Type (%FormalExpressionType;) #REQUIRED
```

```
>
<!ELEMENT DataValues EMPTY>
<!ATTLIST DataValues
  ExValue CDATA #IMPLIED
>
<!ELEMENT HL7v2xTables (MetaData?, HL7v2xTable*)>
<!ELEMENT HL7v2xTablesRef EMPTY>
<!ATTLIST HL7v2xTablesRef
  Identifier CDATA #REQUIRED
>
<!ELEMENT HL7v2xTable (Annotation*, HL7v2xTableElement*)>
<!ATTLIST HL7v2xTable
  CodeSystem CDATA #REQUIRED
  CodeSystemName CDATA #REQUIRED
  CodeSystemIdentifier CDATA #IMPLIED
  CodeSystemVersion CDATA #IMPLIED
  Type (HL7 | User | Local | External | Realm) "HL7"
>
<!ELEMENT HL7v2xTableElement (Annotation*)>
<!ATTLIST HL7v2xTableElement
  Code CDATA #REQUIRED
  DisplayName CDATA #REQUIRED
  Source (HL7 | User | Redefined | Realm) "User"
  Usage (R | O | X) "O"
>
```

2.B.13 Outstanding Issues

The following items are being discussed in the Implementation/Conformance Work Group for addition to future versions of HL7:

1) Conformance Based Queries use of the Message Profiles. Current status of this can be found on the Implementation/Conformance Work Group web page on the HL7 web site. (http://www.hl7.org).