Chapter 10
HL7 Dynamic Model and IHE XDS

This chapter is primarily devoted to the HL7 Dynamic Model, which describes a set of related interactions, which together perform one or more use cases. It also includes a short section on the IHE cross-enterprise document sharing (XDS) profile.

Safety is paramount in health care. Examples of safety procedures include:

- Acknowledgements sent at both transport level (message received) and application level (message processed).
- Explicit validation by both sender and receiver systems.
- Use of automatic patient matching, with fallback of manual matching if not entirely unambiguous.
- Routing messages to alternative recipient if not actioned within a specified time (e.g., if a named recipient is on leave).
- Messages are not removed from a task list until all actions specified have been performed.
- If any user edits a message the original is kept unchanged (deletionless messages).
- A full audit trail is maintained.

10.1 HL7 Dynamic Model

The following aspects of the Dynamic Model have to be specified (Fig. 10.1):

- Trigger events
- Application roles (sender and receiver) and their responsibilities
- Interactions (defined as a one-way transfer of information)
- Message type(s)
- Interaction sequence
- Message wrappers
- Acknowledgments

T. Benson, *Principles of Health Interoperability HL7 and SNOMED*, HI,
DOI 10.1007/978-1-84882-803-2_10, © Springer-Verlag London Limited 2010

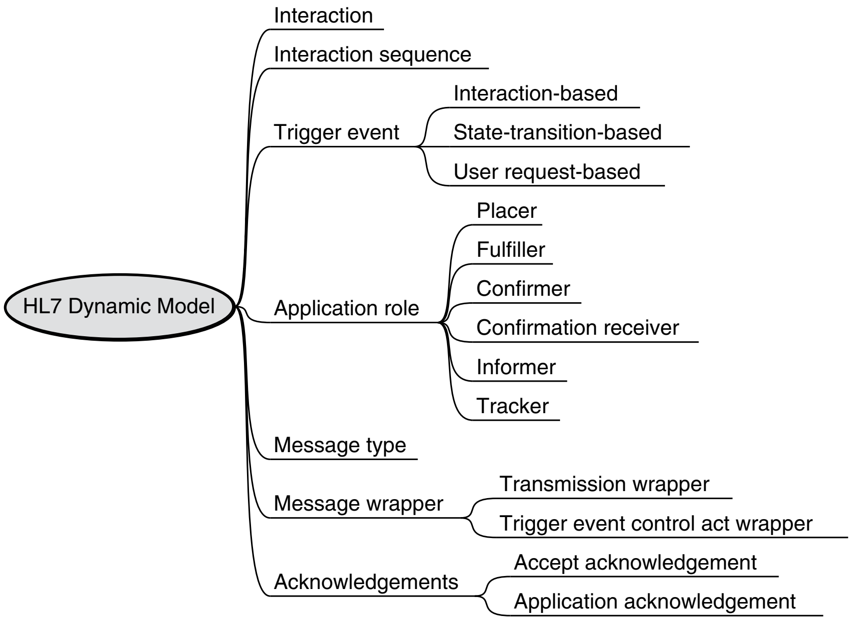

Fig. 10.1 HL7 Dynamic Model

10.2 Trigger Event

A trigger event is an explicit set of stated conditions, which can be recognized by a computer system that initiates an interaction. It may be the result of human action, such as a mouse click, a state transition for an information object (such as the successful completion of a business transaction), an exception condition (such as an error), or be specific to a point in time (e.g., midnight).

The context for each trigger event is specified in use cases, and storyboards, which form part of the requirements specification.

Trigger events may be classified as following:

- Interaction based: Trigger events can be based on another interaction. For example, the response to a query (which is an interaction) is an Interaction-based trigger event.
- State-transition based: Trigger events resulting from a state transition as depicted in the State Transition Model for a particular message interaction. The trigger for canceling a document, for example, may be considered a state-transition-based trigger event.
- User request based: Trigger events may be based on a user request. For example, the trigger event that prompts a system to send all accumulated data to a tracking system every 12 h is considered user based.

10.3 Application Role

An Application Role is a collection of communication responsibilities intended to be implemented as a group. Communication responsibilities are identified as the interactions that the system is able to send or receive. Application Roles may be specializations of other existing Application Roles, inheriting the responsibilities of their parent, with additional or more specialized responsibilities added, or they may be the merging together of other application roles acting as components.

From the Application Role definitions, the reader can identify the purpose of information flow between two health care applications and the roles they play in that exchange.

The Application Role description sets out what one application does, with respect to information exchange. It lists all of the interactions, sent or received, consequent to one particular trigger event. It is silent about the application functionality behind it – and how this is achieved.

Application Roles have responsibilities, which are restricted to sending messages (interactions). Any other responsibilities and actions are outside the HL7 model. The sender role has the responsibility to send a message in response to a trigger event, and the receiver role may have responsibilities to initiate further transactions such as an acknowledgment, error report, and response to query.

The Application Role is a key element in specifying conformance and for contractual arrangements between users and service providers. It is the intent of HL7 that healthcare systems should be able to declare conformance to the HL7 specification by creating an implementation profile that identifies the Application Roles supported by that implementation. Conformance to an Application Role means supporting each of the interactions specified.

Typically, one Application Role supports several interactions. For instance, a query is meaningless unless it includes a response; so the Application Role for the query questioner requires at least two interactions (query and response) to be supported, and similarly for the query answerer.

The names given to Application Roles provide one of the best ways of finding the transaction sets, already defined, which meet a particular requirement. The naming convention is to state the subject of the interaction (e.g., Residential Address) followed by the Application Role category.

HL7 uses the following generic terms for Application Roles:

- Placer: An application that is capable of notifying another application about a significant event, and expects the receiver to take action.
- Fulfiller: An application that is capable of receiving a request from a Placer application.
- Confirmer: An application that is capable of accepting a request from a Fulfiller application.
- Confirmation Receiver: A role implemented by a placer indicating what types of confirmations it accepts.

- Informer: An application that is capable of notifying another application about a significant event, but does not expect any action on the part of the receiver. Paired with "Tracker."
- Tracker: An application that is capable of receiving information about a significant event, but is not expected by the receiver to perform any action.

In theory, Application Roles should be helpful to the reader in understanding the business roles and functionality provided by a set of interactions. However, the use of abstract terms, such as manager, tracker, placer, and filler, makes this less useful than it might be.

10.4 Interaction

An interaction is the smallest unit (atomic) of communication that can stand on its own. It is a one-way transfer of information and ties together HL7's static models of payload content and the dynamic model of information flow and system behavior.

Formally, an interaction is a unique association between a specific message type, a particular trigger event that initiates or triggers the transfer, and the Application Roles that send and receive the message type.

In HL7 Version 3, each interaction is described in a table with its name and artifact ID, together with the sending and receiving Application Roles, the trigger event, the message type, the trigger event type, and the wrapper types and their artifact identifiers.

10.5 Message Type

A Message Type is the most precise specification of a message, with explicit constraints about what data elements are sent and what values each data element may have. These constraints should be as tight as possible to minimize any chance of ambiguity.

Message Types are derived by the intersection of specific interactions, application roles, and trigger events. The same Message Type may be associated with any number of Application Roles and be used in response to many different trigger events. However, an interaction can only ever have one trigger event and one Message Type.

10.6 Interaction Sequence

The precise flow of messages may be represented using a UML sequence diagram, which shows the Application Roles and the flow of Message Types between them in sequential order.

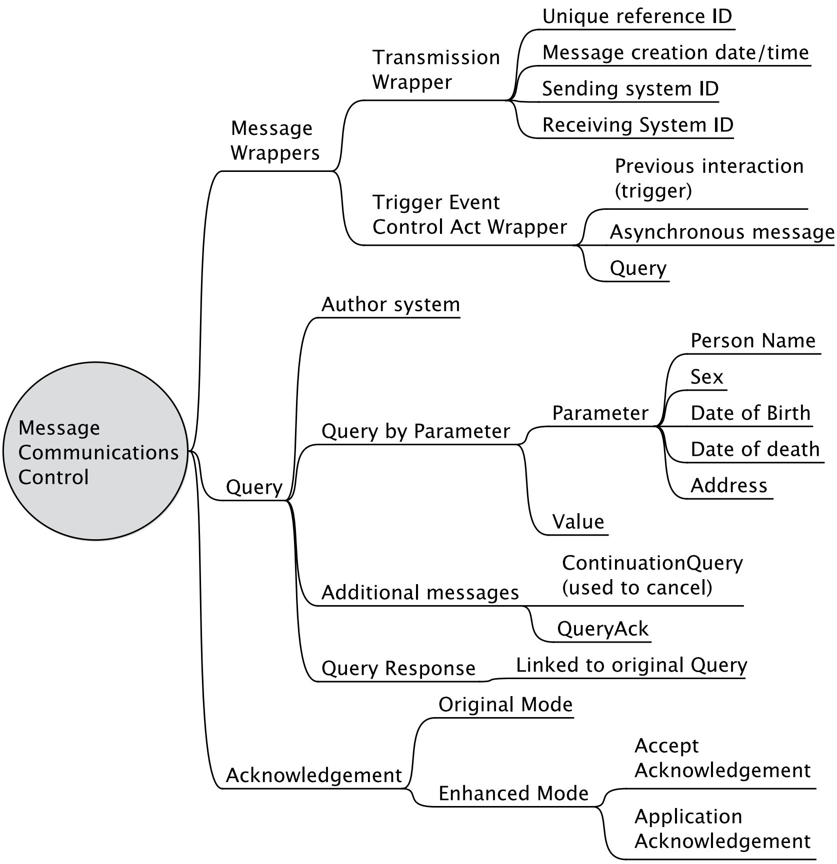

Fig. 10.2 Message communications control

10.7 Message Wrappers

Whenever domain content (as a payload) is transmitted in the form of messages they use message wrappers, analogous to a letter's envelope. HL7 defines two types of wrapper (Fig. 10.2):

- Transmission wrapper
- Trigger Event Control Act wrapper

Each HL7 Version 3 message typically consists of a Transmission wrapper, a Trigger Event Control Act wrapper and the Domain Content.

The Transmission wrapper includes a unique reference ID for each message instance sent, the precise date and time the message was created and the identity of the sending and receiving systems.

The Trigger Event Control Act wrapper sits inside the outer Transmission wrapper and may include details of a previous interaction, which has triggered this interaction. Different variants of the Trigger Event Control Act wrapper are used for asynchronous messaging and for queries, where the response needs to be coupled with the query.

10.8 Query

Queries are used to interrogate databases, such as to retrieve patient identification details from a patient master index.

The standard query message is an extension of the control act, using query by parameter. For a simple patient demographics query, the parameters could be patient name, sex, date of birth (and/or death), and address.

The query response is linked to the original query message.

10.9 Acknowledgment

Most HL7 transactions involve two or more messages: an originating message and an acknowledgement, in one of two modes: original mode or enhanced mode.

In original mode acknowledgment there are just two messages: the first, originating message comes from the sending system and the second, an acknowledgement is sent by the receiver saying whether it was able to process the originating message.

Enhanced mode is more complex, but is suited to a multi-hop environment that uses an intermediary such as an interface engine between the sender and the final recipient. In enhanced mode acknowledgement, two separate acknowledgements are sent.

The first, the "accept acknowledgment" is a message indicating whether the receiving system, which could be an interface engine, was able to take custody of the sender's message, but does not indicate whether it was able to process the information contained within it.

The second, the "application acknowledgment" indicates whether the final receiving application was able to process the sender's message successfully.

Original mode acknowledgement is more straightforward to implement, especially for simple point-to-point interfaces.

10.10 Implementation Technology Specification (ITS)

The XML implementation technology specification describes how individual instances of message types shall be rendered in XML for serial transmission over the network and the structure of schemas used to validate each instance. Note that the HL7 generated XML schemas are not able to test all of the constraints defined in a HL7 message definition.

The generation of the schemas and message representation tends to be done automatically. Those not directly involved in that process do not need to understand the technical details. Hinchley summarizes the key points as follows:

- An XML element is defined to correspond to each row in the HMD grid, with the exception of structural attributes, which are expressed as XML attributes.
- Each Data Type has a defined XML representation. The "restriction base" feature in an XML schema is used extensively to define how data types are implemented.
- The schema files for CMETs are supplied separately and then used by each message schema as required.
- The XML schemas defined support V3 Data Types, and Data Type refinement, through the use of the W3C Schema restriction element. Additional standard schema sections support RIM classes and the HL7-defined vocabulary definitions. These schema sections can be selectively combined with a specific message schema through the "include" function in the XML schema standard.
- HL7 messages all share the same XML namespace.
- Message version information is conveyed as attributes within the message rather than by changes to the namespace identifier.

10.11 IHE XDS

IHE Cross-Enterprise Document Sharing (XDS) allows health care documents to be shared over a wide area network, between hospitals, primary care providers, and social services. The idea is to build virtual patient records on the fly from a variety of clinical documents created by different healthcare organizations (Fig. 10.3).

The main innovation in XDS is the logical and physically separation of the indexing information (the metadata) used to retrieve documents from the actual content. This allows XDS to handle any type of content and simplifies the addition of an XDS export function to existing systems.

XDS enables users (document consumers) to retrieve different types of documents (letters, results, images, and folders) contained in one or more repositories in a quick and consistent way. Each document is viewed in its original form, which may include structured data. XDS provides a foundation on which to build virtual patient records on the fly from a variety of clinical documents created by different healthcare organizations.

XDS has four distinct roles or actors: document source; document repository; document registry; and the document consumer.

- The document source produces original documents, submits these to a document repository (each organization may maintain its own), and generates metadata about each, which is sent to the centralized document registry. That is all that is required.

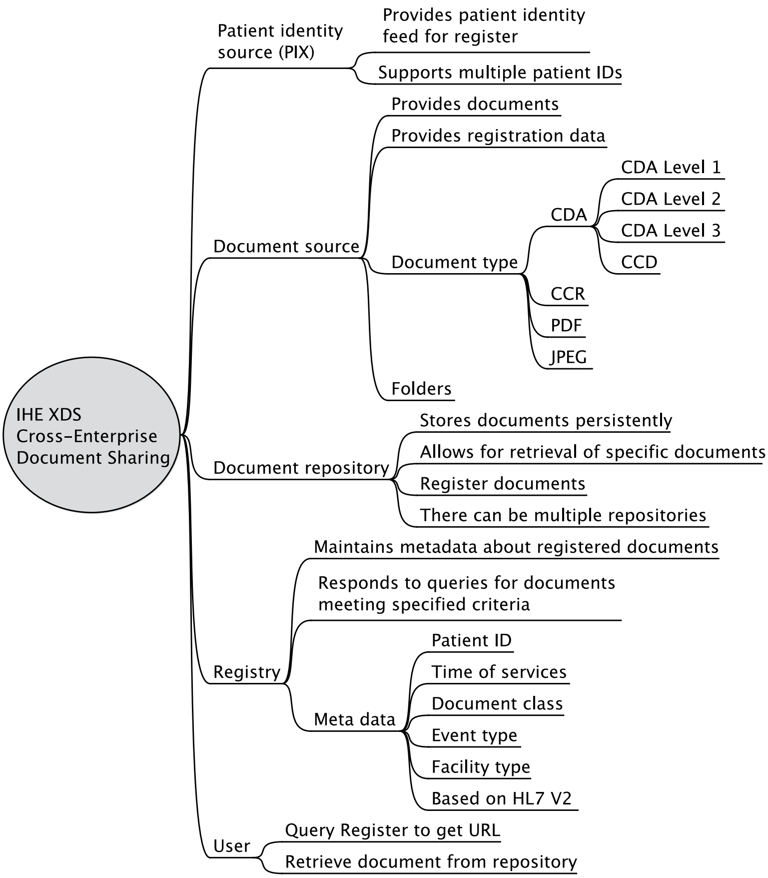

Fig. 10.3 IHE XDS

- Each document repository (there can be more than one) provides secure document storage and supports document retrieval; documents may be in folders. Documents may be stored in a number of different formats, such as CDA or PDF.
- The central document registry indexes documents, supports searches, and maintains a URI link back where the document is stored in a document repository.
- The document consumer is a user system which initiates search of the register, retrieves, and displays selected documents from their repository(ies).

To maintain security, the XDS registry has no access to the content of any document, but relies entirely on standardized metadata provided by the source to retrieve relevant items. The standardized set of metadata is the key to XDS and includes patient ID, dates/times, clinical event context, document type, author, size, format, and location.

The basic XDS has been refined to support special requirements for DICOM images (XDS-I), HL7 CDA medical summaries (XDS-MS), and structured laboratory reports (XDS-Lab).

Technically, the XDS document registry is a subset of the ebXML Registry standard (ISO 15000 parts 3 and 4). Documents are exchanged using SOAP and HTTP, while SQL is used for information retrieval. Various document formats may be used, including HL7 CDA Release 2, DICOM, and PDF. The format of the metadata is largely based on HL7 Version 2.[1]

[1]IHE IT Infrastructure Technical Framework Volume 1 (ITI TF-1), Chapter 10, 69–99. http://www.ihe.net/Technical_Framework/upload/ihe_iti_tf_2.0_vol1_FT_2005-08-15.pdf