Health Level Seven
www.hl7.org

Healthcare Development Framework (HDF)

Version 1.2

Published on: March 31, 2008

Revision History:

Date	Authors	Version	Change Reference
Oct. 25, 06	Ioana Singureanu	0.1 Draft	Initial Draft
May 22, 07	Ioana Singureanu	0.1 Draft	Migrated Project introduction, initiation, and profiling
July 27, 07	Ioana Singureanu	1.0	Finalized the Change Review Process Design
August 17, 07	Ioana Singureanu	1.0.1	Updated model publication
November 23, 07	Ioana Singureanu	1.2	DSTU Publication
March 11, 08	Freida Hall, Ioana Singureanu	1.2	Reconciled ballot comments, updated Project Lifecycle, updated Project Approval and Initiation to represent the approved process.

CONTENTS

Healthcare Development Framework (HDF) 6
Authors: 6
Change History 7
1. HDF Background, Objectives, Overview 8
1.1 Introduction 8
1.1.1 Scope 8
1.1.2 HL7 Version 3 9
1.1.3 Unified Modeling Language 9
1.1.4 Healthcare Development Framework 9
1.1.4.1 Scope of Coverage 10
1.1.4.2 Use of UML Notation 10
1.1.4.3 HDF Lifecycle Management 10
1.2 HDF Overview 10
1.3 Project Lifecycle Process 12
1.3.1 Project Life Cycle for Product Development (PLCPD) 13
1.3.2 Project Initiation and Sunset 13
1.3.3 Project Management Approach 14
2. Project Initiation Process (PIP): Initiation, Planning, and Approval 14
2.1 Overview 14
2.2 Context 15
2.2.1 HL7 Work Effort 16
2.3 Roles and Responsibilities 17
2.4. Process 17
2.5 Quality Criteria 17
2.6. Tools 17
2.7 Artifacts 18
3. Domain Analysis Process (DAP): Analysis and Requirements Documentation 18
3.1 Overview 18
3.2 Context 20
3.3. Roles and Responsibilities 20
3.4 Process 20
3.4.1 Business Context Analysis 21
3.4.2 Use Case Analysis 22
Additional Guidance 23
3.4.3 Process Analysis 23
3.4.4 Information Analysis 24
3.4.5 Business Rules Analysis 24
Additional Guidance: 24
3.5 Quality Criteria 25
3.6 Tools 25
3.7 Artifacts 25
DAM Artifact Samples 30
4. Specification Design Process (SDP): Design and Harmonization 30
4.1 Overview 30
4.2 Context 31
4.3 Roles and Responsibilities 32
4.4 Process 33
4.4.1 Information Model Design 33
4.4.2 Dynamic Model Design 33
4.4.3 Design Harmonization 34
4.4.4 Design Localization 34
4.4.5 Design Publication 34
4.5 Quality Criteria 34
4.5.1 Completeness 34
Alignment with HL7 references 35
4.5.2 Consistency 35
4.5.3 Quality Validation 35

4.6 Tools....36
4.7 Artifacts....36
4.7.1 Information Modeling Artifacts....36
4.7.2 Dynamic Modeling Artifacts....37
4.7.2.1 Sample Diagrams for Dynamic Model Design....40
4.4.2.1(a): State Transitions....40
4.4.2.1(b): Sending process....40
4.4.2.1(c): System Interfaces....41
4.4.2.1(d): System Interactions....42
4.4.2.1(e): Receiving Process....43
4.4.2.1(f): Healthcare Services Specification....44
4.7.3 Publication Artifacts....45
5. Standard Profiling Process (SPP): Constraints, Extensions, and Annotations....45
5.1 Overview....45
5.2 Context....46
5.2.1 Conformance using Profiles....47
5.2.2 Profile Types....48
5.2.2.1 Profiling Methods....49
5.2.3 Layered approach to profiling....51
5.2.3.1 Level 1 - RIM Conformance....52
5.3 Roles and Responsibilities....52
5.4 Process....52
5.5 Quality Criteria....53
5.6 Tools....53
5.7 Artifacts....53
6. Technology Specification Process (TSP)....53
6.1 Overview....53
6.2 Context....54
6.3 Roles and Responsibilities....54
6.4 Process....54
6.5 Quality Criteria....54
6.6 Tools....54
6.7 Artifacts....55
7. Change Control Process (CCP)....55
7.1 Overview....55
7.1.1 Change control using HL7 Project Homebase....55
7.2 Context....56
7.3 Roles and Responsibilities....57
7.4 Process and Tasks....57
7.5 Quality Criteria....58
7.6 Tools....58
Screen Shots....59
7.7 Artifacts....68
Annexes....68
Annex A. Domain Analysis Example....68
Domain Analysis Model....69
Business Process Model Analysis....70
Process Flow (Activity Diagram)....71
License Renewal Process - main flow....71
License Renewal Process - alternate flow....72
System Interactions....73
Use Case Model....74
Information Model....76
Business Trigger Analysis....78
Specification Development....78
Design Information Model - DIM....80
Message Structure - CIM....81
Message Envelope....83
Localization - LIM....83
Dynamic Model....83
Functional Model....84
System....84
Technology-Specific Artifacts....84

BPEL....84
WSDL....84
XSD....84
Annex B. System Interaction Examples....84
Annex C. References....85
Glossary....85

Healthcare Development Framework (HDF)

This specification describes the methodology and artifacts required to develop implementable interoperability standards for healthcare.

The following diagram provides an overview of the HDF content. For more details on the HDF project refer to **http://hl7projects.hl7.nscee.edu/projects/hdf/.**

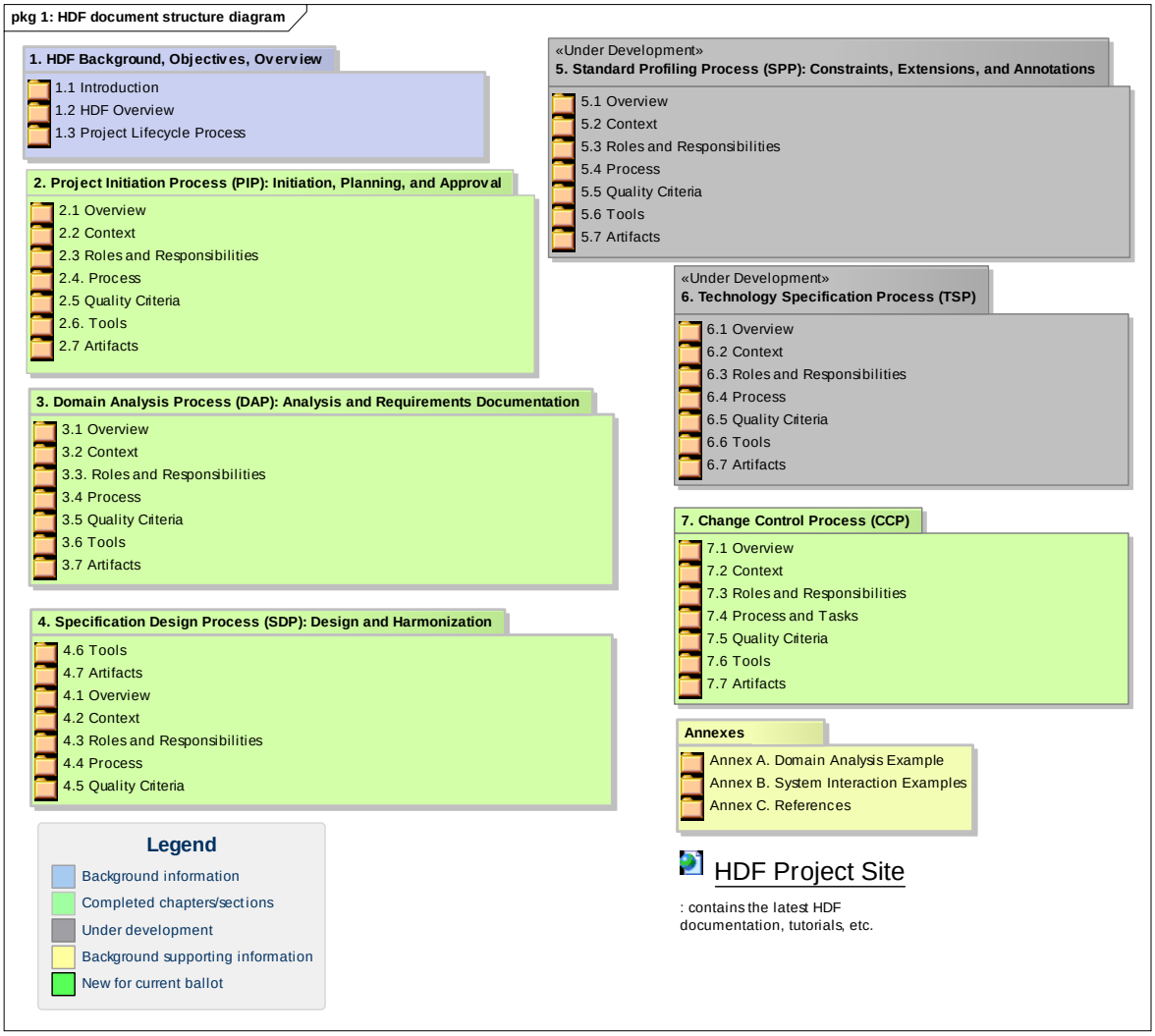

Figure 1: HDF document structure diagram

Authors:

Modeling & Methodology Co-Chair, Editor:
Ioana Singureanu
ioana@eversolve.com
Eversolve, LLC

Contributor:

Lee Coller
lee.coller@oracle.com
Oracle

Contributor:
Norman Daoust
norman@daoustassociates.com
Daoust Associates

Contributor:
Len Gallagher
lgallagher@nist.gov
NIST

Contributor:
Charlie Mead
mead_charlie@bah.com
Booz Allen Hamilton

Contributor:
Freida Hall
Freida.Hall@va.gov
Department of Veteran Affairs

Contributor:
Kenneth H. McCaslin
Kenneth.H.McCaslin@questdiagnostics.com
Quest Diagnostics

Last Published: **3/23/2008**

HL7® Version 3 Standard, © 2006 Health Level Seven®, Inc. All Rights Reserved.

HL7 and Health Level Seven are registered trademarks of Health Level Seven, Inc. Reg. U.S. Pat & TM Off

Change History

The Change History represents a record of changes to the specific document itself. The current document and the underlying artifacts is expected to be revised as the HL7 methodology is enhanced, clarified, peer reviewed and receives feedback through use.

1. May 2006: Requirements Analysis and Specification Modeling with annex - for comment

2. September 2006: Added Standard Profiling, ballot reconciliation for Requirements Analysis and Specification Modeling. Added the Project Initiation section approved as informative in September 2005.

3. January 2007: Resolved comments and enhanced Standard Profiling section.

4. May - September 2007: The HDF is migrated to UML2 and each process is enhanced with diagrams. The Change Control Process was added. The Specification Design was refactored. The Technology Specification chapter was added as a place-holder for future ballots.

5. January 2008: Project Definition, Planning, and Approval revised to reflect the new HL7 organizational structure and Harmonization Process added.

6. May 2008: Renamed document "Healthcare Development Framework" to allow its reuse by other SDOs.

Resolved ballot comments from the first DSTU ballot.
Updated both Lifecycle and Project Approval processes (approved by TSC).

1. HDF Background, Objectives, Overview

1.1 Introduction

The Healthcare Development Framework (HDF) specification is a product of the methodology specification project lead by the Modeling and Methodology work group. The purpose of the HDF Project is to analyze, design, and document the processes, policies, and artifacts associated with development of HL7 standards.

This document details the process of the HL7 standard development methodology. The HDF is a framework of modeling and administrative processes, policies, and deliverables used by HL7 to produce specifications for enabling interoperability among computerized healthcare information systems.

The HL7 *Message Development Framework* (MDF) was produced by HL7 in 1997 to describe the model driven development methodology for producing HL7 Version 3.0 message specifications; it will be superseded by the HDF. The specifications produced by HL7 target multiple facets of the interoperability challenge. They include specification of information models, data types, and vocabularies; messaging, clinical documents, and context management standards; and implementation technology, profile, and conformance specifications. Despite the diversity in depth and scope of HL7 specifications a common thread is the use of a model driven methodology and the derivation of specifications and interim work products from a common set of reference models.

The models used in the HDF development methodology use the Unified Modeling Language (UML) as the preferred syntax. The HDF is a replacement for and an extension to the Message Development Framework (MDF). The HDF differs from and enhances the MDF by more closely aligning the underlying metamodel governing well-formed HL7 models with the metamodel of UML and applying the model driven process to all of the technical specifications of HL7, not just messages.

The HDF project has multiple phases and anticipated deliverables. This documentation of the HDF development methodology has as companion documents a specification of the HDF Metamodel (UML Profile) and Developer's Guides (see Annex c). This specification will be updated periodically and additional companion documents will be created as the project continues. References to other information sources on related topics are included in the appropriate sections of this specification.

1.1.1 Scope

The HDF includes several sections or chapters that document the processes required to complete HL7 artifacts and standards. In addition to these process definitions the HDF contains sample artifacts and references to supporting documents and style guides

1.1.2 HL7 Version 3

In 1992 HL7 made a fundamental shift in the methodology it uses to develop its standards specifications. The new methodology, referred to as HL7 Version 3.0 or V3, is a model-driven methodology based upon modern object-oriented software development practices. HL7 spent four years creating the methodology that adapts modern analysis techniques from system building to message design.

In 1992, HL7 Executive Committee chartered an independent task force to establish an approach. In January 1996, the Technical Steering Committee agreed to adopt the main features of the approach and take over its management. In the spring of 1997, all the HL7 Technical Committees began to use the V3 process.

HL7 version 3.0 is the most definitive HL7 standard thus far, incorporating more trigger events and message formats than any previous version. It uses a Reference Information Model (RIM) as a common source for the information content of specifications. The RIM is an essential part of the HL7 Version 3.0 development methodology. It provides an explicit representation of the semantic and lexical connections that exist between the information carried in the fields of HL7 messages. As part of version 3.0, the HL7 Vocabulary Technical Committee developed methods that allow HL7 specifications to draw upon codes and vocabularies from a variety of sources.

The use of standardized vocabulary ensures unambiguous interpretation of the code sources and code value domains across systems. HL7's primary goal for version 3.0 is to offer a standard that is definite and testable, and to provide certification of vendor's and implementer's conformance.

The HL7 Version 3.0 development methodology is a continuously evolving process that seeks to develop specifications that facilitate interoperability between healthcare systems. The HL7 RIM, vocabulary specifications, and model-driven process of analysis and design combine to make HL7 Version 3.0 an exemplary methodology for development of consensus-based standards for healthcare information system interoperability. The HDF is the most current rendition of the HL7 V3 development methodology.

1.1.3 Unified Modeling Language

The models used in the HL7 V3 process are based upon the Unified Modeling Language (UML). The UML is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The UML is an Object Management Group standard that represents the unification of best practices in practical object-oriented modeling. The Object Management Group (OMG) is an open membership, not-for-profit consortium founded in 1989 that produces and maintains computer industry specifications for interoperable enterprise applications.

The version of the UML specification used by this project is UML 2.1. Under the stewardship of the OMG, the UML has emerged as the software industry's dominant modeling language. It has been successfully applied to a wide range of domains, ranging from health and finance to aerospace. Its extensive use has raised numerous application and implementation issues by modelers and vendors. Over 500 formal usage and implementation issues have been submitted to the OMG for consideration and they have been resolved.

1.1.4 Healthcare Development Framework

The HL7 Development Framework (HDF) is a replacement for and an extension to the HL7 Message Development Framework (MDF). It is the primary deliverable of the HDF project. The HDF differs from the MDF in terms of the following:

1.1.4.1 Scope of coverage;
1.1.4.2 Use of UML Notation;
1.1.4.3 HDF Lifecycle Management

1.1.4.1 Scope of Coverage

The HDF documents the processes, tools, actors, rules, and artifacts relevant to development of all HL7 standard specifications, not only messaging. This initial version of the HDF methodology specification will address updates to messaging specification, and will be applicable to structured documents and context management. The HDF is intended to encompass all of the HL7 standard specifications, including any new standards resulting from analysis of electronic health record architectures and requirements.

1.1.4.2 Use of UML Notation

The meta-model of the artifacts to be produced as part of the development processes documented in the HDF will be UML 2 conformant. The differences between the HL7 Meta model and the UML meta model have been analyzed and resolved by an HL7 UML Profile (published starting in 2006) that leverages the UML extension capabilities.

1.1.4.3 HDF Lifecycle Management

In addition to documenting the processes for development of HL7 standard specifications, the HDF, in conjunction with HL7 Bylaws and Policies and Procedures, also documents the policies and procedures that govern the enhancement and version management of those processes.

The HDF Methodology Specification is a "living document" intended to keep pace with the improvements in methodology. The HDF Life cycle management is uses proven techniques in document management, configuration management, and version control. The HDF is intended to be approved as an informative, background document as a part of the HL7 specification approval process and it will be re-balloted as it changes over time.

1.2 HDF Overview

The process used in the development of HL7 specifications and documented in the HDF consists of the following process categories:

The HDF includes the following sections; the current status is indicated in brackets:

HDF Background (including Project Lifecycle for Product Development) - approved
Project Initiation, Planning, and Approval - approved
Requirements Analysis and Documentation - for review
Specification Design - for review
Specification Profiling - for review
Technology Specification - under development
Change Control Process - for review

Annexes:
A: Domain Analysis Example
B: System Interaction Examples

C: References

Although the activities are presented in a linear fashion, they should be thought of as independent of any predefined order of occurrence. There are implied dependencies between the activities as work products produced in one activity are consumed as input to other activities. However, the process is intended to be followed in an iterative fashion with feedback loops from later activities to their predecessors to revise, append, or otherwise enhance earlier deliverables.

The following diagram provides a high-level view of the processes described in the HDF methodology specification. The diagram illustrates the key processes in the context of a product lifecycle process and on-going change control (e.g. technical corrections, enhancement requests, new standard creation) to the HL7 standards products.

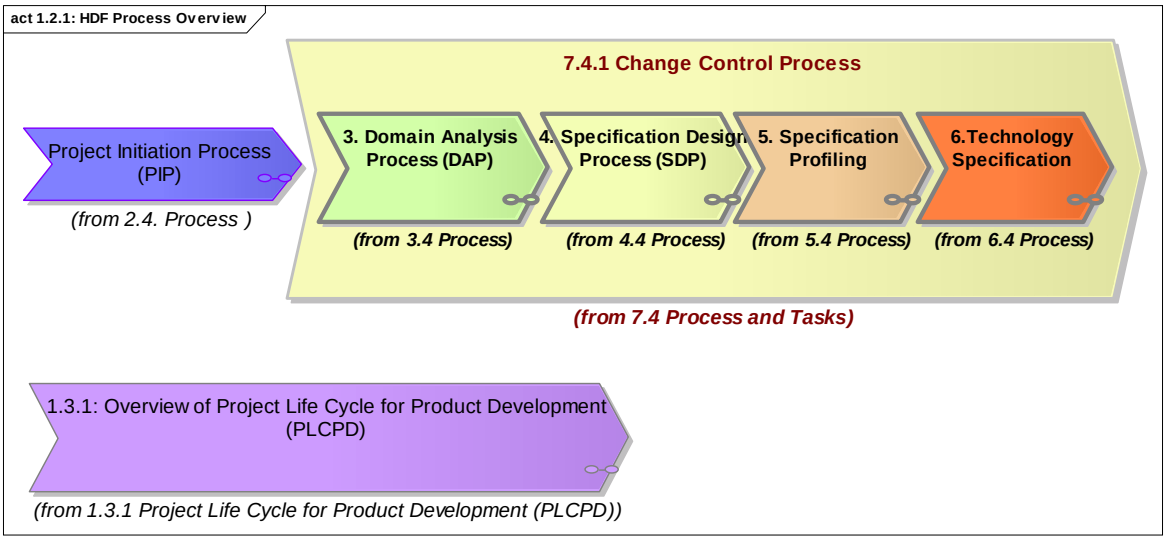

Figure 1.2.1: HDF Process Overview

The following diagram provides an in-depth view of the processes along their input, outputs, and triggers. As seen in this diagram, the HL7 standard development starts after a project was approved.

1) The first phase of any project consists of analyzing the standard development project requirements and fulfilling the steps identified in "Domain Analysis" process. The input requirements may identify business processes that are automated, they may specify information and coded terminology to be shared between systems, and specific operations that are used to exchange the information and exposed system capabilities The artifact produced from requirements analysis is a Domain Analysis Model that provides a synthesis of both information and behavioral requirements in a formal language (UML) using a common style across committees.

2) The Domain Analysis Model (DAM) is used as the basis for "Specification Development". The specification may be a functional model, a service specification, a message definition, etc. depending on the type standard targeted by the project. The standard specification may be directly based on reference information models and on the contents (the information and operations) specified in the DAM or derived from it through mappings or transformation. Both the DAM and standard specifications will rely on models and diagram to manage the contents and to provide views of the information and interactions between systems.

3) The standard specification undergoes a process of transformation into a specific technology artifact (e.g. XML schema, Web Service Description Language files, Java code) based on the standard

specification.

4) Standard Profiling is the process of adapting, localizing, or extending a standard specification for the purposes of implementing the standard in a project or solution.

NOTE: If you are viewing the web publication of this document, any processes that appear in bold letters or are represented as structured elements are linked to detailed diagrams.

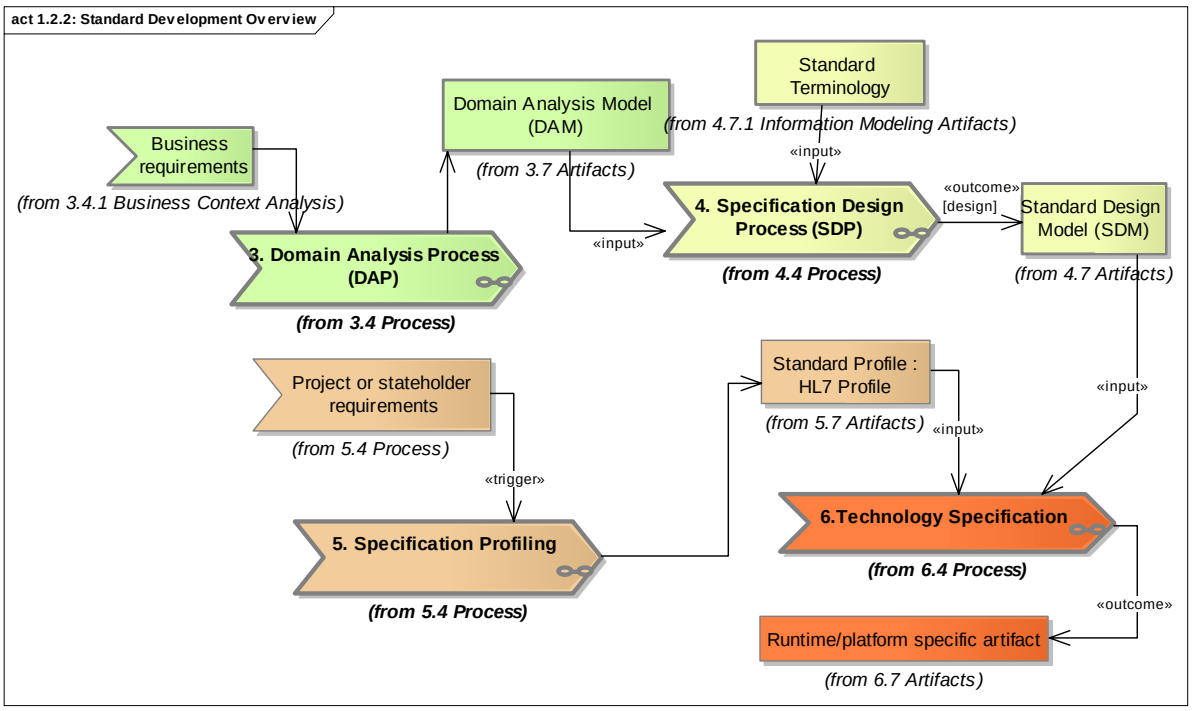

Figure 1.2.2: Standard Development Overview

1.3 Project Lifecycle Process

Note: The Project Lifecycle Process is maintained by the Project Services Committee working in conjunction with other HDF editors and contributors.

The objective of the HDF methodology specification is to document the context, processes, and work products comprising the entire HL7 standard development Life cycle. The HDF methodology consists of abstract processes that apply to any particular kind of specification or development project. The HDF introduces the processes, describes their deliverables, and documents their interdependencies. Secondarily, HDF serves as input to the development of more specific developer guides focused on particular types of specifications and projects.

References to additional information sources are provided as an aid to those interested in further detail or elaborations on the topics presented in this specification as an annex or reference.

This approach to documenting the methodology was taken for the following reasons:

Management of the size and complexity of this specification.
This specification is expected to be used by a large audience with diverse interest in detail and with varying levels of experience with the processes and techniques used in the methodology. In order to

allow the document to be useful to a larger audience we have elected to include references and linkages to more detailed documents so as to reduce the complexity of this specification.

Insulation of this specification from changes in dependent specifications.
The HDF methodology continues to evolve thus changes in work products, tooling, and policies are inevitable. By adopting a relatively high level of abstraction and isolating detailed specifications to other publications this overall description of the HDF methodology is insulated from changes in the detail.

Anticipation of publishing the methodology as a collection of interrelated linked documents.
The HDF methodology specification and its related companion documents are published or linked to the HL7 web site and ballot site. By linking related documents to this specification we avoid any duplication and possible conflicting content when we render these documents as web artifacts.

1.3.1 Project Life Cycle for Product Development (PLCPD)

The following diagram details the product or project lifecycle that the alignment between the lifecycle process and the HDF which deals with the creation of standard artifacts. As seen in the following diagram, the Project Initiation process defined in HDF supports project initiation (step 5) in the project lifecycle for Product Development. Similarly "Requirements Analysis" and "Specification Design" are clearly supporting the overall lifecycle process.

The following diagram details the product or project lifecycle. The Standard Development Process Steps are further detailed in the HDF. Each activity in this process is identified by a numbering scheme and a unique name (e.g. "(2) Approvals"). The activities are further described in the table below.

Note to the reader: The Project Life Cycle for Product Development (PLCPD) provides the strategic process for HL7 standards while the HDF focuses on the tactical development processes and artifact for processes included within the PLCPD. Additionally, some terms are described in the glossary.

1.3.2 Project Initiation and Sunset

Upon recommendation by the steward domain committee, the decision to sunset a product must be approved by the Steering Division, TSC, or other appropriate approval bodies, as defined by the HL7 organization and policy. Product sunset requires a project, but with significantly fewer steps than required to create a product.

a. Project Initiation
Same as Step (5) in section 1.3.1, but typically does not require funding and has no ballot requirements, but does require a transition plan for the news standard

b. Appropriate Approvals to move forward
Same as Step 2 in section 1.3.1.

c. Review Documentation
Similar to Step (19) in section 1.3.1, but requires technical corrections to the retiring standard.

d. Announce and set sunset Date
Inform Marketing an announce sunset to HL7 Membership.

e. Product Sunset
The project artifacts are archived and any subsequent work ends.

1.3.3 Project Management Approach

HL7 will define and recommend a project management approach for the development of new products or enhancement of existing products that values volunteer culture by providing appropriate tooling to support project management.

In a sense, the old adage "form follows function" applies here in that the product-project lifecycle sets forth the framework for the project management activities. Within the HL7 volunteer culture there are other factors that impact on the project management approach, including:

Streamlined: The approach must support macro- rather than micro- management. HL7 has neither the need nor the resources to micro-manage projects. Further, the volunteer culture simply does not lend itself to micro-management. At the same time, as indicated in the previous section, certain lifecycle steps such as project initiation will require the capture of specific criteria that need to be tracked (schedule, resources, objectives/outcomes, etc) in order to measure progress and evaluate status.

Supportive: Because project management skills are not currently readily available in the HL7 community the project management approach must not impose significant work requirements on the project team. Staff consulting resources must be available.

Effective: Must provide appropriate level of sophistication to enable the project manager to determine project status and progress.

Complete: Must be able to track and monitor all the key criteria.

Thus project management tools must be:

Easily understood
Able to track and periodically update status of project criteria attributes
Able to track committed resources
Able to determine whether a project is ahead of or behind the proposed schedule
Able to capture progress notes and comments
Able to interface with other projects where there are dependencies (i.e. elements that are required for use in the subject project)
Accessible via the Web with user rights controls

Does not need to be able to:

Support elaborate critical path scheduling
Support resource requirements by type monitoring at the task level
Support resource usage time reporting

2. Project Initiation Process (PIP): Initiation, Planning, and Approval

2.1 Overview

Note: The Project Initiation, Planning, and Approval Process is maintained by the Project Services

Committee working in conjunction with other HDF editors and contributors.

Project Initiation is the first step in the development of all Health Level 7 (HL7) specification projects. Project Initiation and the subsequent ongoing management helps ensure the entire HL7 community can track what specifications are developed and how they are progressing. The Project Scope Statement identifies the business need that the project was undertaken to address and a high level description of the project. The Project Scope Statement (PSS) is a communication tool that enables HL7 to make the most effective use of the collective effort expended. The objectives are to avoid any duplication of effort, encourage collaboration in areas of common interest and ensure resulting specifications contribute to a coherent approach across all HL7 products.

Once the project scope statement is approved, ongoing project status is tracked in HL7's project management application (Project Insight) which can be used online and it available at "http://healthlevelseven.projectinsight.net". As the project progresses and new information becomes available, adjustments may need to be made in its scope, milestones, deliverables, dependencies or any other aspect that changes the understanding of what the project is intended to accomplish. The Project Services Committee will develop additional guidance (e.g. when to update a Project Scope Statement and when it is appropriate to initiate a new project instead).

Sub-projects may be initiated to undertake some related tasks, such as tooling, training, or joint work to combine efforts to solve common problems. Each sub-project must have its own project scope statement, indicating the relationship to the parent project and its own objectives and expected deliverables.

The project is subject to varying degrees of approval depending upon the type of specification being produced, participants involved, and financial implications. Once approved, the project is formally recognized and is appropriately aligned to the ongoing work of the organization.

This process document is intended an adjunct to a work group's decision making practices which governs operational items such as quorum determinations, agenda setting, notification, etc.

:

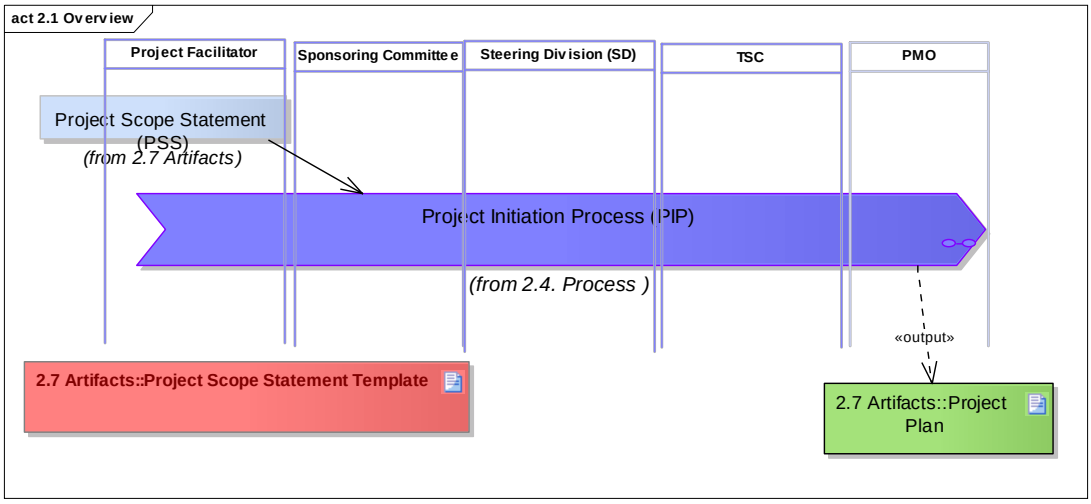

Figure 2.1 Overview

2.2 Context

HL7 is a standards development organization (SDO). HL7 products are specifications that are balloted to

become standards that implementers use to enable interoperability among information systems in healthcare.

HL7 also produces educational material, develops best-practice processes, develops tools and evaluates specifications/tools.

Standards development is inherently iterative that requires a complete development cycle. The cycle includes steps of planning, requirements, analysis, design, balloting and publication of the specification. Some projects may also have a construction, testing and deployment phase. Refer to section 1.3 Project Lifecycle for additional information.

HL7, or HL7 in collaboration with external parties, typically do the planning, requirements, analysis, design, balloting and publication of the specification
External parties, using HL7 specifications, typically do the construction, testing and deployment steps; HL7 does not build applications but provides specifications intended to enable interoperability between applications provided by any vendors.

“Multiple System”/”Multiple Context” is the “problem space” that HL7 specifications are working with. It also characterizes the environment in which HL7, including HL7 Affiliates, is seeking to develop specifications. HL7 is a group of individuals dispersed around the world working concurrently, largely through electronic communication, to produce specifications that are congruent across the full set of products.

A full set of project scope statements representing the collective work of HL7, is part of the necessary minimal infrastructure needed to manage within an inherently complex environment. Experience in complex environments has demonstrated repeatedly that a minimum level of consistent communication is necessary to avoid wasting scarce human attention on efforts that do not contribute to the congruent outcome.

2.2.1 HL7 Work Effort

An HL7 work effort represents an activity being undertaken by an existing Technical Committee (TC), Special Interest Group (SIG), or Board appointed committee (hereafter, in this document, collectively referred to as work group) to achieve specific objectives or to produce specific work products.

A work group shall consider a work effort to be a project if one or more of the following is true of that work effort:

• involves a group outside of HL7 (may require Board approval)
• requires external funding (may require Board approval)
• is going to be balloted
• requires cross-committee participation
• is determined to be a project by the committee

An HL7 project:

• has an objective (statement of what is going to be produced),

• will have a finite existence (the end date to be determined by the resources available and the start date), and

• if additional funding is required, it will have a budget (including resources and funding sources)

• will have at least one participant available to contribute and must have a project leader, if only an interim

to get the project started

• will have at least to implementers (unless the project is intended to support the HL7 infrastructure)

• will have an estimated schedule

2.3 Roles and Responsibilities

This section describes the various roles involved during the project definition. It may be possible for the one individual to fulfill several roles if the skills and schedule allow it.

Name	Description
Project Facilitator	The facilitator is involved in the project initiation on behalf of the group (committee or project) responsible for the work products in the scope statement.
Sponsoring Committee	The sponsoring committee is Technical Committee or Special Interest Group (work group) that owns the subject area that is at the core of the project.
Steering Division (SD)	A Steering Division (SD) that includes among its members the Sponsoring Committee
TSC	Technical Steering Committee (TSC), the body responsible for approving all HL7 projects.
PMO	HL7 has staffed Project Management Office (PMO) to support HL7 committees. The PMO is responsible for coordination and project management support for HL7 projects.

2.4. Process

This section describes the steps of the Project Approval and Initiation process. This process was updated to reflect the changes made to the HL7 organizational structure in 2007.

This process was approved by the HL7 Technical Steering Committee (March 2008).

2.5 Quality Criteria

The project scope statements submitted by project teams are subject to quality verification as they advance through the approval process. If the required information is missing or incomplete, the scope statement may be rejected and updated by the project facilitator(s). The Project Scope Statement template indicates what information is mandatory.

2.6. Tools

This section identifies the tools required to complete this process successfully.

Name	Description
Word Processor	A word processor will be used to create the scope statement using the document

	template provided in the "Artifacts" section. The word processor must be capable of using the Project Scope document template published by HL7.
HL7 Project Homebase	The GForge-based HL7 Project Homebase is used to track project scope statements and TSC approval. The TSC Issue Tracker is located at: http://hl7projects.hl7.nscee.edu/tracker/?atid=313&group_id=52&func=browse After this submission, the Project Facilitator and the Sponsoring Committee can monitor the status of the PSS by on the TSC Issue Tracker above
Project Management Tool	The information provided in the project scope statement will be entered in the HL7 project management tool (Project Insight). This tool is used to track the project's progress towards meeting its stated objectives and its estimated milestones. Project Insight is accessible on the Web at http://healthlevelseven.projectinsight.net.

2.7 Artifacts

This section describes the structure of artifacts produced during project definition and approval process.

Name	Description
Project Scope Statement Template	The scope statements provided by project teams must follow a pre-defined template format in order to be approved. The template contains specific required information (e.g. team, implementers, etc.) that is mandatory for initiating an HL7 project. The PSS document template is maintained by the Project Services Committee at: http://hl7projects.hl7.nscee.edu/docman/?group_id=55.
Project Scope Statement (PSS)	The Project Scope Statement (PSS) is created and revised by the Project Facilitator. The PSS is created using the Project Scope Statement document template.
Project Plan	Project plan is maintained in the HL7 Project Management tool (Project Insight) and it used to track the progress of each project. The project plan is based, initially, on the estimated milestones provided in the project scope statement and may be revised as the project progresses.

3. Domain Analysis Process (DAP): Analysis and Requirements Documentation

3.1 Overview

Domain Analysis produces a set of artifacts that clearly describe the healthcare business in a given domain in terms familiar to the people who work in that business area. This set of artifacts is known as Domain Analysis Model (DAM). HL7 workgroups use these artifacts to develop HL7 V3 standard specifications. Each artifact in the DAM must be unambiguously stated in a way that can be well understood both by the domain experts and by the HL7 project members who are responsible for developing the specification.

This chapter presents a set of internally consistent processes and techniques for analyzing and

documenting interoperability requirements and enable domain experts to explicitly define the problem in a manner consistent with HL7 design methodologies. These processes make extensive use of the UML 2.1 standard notation and tooling (see sections 3.6 and 3.7). The process encourages project teams to focus on the underlying healthcare information and process requirements before designing a standard specification.

While the focus of this chapter is to identifying interoperability requirements for standard development, this process could be used for to analyze requirements for other purposes or projects.

Requirements/Domain Analysis is essentially a task for domain experts and business analysts who represent the users and understand their system interoperability needs. The problem space for HL7 is defined by the interoperability requirements of stakeholders in a given domain of healthcare delivery or administration. This includes all sharing of information among healthcare stakeholders that may be required for the collection, aggregation, reporting, and other analysis of any clinical, administrative, and financial data information that is relevant to the business.

A DAM defines what needs to be done, not how to do it. It is important to separate the description of requirements from that of the design of the solution. Prematurely including technical and implementation details will compromise the clarity of the original problem and will result in standards that fall short of the business needs.

The primary deliverable produced during requirements documentation is the Domain Analysis Model (DAM) that captures the requirements is:

Business Process subject to automation through interoperability
Information exchanged (including any controlled vocabularies)
Application/system Interactions, Triggers, and Constraints
Business Rules

The DAM is then used to create standard specifications by harmonizing it with HL7 references: Reference Information Model (RIM), structural vocabulary, and application roles.

The following is a high-level process overview of the Domain Analysis process. The diagram is showing a reference to the sample DAM included in the Annexes section:

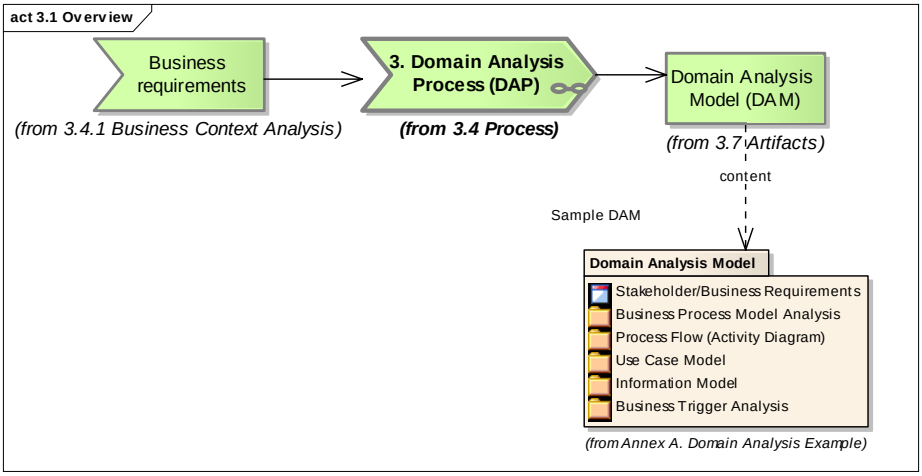

Figure 3.1 Overview

Name	Description

Text	Sample DAM

3.2 Context

Healthcare consists of a number of complex problem domains defined from multiple perspectives. This results in an ongoing demand for integrated clinical, administrative, and financial information systems. There is near universal agreement among healthcare professionals that integrated information systems are an essential tool for maximizing both the efficiency and quality of healthcare delivery.
HL7 is committed to producing requirements that promote computable semantic interoperability. The scope of this interoperability must span a variety of contexts—clinical, administrative, financial, individual patient, aggregated populations, non-human care, community care, multiple reimbursement model—and be defined in a technology neutral format.

3.3. Roles and Responsibilities

This section describes the various roles involved during the development of the requirements specification. It may be possible for the one individual to fulfill several roles if the skills and schedule allow it.

Name	Description
Domain Expert	A Domain Expert, sometimes known as a Subject Matter Expert (SME) or Subject Matter Specialist (SMS), has detailed knowledge and hands-on experience in the domain of interest. This role does not require detailed knowledge of HL7 but it does require high level understanding of interoperability concepts. During the course of Requirements Analysis, a domain expert will acquire working knowledge of UML in order to communicate effectively with the Business Requirements Analyst. The SME associates actors with the activities they perform, specifies when they perform them, and what information is required. The SME will provide data element definitions and terminology definitions, where appropriate
Business Analyst	The business analyst is knowledgeable about the interoperability needs in a certain domain and the systems that are involved. The analyst must have knowledge of business processes and how those business processes are automated through the use of integrated systems. The analyst and domain expert are expected to analyze the information requirements and business process requirements needed to fulfill the scope of the project.
HL7 Modeling Facilitator	The HL7 Modeling Facilitator is knowledgeable in applying the HL7 Requirements Analysis process described in this chapter. This person is responsible for guiding the development of the requirements specification and for coordinating all of the activities associated with the analysis of project requirements. The facilitator is skilled in the use of the UML tools and in creating models and view during requirements analysis and documentation.

3.4 Process

The examples included in this section are intended for educational purposes and are not complete and may not represent any existing system. The examples also represent only a single iteration through the design process. During actual Requirements Documentation, several iterations would be required to identify additional models and requirements that are needed.

During requirements analysis phase the problem domain is defined, a model of the domain (or problem space) is produced as the Domain Analysis Model (DAM) consisting of static and dynamic model artifacts. Domain, this case, refers to the business domain addressed by the standard. The following diagram shows the major elements of domain analysis. Each step represented in this diagram is further detailed in a separate sub-section and an activity diagram.

While the activities are in sequence, the process of Domain Analysis is iterative and is not a manifestation of the Waterfall approach to "gather all the requirements before proceeding to design or implementation". As one moves from activity to activity during the Domain Analysis, process there will be situations that require revising or expanding previous artifacts.
Even after this process has delivered the Requirements Specification/DAM to the next process, it will usually be necessary to revisit the requirements when issues are discovered during subsequent HDF processes. There may be missing requirements, ambiguities, inconsistencies, and things that are just wrong. While it is important that requirements be stable before model development begins, it is also important to modify the DAM as issues are discovered .The requirements developed during this phase should be traceable through the design and implementation process.

3.4.1 Business Context Analysis

The first sub-process in the Requirements Documentation process is to analyze specific issues or requirements in the context of the healthcare business process that is to be improved either by developing new software or through HL7-based interoperability. This is accomplished using one or more story boards. A story board is a narrative that describes a representative scenario that illustrates the problem or requirement as well as identifying the interchange of information and the various actors involved.

The purpose of this sub-process is to capture the domain expert's knowledge in a simple fashion and to document the business context for message exchange.

Name	Description
Analyze Business Context	Document Business context for interoperability from the stand-point of end-users. The following diagram details the analysis how the business context, system behavior and Information exchanged between the systems should be documented. The following diagram shows the detailed steps involved in analyzing the business context where interoperability is required.

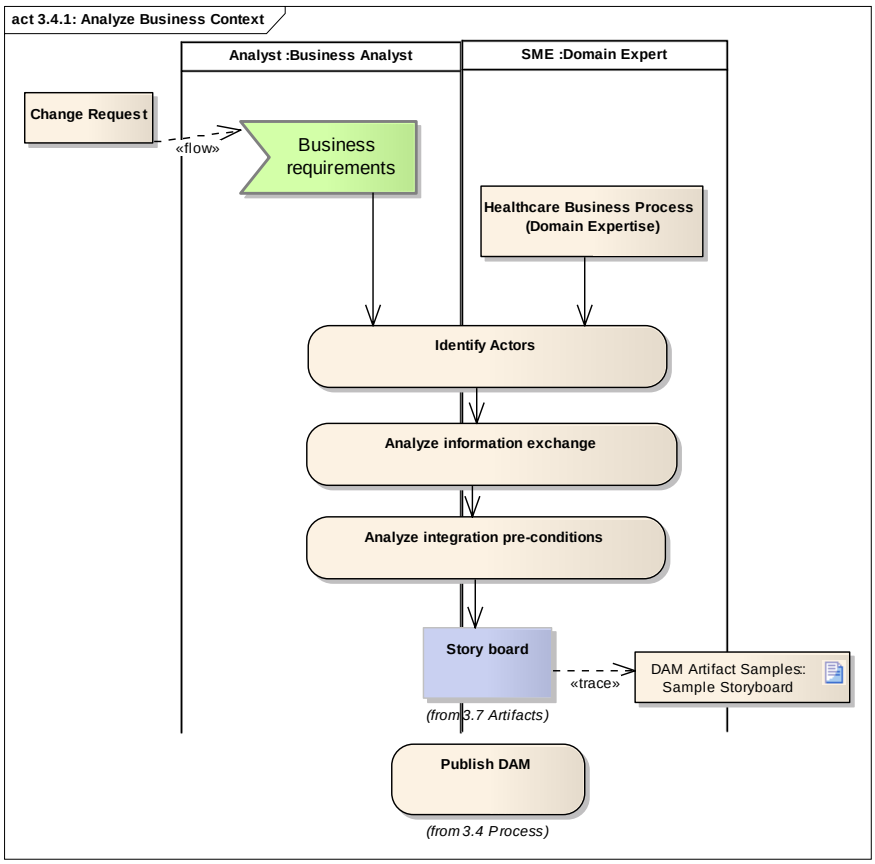

Figure 3.4.1: Analyze Business Context

Name	Description
Change Request	A stakeholder's request to modify or add a standard specification.
Business requirements	The business requirements are derived from the original project charter or on a subsequent change request.
Identify Actors	Identify the actors involved in the business process. This includes human actors, organizations, and systems. One should concentrate on the various roles that may be played by these actors. In some cases, the same person, organization, or system may play multiple roles.
Analyze information exchange	Identify the exchange of information that occurs during this process and the flow of events.
Analyze integration pre-conditions	Describe the conditions under which information is exchanged and the responsibilities of each actor in regards to information sent or received.

3.4.2 Use Case Analysis

The purpose of the Use Case analysis is used to identify the integration scenarios a project or artifact is intended to support.

The Use Case Model formally identifies the Actors and Use Cases illustrated by the Story boards and associates the Actors with the Use Cases they participate in. It enables the project or committee to clearly

identify the functional areas the system will cover and the actors involved. A Use Case Model may consist of multiple Use Cases and multiple Actors.

Each use case within a model provides one or more scenarios that convey how the system should interact with the end user or another system to achieve a specific business goal. Use cases typically avoid technical jargon, preferring instead the language of the end user or domain expert.

Each use case focuses on describing how to achieve a single business goal or task. Actors are parties outside the system that interact with the system; an actor can be a class of users, roles users can play, or other systems.

Use cases treat the system as a "black box", and the interactions with system, including system responses, are as perceived from outside the system. This is deliberate policy, because it simplifies the description of requirements, and avoids the trap of making assumptions about how this functionality will be accomplished.

A Use Case will identify:

- Actors participating in the use case.
- Preconditions
- Flow of Events
- Post-conditions
- Derived Events/Interactions

Activity and State Diagrams further elaborate the Use Case analysis.

The purpose of the following diagram is to describe the process used to identify the Actors and Use Cases in a formalized structure that enables Domain Experts and Business Requirements Analysts to identify the functional interoperability scenarios.

Additional Guidance

- Consider writing use case text only for those use cases whose steps include interoperability to keep the analysis manageable

Guiding Principle: Create the use case model with respect to the system that initiates the information exchange. The system that receives the message of interest is a secondary actor in the use case.

- After writing the use case text, one or more Use Case Scenarios may be written for a particular use case.
- Create Use Case Scenarios for the use case messaging success condition, and for the primary use case messaging failure conditions if there are any and they affect the contents of the message. A Use Case scenario illustrates a single instance of the flow through a use case. It typically does not include any branching or options.

3.4.3 Process Analysis

The process flow show the place of information exchange between systems in the healthcare business process. In UML, Activity Diagrams are used for visualizing the activities and flow of a healthcare business process as described by Use Cases. Use Cases, which refine Story boards to single points of interest, are clarified and expanded into Activity Diagrams. For each Use Case there should be one Activity Diagram.

An Activity Diagram is defined in UML as - "An Activity (Graph) Diagram is a variation of a state machine in which the states represent the performance of actions or sub-activities, and the transitions are triggered by the completion of the actions or sub-activities. It (therefore) represents the state machine of a procedure (or process) itself. The purpose of this diagram is to focus on the flows driven by internal processing (within a system or subsystem)."

The following diagram shows the detailed steps involved in this sub-process.
The purpose of this sub-process is to document how a project or committee may capture the behavior described in the business process in a structured notation (UML) using UML tools.

3.4.4 Information Analysis

One of the most important aspect of requirements analysis is to develop a clear understanding of the business object of interest, their associations, and their attributes.

The purpose of this sub-process is to document the information shared between systems in order to support healthcare business processes. UML class diagrams are used to describe the information required to appear in messages.

Documentation of the structure of a particular business process is done using a combination of an information model, represented as a UML Class Diagram, and a carefully written glossary of the terms used by domain experts to define the static elements within that process.

The information model and the associated diagrams document the static syntactic and semantic relationships of importance in the healthcare business process including the responsible parties/entities and the various data elements/structures required by the process. The semantic meaning of each item and attribute in the information model is described in the model documentation.

3.4.5 Business Rules Analysis

The main purpose of this process is to describe how a committee or project should document additional business rules that are important in creating the message specifications such as business triggers for messages.

The next step in the process is careful description of the business rules associated with the exchange of data that has been identified in the business process. This structure is added to the Activity Diagram using the "object/instance" iconography (in the case of HL7 Specifications, most objects that are exchanged are data objects that have no inherent behavior).

The following diagram describes the steps required to describe the business rules and triggers.

Additional Guidance:

Given that the Information Model, Class and Attribute Glossary, and state charts may document some of the same business rules, avoid duplicating effort and don't document here business rules adequately covered by these other models.
Guiding principle: Focus on those business rules that will be important for the development of the message exchange to be published in the standard.
Business rules may be written in an informal style or in a consistent formalized style. It’s generally advisable to begin capturing business rules in an informal style; a more formal style can be created later,

if needed.
The business rules community has various ways of classifying business rules. If you end up needing many business rules, adopt one of those classifying schemes

3.5 Quality Criteria

- Requirements specifications must be complete, correct, unambiguous, testable, verifiable, traceable, and internally consistent.
- If complete, a requirements specification must specify all information needed to develop the static and dynamic logical and implementation models. It should not include information that is not needed. It must provide the detail necessary to develop the HL7 model specification. For example, a requirements specification “send prescription data to the pharmacy” is not complete. What is the prescription data What is the pharmacy supposed to do with it
- If correct, the requirements specification must correctly specify the conditions and limitations that will be encountered in building the model. How should the model behave if the prescribed medication is not available in the pharmacy
- An unambiguous requirements specification does not leave room for doubt. Data that is exchanged or stored needs to be clearly defined. Message elements that are not well defined leave room for interoperability issues.
- A testable requirements specification is one where each requirement can be examined once the HL7 model is complete to determine if the model does indeed fulfill the requirement. As such, requirements need to be specified an objective fashion. Using UML to represent the requirements for the static and dynamic models will help deal with this issue.
- A requirements specification is consistent as one examines the transitions from lesser to greater detail.
- A traceable requirement is one that can be traced backwards and forward through the Requirements Documentation process.
- Requirements must be internally consistent. One requirement must not conflict with another

3.6 Tools

This section identifies or describes the tools required to develop the artifacts.

Name	Description
UML2 Tool	A UML 2.1 tools may be use to analyze the integration requirements that intended to be met by standard specifications and supports the creation of DAM artifacts outlined in section 3.7. HL7 recommends the use of Eclipse-based UML 2.1 tools that can be extended easily using open-source plugins and features.

3.7 Artifacts

This section describes the structure of artifacts produced during the domain analysis and requirements documentation process.

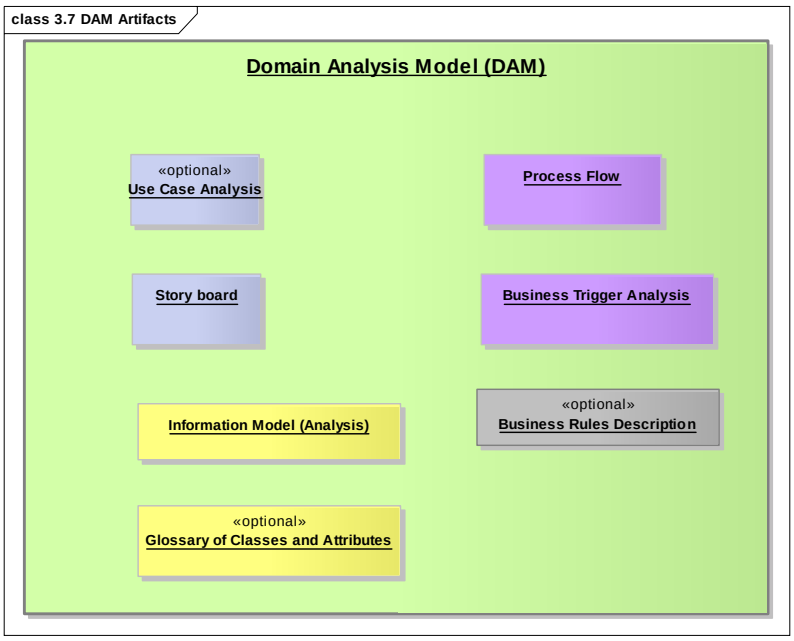

Figure 3.7 DAM Artifacts

Name	Description
Domain Analysis Model (DAM)	A **Domain Analysis Model (DAM)** is an analysis model that describes business process, use cases, process flow, business triggers, and the information exchanged derived from a project's requirements. A sample domain analysis model is available in the Annex A. A Domain Analysis Model is equivalent with a Requirements Analysis Specification but it introduces a consistent notation (UML 2) and style in place of narrative analysis information. Domain Analysis is a well-established discipline in software development that intended to improve the way in which requirements are related from SMEs, to analysts, and integration solution implementers. The artifact describe here adapts the best-practices established in software development to the development of healthcare standards. Having well-documented and thoroughly analyzed the requirements of a standard design specifications, HL7 work groups will be better prepared to respond to ballot comments and questions. **Domain Analysis References:** Carnegie-Mellon University has developed its own methodology known of "Feature-oriented Domain Analysis" that has been applied widely in public/federal and private industry projects. Kyo C. Kang, Sholom G. Cohen, James A. Novak, William E. Hess, and A.Spencer Peterson, "Feature Oriented Domain Analysis (FODA) Feasibility Study", CMU/SEI-90-TR-21, Software Engineering Institute, Pittsburgh, PA, November, 1990 http://www.sei.cmu.edu/domain-engineering/domain_anal.html http://www.sei.cmu.edu/str/descriptions/foda.html http://www.sei.cmu.edu/str/descriptions/deda.html

	Domain Analysis and analysis models are the subject of several books: http://books.google.com/books?id=RxwX6yk72YoC&pg=PA138&lpg=PA138&dq=%22Domain+Analysis+Model%22+Requirements+Analysis&source=web&ots=xZtAsUBswK&sig=padEIDtuGgBwo6YcO2_EEze7uUY&hl=en However, the focus of this methodology is specifying how we use DAMs for HL7 standard development. It is important to mention that a DAM is not only an information model but an analysis model which includes business processes and system interactions. The behavioral/dynamic aspect of a DAM is often overlooked.
Use Case Analysis	The Use Case Analysis describes typical scenarios of end-users interacting with systems for the purposes of sharing or looking up information. In HL7, the Use Case Analysis is a documented as a UML model or package in the DAM that describes a interoperability requirements in terms of use cases. It consists of all the actors of the systems involved in interoperability and all the various use cases by which the actor interact with the system, thereby describing the functional and integration behavior of the systems involved. There are two types of entities identified in the use case approach: actor and use cases. 1. Actors An actor is a person who will use the system we are designing. A bank customer interacting with the software of an automated teller machine is an actor. An astronomer inputting the coordinates of a star to a telescope aiming program is an actor. A bookstore clerk checking the computer to see if a particular book is available is an actor. Usually an actor initiates some operation, although sometimes the actor may act in other ways, such as receiving information or assisting in an operation. In a large project, just identifying all the actors may be difficult. The designer needs to look for people or other systems that: Provide information to the system Need information from the system Assist other actors 2. Use Cases A use case is a specific task, usually initiated by an actor. it describes a single goal the actor wants to attain. Examples are the applying for driver's license, renewing a driver's license, etc. An detailed example of use case analysis is provided in Annex A.
Story board	A Story board is a narrative description of a series of steps involving some exchange of information between different participants to achieve the objectives of a healthcare business process. The list of steps can be in generalized, abstract terms, or in the form of a real-world example. A Story board is intended to illustrate the basic path, simple path, alternate, or error path and intended primarily from the stand-point of the Domain Experts. • Story boards should be written using business terminology to illustrate the business context for the message exchange, functional model, etc. • The content of the initial Story boards should primarily be representative of normal business processes. Avoid exception cases. Attempting to document all the exception cases in a business process can be an exhaustive task that diverts focus

	from the typical case, particularly at this early stage of the requirements process. • A Story board may be imprecise, incomplete, etc., particularly in its initial draft. It may be revised over time if that is deemed important. They typically have no branching or options (e.g. if this condition occurs, then ?). The information in a Story board will typically be made more precise and more general when the corresponding Activity Diagram is created in the next activity. • The Story board may include examples (e.g names of people, organizations, systems, etc. and data values) as appropriate. This guideline helps to make the Story boards illustrative of the real work and also to make clear that items of interest may be of different types than assumed (e.g. that a patient in some cases may be an animal, that a guarantor may be an organization). • In the text, avoid acronyms, abbreviations, etc., because the intended audience is a diverse group, some of whom would likely be confused. If acronyms and abbreviations are deemed important for the intended audience, they can be included in parentheses after the term. For example, Department of Motor Vehicles (DM).
Glossary of Classes and Attributes	The Glossary focuses on the business object of interest and their attributes. This artifact could be generated as a model report. The model describes only the most important classes of objects and the attributes of their classes. It uses business-specific class names and attributes names with unambiguous definitions provided by the SMEs and analysts. The definitions must be comprehensible by the Domain Experts and their end users. To jump-start the modeling process, it's sometimes helpful to think of the events, the relevant parties (persons and organization), places and things that participate in those events, and how they participate in those events; consider as well any "catalogs" of events and things. The data objects from the activity diagram are a primary source for this activity. Some data objects will have been identified in the Message Requirements. It is generally most beneficial to go for breadth first, and depth later. Thus it's typically best to capture the classes and only enough attributes to distinguish them before attempting to identify all the attributes. Similarly, it is typically best to just capture relationships between classes before attempting to document their cardinality (can there be more than one) and optionality (must there be one). Make this model just "good enough". Avoid the temptation to make this model "perfect" or exhaustive. Don't get too wedded to this model. As the process progresses you'll learn more information and the later resulting models will change and improve. While most models benefit from modeling conventions, do not invent new conventions to the detriment of creating the model. Use modeling conventions that do not conflict with the modeling tool used. Using existing quality modeling conventions is more efficient than creating them from scratch. Whenever possible, use the features of the UML modeling tool to create the Domain Glossary. Many UML modeling tools can generate a document of the class and attribute definitions. Review definitions from glossaries of other standards development organizations, industry consortium, dictionaries, etc. whenever feasible. HL7 encourages collaboration with other Standard Development Organizations, industry consortium,etc. and respects their intellectual property. Always reference to the original source in the definition. Having good quality examples from a different domain to use as samples

	frequently helps the process. Creating high quality definitions is not easy for some people. Candidate states for a state chart diagram for an event could be "**active**", "**completed**", "**suspended**", and "terminated". Typically, the activities of creating a DAM and producing a comprehensive Activity Diagram "feed" each other. They may be performed in parallel, by switching back and forth between one and the other, or some combination.
Information Model (Analysis)	The UML models that contains classes, attributes, associations, and packages required to describe the information shared by systems in order to support the business requirements for a standard specification. In this analysis model, local value sets may be expressed as classes called enumerations and assigned to coded attributes. This is not the actual specification of a payload structure (typically, it does not include the order of the data items in the message, the exact formats of the individual data items, or the exact location of the data items within the message, and it might not include the exact data types of the data items in the message, etc.), but rather an analysis of the information required to support the underlying requirements expressed in a way in which SMEs can related to the information, semantics, and structure. In conclusion, this model is not an HL7 static model but an element of the analysis model (DAM). It describes the information exchanged to support a project's requirements (for a project intended to produce interoperability standards).
Business Trigger Analysis	This artifacts documents the allowed states and the transitions that correspond to message triggers. These "state change" triggers are similar to the trigger events defined in HL7 specifications but they relate to events in the "real world". While HL7 has pre-defined a set of state and transitions, the business area analyzed may introduce a different level of granularity. These states and transitions will eventually be mapped or related to HL7 reference states and transitions for each type of Reference Information Model (RIM) class (e.g. Act, Managed Participation, Role, Entity).
Process Flow	The process flow represent a dynamic view of end-user and system interactions and it often the best way to represent the interactions between systems, to derive the application roles, and trigger events. Process flows rely on UML Activity Diagrams to represent the sequence of user and system actions required to fulfill specific business processes. An Activity Diagram identifies a sequence of steps and the information that is transferred from one participating role/actor to another. Sometimes called a "Swim-lane Diagram", the diagrams represent the flow of control among the steps and help identify when information is required to be transmitted to achieve the objectives of the Story board. These diagrams use a consistent notation (UML 2.1) to represent actions, activities, decisions, control, and information flows between users and systems.
Business Rules Description	The business rules may be described using a variety of dynamic views (business process diagrams, activity diagrams) , narrative text, or constraint language statements (e.g. OCL) based on the DAM. These business rules may be used to apply additional constraints to the information

	exchanged or processing rules. . Example Business Rules using Natural Language: • At any point in time, a person can hold a drivers license (active or suspended) in only one state. • In order to obtain or renew a drivers license, a person must have no moving motor vehicle violations in the past three years. • In order to obtain or renew a drivers license, a person must have no health issues that would adversely affect their ability to operate a motor vehicle. • The Department of Motor Vehicles accepts the following payment methods: cash; check with acceptable picture id; credit/debit cards of the following types: Mastercard, Visa.

DAM Artifact Samples

Name	Description
Sample Storyboard	The following is a sample story board for Driver's License renewal. **Example 1**:" Darlene Dobson goes to the local office of her state Department of Motor Vehicles to renew her driver's license and fills out a Drivers License Renewal form. The license clerk requests her current drivers license and using the drivers license number, S057864389, inquires on the status of her current drivers license, sees that it is active but scheduled to expire in three weeks, verifies that Darlene's date of birth and current address are the same as the data currently in the system, enters Darlene's responses from the form indicating that she has not been convicted of a moving motor vehicle violation in the previous three years and that she does not have any health issues that affect her driving ability. After entering that information the system sends an inquiry to the state court system for moving motor vehicle violations in the previous three years and after receiving a response that there have been none, requests payment information. Darlene indicates she would like to pay with a MasterCard debit card and gives the card to the license clerk. The clerk enters the MasterCard number and expiration date. The registry system sends an electronic request to MasterCard to have the $35 driver's license renewal fee debited from Darlene's account and credited to the Department's bank account. After receiving an acceptance response from the MasterCard system, the registry system prints Darlene's new driver's license good for three years and a payment receipt. Given that the entire visit only took 10 minutes."

4. Specification Design Process (SDP): Design and Harmonization

4.1 Overview

This chapter describes the process by which HL7 specifications are designed. The requirements specification and any mappings to reference models are input to the specification design and packaging

process. Existing specifications from earlier or concurrent design activities are also input to this process. The result from this process is a set of one or more of the following artifacts:

Information Models (Harmonized)
Dynamic Models(Harmonized)
Functional Models
Vocabulary Specifications

The artifacts produced during this specification design process are intended to be balloted as standards (e.g. informative, normative, and draft for trial use) however, some projects may simply publish their artifacts as reference specifications.
Some specifications are produced as further refinements or derivative works based upon earlier design specifications. All specifications are produced in response to requirement specifications and make use of the HL7 reference models and earlier design work.

The following is a high-level process overview:

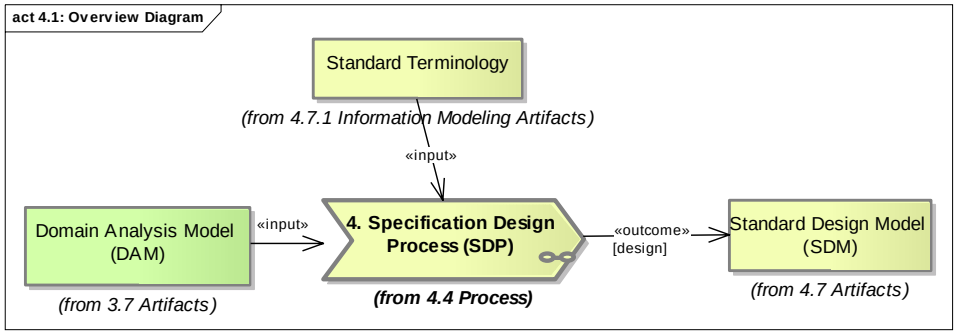

Figure 4.1: Overview Diagram

4.2 Context

HL7 produces a wide variety of specifications. Each specification is designed to assist the users of HL7 standards in overcoming barriers to information system interoperability. HL7 specifications include, but are not limited to, specifications for information exchange / interoperability for the collection, archiving, aggregation of any clinical, administrative, and/or financial data/information that is relevant to a given healthcare stakeholder.

The context in which a specification is defined will determine how the refinements to static model, dynamic models, and vocabulary domains are applied.

The following diagram illustrates how the result of requirements analysis (Domain Analysis Model) is used to develop design information models.

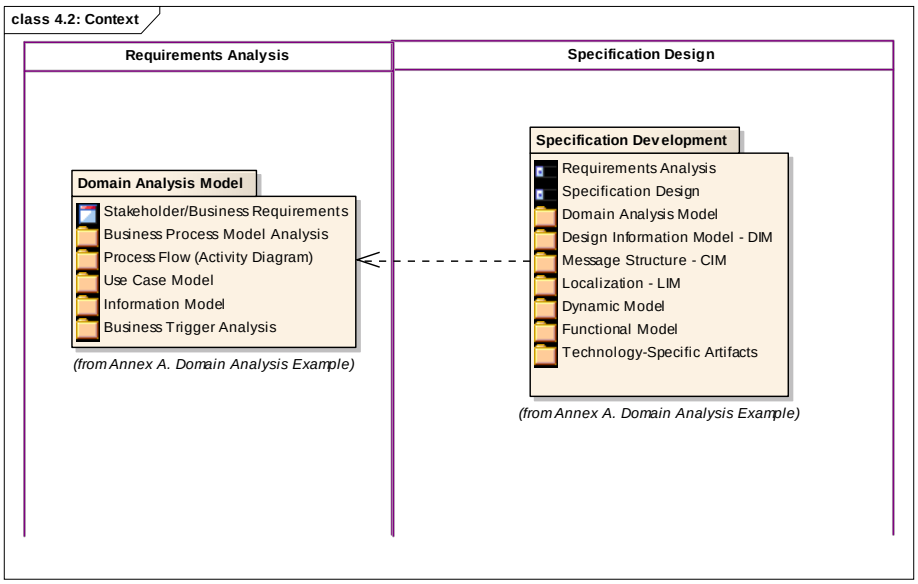

Figure 4.2: Context

4.3 Roles and Responsibilities

This section describes the various roles involved during the development of specification design and modeling. It may be possible for the one individual to fulfill several roles if the skills and schedule allow it.

Name	Description
Affiliate	HL7 Affiliate organization or consortium that creates designs artifacts localized for a locale or consortium.
Committee Stewart	Person that represents a project or committee in regards to reference model harmonization requests.
Business Analyst	This roles requires knowledge of the HDF and domain expertise This person is responsible for collecting interoperability requirements analysis and seeing to their inclusion in the standard specification. This role requires knowledge business rules surrounding the process that is the focus of the specification. The analyst is an individual skilled in the use of the artifacts produced during requirements analysis.
HL7 Modeling Facilitator	The HL7 Modeling Facilitator is knowledgeable in the HDF, knowing the processes that must be performed to produce an HL7 Requirements Specification. This person is responsible for guiding the development of the standard specification. HL7 Modeling facilitators will that the proper use of the HDF is done consistently across domains and standard specifications.
Work group or Project Team	The members of the work group (TC or SIG) or project that are involved in validating the contents of design specifications for HL7 standards. Work group chairs are typically involved in validating domain-specific requirements

	and refinements are correctly represented in a harmonization proposal or design specification.

4.4 Process

During specification modeling reference models are constrained into design models through a process of iterative refinement driven by requirements specifications and following specification design rules, conventions, and guidelines. The primary deliverable produced during specification modeling is a set of specification design models and implementation artifacts.

Design models for static information (based on Domain Analysis)
Design models for dynamic integration (based on Domain Analysis)
Reusable design model components
Construct implementation artifacts (e.g. source code stubs, XML Schema Definitions, Web Services Definition Language, etc.)

During specification modeling reference models are constrained into design models through a process of iterative refinement driven by requirements specifications according to design rules, conventions, style guide, and guidelines. The primary deliverable produced during specification modeling is a set of standard specifications and other standard artifacts.
The following diagram provides an overview of the processes and sub-processes involved in specification design.

4.4.1 Information Model Design

A Design Information Model (DIM) is derived from the Reference Information Model (RIM). It is comprised of classes that are derived from RIM classes. The DIM provides a solution to the information requirements of a particular problem domain. The mapping of the requirement's domain information model to the RIM is used to identify which RIM classes need to be included in the DIM. In some cases it may be necessary to include multiple clones of the same RIM class. Each class derived from a RIM class is given a unique name that is representative of its business use.

Note: Section 4.7.1 provides detailed descriptions of each artifact referenced in this process description.

The development of the derived Information Models is done through a process of restriction, constraint, and extension.

The following diagram describes the details of designing HL7 standard information models. The generation of the artifacts mentioned in this diagram is not done sequentially, but rather iteratively; these steps can be worked sequentially, parallel or concurrently depending on how the analysis is approached .

Note: Model traversals such as Hierarchical Message Descriptions (HMD) are not in scope for this methodology. An information model is traversed in order to generate a technology-specific specification. For the purposes of design, only one representation of information models is required.

4.4.2 Dynamic Model Design

The integration behavior of systems is defined iteratively by refining the collaborations and interactions identified during requirements analysis.

Note: Section 4.7.2 provides detailed descriptions of each artifact referenced in this process description.

4.4.3 Design Harmonization

Harmonization of design models with HL7 Reference Models is a formal collaboration among the technical committees that have steward responsibilities for domains. There are two levels of harmonization, domain specific and extensions to the HL7 Reference Models.

Note: Section 4.7 provides detailed descriptions of each artifact referenced in this process description.

The process documented here is not completed sequentially, but rather iteratively and interactively; these steps can be worked sequentially, parallel or concurrently depending on how the analysis is being done Joint design is conducted through Working Group Meetings,
held three times per year or Harmonization Meetings, held three times per year between regular working group meetings.

The following diagram describes the details of harmonization for HL7 design models.

4.4.4 Design Localization

The following describes the steps undergone to localize information model designs (including terminology) for a specific locale. The following describes the steps required to localize an information model by an affiliate or consortium.

4.4.5 Design Publication

The specification design publication should use the UML or other model artifacts to generate model reports (documentation in "book" form) and a web publication (html files for on-line publication). The following describes the publication process as an automated report and web publication generation.

Note: Section 4.7.3 provides detailed descriptions of each artifact referenced in this process description.

4.5 Quality Criteria

This section describes the quality criteria that have to be met by an HL7 specification in order for that specification to be both relevant and complete. These criteria are intended to lead to the development of interoperability standard with immediate applicability and ready for implementation by the healthcare industry. This section is not intended to address the conformance of a given to an HL7 specification but to address those criteria that make a specification viable.

4.5.1 Completeness

The specification must have the level of detail required for consistent implementation. A complete HL7 specification must be complete in that it must provide the following either implicitly or explicitly:
- Clear definition of dynamic model addressed by each specification that has a system behavior component such as messaging, APIs, and functional specifications.

Complete static model definitions for all the interaction inputs and outputs. This will include:

Complete class descriptions
Complete attribute descriptions
Complete association descriptions
Complete interface/application role descriptions
Complete operation descriptions

Alignment with HL7 references

Currently, RIM alignment assumes that any information models developed in HL7 will be based on the Reference Information model and follow any structural vocabulary specifications.
HL7 specifications must conform to the HL7 RIM; they should not diverge from it. Any divergent requirements deemed valid will have to undergo the process of Harmonization.

Interaction triggers must be aligned with the HL7 state charts for RIM Foundation classes (Act, Managed Participation, Role, Entity state charts).

4.5.2 Consistency

The specifications shall use consistent styles, glossary, and methodology regardless of domain/service profile.

Specifications shall address versioning and backward compatibility even if they are not supported
Specifications shall be reviewed for completeness and consistency with the HL7 Version 3 style guide requirements (e.g., self-standing?no references to non-existent content, no missing sections, term usage consistent with the glossary, etc).
Specifications shall be maintained to be consistent with the evolving HDF.
Where feasible, a service or functional specification must reference HL7 messaging interactions that are comparable or compatible in both use and functionality.

4.5.3 Quality Validation

HL7 Modeling facilitators and committee chairs play a very active role in providing active quality verification throughout the development process.

The following process will be used by each team/subgroup to validate the HL7 specifications they develop:

Formal Peer Review of the Submitting TC involving chair(s) and Modeling Facilitator is conducted
Resolution of Submitting TC peer review comments
(Optional) Internal committee peer-review to ensure the quality checks.
Assign formal quality roles for Review
Conduct Formal Peer Review of Release Candidate
Resolve identified Peer Review Comments
Approve release recommendation to committee for ballot

4.6 Tools

HL7 projects are using a variety of tools to develop standard design specifications.

Information models and messages are designed using the Visio R-MIM designer.
Business process models and functional specifications are documented using documents and spreadsheets
Clinical Document Implementation guides are documented using a set of narrative and computable constraints (Schematron).

Name	Description
Rose-Tree and RIM repository	The RoseTree is a Visual Basic (VB) application that functions as an interface to the HL7 repository; provides a browser for the HL7 RIM and Vocabulary; builds CIMs from Visio designs and manages the repository storage of these artifacts: http://hl7projects.hl7.nscee.edu/projects/rose-tree/ This tool uses the design RIM repository (Access database): http://hl7projects.hl7.nscee.edu/projects/design-repos/
Visio R-MIM Designer	Visio add-in component that supports custom HL7 shapes and extensions. This tools is widely used for information model design. http://hl7projects.hl7.nscee.edu/projects/visio-rmim-desi/

4.7 Artifacts

Name	Description
Standard Design Model (SDM)	A standard design model may be a functional model, a dynamic model, an information model produced by an HL7 project to resolve specific interoperability requirements.
RIM Harmonization Request	Harmonization requests document any changes to the RIM or to the HL7 structural vocabulary requested by a project team.
Runtime Artifact Design	These are description files or code stubs based on the contents of standard design models. The HL7 data type implementation technology specification defines constraints and operations for data types and describes how data types are to be represented using a certain technology (e.g. XML, Java, WSDL). Section 6.7 documents the current state of runtime artifacts generated from HL7 standard design models.

4.7.1 Information Modeling Artifacts

Design Information Model (DIM)	The Design Information Model contains the classes, attributes, and value sets for coded attributes for a the current domain. It may or may not re-use classes defined in other clinical or administrative information domains. The DIMs contain specific Focal classes (those classes identified to be “of interest” in the requirements analysis phase). Objects instantiated from these

	classes have the distinction that their state transitions trigger message transmissions and are correlated to higher-level business/application triggers. E.g. "ActGenericReference" is indicated as the focal or entry class for this domain.
Localized Information Model (LIM)	This information model design is based on a standard Design Information Model and meets the locale-specific requirements for structure and terminology.
Constrained Information Model (CIM)	Constrained Information Model; i.e. (a sub-set of the DIM) - packaged design information model clones in a logical fashion introducing new clones and additional refinements as needed. It may or may not re-use classes defined in other clinical or administrative information domains.
Annotated CIM	Annotated CIM are intended to replace Hierarchical Message Descriptions if the way a model is traversed is important for creating other design specifications.
Standard Terminology	A vocabulary domain consists of a set of concepts, not a set of words or codes. In different contexts, the same concept could be represented using different coding systems. Thus, each concept in a vocabulary domain has a one-to-many relationship to codes that might be used as representations for the concept in a message instance. Vocabulary Domains apply to coded attributes of RIM classes as follows: Structural Vocabulary Domains apply to those coded HL7 invariant such class type, mood, status, etc. Structural codes are part of the HL7 syntax and thus may not be constrained routinely during the process of developing specification. These codes may be constrained or extended only if the changes are also subject to harmonization. The Extensibility Qualifier (coding strength) for these domains is specified as "Coded, no extensions" (CNE) Clinical Vocabulary Domains apply to those coded attributes that describe healthcare coded concepts and not syntax. Depending on the context, Extensibility may be set to "Coded with extensions" (CWE) or CNE. External Vocabulary Domains apply to those coded attributes that are based on an externally defined Code System which is simply referenced by HL7. The context in which a vocabulary domain is used specifies the extensibility rules for a domain. If a domain allows values other than those specified by its value sets, the extensibility qualifier will be "Coded with extensions". Otherwise the value sets must be used as specified and the extensibility qualifier must be "Coded, no extensions". CS (coded specifier) data types are by default CNE. If a domain is specified as "Coded, no extensions" (CNE), any of its restrictions must maintain the same extensibility qualifier.

4.7.2 Dynamic Modeling Artifacts

HL7 standards must specify system interfaces and provide clear views of how the interfaces are used to support the underlying requirements. These dynamic views/diagrams are very important to communicate intended system behaviors to user and implementers of HL7 standards.

Name	Description
Functional Model	The functional model currently supported by the HDF refers specifically to the Electronic Health Record System (EHRS) Functional Model. The proposed functional model is based on two axes: functions and care settings. The functional axis is a hierarchy of the essential, desirable and optional EHRS functions across all care settings, with functions organized into care setting and infrastructure categories. Each care setting (e.g., inpatient, ambulatory) has an accompanying normative profile to define how it uses the defined functions and identifying any care setting specific functions.
State Transitions	The state transitions of focal objects are directly related to the trigger events and focal classes. A Focal Class is a special domain class whose state changes are directly responsible for a trigger event (transition-based trigger). Transitions (edges in state charts) map one-to-one to interaction triggers and thus they are very important to the standard design specifications. The components of a State Transition Diagram are States and Transitions. A transition is a named, directional association from a state instance, termed the source state, to a state instance, called the target state. States may be nested with the "super-state" specified as abstract This document treats state transitions, and their component states and transitions, as independent concepts that can be referenced by other dynamic model objects. State transitions do not carry artifact identifiers; instead, state transitions are bound to a single static model class (a focal or entry class of a DIM). **State transitions:** A state transitions is a directed graph consisting of states (as nodes) and transitions (as directed edges). Each state represents the state (sometimes called status) of a healthcare object. Each transition represents a potential change of state in that healthcare object. In HL7, each healthcare object is represented by a focal class, defined in the static model, which is derived from one of the following RIM classes: Act, Entity, Role, or Managed Participation. Valid states and valid transitions for each of these four root classes are defined in the RIM. When a domain committee defines a class that is derived from one of these root classes, it is allowed to define a refinement of the states and transitions that apply to that class. The resulting class-specific state transitions identify the valid states and transitions for each healthcare object having that class as its focal class. State transitions reference information specified in the static model or in the RIM which specify the allowed state transitions for RIM classes using state-transition diagrams. 1. State-transition diagram for Act class 2. State- transition diagram for Entity class 3. State- transition diagram for Role class 4. State- transition diagram for ManagedParticipation class
Interface Specification	This artifact consists of UML interfaces and operations that use CIM and LIM structures as input parameter (request), return types (response), and error status information (if applicable). System interfaces are also known as "application roles" or "service interfaces". The interactions subsumed by an interface correspond to

	operations. Within the process, HL7 has primary responsibility for identifying the functional requirements, mapping those requirements to the EHR Functional Model, enumerating the valid information for the services, and establishing conformance criteria. Based on the HL7 Development Framework (HDF), the HSSP Service Development Framework is a methodology for producing service specifications for Healthcare. The following describes how Service specifications leverage the based Design Models for Behavior. Interfaces are used to describe a the capabilities provided by a system (or a certain type of system).
System Interactions	This artifact consists of UML sequence diagrams that describe the interactions between systems based on the process flow documented in the DAM. This activity assumes the use a set of standard conventions regarding naming of objects and the depiction of inter-object linkages and associated message. The messages or operations in the sequence diagram is intended to reference the operations specified in the interface/system role. An interaction is the concept that ties together static models and application/system behavior. A set of interactions organized in application roles characterize the behavior of healthcare applications. An interaction will be sent by a sending application and received by a receiving application. Each interaction is related to one or more sending roles, one or more receiving roles, a trigger event, and a set of constrained serialized information models. Interaction is identified by an artifact Id, a formal name, and a description. In addition, each interaction will have relationships that identify potential sending and receiving application roles, a unique trigger event whose firing causes the interaction to be sent, the receiver responsibilities of the receiving application role and a single constrained information model with optional bindings to other constrained information models which together define the content which is transmitted by the interaction. The root constrained information model always has a focal class derived from the RIM Transmission class. This static model is frequently bound to other static models which might define the Control Act for the interaction (which conveys data related to the trigger event), as well as query definitions, response payloads, etc.
Sending process	This artifact document the steps preceding the message exchange. This may include an end-user need for information or an end-user action (e.g. co-signing a pharmacy order). The sending process describes the pre-conditions for a set of system interactions using a graphical notation (UML 2) in order to remove any ambiguity and improve the quality of a standard design.
Receiving process	This artifact documents how a notification, request, or query is processed by the tracker or fulfiller system. These receiver responsibilities specify different choices and responsibilities a receiving application may have upon receipt of a message. Some interactions, such as notifications, will have no receiver responsibilities. In general for request interactions, the receiving application may accept or reject the incoming message. If rejected, a reason for the rejection is normally required. If accepted, the receiving application may be required to make modifications to healthcare objects under its care, and additionally may be required to respond to a request either by confirming the state of something, promising to do something, reporting that something has happened, answering a query, scheduling an activity,

	or delivering the results of an observation on a sample. All such potential requirements are modeled as a collection of potential receiver responsibility instances. Each individual responsibility is modeled as having a guard predicate that will be evaluated at execution time. The assumption is that exactly one guard predicate will evaluate to true and the responsibility associated with that guard becomes the required receiver responsibility of the receiving application.

4.7.2.1 Sample Diagrams for Dynamic Model Design

4.4.2.1(a): State Transitions

The following are example state transitions that may trigger notifications or may be initiated by requests to effect state changes in a focal class.

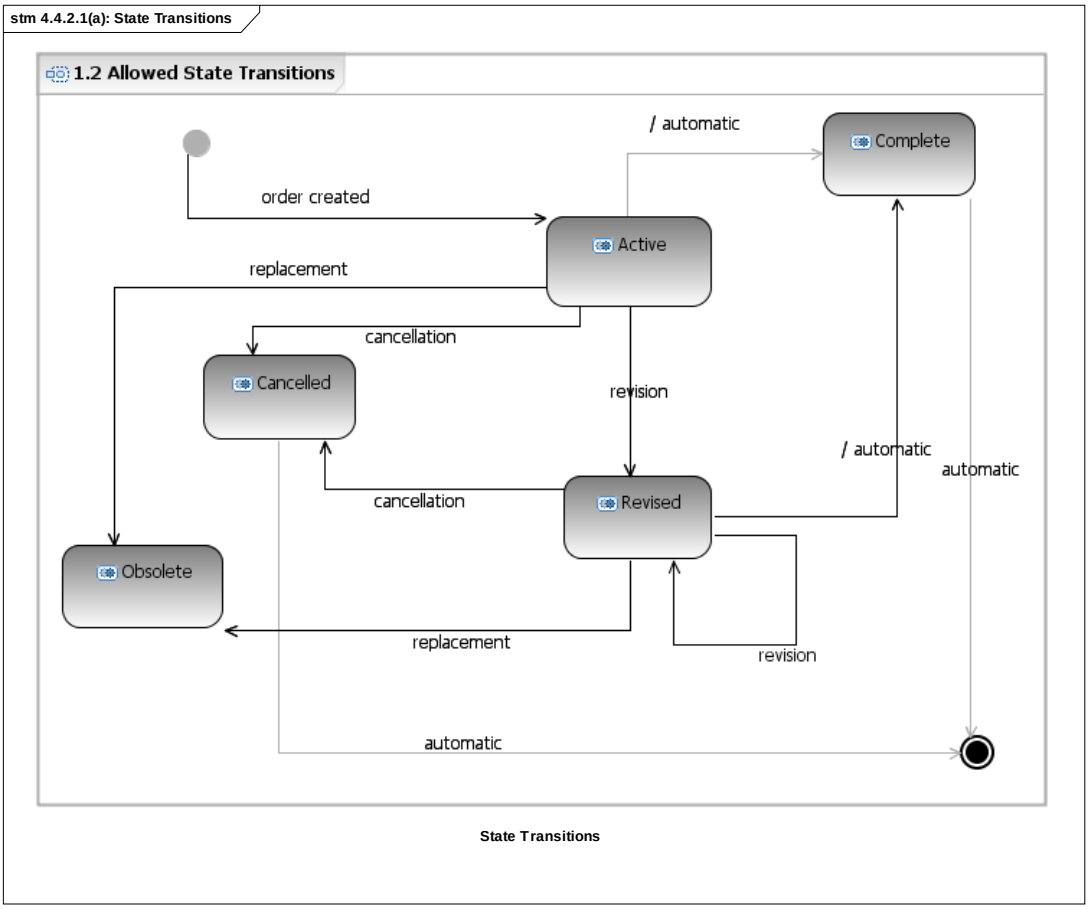

Figure 4.4.2.1(a): State Transitions

4.4.2.1(b): Sending process

The following illustrates the end-user activities that trigger a new order request.

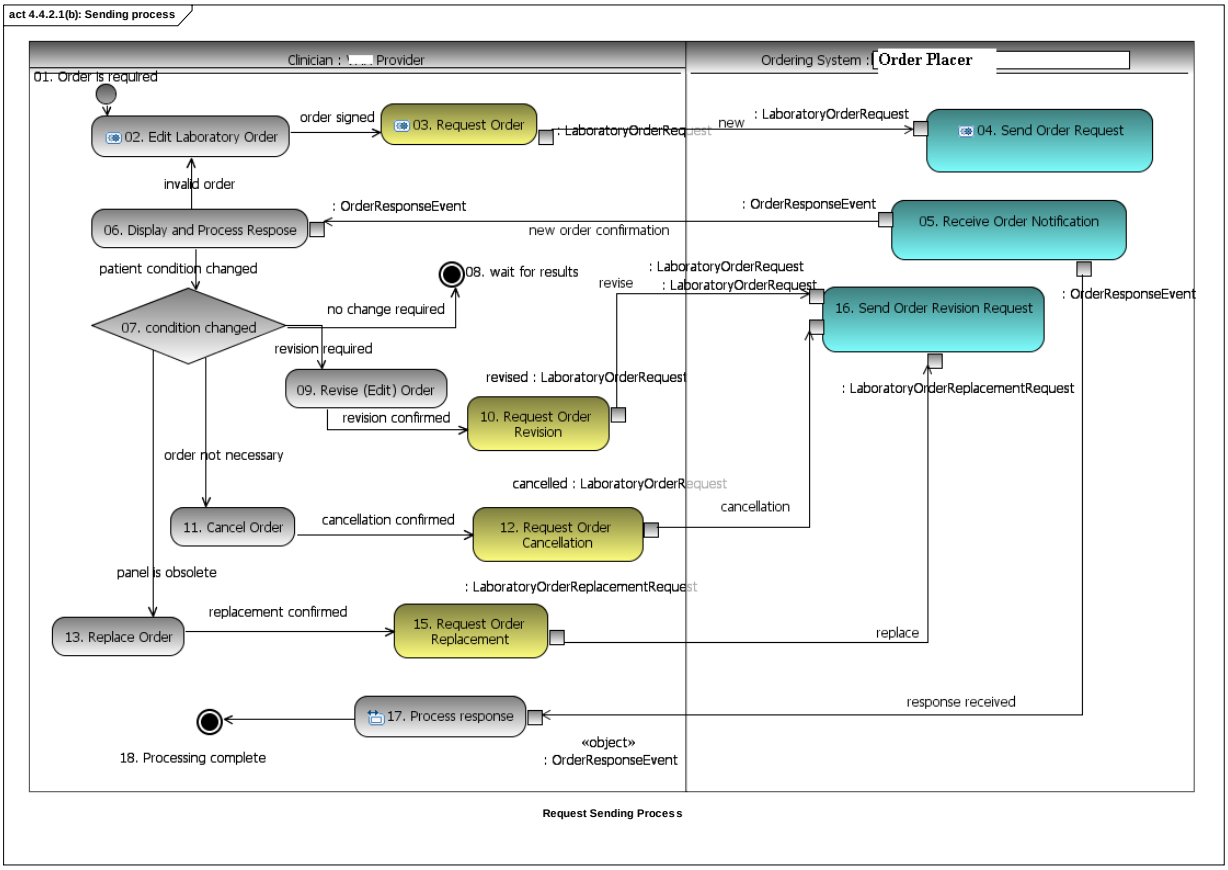

Figure 4.4.2.1(b): Sending process

4.4.2.1(c): System Interfaces

The following demonstrates the use of class diagram to visualize the operations supported by a certain system role (or interface). In this case the system role (or interface) is "LaboratoryResultInformer".

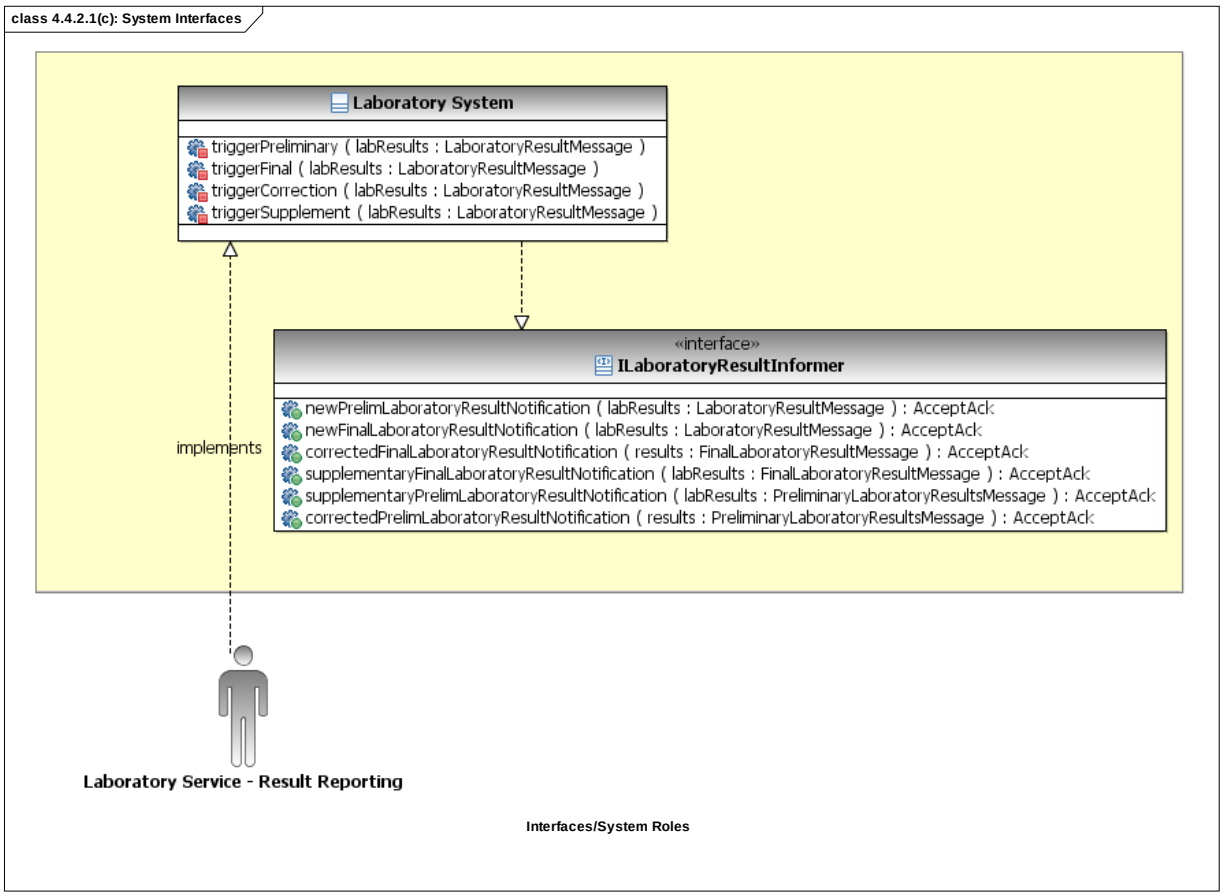

Figure 4.4.2.1(c): System Interfaces

4.4.2.1(d): System Interactions

The following is an example sequence diagram used to specify system interactions. As seen in this diagram, the prior order request is a pre-requisite for a status change notification.

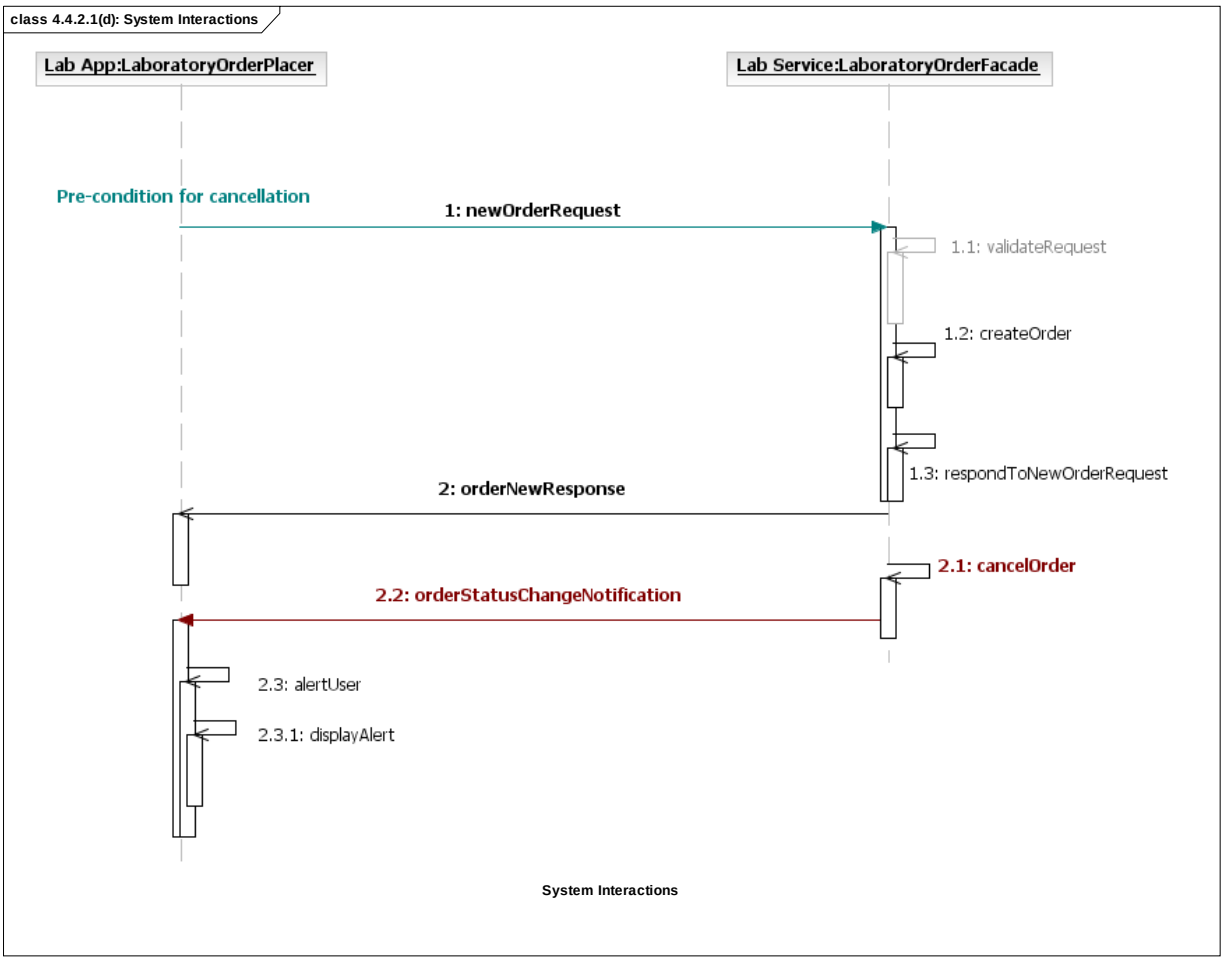

Figure 4.4.2.1(d): System Interactions

4.4.2.1(e): Receiving Process

The following activity diagram is describing the receiver's responsibilities to process an inbound message.

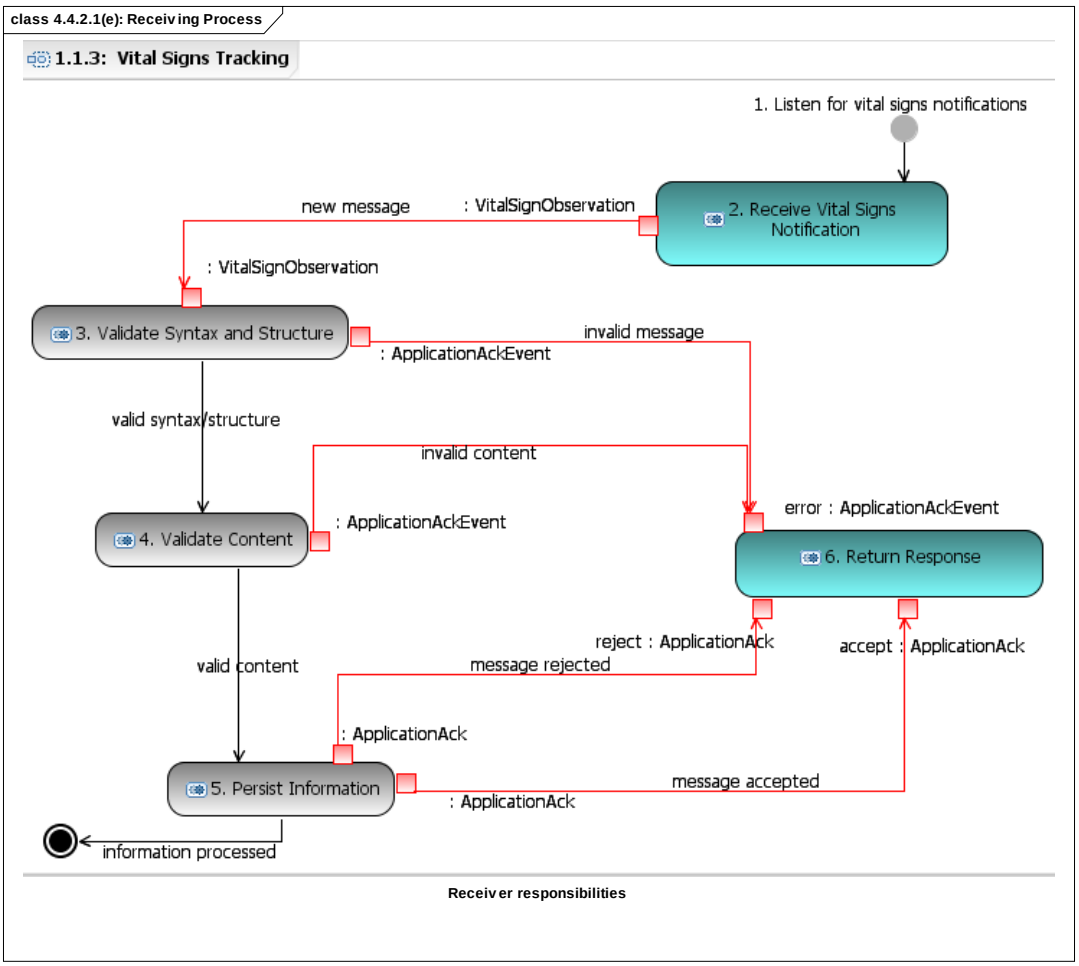

Figure 4.4.2.1(e): Receiving Process

4.4.2.1(f): Healthcare Services Specification

The HSSP Service Development Framework is a methodology for producing service specifications for Healthcare. The following describes how a service functional specifications can leverage application roles and interaction to specify service interfaces. HL7 work groups have a responsibility to relate their artifacts to HL7 reference information and behavioral models. This diagram demonstrates how service functional specification can be related to a set of application roles and interactions.

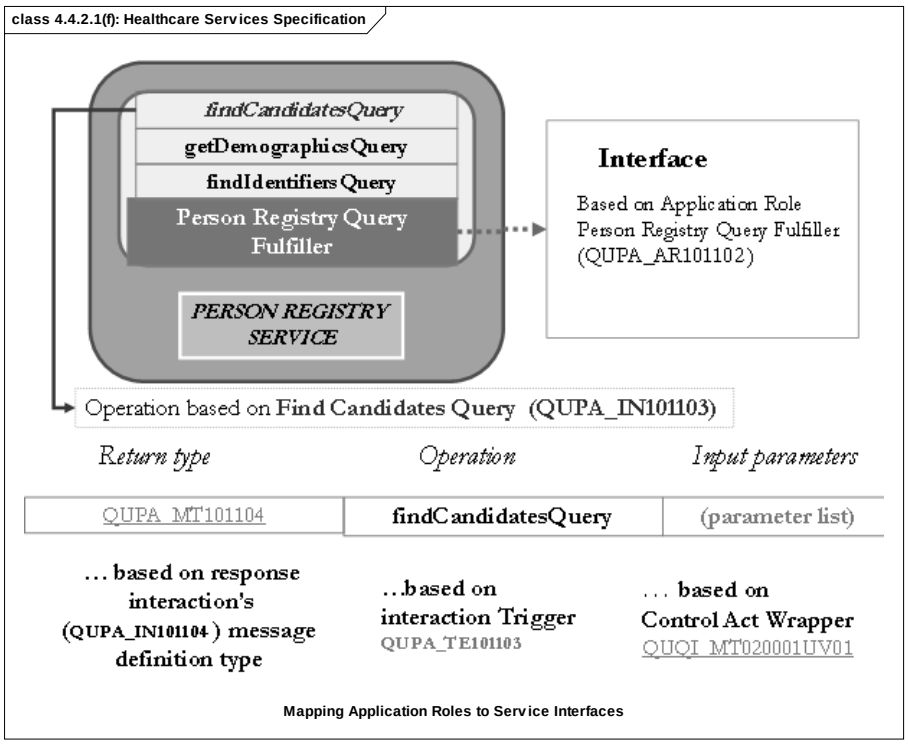

Figure 4.4.2.1(f): Healthcare Services Specification

4.7.3 Publication Artifacts

Standard Design Models must be presented to end-users for review and ballot, therefore the modeling tools used to create the standard designs must also produce readable end-user documentation, reports, and even web site content.

Name	Description
Model Report	Book-form (e.g. Word document) user document describing the design specification.
Web Publication	HTML publication of the specification design. This specification is intended to be viewed using a Web browser and published on the HL7 web site.

5. Standard Profiling Process (SPP): Constraints, Extensions, and Annotations

Implementers are expected to use a common process of constraining and clarification of the standard artifacts in order to leverage the standard specifications developed by HL7.

NOTE: This chapter is under development.

5.1 Overview

The HL7 standard provides a rich set of constructs to support communications in a variety of health

endeavors. The standard faces two challenges:

1) it can always be made more specific in order to provide a more precise solution to a particular requirement.
2) it will not contain all of the data needed in every environment, particularly when international requirements are considered.

These challenges lead to a pair of complementary requirements: the ability to constrain the standard in more detail, and the ability to extend the standard in a controlled fashion. To support the needs of implementers we need two critical capabilities:

the ability to establish a testable statement of conformance to the standard, and
the ability to create a "profile" of the standard that formally defines how the standard will be implemented in a particular setting.

The profiling and conformance statement declaration require refining the standard through constraint and extension, a definition of constraint and localization profiles, well-defined criteria for establishing a conformance statement, and principles guiding how HL7 standards may be extended. Just as the standard is developed using a series of constraints, the design of applications to implement these constructs, the definition of conformance claims, and the definition of localization specifications, all require further constraint and extension of the base standard.

During specification profiling, specification models are further refined and specifications further constrained following the same set of design rules, conventions, and guidelines used in the development of the specification to produce a profile of the specification for use in a particular environment by a defined community of users. The primary deliverable produced during specification profiling is a set of specification profiles and conformance statements.

1. Identify profile stakeholders/community of interest
2. Further refine, constrain, or extend specification design models
3. Publish specification profile and maintain the specification(s) over time

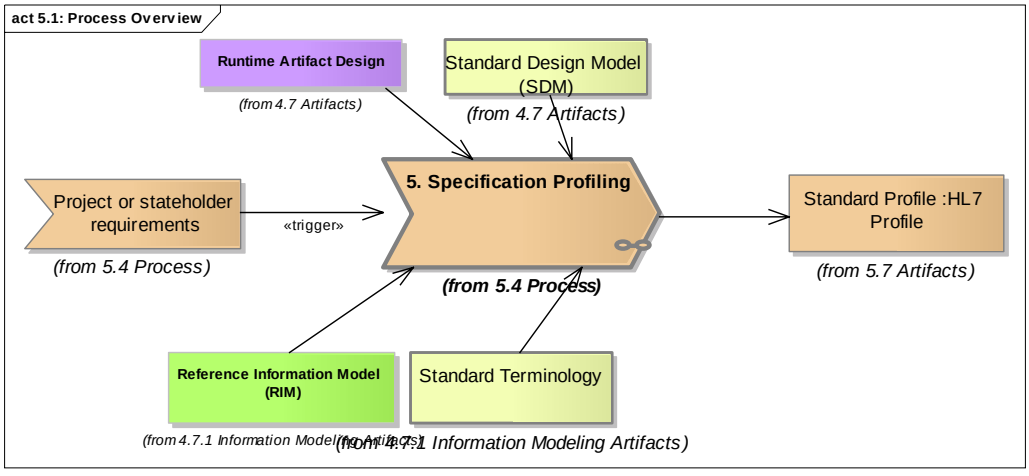

Figure 5.1: Process Overview

5.2 Context

The processes are used to define profiles of standards and associated conformance statements, to be used for implementation. These profiles have a variety of anticipated uses that include:

An HL7 International Affiliate organization to specify the localization of the standards to be used in the affiliate's region
An application developer to document the conformance of their application to the HL7 specification
An organization or individual to document the requirements of their environment prior to searching for an application.
A user to document the specific implementation of the standard user's system(s) or interoperability solution

5.2.1 Conformance using Profiles

This section presents the methodology for producing a precise and unambiguous specification called a "profile". Implementations that adhere to the constraints of a profile are said to be conformant to the profile. For conformance to be measurable, the profile must specific in regards to the following:

1. The information exchanged by systems in order support the end-users' requirements
2. The implementation technology in which the information is exchanged
3. The dynamic behavior (triggers, input, and operations) and acknowledgement responsibilities of the systems

A conformance statement is a claim that the behavior of an application or application module agrees with the constraints stated in one or more profiles.

The following is showing the relationship between a standard conformance statement, a profile, and the base specification on which it is based and it profiles.

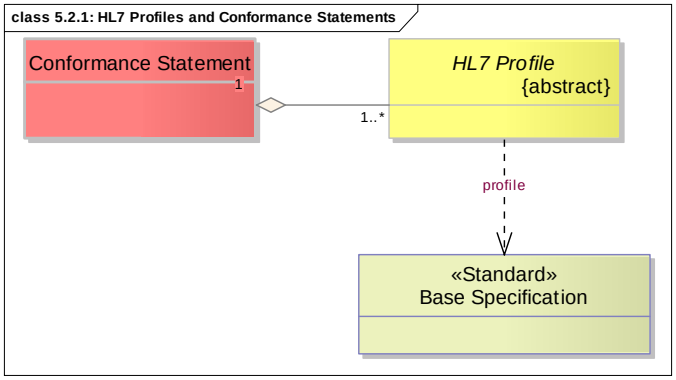

Figure 5.2.1: HL7 Profiles and Conformance Statements

Name	Description
Base Specification	Standard specification that is the subject of profiling according to specific integration requirements.
HL7 Profile	An HL7 profile is an unambiguous specification of one or more HL7 standards that have been analyzed for a particular use case. It prescribes a set of precise constraints upon one or more standard HL7 artifacts. An HL7 profile is conformant, in all aspects, with the HL7 defined specification used in the profile according to the constraints or extension rules. It may specify constraints on the standard HL7 definition. An implementation profile fully describes an interoperability interaction between two or more systems through the combination of the following:

	1. one use case analysis, 2. one or more dynamic definitions, and 3. one or more static definitions.
Conformance Statement	A conformance statement is a claim that the behavior of an application or application module agrees with the constraints stated in one or more profiles. A Conformance Statement is documentation of the degree to which a particular application conforms to the specification. Part of that document will be a profile expressing the requirements relevant to a particular standard. Standard Profiling is based upon the consistent application of constraints to a set of base specifications. This document outlines the processes that govern the definition of profiles and conformance statements.

5.2.2 Profile Types

The following describes the types of profiles allowed for HL7 standards.

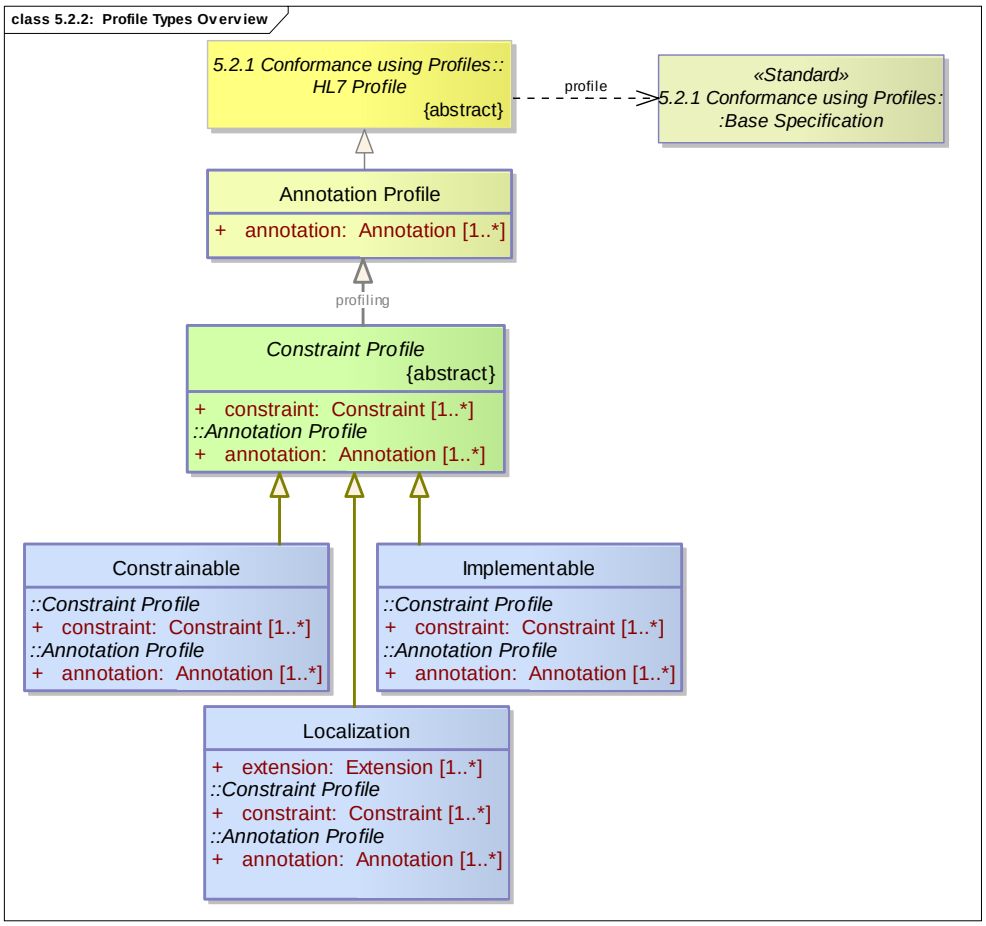

Figure 5.2.2: Profile Types Overview

Name	Description	
Annotation Profile	Annotation profiles are exactly the standard but with more information; an annotation profile will usually enhance the descriptions of the elements of the base specification in order to relate them to a particular context	
Attribute		**Notes**
annotation Annotation Public Collection [1..*]		
Constraint Profile	Constraint profiles may contain unchanged, annotated, and constrained elements such a constraint profile reduces the uncertainty of the base profile in order to make the specification more exact and to approach "plug-and-play" interoperability. Depending on the level of constraint, these profiles may be Constrainable or Implementable.	
Attribute		**Notes**
constraint Constraint Public Collection [1..*]		
Constrainable	Constrainable profiles may be developed for a vendor or a certain context of use with the explicit intention to allow further constraints based on site-specific or deployment models. A constraint profile is an expression of local constraints to be applied to a formally specified HL7 message type or a constraint thereof. Every instance message that is valid for the message type defined by the profile must also be valid for the base message type.	
Localization	Localization profiles may contain unchanged, annotated, constrained, and extended elements; a localization profile is usually designed to meet the same objectives as a constraint profile, except that the demands of the intended implementation context mandate additional elements, as represented in the extensions. Typically, extensions will represent the minimum changes to the base specification needed to meet the requirements of an affiliate's domain.	
Attribute		**Notes**
extension Extension Public Collection [1..*]		
Implementable	Implementable profiles are consist of standard specifications constrained such that their runtime artifacts and implementation technology-specifications are fully implementable.	

5.2.2.1 Profiling Methods

This section identifies the types of constraint methods used to profile a standard for implementation or localization. The following diagram describes the relationship between profiling techniques.

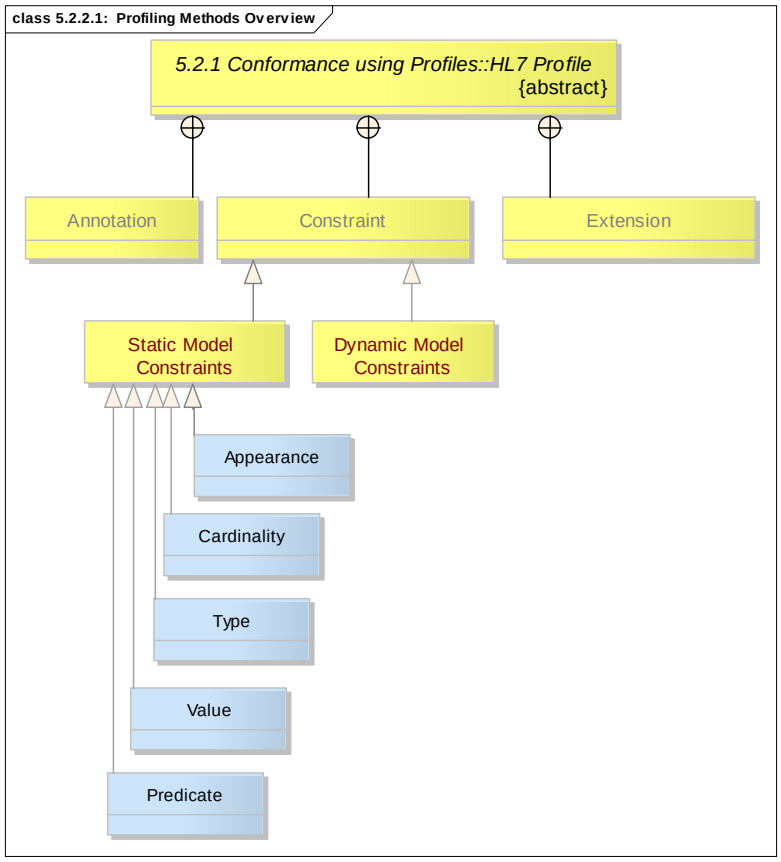

Figure 5.2.2.1: Profiling Methods Overview

Name	Description
Annotation	An annotation is a description or documentation used to enhance the descriptions of the elements of the base specification in order to relate them to a particular context
Constraint	A limitation or restriction applied to a standard in the process of profiling.
Extension	A local extension applied to a standard in the process of profiling. Extensions are supposed to be submitted to HL7 as an enhancement request in a future ballot.
Static Model Constraints	Static model constraints are addressed in detail in the "Constraints, refinements, and localization" specification: http://www.hl7.org/v3ballot/html/infrastructure/conformance/conformance.htm. Developing a Constraint or Localization profile will involve the following constraints against the constrained design models, as required.
Appearance	Mandatory Declaration & Conformance Indicator that refer to the visibility/appearance of a model element,
Cardinality	This type of constraint restricts the number of repetitions that may occur for an element.
Type	This type of constraint allows type substitution and or a type refinement.

Value	This type of constraint restricts the set of concepts that can be taken as valid values or default values in an instance of a coded attribute or field
Predicate	This type of constraint restricts the values of an instance of class or attributes using text or testable expressions to represent "business rules" that apply to it.
Dynamic Model Constraints	Dynamic model constraints to specific interactions refer to the use of trigger events, message envelopes, update modes, and receiver responsibilities.

5.2.3 Layered approach to profiling

The layered approach to conformance recognizes that early V3 adopters have implemented solutions based on earlier version of the Reference Information Model (RIM).

The layered approach to conformance proposes a set of levels of conformance to the loosest (level 1) to the most rigorous (level 4). All levels assume RIM conformance is met.

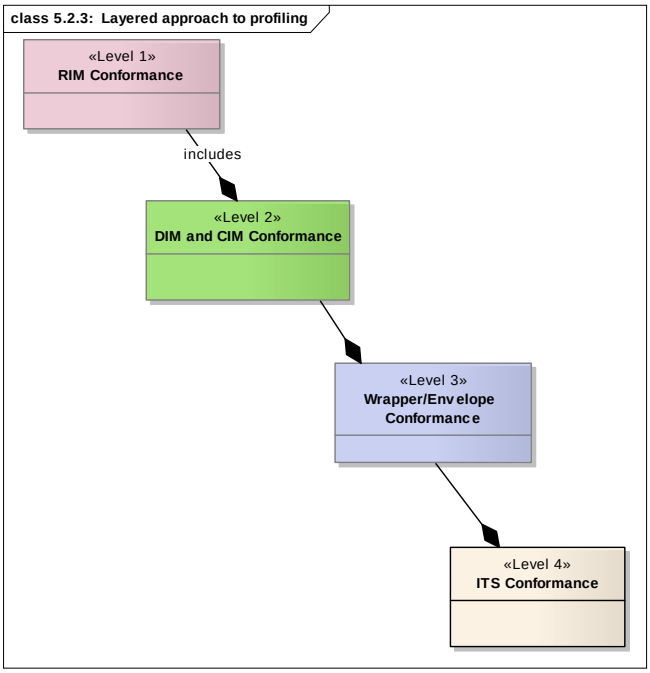

Figure 5.2.3: Layered approach to profiling

Name	Description
RIM Conformance	This level is met by a profile that uses RIM as the abstract information model for information exchange.
DIM and CIM Conformance	This level is met by a profile that uses HL7 Domain Information Models (DIMs) or a Constrained Information Model(CIM) abstract payload model for information exchange.
Wrapper/Envelope	This level is met by a profile that uses HL7-defined message wrappers/envelopes (conceptually represented as DIM/CIM) to exchange information. However, at this

Conformance	level the technology implementation is still undefined. **Note:** Wrapper/envelope models are subject to the same type of constraint as any other CIMs.
ITS Conformance	Uses all of the HL7 Static and Dynamic Models. In addition to the static and dynamic profiles, an implementation profile must indicate: 1) a technology specifications for information and behavior 2) a technology specification for transport

5.2.3.1 Level 1 - RIM Conformance

This level is met by a profile that uses the RIM as the abstract information model for information exchange. Conformance at this level assumes, therefore, conformance to the RIM. The following diagram shows the high-level RIM pattern that is recognizable in any RIM-conformant profile.

Note: The other conformance levels will be detailed in a future version of the HDF.

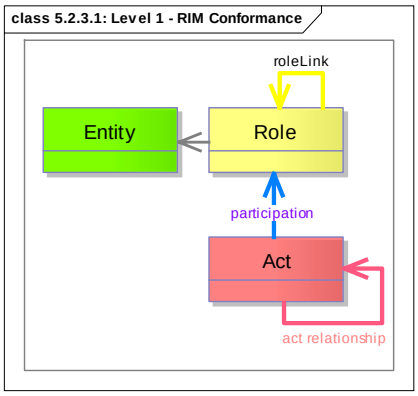

Figure 5.2.3.1: Level 1 - RIM Conformance

5.3 Roles and Responsibilities

This section describes the various roles involved during the development and modeling of HL7 standard specifications. It may be possible for the one individual to fulfill several roles if the skills and schedule allow it.

Name	Description
Integration Project Team	The project team involved in designing, implementing, and validating the project or solution that includes HL7-based interfaces.
Integration Specialist	Integration Analyst or Architect that works with a project interested in standard-based solution. This individual advises the project team on the use of healthcare interoperability standards produced by HL7.

5.4 Process

Standard Profiling is based upon the consistent application of constraints to a set of base specifications. This document outlines the processes that govern the definition of profiles and conformance statements.

During implementation or localization specification, models are further refined and specifications furthered constrained according to rules, conventions, and guidelines provided by HL7. The primary deliverable produced during specification profiling is a set of specification profiles and conformance statements. The steps to produce these profiles and conformance statements include:

1. Identify stakeholders/community of users
2. Further refine,constrain, or extend standard specifications
3. Publish specification profile(s)
4. Maintain specification profile(s)

5.5 Quality Criteria

To be defined

5.6 Tools

Standard implementers may use an Eclipse-based, UML 2.1 to import the HL7 standard specifications and apply constraints. In other cases, the constraints may appear as constraints predicates (e.g. Schematron).

5.7 Artifacts

Name	Description
Standard Profile	An HL7 profile may represent constraints applied to a standard specification. Information Model Profiles: http://www.hl7.org/v3ballot/html/infrastructure/conformance/conformance.htm#CECprofiles1 Functional Profiles for EHRS Implementation Guides for CDS

6. Technology Specification Process (TSP)

This section describes the process of developing of standard-based technology-specific artifacts for runtime testing. These artifacts may include XML Schema Descriptions, Web Service Descriptions.

NOTE: This chapter is under development.

6.1 Overview

This is a placeholder for a detailed description of the tooling and process involved in generating runtime

artifacts from design specifications. The process outlined in this chapter is intended to apply to both standard designs and profiles.

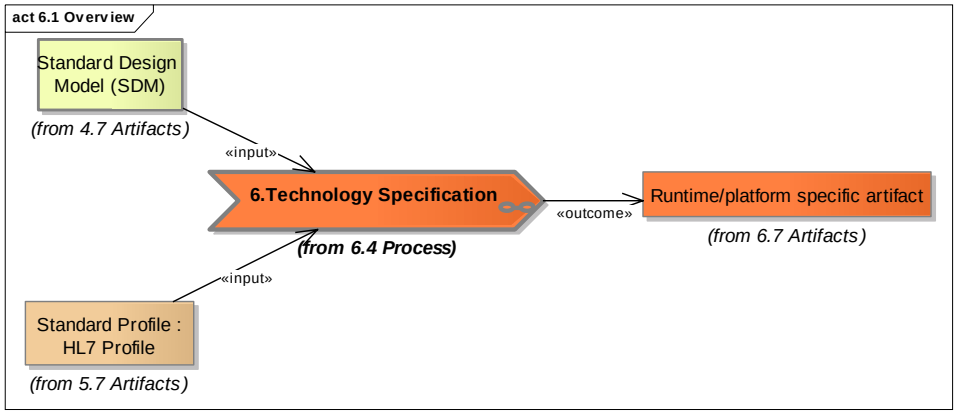

Figure 6.1 Overview

6.2 Context

The Technology Specification Process (TSP) is crucial in the uptake of a standard specification. While a conceptual specification is important to reaching consensus, an implementation specification is necessary to ensure that interoperability solutions conform to HL7 standards.

6.3 Roles and Responsibilities

This section describes the various roles involved during the development technology-specific artifacts for runtime testing. It may be possible for the one individual to fulfill several roles if the skills and schedule allow it.

6.4 Process

The process of designing technology specifications is under development. The following diagram is intended to describe the details of the TSP.

6.5 Quality Criteria

A quality standard profile must fulfill the requirements of the product or project for which it is intended. The actual criteria may vary depending on the type of specification. For information models, clinical document specification, and message specifications the profile must add the precision required to provide software development

6.6 Tools

HL7 uses specific tools to generated run-time artifacts. The default representation of HL7 information

models and interactions is XML. Therefore, the TSP tool right now is "V3 Generator":
http://hl7projects.hl7.nscee.edu/projects/v3-generator/

_

6.7 Artifacts

The following section summarizes the artifacts produced when a standard design is used to generate technology-specific implementation specifications and artifacts.

Name	Description
Runtime/platform specific artifact	This chapter is under development but currently, the only run-time artifacts created from HL7 standard design specifications are: XML Schema Descriptions (XSDs) for HL7 information models, interactions, and vocabulary. Schematron statements that use on Extended Style sheet Language (XSL) transformations to validate instances of clinical documents.

7. Change Control Process (CCP)

.

7.1 Overview

Change control is the process to be used when requesting and managing changes for product development performed by the members of HL7. The process will (1) facilitate communication about requested changes amongst the stakeholders of HL7, (2) provide a common process for resolving requested changes and reported problems, and (3) reduce any uncertainty surrounding the existence, state, and outcome of a change that has been requested in a work product.

This chapter describes the context, process, quality criteria, and tools used to support a consistent change control process across projects and committees in HL7 regardless of the type of standard they create.

This process is used to address issues/defects and enhancements to current approved or draft standards. During the validation of a certain standard, the early adopters, stakeholders, and other implementers will make use of the process and the tools that automate if

7.1.1 Change control using HL7 Project Homebase

The purpose of this section is to describe the options available to HL7 members to automate issue tracking and change control in order to provide better quality standards. HL7 provides to its members an integrated web portal (HL7 Project Homebase) that supports change control, task management, surveys, release management, document management, and source control in an integrated, web-based tool. As the name indicates, HL7 Project Homebase can be used as the home page for HL7 projects and committees in a way similar to Sourceforge.net, one of the most successful open-source project hosting portals.

HL7 Project Homebase (hl7projects.hl7.nscee.edu/) is based on the GForge open-source product (www.gforge.org) that can host the artifacts of an HL7 project including the mechanism to manage the changes to those artifacts. The benefits of the Homebase portal include seamless integration of issue tracking and task management with version control and release management. In this capacity, it can support all the phases of an HL7 standard development project.

Name	Description
Background	There is considerable overlap and confusion surrounding the change management, configuration management and change control modalities. Change control today consists of six steps, where each step may have another process associated with it. These start with the receipt of the Request for Change or RFC, or a Request for Service (RFS): 1. Record / Classify 2. Assess 3. Plan 4. Build / Test 5. Implement 6. Close / Gain Acceptance
Training	HL7 provides free training classes that introduce the CCP, project management, and associated tool training during the HL7 working group meetings (the next training session will be available in January 2008).

7.2 Context

The change control process (hereafter referred to as CCP) is used in the larger context of a project life-cycle. Any HL7 standard stakeholder can submit the following types of issues to the change control system:

- Requests for requirement changes (additions, deletions, modifications, deferrals) to standards currently under development
- Reports of problems in current specifications and runtime artifacts
- Requests for enhancements in current artifacts
- Requests for new development projects and artifacts

The CCP will apply to baselined work products created or managed by the members of the HL7 committee or project, including:

- Standard artifacts that have been approved or draft for trial use
- Requirements specifications for methodology (procedures and processes)
- User and technical documentation

The following work product classes are exempted from the CCP:

- Work products that are still under development except requirements changes requested for new projects
- Interim or temporary work products created during the course of a project
- Any work products intended for individual use only

The following diagram provides a high-level view of the processes described in the HDF specification to support the project and product lifecycle. Managing change is an intrinsic aspect of creating and improving specifications in a collaborative environment.

The key processes required by the HL7 Development Framework (HDF) are represented from a high-level as components of a larger product lifecycle process. All the processes in the HDF require a rigorous

approach to managing changes (i.e., technical corrections, enhancement requests, and new standard creation) suggested to the body of HL7 standards.

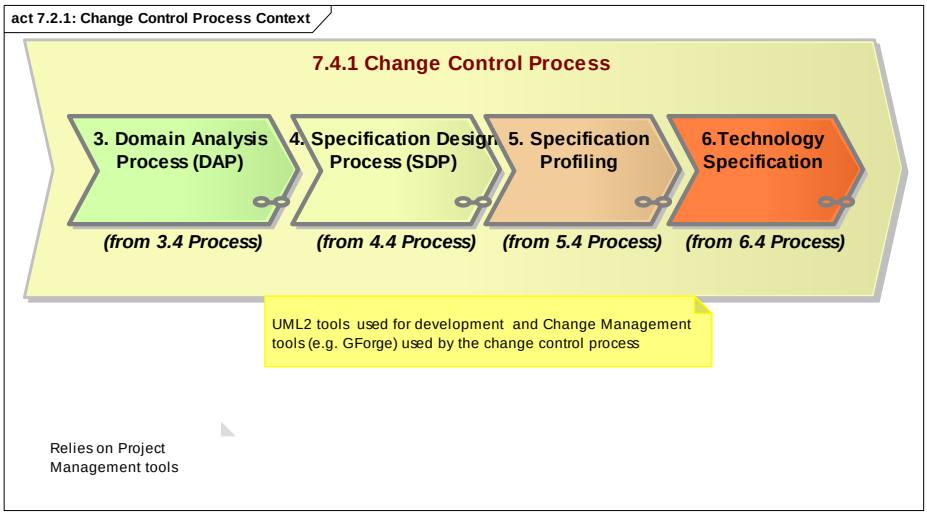

Figure 7.2.1: Change Control Process Context

Name	Description
Note	Relies on Project Management tools

7.3 Roles and Responsibilities

The following roles and responsibilities are used within the CCP for HL7 artifacts. Note that one individual may play multiple roles in the context of this process.

Name	Description
Submitter (Requester)	The submitter is a person, project, or committee that initiates the change request. This may be a user of a standard, a DSTU implementer, or a member of the committee or project responsible for the artifact that requires the changes.
Evaluator	The evaluation of a Change Request will be done by a group similar to a Change Control Board (CCB). This group may consist of the co-chairs or of a volunteer designated by the work group or project that receives the Change Request. The Evaluator or CCB may consist of the co-chairs or a designated co-chair or a volunteer as decided by the committee or project. The committee may be Technical Committee, a Special Interest Group, or project affiliated with a committee.
Modifier	The person assigned to address the change request. This committee member is responsible for completing the solution required in the change request.
Reviewer	The reviewer is a member of a project team that is involved in reviewing an artifact that was changed in response to defect report or enhancement request..

7.4 Process and Tasks

This section describes the process, decision points, activities, deliverables, and tools required to support CCP.

A change is "an event that results in a new status of one or more configuration items (CI's)", and is approved by management, is cost effective, and enhances business process changes (fixes) with a minimum risk to IT infrastructure. The main aims of Change Management are:

- Minimal disruption
- Reduction in back-out activities
- Economic utilization of resources involved in the change
- Efficient use of resources

CCP is a formal process used to ensure a product, service or process is only modified in line with the identified necessary change. It is part of the lifecycle. It is particularly critical to the early development phase of the software development/engineering process to ensure that unnecessary requirements were not created as artifacts in the code creation phase. Quite often unnecessary requirements and changes introduce defects which require expensive rework. CCP has evolved to become a fundamental process in quality control too. A change "freeze point" was introduced to suspend any further changes until after the completion of the initial project. CCP is also used to help determine the impact of a change and if it introduces severe risk and/or financial consequence. Typical examples from the computer and network environments are the upgrade of operating systems, network routing tables or the electrical power systems supporting such infrastructure, not forgetting changes to the contract itself.

The following diagram describes a simple CCP supported by HL7 GForge, the underlying implementation of the HL7 Project Homebase. The actual process may be more complex but this represents a comprehensive process that takes advantage of existing tools to automate the steps and ensure that the process is executed correctly.

7.5 Quality Criteria

The quality criteria for this process ensure the process was followed and the results are beneficial to the quality and speed of developing and correcting HL7 artifacts.
- the change requests are reviewed in a timely manner
- the approved requests are resolved in a ballot or technical correction in accordance to its severity.

Name	Description
7.5.1 Traceability to requirements	The principal way of demonstrating that a change request was fulfilled is to demonstrate traceability to the requirement or issue that originated it. A careful requirements analysis based on enhancement requests ensures that the changes made to the standard specifications can be traced to the change request initiated by the submitter.

7.6 Tools

This section demonstrates how we can use the HL7 Homebase portal and its ancillary tools (CCP) to record change requests and access the artifacts published by committees in response to new requirements or defect reports from HL7 membership.

The tools described in this section are used to automate the CCP. Unlike other web-based collaboration tools (such as wiki, web-sites, etc.) they provide intrinsic support for change management functions. This is an important difference that enables the seamless automation of CCP.

Name	Description
HL7 Homebase SCM	Software Configuration Management (SCM) is part of configuration management (CM). Currently version control is provided by Subversion(SVN), an open-source

	SCM tool. Roger Pressman states in his book, *Software Engineering: A Practitioner's Approach*, that software configuration management (SCM) is a "set of activities designed to control change by identifying the work products that are likely to change, establishing relationships among them, defining mechanisms for managing different versions of these work products, controlling the changes imposed, and auditing and reporting on the changes made." In other words, SCM is a methodology to control and manage a software development project. SCM concerns itself with answering the question: "Somebody did something, how can one reproduce it?" Often the problem involves not reproducing "it" identically, but with controlled, incremental changes. Answering the question then becomes a matter of comparing different results and analyzing their differences.
Task	The project management tool is intended to track the progress of ballots and projects in HL7. The project management tool available to committee members and change control board is currently Project Insight.

Screen Shots

This section contains sample screen shots of the tools used to support the change control process.

HL7 Project Homebase is a GForge-based tool used to manage project issues, version control, documentation, and release publication.

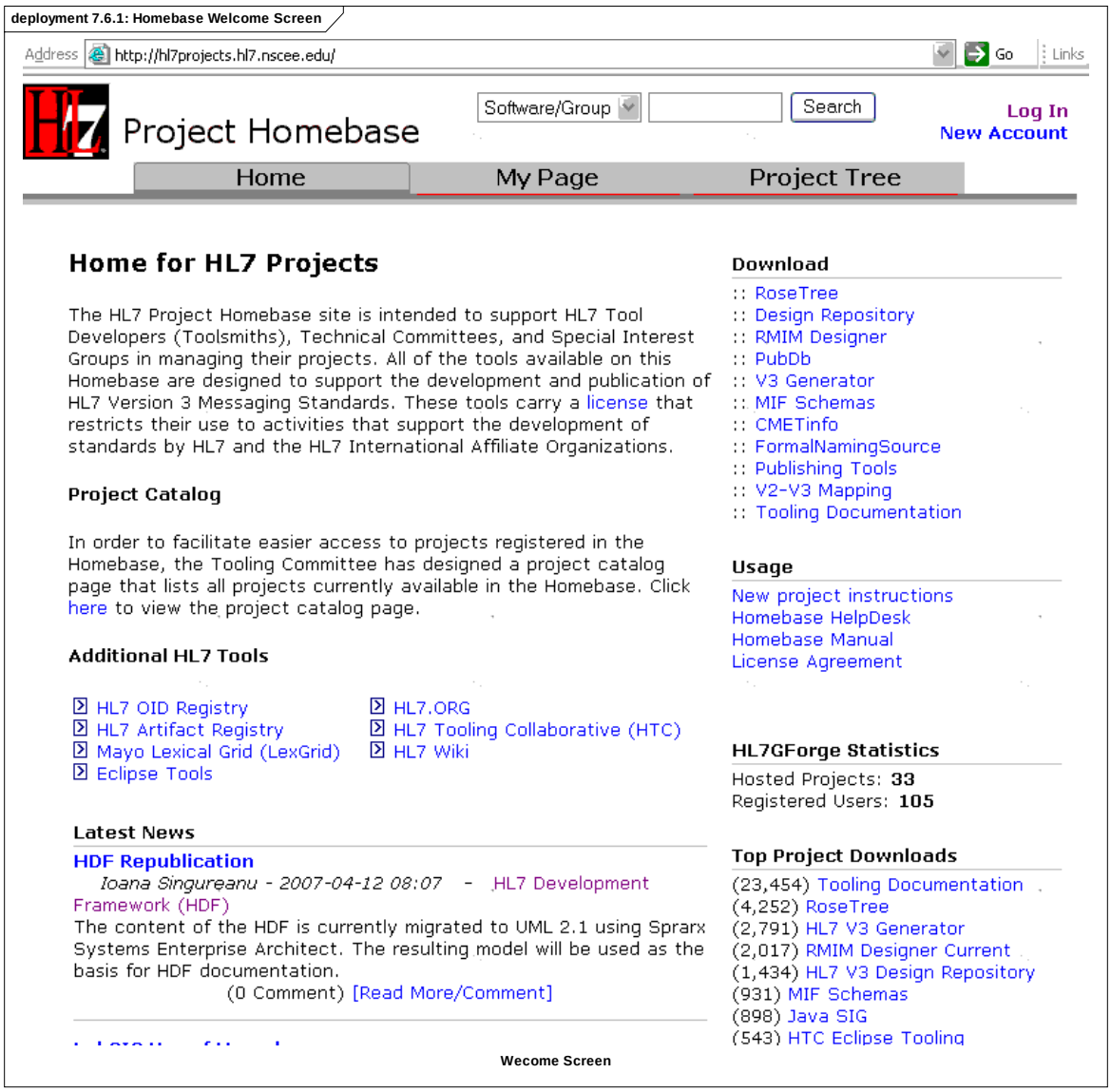

Figure 7.6.1: Homebase Welcome Screen

Once the user logs in, the following page will be visible. Displayed in this page are projects of interest and the change requests originated by this user.

Figure 7.6.2: My Personal Page

Figure 7.6.3 shows the HDF Summary tab in GForge. This screen may be used to select whether a Bug (Defect or Technical Correction), Support Request (e.g. a publication error, missing files, etc.), or a new Feature is requested. To enter a new issue or change request, the users must select the "tracker" tool.

Figure 7.6.3: Project Space

Figure 7.6.4 shows the "Feature request" screen. Each project may customize the drop-down lists and the fields that the submitter must complete. In this case HDF is used as an example. The submitter is selecting an Methodology area", specific to HDF, to indicate where the enhancement is needed. In addition to pre-defined fields, the submitter may upload documentation before pressing the "Submit" button to complete the request.

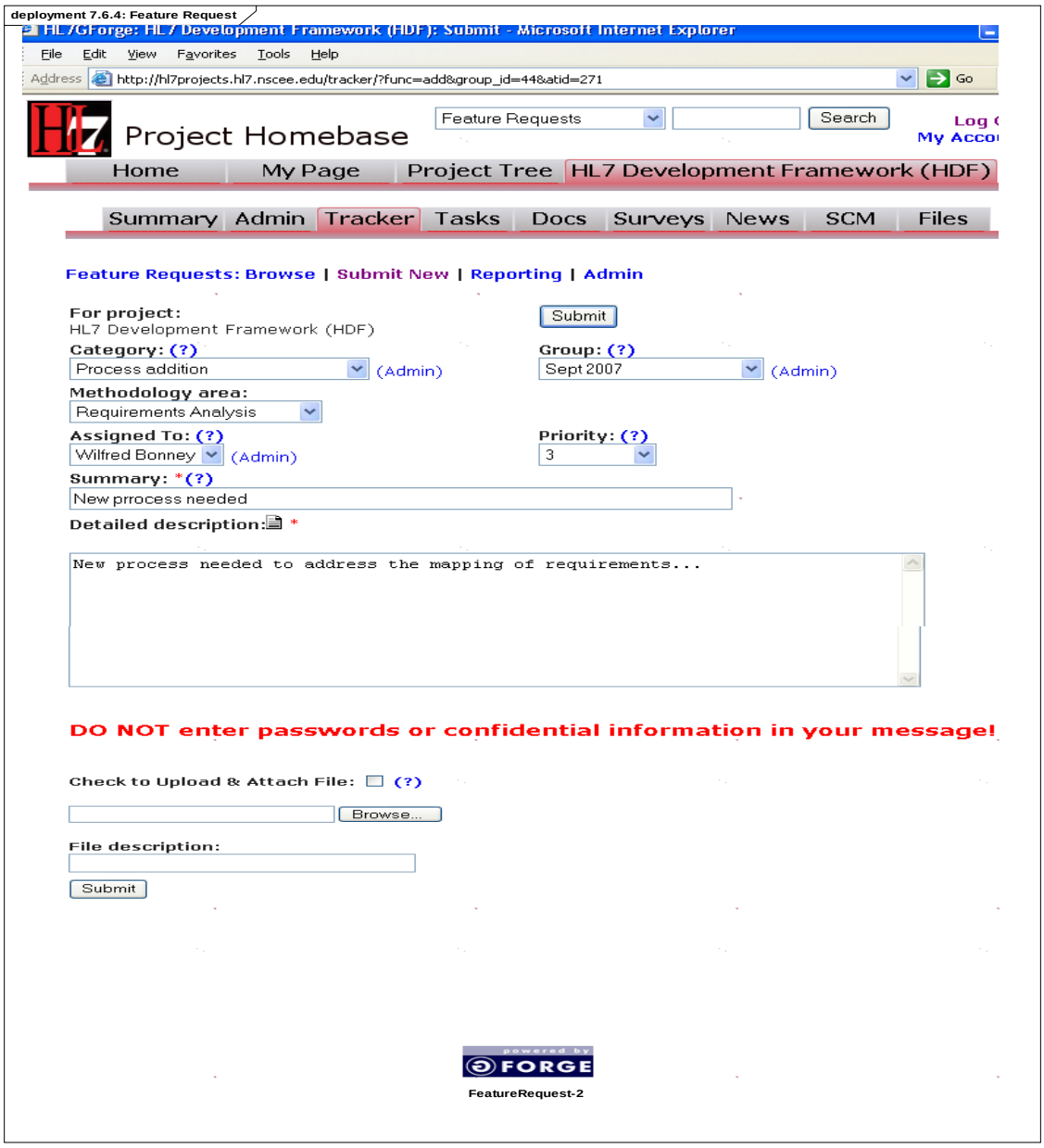

Figure 7.6.4: Feature Request

Once a change request is approved and determined to be in scope for the project by the Evaluator, its resolution will be part of a task, which in turn is part of a ballot cycle or release. The project team may create a new task in the current/future ballot or assign the issue to an existing task. The Task tools allow us to assign one or more issues from the tracker to the task. This way we can validate that all issues are resolved when the task is completed.

A task is a work item for the project team for a specific ballot. Unlike activities and milestones tracked in project management, these are very specific activities that change standard artifacts and tools. The project management tasks identify high-level life-cycle milestones rather than specific development, changes, and enhancements to standard specifications.

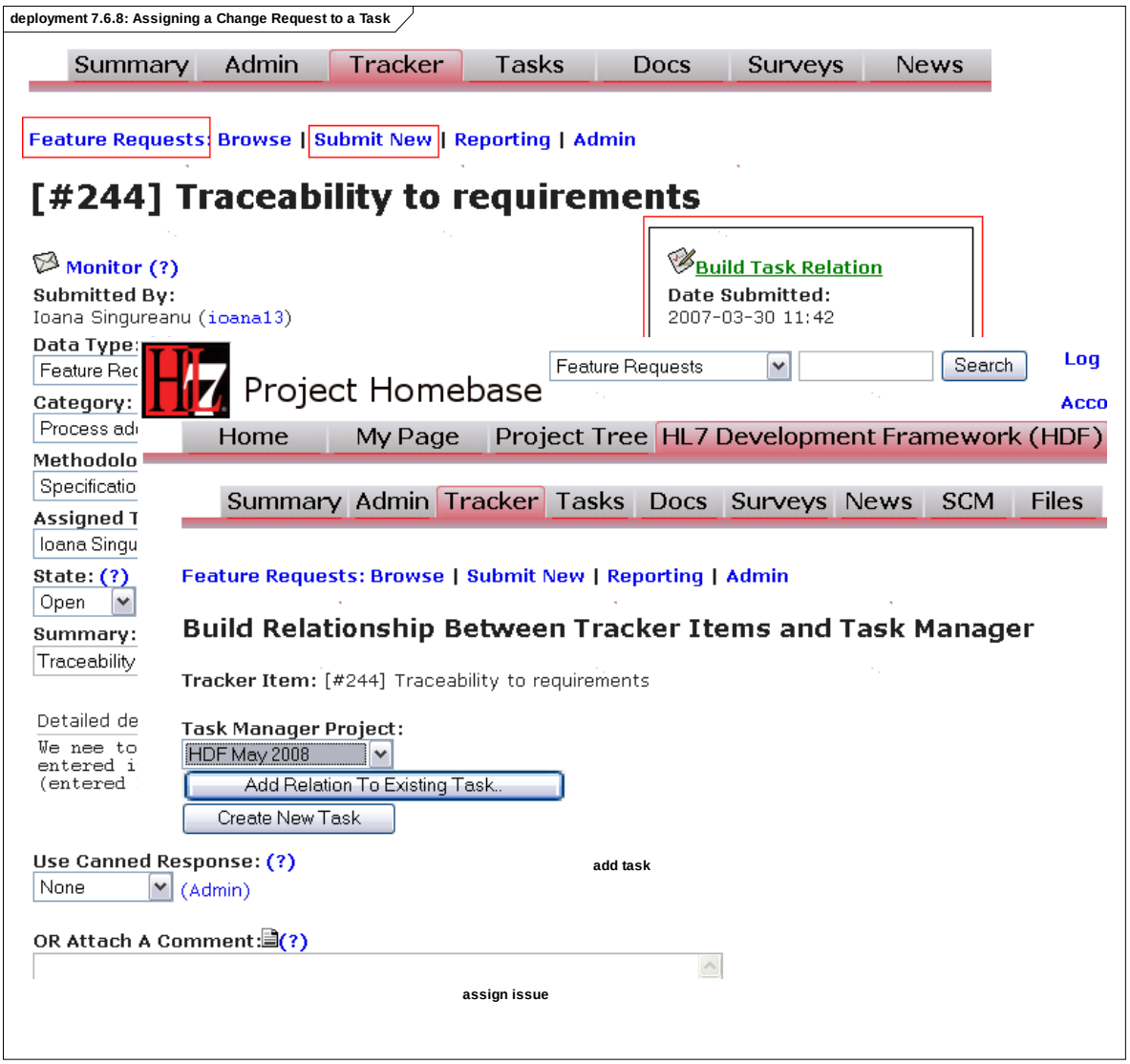

Figure 7.6.8: Assigning a Change Request to a Task

The projects and committees may provide a multitude of trackers (available in the "Tracker" tab). By default, the HL7 Homebase provides three trackers:

- "Bugs" (used to track defects and technical corrections)
- "Feature Requests" (for enhancements to existing standards or tools)
- "Support requests" (for problems using the standard publication or tools)

For change control, only the "Bugs" and "Feature Requests" trackers are relevant. As seen here these trackers may be used to view/change current items, enter new items, and create reports on the existing items. The trackers are used by the committees to assign a change request (bug or feature request) to a member of the project for impact analysis. When the impact is completed, the item is related to a project task.

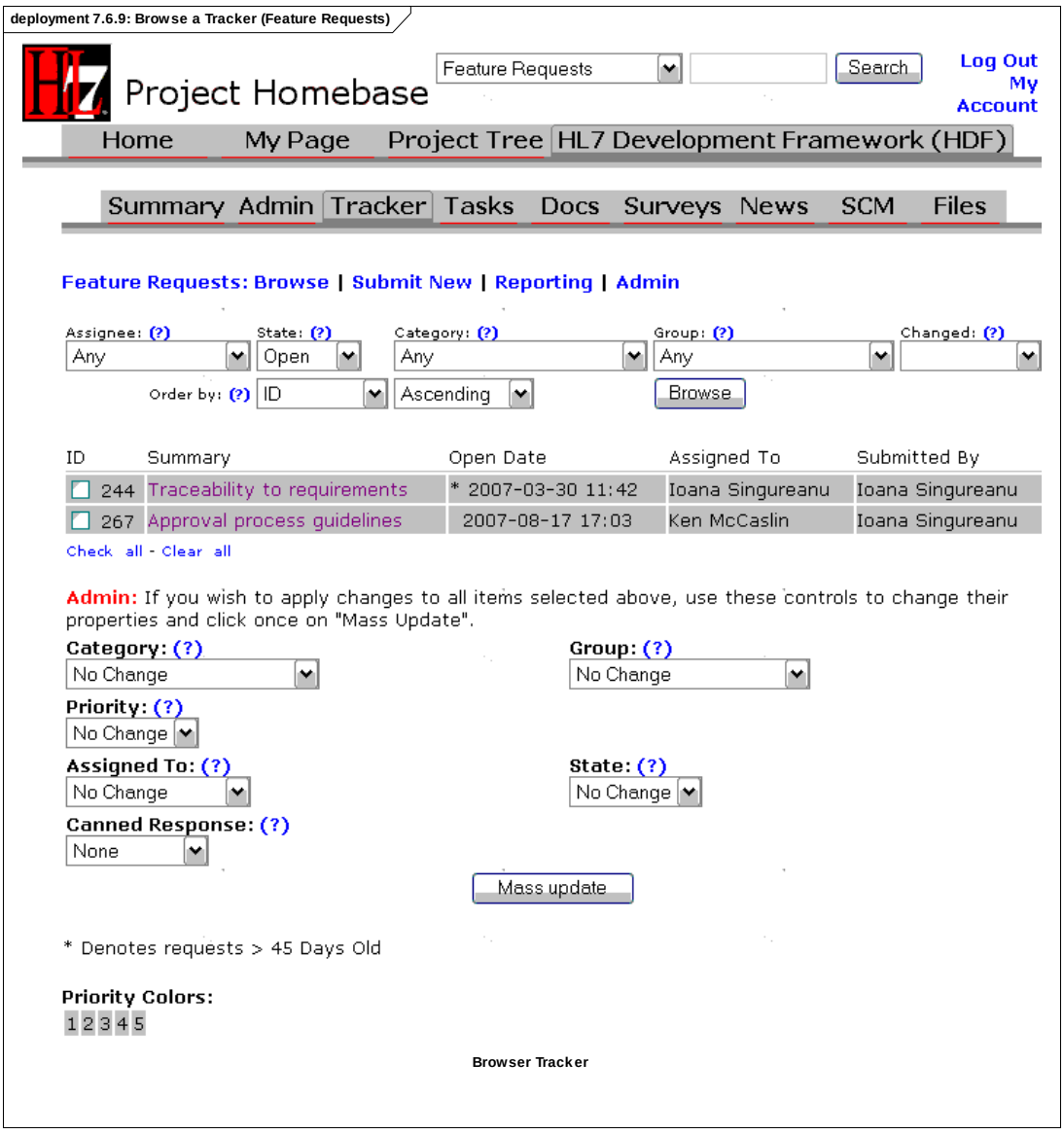

Figure 7.6.9: Browse a Tracker (Feature Requests)

A task is intended to address one or more change requests. The relationship between a task and one or more change requests is indicated in the "Related Tracker Items" section of the task detail dialogue.

One of the great benefits of the Homebase tools is the provision for one or more change requests to be associated with a task and assigned to a project member. Unlike other tools the GForge "Tasks" tools allows us to group together several change requests in a single task. This allows us to address duplicate change requests or related change request. Before the tasks can be completed all the requirements referenced in the "Related Tracker Items" must be resolved.

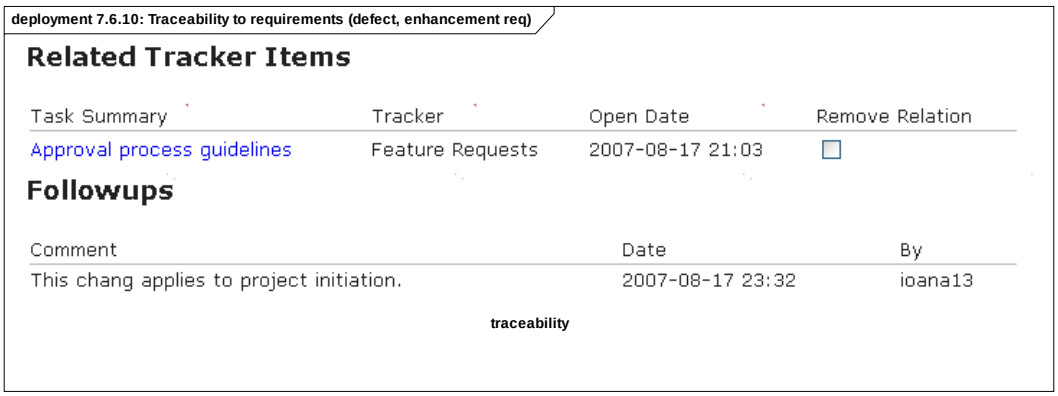

Figure 7.6.10: Traceability to requirements (defect, enhancement req)

Each ballot may appear as a different "sub-project" under the "Tasks" tool. The committees/project may assign a task to any one of the available releases.

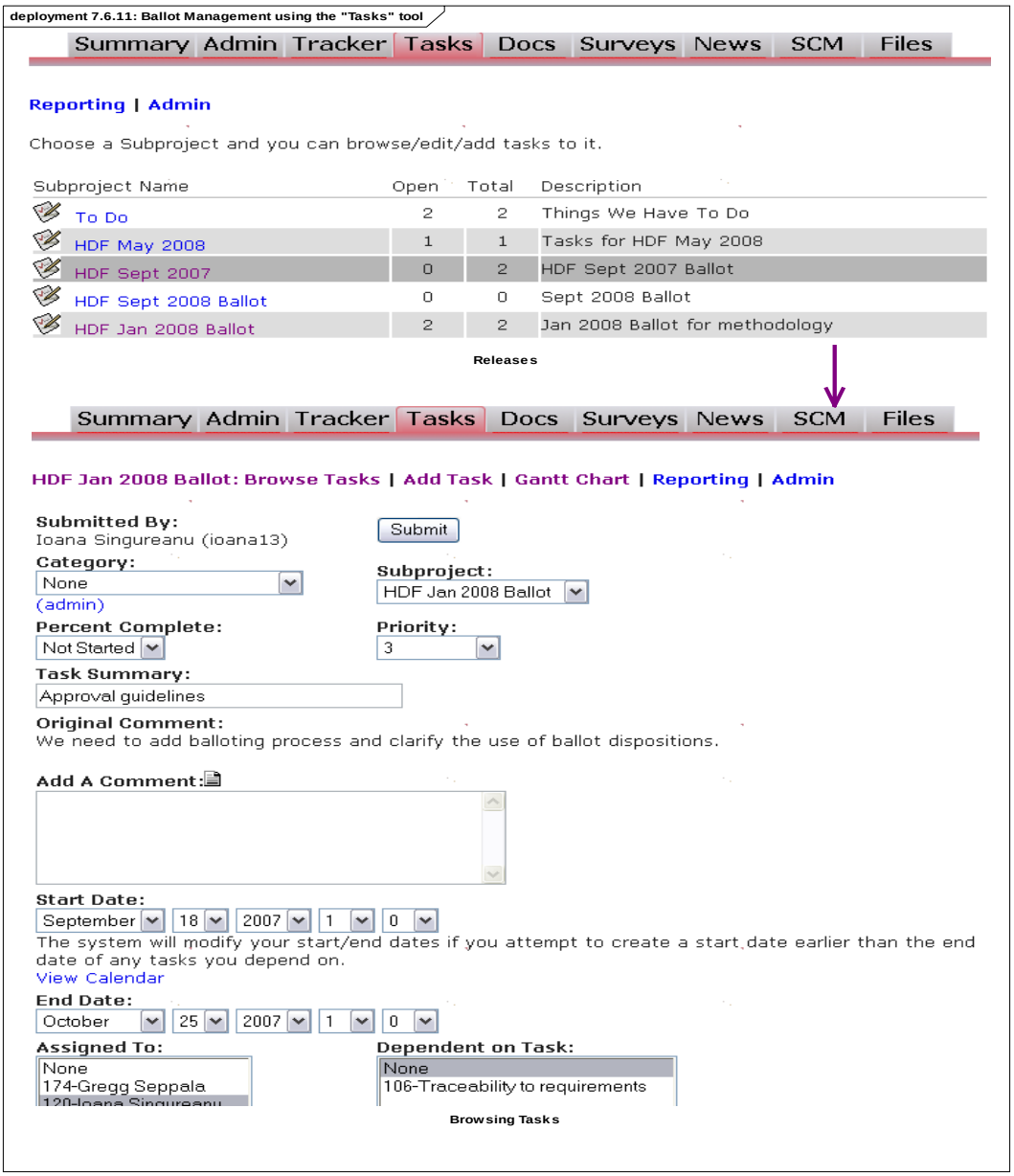

Figure 7.6.11: Ballot Management using the "Tasks" tool

Based on a subproject/ballot, the project team may create Gantt charts to communicate dependencies, progress, and identify scheduling issues in a graphical way. This Gantt chart addresses a single ballot and does not duplicate the type of chart created by the project management tools for the entire project/product lifecycle.

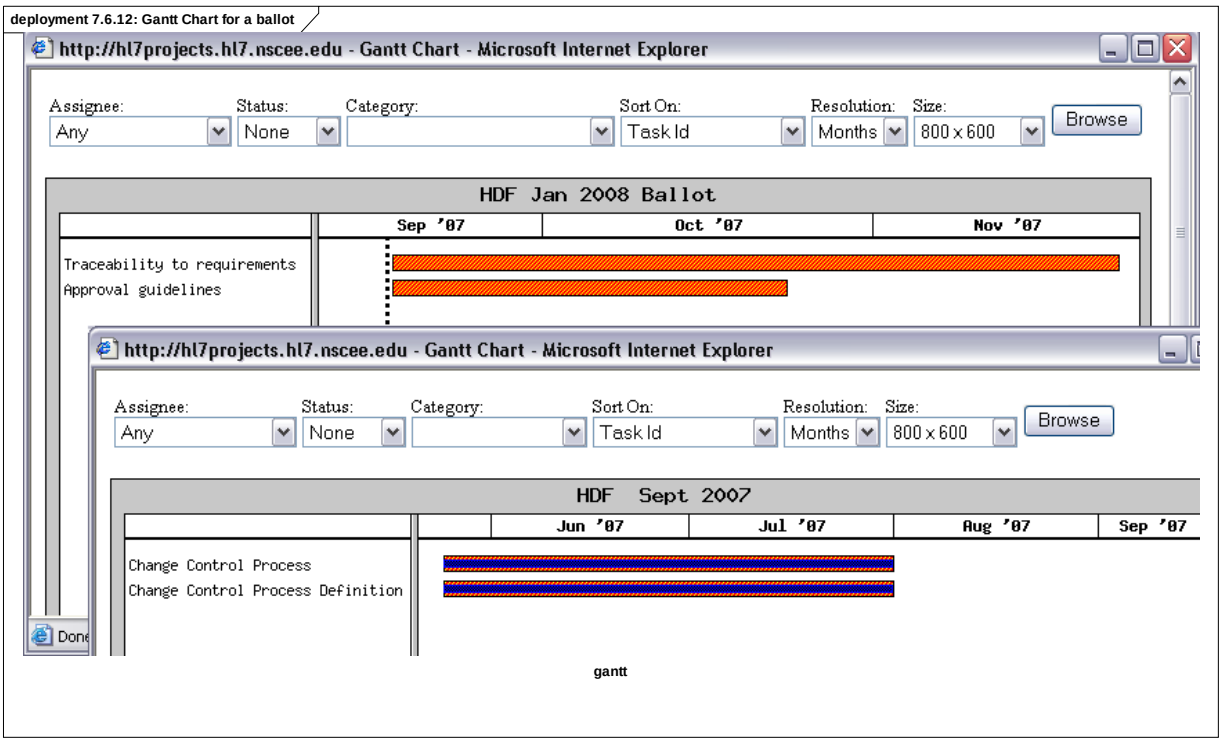

Figure 7.6.12: Gantt Chart for a ballot

7.7 Artifacts

The following section describes the artifacts produced to process a change request.

Name	Description
GForge Change Request Ticket	The ticket is used to track the requester's change request through the resolution process. If approved for inclusion into a future release/ballot, then the ticket is transformed into a task assigned to a specific committee or project member.
Task	The task is associated with a ballot/release. The task is assigned to the person who is expected to make the artifact changes. The task may be part of a future ballot or the current ballot depending on the urgency of the issue. The task is part of a ballot schedule.

Annexes

The following annexes are intended to clarify the document and provide references to other documents.

Annex A. Domain Analysis Example

The following is an example domain analysis based on a real-live example: driver's licence renewal process. The choice was made based on the fact that the process is both familiar and not healthcare-specific.

The following diagram outlines sample artifacts produced during the DAM (Domain Analysis Model) and specification design/development phases (Standard Specifications).

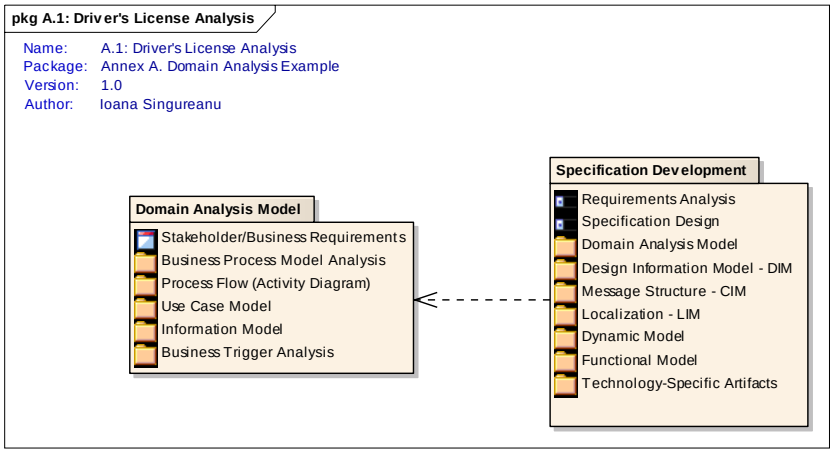

Figure A.1: Driver's License Analysis

Domain Analysis Model

The Domain Analysis Model contains the information and interactions needed to support the needs of the projects - from the stand-point of the business stakeholder (SME).

The Requirements Analysis phase produces a Domain Analysis Model like the one below:

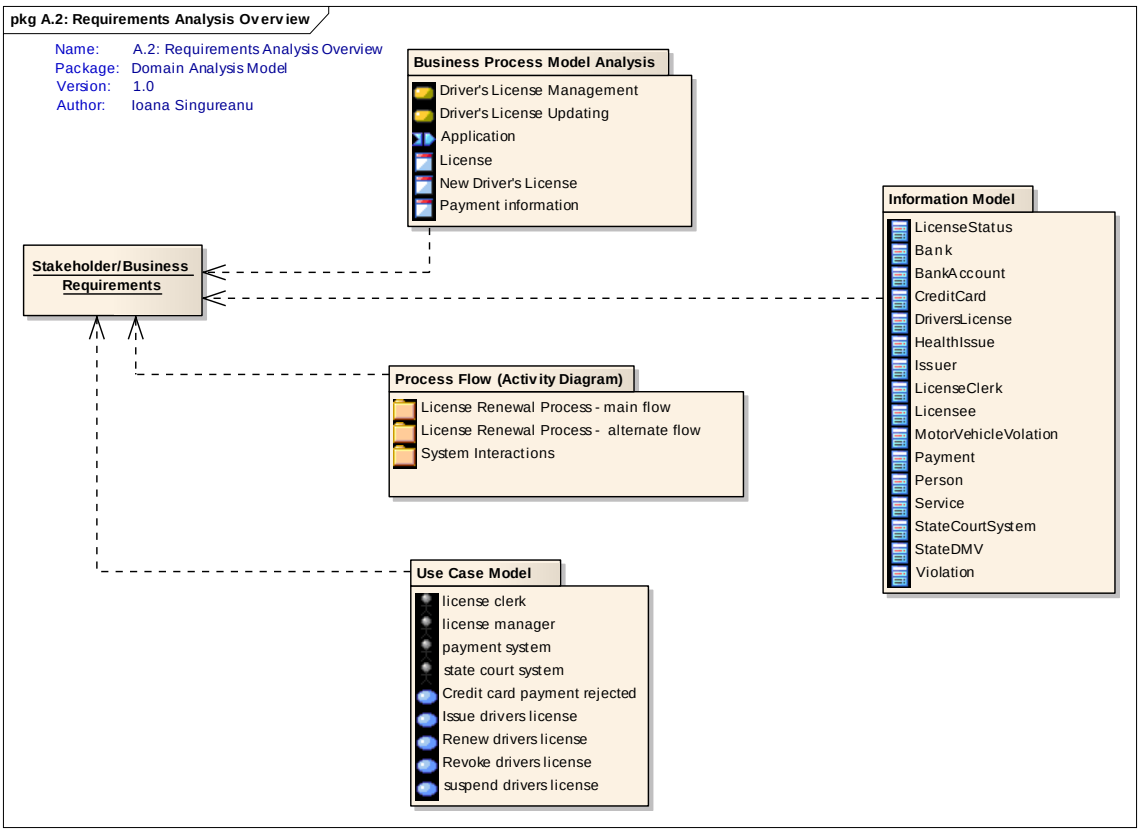

Figure A.2: Requirements Analysis Overview

Name	Description
Stakeholder/Business Requirements	The original problem description or requirements may be included in the project scope or added as change requests during the project's lifecycle (using the Change Control Process).

Business Process Model Analysis

The following illustrates the use of business process models to document business rules during the requirements analysis phase.

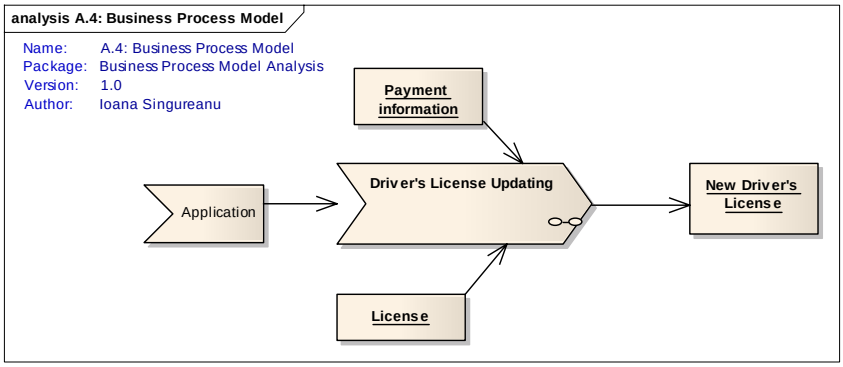

Figure A.4: Business Process Model

Driver's License Updating	The following diagram summarizes the update/renewal of a driver's license.

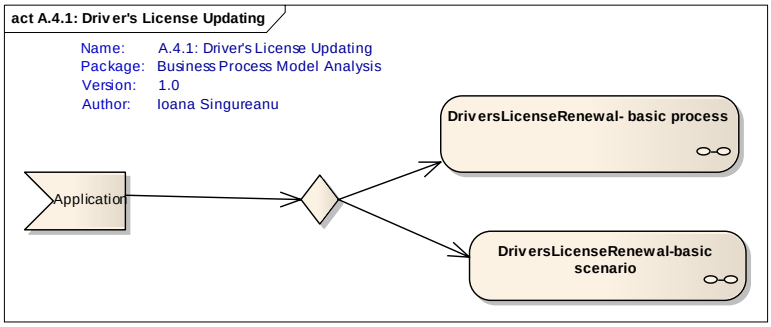

Figure A.4.1: Driver's License Updating

Process Flow (Activity Diagram)

The following is an example showing several vies of the process flow.

License Renewal Process - main flow

The following is basic flow for license renewal.

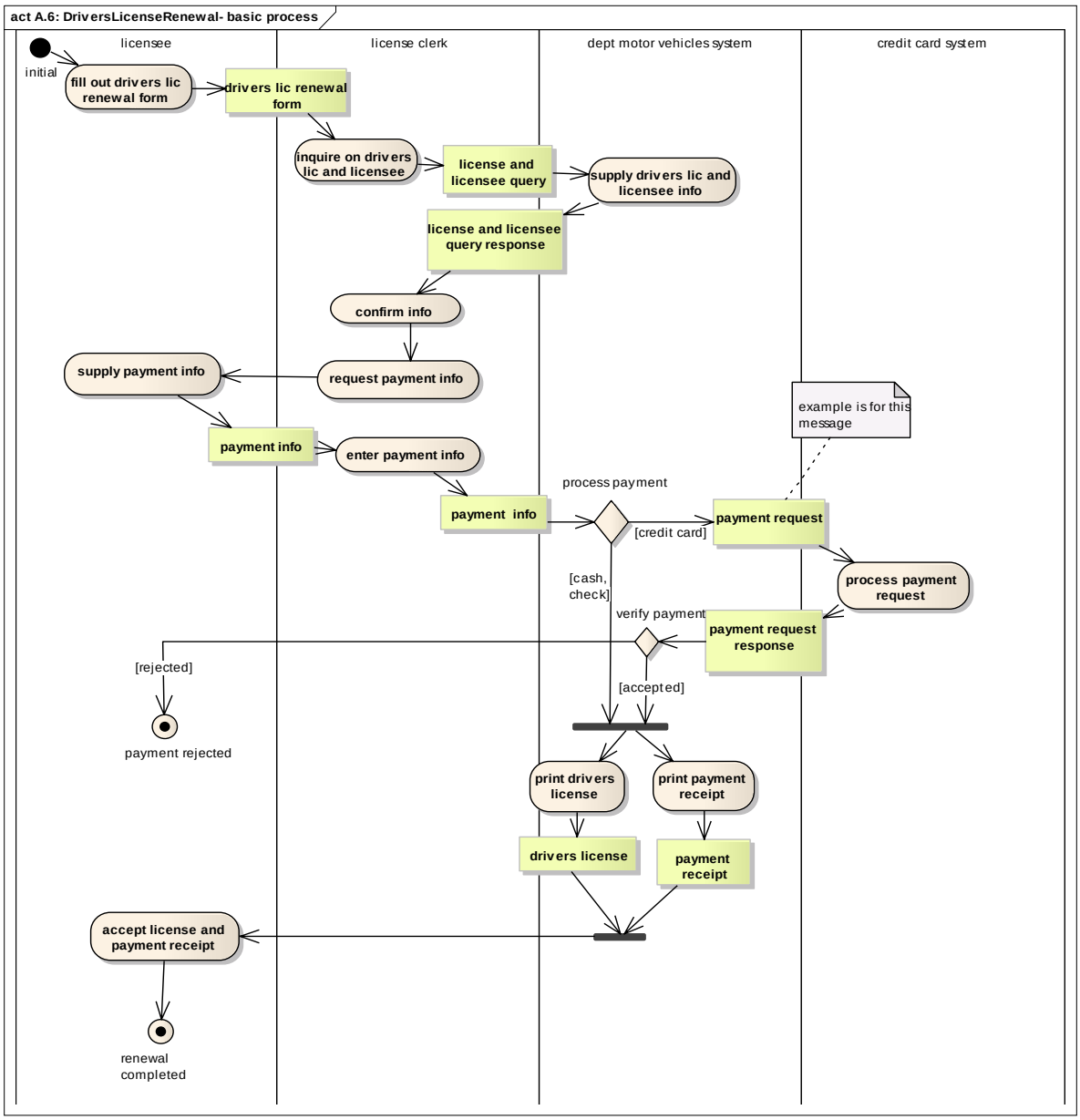

Figure A.6: DriversLicenseRenewal- basic process

License Renewal Process - alternate flow

The following section describes an alternate process flow for the Driver's License Renewal use case. This is an alternate scenario for a typical license renewal.

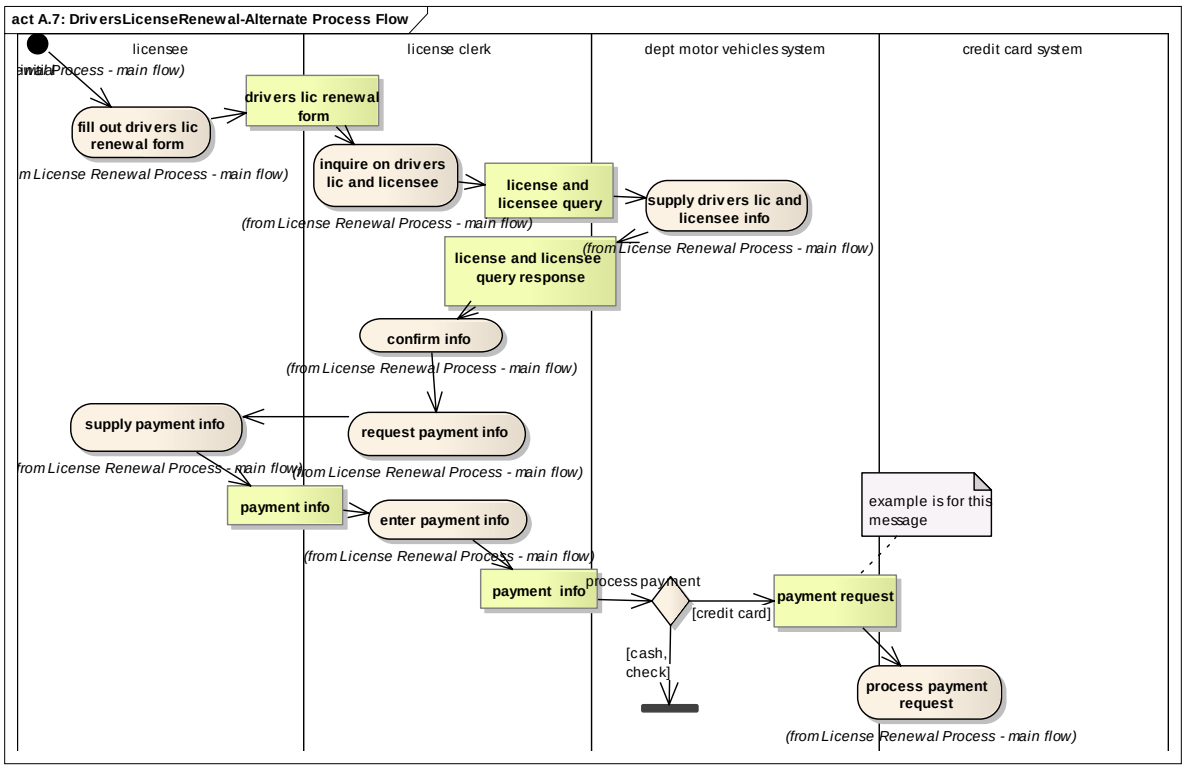

Figure A.7: DriversLicenseRenewal-Alternate Process Flow

System Interactions

The following section identifies the interfaces and interactions required to support the process and the use cases documented in the DAM.

The following summarizes the system interfaces that are used in the sequence diagrams.

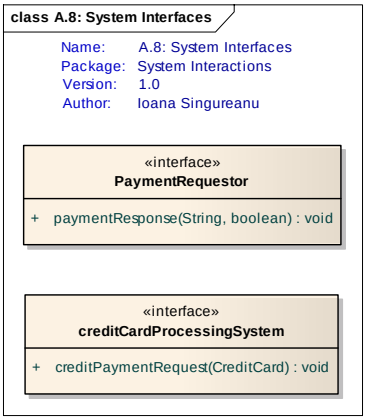

Figure A.8: System Interfaces

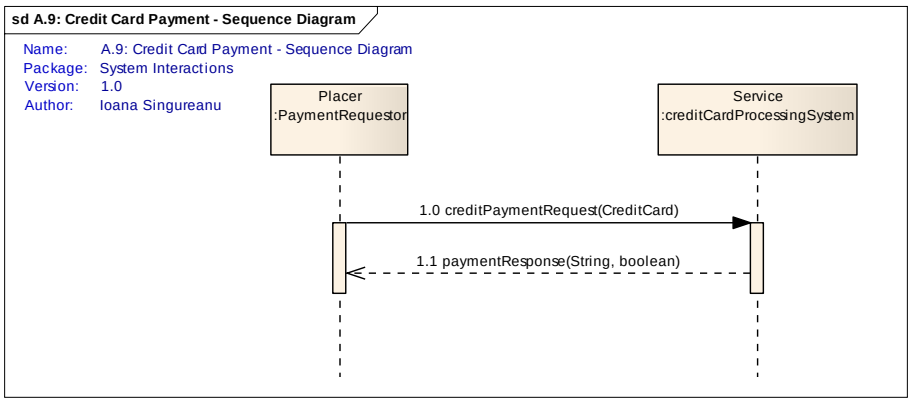

Figure A.9: Credit Card Payment - Sequence Diagram

Use Case Model

The following is an example use case model. It documents the features required to support driver's the entire lifecycle of a driver's license. The diagrams included in this section are intended to demonstrates the use of the UML notation to identify use cases and actors.

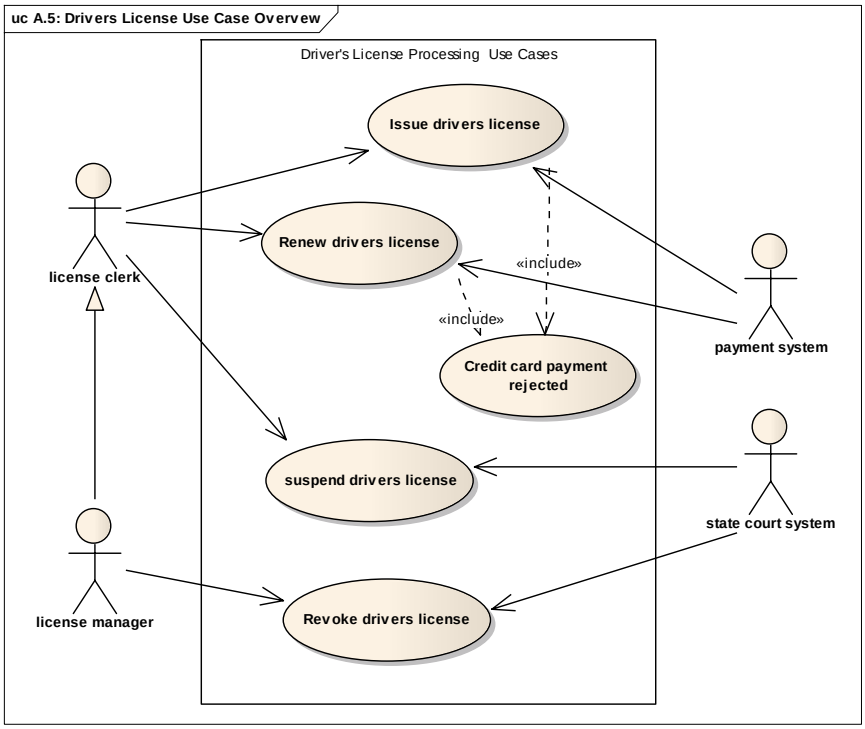

Figure A.5: Drivers License Use Case Overvew

Name	Description
Credit card payment rejected	**Use Case Scenario #2:** credit card payment rejected After Darlene Dobson has provided her credit card information for payment of her drivers license renewal to the license clerk and it was entered into the Department of Motor Vehicles system, the system sent a payment request to the credit card

	system. The response indicated the payment request was not accepted; no reason was provided. Darlene's license was not renewed during this visit. Note: in a real world scenario, the licensee would likely be given the opportunity to pay in cash; merely to make the Activity Diagram small enough to easily display, that Activity Diagram did not specify that option; this Use Case Scenario thus ends without the successful renewal of the drivers license
Renew drivers license	This use case requires interoperability. Each use case identifies preconditions, post conditions, and the flow of events. **Use Case Scenario:** Renew Drivers License Use Case Name: Renew Drivers License **Primary Actor:** license clerk **Goal in Context:** renew the licensee's drivers license **Precondition(s):** the license clerk is logged into the system Success Guarantee(s): the system has renewed the person's drivers license, printed the new license and a receipt **Main Success Scenario:** Using the filled out Drivers License Renewal Form the licensee has filled out, the license clerk inquires on the status of the person's drivers license • The system sends an inquiry to the state court system to confirm the licensee has no moving motor vehicle violations in the past three years. • The system receives the response from the state court system • The system returns the person drivers license information • The license clerk verifies the person's address on file and enters the payment method (cash, check, credit card) and if paying by credit card, the person's credit card information. • If the payment method is by credit card, the system sends a financial transaction to the credit card system and receives the response • The system prints out the new drivers license and a payment receipt. **Extensions:** 3a. If the licensee had any moving motor vehicle violations in the previous three years, the use case ends. **Business Rules:** drivers license renewals are for a period of three years if a licensee had any moving motor vehicle violations within the previous three years as recorded in the state's court system, they are not allowed to renew their drivers license. **Secondary Actor(s):** state court system, credit card system

Information Model

This model describes the key information to be shared to achieve the objectives detailed in the documented cases.

The following illustrates the first pass at analyzing information requirements for a typical driver's license renewal. Note that the classes are identified and defined by the SMEs but the details and attributes have not been identified yet.

.

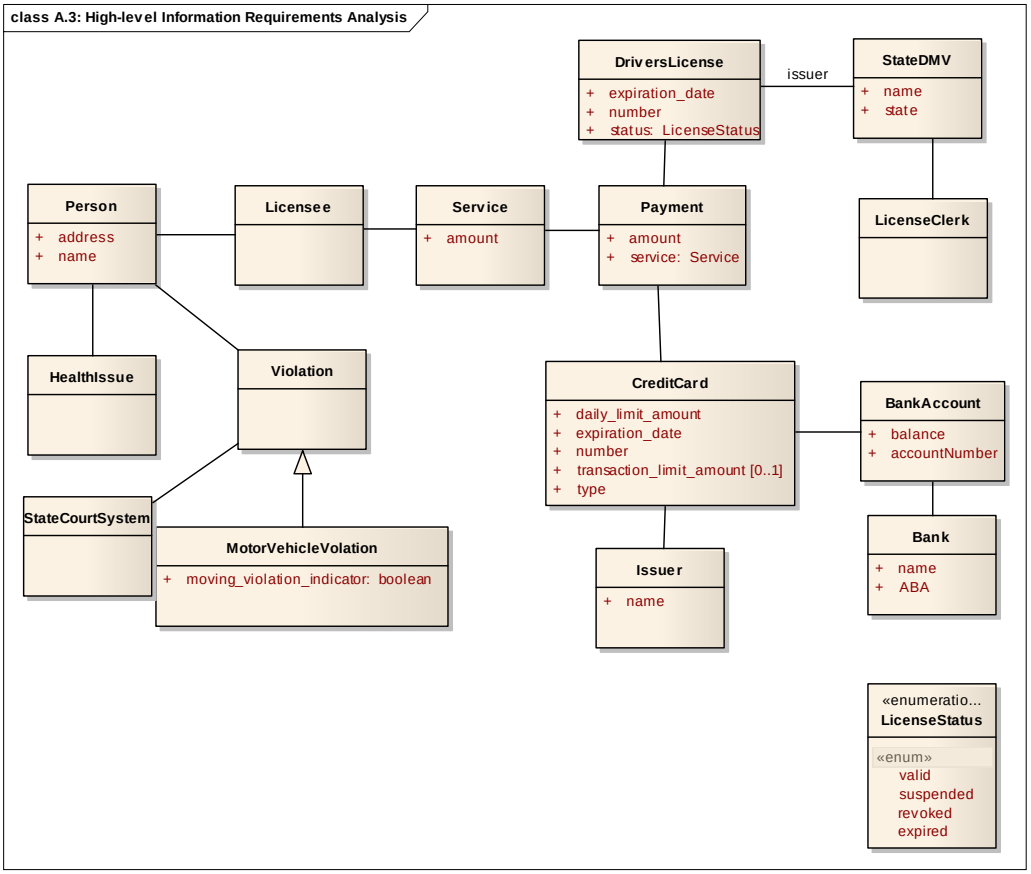

Figure A.3: High-level Information Requirements Analysis

BankAccount	A financial entity representing the sum of monetary credits and debits. It involves two parties; one typically referred to as the holder, the other frequently a financial institution such as a bank.
Attribute	**Notes**
balance Public	Account balance, if available
CreditCard	An item issued by a financial organization to a person or organization allowing them to purchase items. The physical plastic wallet-sized card includes information about the credit card account including credit card

	number, expiration date, issuer name (e.g. Mastercard, Visa).

Attribute	Notes
daily_limit_amount Public	

DriversLicense	An authorization given by a US state department of motor vehicles to a person granting them the legal right to operate a motor vehicle.

Attribute	Notes
expiration_date Public	Specifies the date when the license expires automatically and it requires to be renewed.

HealthIssue	This is a health issue that may affect the driver's ability to operate a motor vehicle.
Issuer	State or country that issued the license

Attribute	Notes
name Public	The name of the department and state that issued the license.

LicenseClerk	A person working at the Department of Motor Vehicles who processes drivers licenses.
LicenseStatus	This enumeration specifies the DriverLicense status.

Attribute	Notes
valid Public «enum»	

Licensee	A person issued a drivers license. Also known as a license holder.
MotorVehicleViolation	A category of violation of a state motor vehicle law that occurs while a person is operating a motor vehicle. Contrast with a parking ticket, a violation of a state motor vehicle law which is not categorized as a moving motor vehicle violation.

Attribute	Notes
moving_violation_indicator boolean Public	

StateCourtSystem	A statewide electronic records system that records violations of the law within the state.

StateDMV	State Dept of Motor Vehicles

Attribute	**Notes**
name Public	
Violation	Violations that may have lead to penalties related to the driver's license.

Business Trigger Analysis

The following is an example showing how to document the state transitions that correspond to business triggers for information exchange. The diagram details the allowed states and transitions for the relevant business objects.

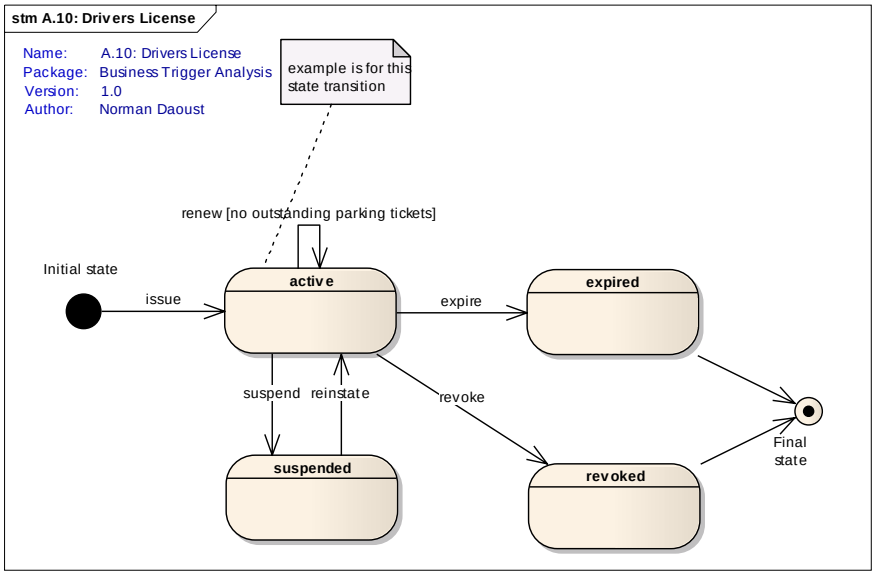

Figure A.10: Drivers License

Specification Development

The following example demonstrates the use of requirements analysis artifacts during the design of HL7 standards. Note that this example is specific to messaging but the same approach applies to any other HL7 standard.

The following diagram shows how the Domain Analysis Model to create design modes for standard development.

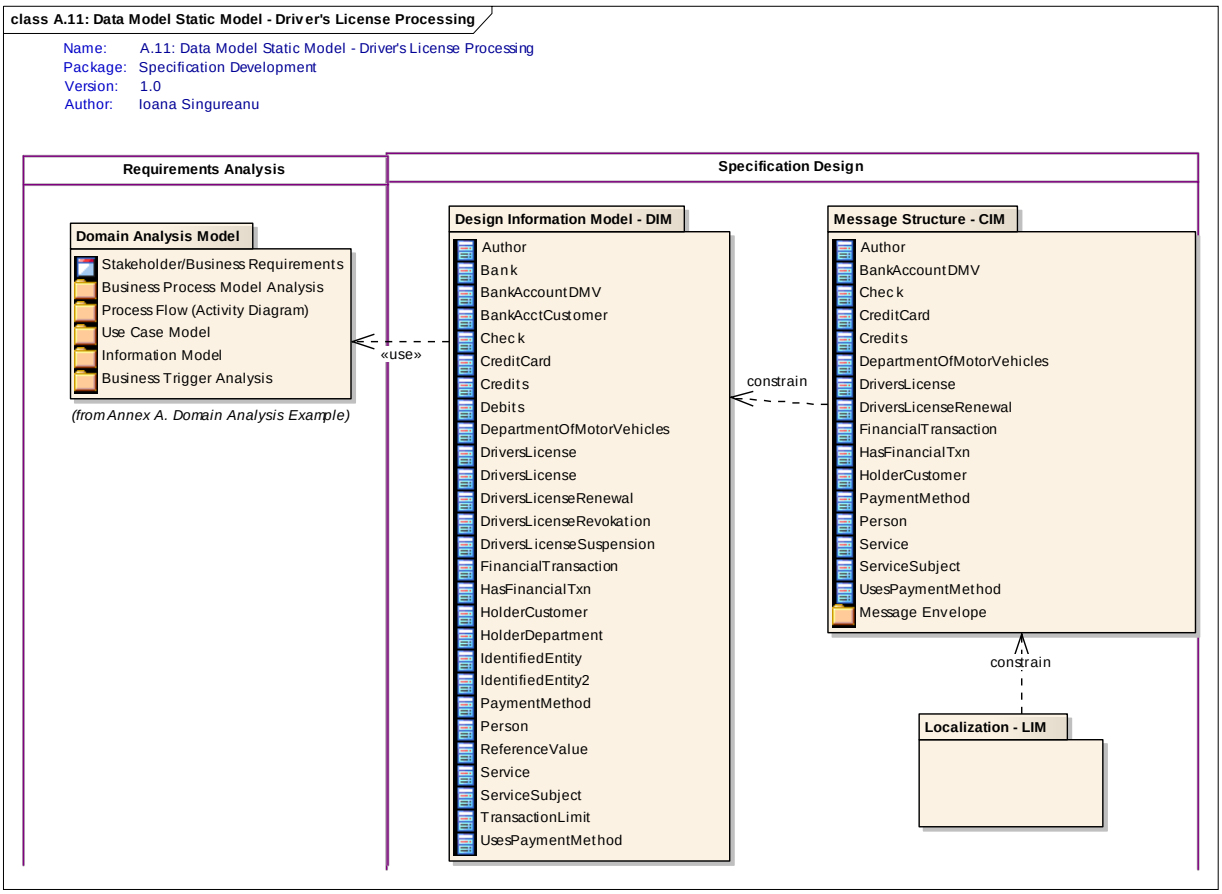

Figure A.11: Data Model Static Model - Driver's License Processing

The following diagram illustrates the uses of Dynamic Model Design artifacts in functional modeling or technology-specific runtime artifacts.
The Requirements Analysis process produces the Domain Analysis Model which is then used for Specification Design.

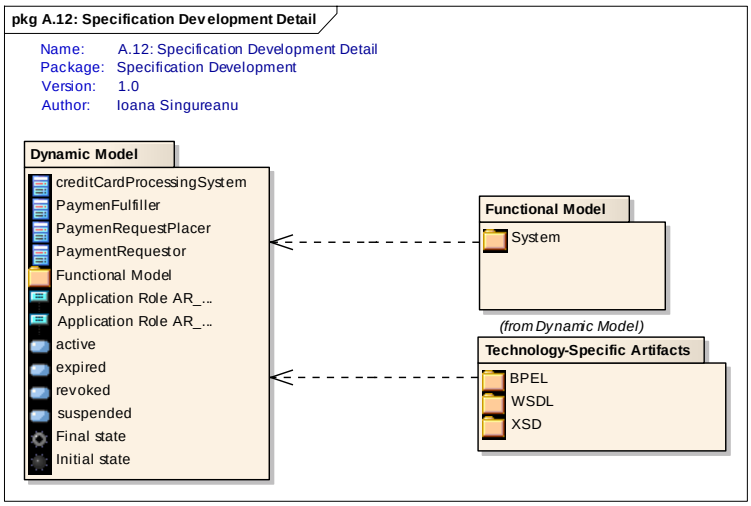

Figure A.12: Specification Development Detail

Design Information Model - DIM

The following contains the classes and diagram associated with processing a driver's license.

State Department of Motor Vehicles
domain information model
class diagram
ND 12/30/2005

Notes: The attribute values after the equal sign represent the constrained domain value (even though they display as initial values according to UML notation); for this travesty I ask for forgiveness. All attributes of type TS have constraints in precision to the date (i.e. time is not included); all attributes of type MO (money) have a constraint to US dollars with a precision of one cent, e.g. $9.99); the Department of Motor Vehicles account is credited, the service recipient's account is debited (for the sake of a smaller class diagram).

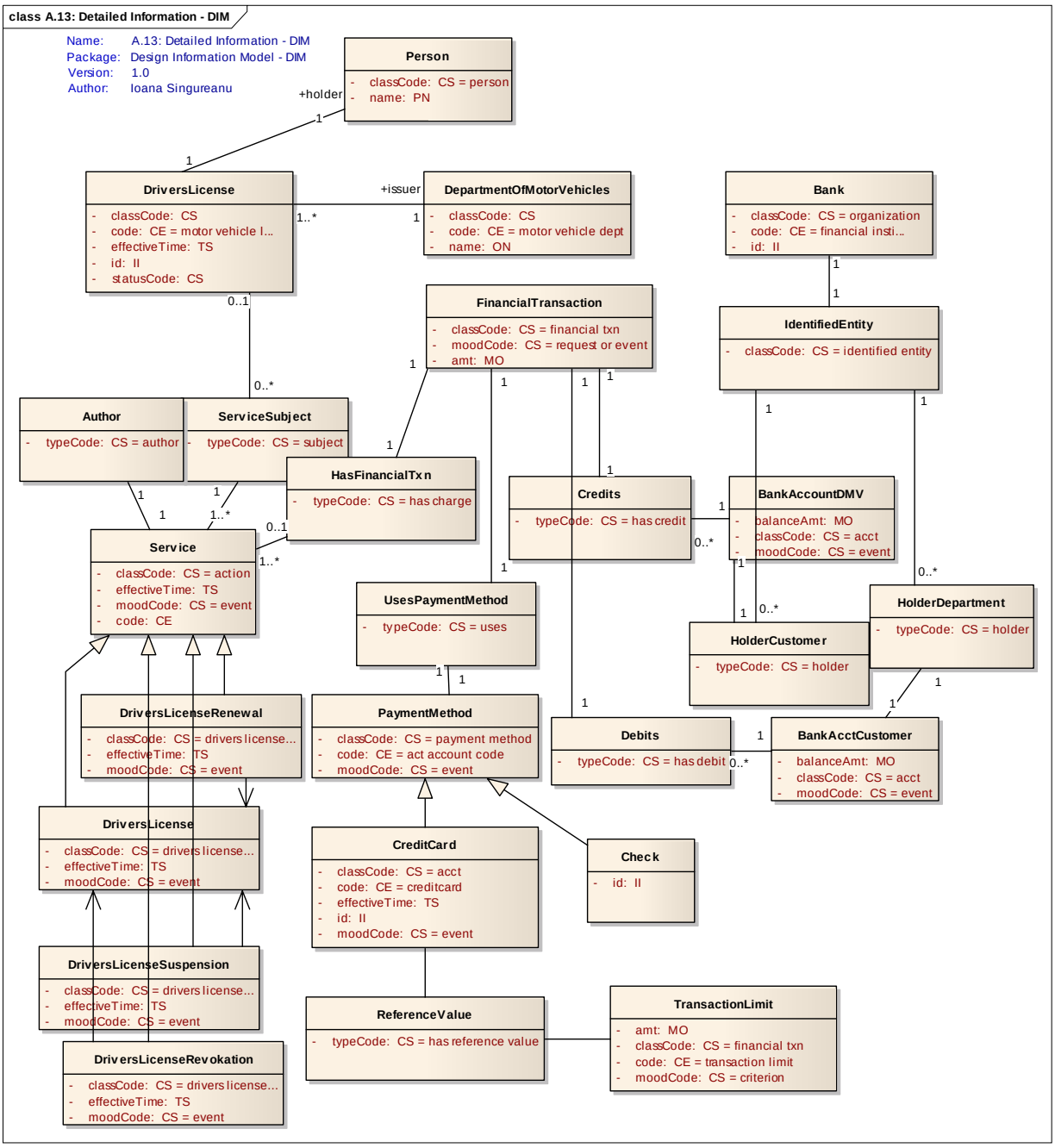

Figure A.13: Detailed Information - DIM

Message Structure - CIM

This is the Message structure that intended to support the interactions required for driver's license renewal

Differences/constraints from domain:

Service class was constrained to DriversLicenseRenewal and specific references to credit card limits and baking were removed as they are not required for this particular message payload.

Notes: The attribute values after the equal sign represent the constrained domain value (even though they display as initial values according to UML notation); for this travesty I ask for forgiveness. All attributes of type TS have constraints in precision to the date (i.e. time is not included); all attributes of type MO (money) have a constraint to US dollars with a precision of one cent, e.g. $9.99); the Department of Motor Vehicles account is credited, the service recipient's account is debited (for the sake of a smaller class diagram).

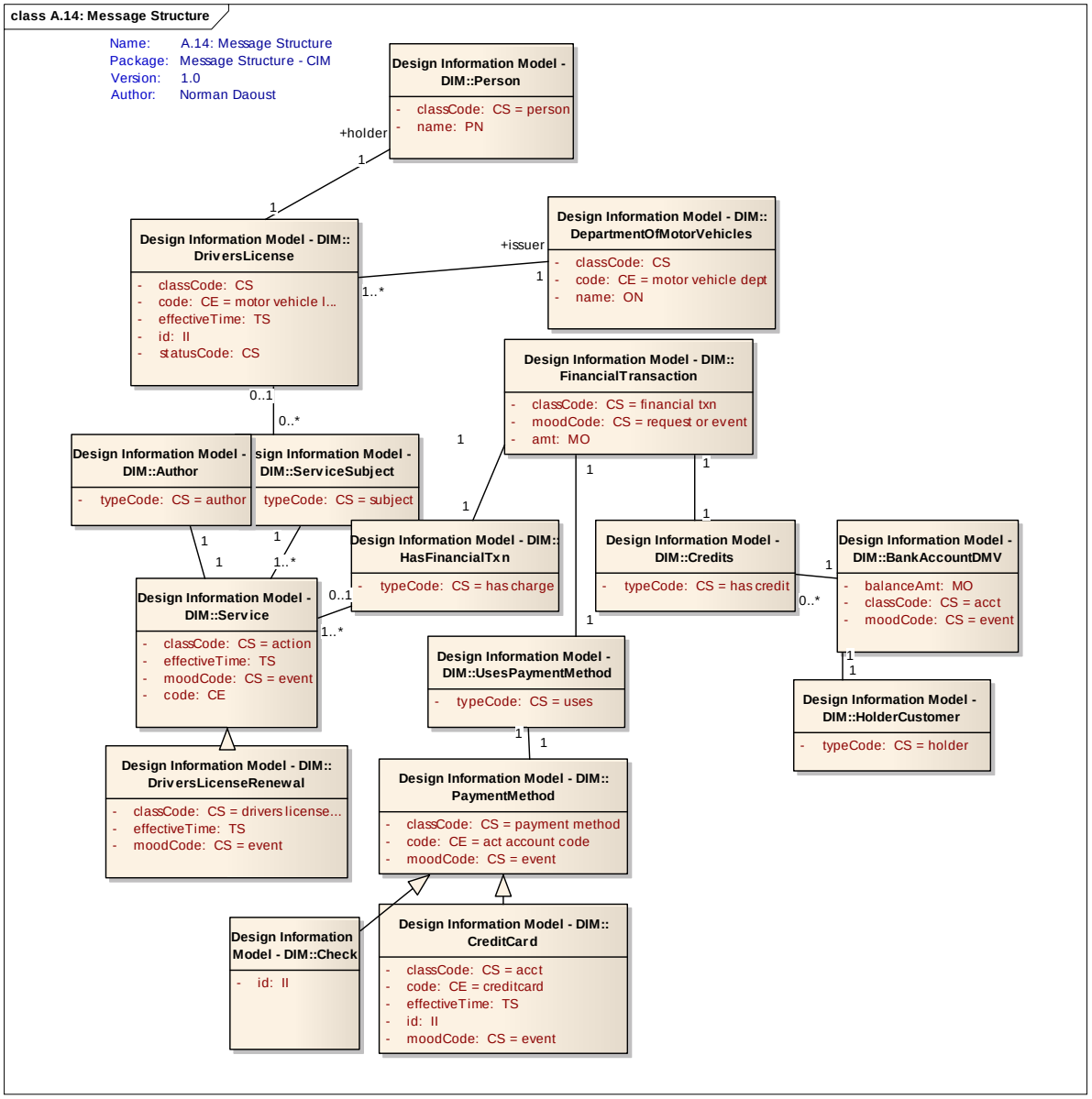

Figure A.14: Message Structure

Message Envelope

Localization - LIM

Dynamic Model

The following shows the allowed states of a focal business object. These state transitions correspond to business triggers for interactions.

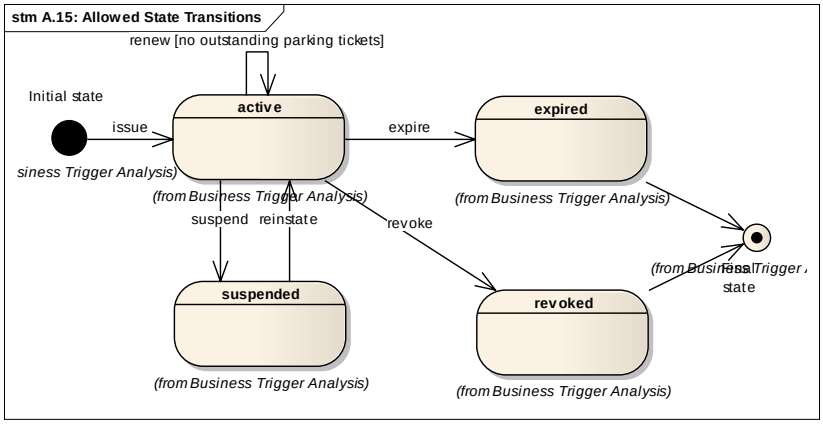

Figure A.15: Allowed State Transitions

The following diagram is describing the structure of application roles developed for an HL7 standard design.

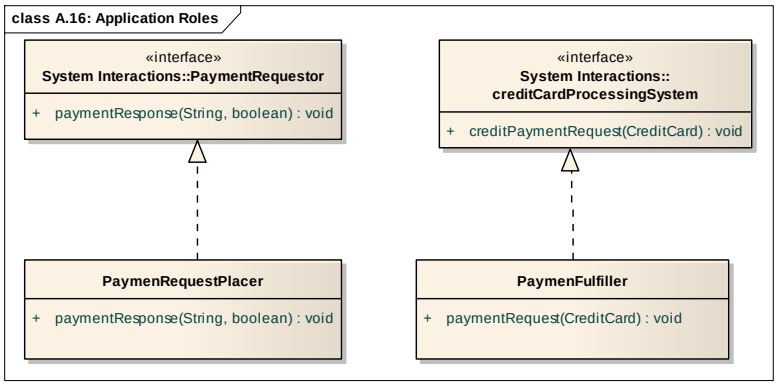

Figure A.16: Application Roles

The following diagram described system interactions based on state transitions and declared application roles.

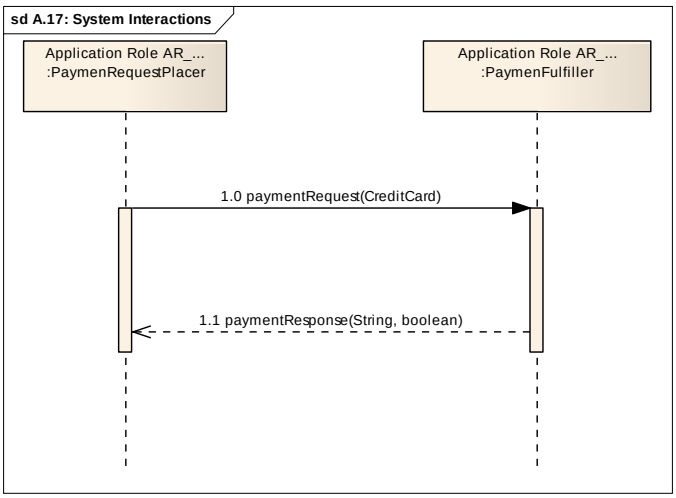

Figure A.17: System Interactions

Functional Model

System

Technology-Specific Artifacts

BPEL

WSDL

XSD

Annex B. System Interaction Examples

The following diagram propose several diagrams or view types to describe system interactions.

The following diagram shows how a sequence diagram may be used to describe the system interactions and receiver responsibilities.

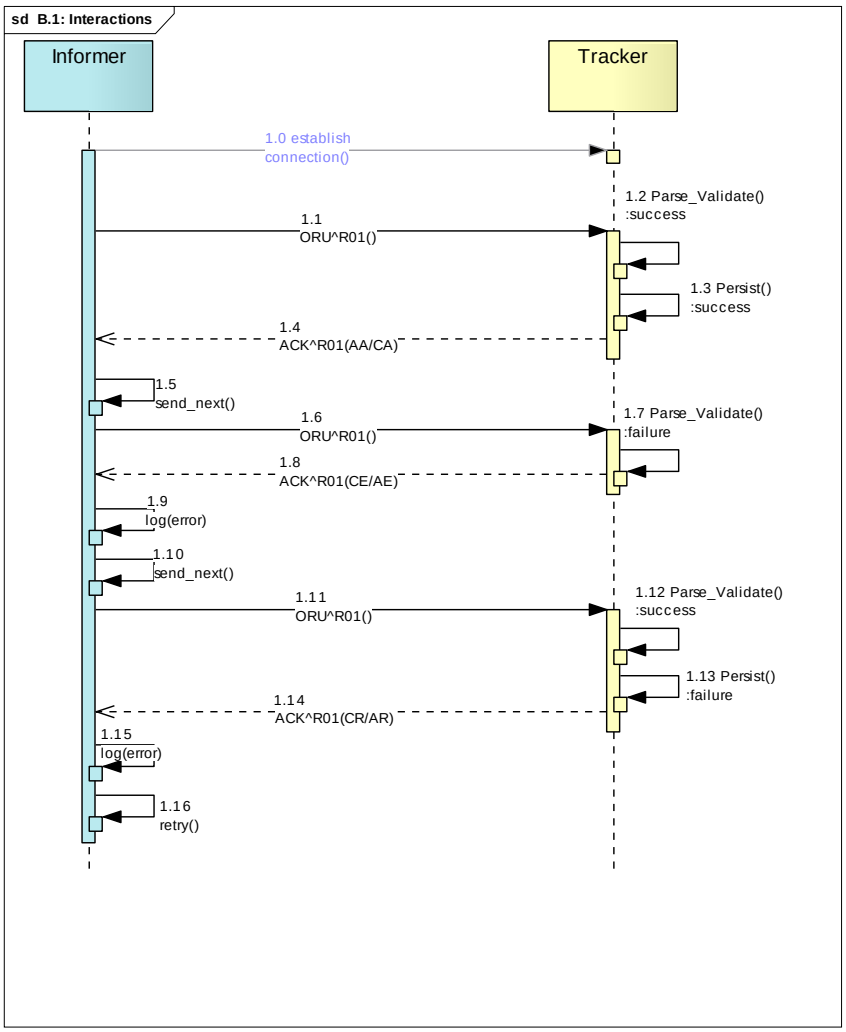

Figure B.1: Interactions

	The following demonstrates the use of sequence diagram to document system interactions and simple receiver processing rules:

Annex C. References

This annex contains external references, template documents, and other artifacts that are required for projects.

• HL7 Version 3.0 Publication Guide:
http://www.hl7.org/v3ballot/html/help/pfg/pfg.htm

• Refinement, Constraint and Localization, Release 2
http://www.hl7.org/v3ballot/html/infrastructure/conformance/conformance.htm

Glossary

Term	Description
Application Role	System interfaces are also known as "application roles" or "service interfaces". An application role is a collection of communication responsibilities intended to be implemented as an atomic 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 its parent, with additional or more specialized responsibilities added, or they may be the merging together of other application roles acting as components. It is the intent of HL7 that healthcare systems be able to declare conformance to the HL7 specification by creating an implementation profile that identifies the application roles supported by that implementation.
Association	A relationship between two or more entities. Implies a connection of some type - for example one entity uses the services of another or one entity is connected to another over a network link.
Balloting strategy	Multiple domains or infrastructure committees may be involved in a project. The project plan should include a plan for engagement of impacted committees and a plan for advancing new or refined content into processes/ballots that are already underway in those committees.
Candidate Standard validation	Executing the Candidate Standard validation approach. HL7 will have a modified open approach to candidate standard validation. All those participants that made a non-binding commitment in step (5) will be included if they choose to honor the commitment. Others may be added to achieve a balance or for other necessities for validation. The previous notwithstanding, HL7 will limit the number of participants to ensure a manageable process and reasonable time frame.
Candidate Standard validation approach	A project step that ensures that the Candidate Standard is validated by external industry resources before it is finalized as a normative standard. Where the standard is for interoperability, it is expected that the validation will include at least two independent entities (vendors, user organizations, etc.) building trial implementations and testing them together. Where the standard serves another purpose the validation approach will involve a trial effort to use the draft standard in the manner for which it was created. At the planning stage the entities willing to test must make a non-binding declaration of their intent to participate in validation. Without such a declaration the project should not be initiated. Comment: This is expected to be a significant hurdle for new project initiation. At the same time it helps to assure that HL7 member resources will be concentrated on efforts that have a good shot at industry adoption
Class	A logical entity encapsulating data and behavior. A class is a template for an object - the class is the design, the object the runtime instance.
Component Model	The component model provides a detailed view of the various hardware and software components that make up the proposed system. It shows both where these components reside and how they inter-relate with other components. Component requirements detail what responsibilities a component has to supply functionality or behavior within the system.
Conformance Statement	A conformance statement is a claim that the behavior of an application or application module agrees with the constraints stated in one or more profiles. A Conformance Statement is documentation of the degree to which a particular application conforms to the specification. Part of that document will

	be a profile expressing the requirements relevant to a particular standard. Standard Profiling is based upon the consistent application of constraints to a set of base specifications. This document outlines the processes that govern the definition of profiles and conformance statements.
Deployment Architecture	A view of the proposed hardware that will make up the new system, together with the physical components that will execute on that hardware. Includes specifications for machine, operating system, network links, backup units &etc.
Deployment Model	A model of the system as it will be physically deployed
Extends Relationship	A relationship between two use cases in which one use case 'extends' the behavior of another. Typically this represents optional behavior in a use case scenario - for example a user may optionally request a list or report at some point in a performing a business use case.
HL7 Profile	An HL7 profile is an unambiguous specification of one or more HL7 standards that have been analyzed for a particular use case. It prescribes a set of precise constraints upon one or more standard HL7 artifacts. An HL7 profile is conformant, in all aspects, with the HL7 defined specification used in the profile according to the constraints or extension rules. It may specify constraints on the standard HL7 definition. An implementation profile fully describes an interoperability interaction between two or more systems through the combination of the following: a) one use case analysis, b) one or more dynamic definitions, and c) one or more static definitions.
HL7 project criteria	HL7 projects shall: • Be consistent with Board strategic direction or, if a hosted project, approved as an exception project • Include appropriate project documentation - project charter, scope, resources, timelines, assumptions, constraints, planned deliverables, etc. per PMO methodology • Be aligned with market demand • Be sponsored by stakeholders intending to implement the product produced by the project • Define a reasonable balloting strategy to meet market demand and implementation timelines • Define how the project will engage with other impacted committees • Follow project approval protocols to ensure appropriate project socialization and sign-off has taken place
Includes Relationship	A relationship between two use cases in which one use case 'includes' the behavior. This is indicated where there a specific business use cases which are used from many other places - for example updating a train record may be part of many larger business processes.
Industry outreach	Depending on the goals of the project this may be as little as a set of announcements of work going on in HL7 targeted at the impacted stakeholder communities. For some projects it may involve scheduling out-of-cycle meetings, scheduling meetings jointly with other stakeholder organizations or some kind of “Town Hall” meetings similar to those used for the EHR Functional Requirements DSTU.
Interface	System interfaces are also known as "application roles" or "service interfaces".
Non-volunteer resources	Beyond the routine support, HL7 headquarters provides for balloting, etc., additional support may be required to assess potential funding requirements. Hosted projects will be expected to provide associated funding

Project Management Office (PMO)	The PMO is staffed by HL7 to provide support for committees executing projects. The PMO is responsible for oversight of project methodology, and tracking the status of HL7 projects. The PMO does not directly manage product related projects.
Quality Criteria	A project commitment to a measure of the quality for each step of the project cycle. It is expected that most projects will use or possibly adapt boiler-plate quality criteria developed as part of HL7's methodology
Quality Review	An evaluation of whether the work products of a step meet the pre-established quality criteria. At most steps the project team will self-assess against these criteria and take a vote (not a ballot) to move ahead to the next step.
SME	Subject Matter Expert or Domain Expert: a person with domain knowledge that represent the users of IT systems and their business needs.
Stakeholder	A person or a company that requests a new standard, a technical correction to an existing standard, or an enhancement.
Substantive Change	A change that requires ballot. The Architectural Review Board (ARB) provides criteria to determine whether or not a change is substantive. HL7 Definition of Substantive The ARB considers that a change to the standard is substantive if it would cause an interface sender or receiver to have its interface fail when a newly specified message was received or attempted to be sent. This is similar to, but more expansive than, the definition of backwards compatibility. In general, for backwards compatibility, a message receiver is expected to receive a new message and be able to ignore added material. On the other hand, if there is new material that needs to be parsed in order to process the message given its new definition, the change is substantial. On the other hand, if a change would create a backwards compatibility problem (as defined in Chapter 2), it is substantive by definition.
Timeline	A GANNT chart or other project diagram that expresses the estimated times required to complete the steps of the project, synchronized with the dates of HL7 trimester meetings.
Use Case	A Use Case represents a discrete unit of interaction between a user (human or machine) and the system. A Use Case is a single unit of meaningful work; for example creating a train, modifying a train and creating orders are all Use Cases. Each Use Case has a description which describes the functionality that will be built in the proposed system. A Use Case may 'include' another Use Case's functionality or 'extend' another Use Case with its own behavior. Use Cases are typically related to 'actors'. An actor is a human or machine entity that interacts with the system to perform meaningful work.