Clinical Interoperability Council (CIC) Working Group

Clinical Domain Analysis Models

Handbook for Developing a Domain Analysis Model
Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>By</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3</td>
<td>October 23, 2007</td>
<td>Mead Walker</td>
<td>Initial Draft</td>
</tr>
<tr>
<td>0.6</td>
<td>October 21, 2010</td>
<td>Mead Walker</td>
<td>Update document to include full set of diagram types. Add description of model components.</td>
</tr>
<tr>
<td>0.7</td>
<td>January 25, 2011</td>
<td>Mead Walker</td>
<td>Address comments from reviewers</td>
</tr>
<tr>
<td>0.8</td>
<td>February 21, 2011</td>
<td>Anita Walden</td>
<td>Added additional support sections</td>
</tr>
<tr>
<td>1.0</td>
<td>January, 2012</td>
<td>Anita Walden</td>
<td>Voted on as Final Document</td>
</tr>
</tbody>
</table>
# Table of Contents

1. **INTRODUCTION** ................................................................. 3
2. **OBJECTIVE** ........................................................................... 5
   2.1 **WHO SHOULD READ THIS DOCUMENT** ...................... 5
   2.2 **THE ROLES OF A DAM** .............................................. 5
   2.3 **RELATIONSHIP TO HL7’S SERVICE AWARE INTEROPERABILITY FRAMEWORK** .......... 6
3. **DAM DEFINITION** ............................................................... 8
4. **RESOURCES** ....................................................................... 9
   4.1 **EDUCATIONAL SOURCES** .......................................... 9
   4.2 **PREVIOUS HL7 DAM PROJECTS (NOT AN EXHAUSTIVE LIST)** ................... 9
5. **TOOLING** .......................................................................... 10
   5.1 **CLINICAL EXPERTS TOOLING** .................................. 10
   5.2 **MODELING TOOLING** ................................................. 10
6. **DOMAIN ANALYSIS MODEL CONSTITUENTS** ..................... 11
   6.1 **DATA ELEMENTS** ..................................................... 12
      6.1.1 **Relationships** ................................................... 13
   6.2 **STORY BOARDS** ....................................................... 13
      6.2.1 **Relationships** ................................................... 13
   6.3 **USE CASES** ............................................................... 14
      6.3.1 **Relationships** ................................................... 14
   6.4 **ACTIVITIES** ............................................................... 14
      6.4.1 **Relationships** ................................................... 14
   6.5 **CLASSES, ATTRIBUTES** ............................................ 15
      6.5.1 **Relationships** ................................................... 15
   6.6 **STATE MACHINES** ..................................................... 15
      6.6.1 **Relationships** ................................................... 16
   6.7 **INTERACTIONS** .......................................................... 16
      6.7.1 **Relationships** ................................................... 16
7. **A WALKTHROUGH OF THE MODEL CONSTITUENTS** ........ 17
   7.1 **DATA ELEMENTS** ..................................................... 17
      7.1.1 **Data Element** ................................................... 17
      7.1.2 **Name** ............................................................... 17
DAM Construction Manual

7.1.3 Lists of allowable values ................................................................. 18

7.2 STORYBOARDS ..................................................................................... 19
7.2.1 Story Board ............................................................................. 19
7.2.2 Storyboard Example ............................................................... 19

7.3 USE CASES ......................................................................................... 20
7.3.1 Use Case Diagram ................................................................. 20
7.3.2 Diagram Example ................................................................. 21
7.3.3 Use Case .................................................................................. 21
7.3.4 Actor ....................................................................................... 22
7.3.5 Associative Elements ............................................................ 22
7.3.6 Use Case Association ............................................................ 23

7.4 ACTIVITY MODEL ............................................................................... 24
7.4.1 Activity ....................................................................................... 27
7.4.2 Swim Lane ............................................................................. 27
7.4.3 Information Object ................................................................. 28
7.4.4 Associative Elements ............................................................ 28
7.4.5 Control Elements ................................................................. 29

7.5 CLASS MODEL ..................................................................................... 30
7.5.1 Diagram Example ................................................................. 31
7.5.2 Package ..................................................................................... 32
7.5.3 Class ......................................................................................... 33
7.5.4 Attribute .................................................................................... 34
7.5.5 Data Type Specification ......................................................... 35
7.5.6 Vocabulary Constraint .......................................................... 36
7.5.7 Associative Elements ............................................................ 36

7.6 STATE MODEL ..................................................................................... 39
7.6.1 Example Diagram ................................................................. 39
7.6.2 State ......................................................................................... 40
7.6.3 State Transition ................................................................. 40
7.6.4 Initiation .................................................................................... 41
7.6.5 Termination ................................................................................ 41

7.7 INTERACTIONS ................................................................................... 41
7.7.1 Example Diagram ................................................................. 42
7.7.2 Actor ......................................................................................... 42
7.7.3 Interaction ............................................................................... 43

8 REQUIREMENTS GATHERING PROCESS & VALIDATION (VETTING) ............... 44
8.1.1 Define scope .................................................................................................................. 44
8.1.2 .......................................................................................................................... Error! Bookmark not defined.
8.1.3 Identify uses .................................................................................................................. 44
8.1.4 ........................................................................................................................ 44
8.1.5 Identify key stakeholders.................................................................................................. 44
8.1.6 ...................................................................................................................................... 44
8.1.7 Gather existing standards or obtain current data elements .................................................. 44
8.1.8 ...................................................................................................................................... 44
8.1.9 Project communication .................................................................................................. 44
8.1.10 ...................................................................................................................................... 44
8.1.11 ...................................................................................................................................... 44
8.1.12 Feedback mechanisms .................................................................................................. 45
8.1.13 ...................................................................................................................................... 45
8.1.14 Balloting ...................................................................................................................... 45
8.1.15 ...................................................................................................................................... 46

9 NEXT STEPS .......................................................................................................................... 46

10 REFERENCES: .................................................................................................................... 47
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1</td>
<td>DAM Foci and Objectives</td>
<td>6</td>
</tr>
<tr>
<td>Figure 2</td>
<td>DAM Components</td>
<td>11</td>
</tr>
<tr>
<td>Figure 3</td>
<td>Sample Use Case Diagram</td>
<td>21</td>
</tr>
<tr>
<td>Figure 4</td>
<td>Use Case</td>
<td>21</td>
</tr>
<tr>
<td>Figure 5</td>
<td>Actor</td>
<td>22</td>
</tr>
<tr>
<td>Figure 6</td>
<td>Actor Primary Participation</td>
<td>22</td>
</tr>
<tr>
<td>Figure 7</td>
<td>Actor Secondary Participation</td>
<td>22</td>
</tr>
<tr>
<td>Figure 8</td>
<td>Generalization Association</td>
<td>23</td>
</tr>
<tr>
<td>Figure 9</td>
<td>Include Association</td>
<td>23</td>
</tr>
<tr>
<td>Figure 10</td>
<td>Extend Association</td>
<td>24</td>
</tr>
<tr>
<td>Figure 11</td>
<td>Sample Activity Diagram</td>
<td>26</td>
</tr>
<tr>
<td>Figure 12</td>
<td>Activity</td>
<td>27</td>
</tr>
<tr>
<td>Figure 13</td>
<td>Swim Lane</td>
<td>28</td>
</tr>
<tr>
<td>Figure 14</td>
<td>Information Object</td>
<td>28</td>
</tr>
<tr>
<td>Figure 15</td>
<td>Control Flow</td>
<td>29</td>
</tr>
<tr>
<td>Figure 16</td>
<td>Information Flow</td>
<td>29</td>
</tr>
<tr>
<td>Figure 17</td>
<td>Fork/Join</td>
<td>30</td>
</tr>
<tr>
<td>Figure 18</td>
<td>Process Initiation</td>
<td>30</td>
</tr>
<tr>
<td>Figure 19</td>
<td>Process Termination</td>
<td>30</td>
</tr>
<tr>
<td>Figure 20</td>
<td>Sample Class Model</td>
<td>32</td>
</tr>
<tr>
<td>Figure 21</td>
<td>Package</td>
<td>33</td>
</tr>
<tr>
<td>Figure 22</td>
<td>Class</td>
<td>34</td>
</tr>
<tr>
<td>Figure 23</td>
<td>Attribute</td>
<td>34</td>
</tr>
<tr>
<td>Figure 24</td>
<td>Association</td>
<td>37</td>
</tr>
<tr>
<td>Figure 25</td>
<td>Generalization</td>
<td>38</td>
</tr>
<tr>
<td>Figure 26</td>
<td>Aggregation</td>
<td>39</td>
</tr>
<tr>
<td>Figure 27</td>
<td>Sample State Model</td>
<td>40</td>
</tr>
<tr>
<td>Figure 28</td>
<td>State</td>
<td>40</td>
</tr>
<tr>
<td>Figure 29</td>
<td>State Transition</td>
<td>41</td>
</tr>
<tr>
<td>Figure 30</td>
<td>Initiation</td>
<td>41</td>
</tr>
<tr>
<td>Figure 31</td>
<td>Termination</td>
<td>41</td>
</tr>
</tbody>
</table>
1 Introduction

This document is to provide HL7 CIC project teams with a guide to developing Domain Analysis Models.

The HL7 Clinical Interoperability Council has several Clinical Domain Analysis Modeling (DAM) projects in development and is co sponsors of DAM projects in other HL7 working groups. DAMs have recently been integrated in to the HL7 balloting process which has brought higher visibility to the content and structure of these models. As a result the CIC would like to provide those project teams who work with CIC guidance on developing DAMs for balloting within HL7.

The purpose of this document is to provide CIC DAM project groups with a guide on establishing a DAM project and the process for developing the DAM. Hopefully this guide will help others learn from the past projects and reducing and streamlining their design and development efforts. A secondary benefit of this document will provide DAMs with similar designs that will increase consistency for smoother harmonization and integration with other DAMs and HL7 artifacts.

This document is designed as an introduction to the constituent parts of a Domain Analysis Model (DAM), to the components of each DAM constituent, and to the diagrams that are used in documenting those constituents.

While the models discussed have an abstract definition within the UML, their representation in the clinical models is affected by the conventions of Enterprise Architect which is the software tool that has been chosen for use. The illustrative diagrams have been created using Enterprise Architect.

Development of a domain analysis model (DAM) is described as the initial stage of HL7 specifications development by HL7 modelers\(^1\). The goal is to develop a representation of the problem space for a new set of specifications that can be commented on, and discussed with subject matter experts without requiring them to become expert in the jargon and minutia of the standard. The HL7 documentation notes:

“During requirements documentation 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, in this case, refers to the problem space for the requirements.\(^2\)”

Broadly speaking, a DAM is constructed to illustrate the scope of a problem domain, and to introduce its information content, the parties involved in creating and managing the information, and the relevant behaviors of those parties. The goal is to document requirements in sufficient detail to guide the way for creation of HL7 specifications. IN some cases, a DAM has been used to introduce existing material to HL7, either to pave

\(^1\) HL7 Development Framework - HDF
\(^2\) HL7 Development Framework – HDF 1.3.2
the way for the creation of interoperability specifications, or to introduce content to the HL7 community for review and comment.
2 Objective

This document will provide project teams with in the CIC working group with training information available within HL7 for DAMs development, to outline general components of a DAM and how they are used, give direction on creating and naming data elements to promote standardization and to provide guidance to modelers who are developing DAMs.

2.1 Who Should Read this Document

The document is aimed at two distinct but related audiences:

- **Model Developers**: DAM developers will normally have experience with HL7 products and with the modeling process. They will use the document for information on model completeness and on modeling style. The discussion of the DAM model components, and of the relationship between them is particularly important.

- **Model Reviewers**: Reviewers can use the document as a primer on reading a UML based model, and on interpreting the diagrams that are provided for the DAM components.

It is important to note that busy subject matter experts who are often health care practitioners are a key audience for the models developed under the auspices of the Clinical Interoperability Council. In order to facilitate review, model diagrams should be focused on the key information that needs to be displayed. The names of items such as classes, attributes, activities should be consistent with the terms used in healthcare practice.

The DAMs that we produce have a non-IT audience with not much time to review. A key notion is that they do not have much time. It’s text needs to be succinct. In some cases, it will be best to share the DAM in sections?

2.2 The Roles of a DAM

There are different reasons for an HL7 Work Group to construct a DAM, and these reasons will have an impact on the final product. To date, there are several different major objectives for DAM construction. This is relevant because it can affect the choice of DAM components to focus on. The table offers perspectives on those objectives we have identified.
DAM Construction Manual

<table>
<thead>
<tr>
<th>Clarify data structure</th>
<th>Class Model</th>
<th>Provide guidance to developers of HL7 specifications: messages, documents, services</th>
</tr>
</thead>
<tbody>
<tr>
<td>List data elements</td>
<td>List of Data Elements</td>
<td>Provide a consensus set of definitions to be used at the point of data collection in healthcare and for secondary uses in registries, surveillance and research</td>
</tr>
<tr>
<td>Explicate Dynamic Behavior</td>
<td>Use Case Model, Activity Model</td>
<td>Clarify the roles of different parties involved in an area. Highlight the needed pattern of data exchange.</td>
</tr>
</tbody>
</table>

**Figure 1: DAM Foci and Objectives**

It is important to note that objectives may overlap. For one thing, virtually all, if not all, HL7 DAMs will seek to support specification development. For this reason, the class model component of the DAM is virtually mandatory.

### 2.3 Relationship to HL7’s Service Aware Interoperability Framework

The HL7 Service Aware Interoperability Framework (SAIF) has defined the Enterprise Conformance and Compliance Framework (ECCF) specification stack as a tool for organizing the products developed by HL7 work groups. The rows of the specification stack (“The specification stack (SS) is a 3-row by 4-column matrix that represents a collection of artifacts, which collectively and unambiguously defines the subject of the collection.”) are particularly relevant to DAM construction. The rows identify the following levels of abstraction:

- **Conceptual level (CIM):** also known as a business domain model. A model that uses a vocabulary that is familiar to the subject matter experts (SMEs). A model in which the “computational details” are either hidden or undetermined.

- **Logical level (PIM):** A model of a software or business system that is independent of the specific technological platform used to implement it. A model in which a system is defined independently of a platform in which it could be implemented.

- **Implementation level (PSM):** A model of a software or business system that is linked to a specific technological platform, language, operating system or database. A model in which the specific details needed to define implementation within the context of a particular platform are included.

---


4 Ibid. Page 10

As a general statement, the evolving HL7 architecture calls for (will call for), the content to be included in a specification to be developed on the conceptual (CIM) level before being published as an HL7 product – usually as a logical level (PIM) artifact. The actual implementation of the data exchange by users of HL7 specifications involves the creation of Implementation level (PSM) items.

A DAM is created at the conceptual level. It is a business domain model. However, when a DAM is being created it is quite possible for HL7 to have already created specifications – logical level (PIM). For example, a DAM that included the concept of laboratory observations would be occupying the same functional area as the Laboratory Observation specifications created in both Version 2 and Version 3. When a DAM is being created it is important to determine whether HL7 has already created specifications within the relevant functional area. If so, those specifications should be examined to determine the functional requirements they are based on. The constructed DAM should end up being consistent with balloted specifications that exist within a given functional area. If the Work Group developing the DAM feels it has uncovered errors in the existing specification, these should be specifically noted. This requirement is consistent with the general rule that, by preference, new HL7 specifications should refer to a DAM within the relevant functional area, and those specifications should be created by mapping that DAM to a form based on HL7’s Reference Information Model (RIM).
3 DAM Definition

This is an analysis model developed to improve communication between stakeholders from different organizations. This requirements document is used to formally define and structure a common set of information or processes for the purpose developing specification within HL7 (ie. Messaging, EHR models, DCMs,…).

According to the HL7 Development Framework Document, the Domain Analysis Model (DAM) is used as the basis for specification design. The design may be a functional model, a service specification, a message definition, etc., depending on the type standard targeted by the project. The
4 Resources

4.1 Educational Sources

The tutorials offered that related to the Domain Analysis Modeling Guide include:
Introduction to the HDF
Domain Analysis Modeling
Introduction to UML
Introduction to Project Insight
Introduction to V3

4.2 Previous HL7 DAM Projects (not an exhaustive list)

Cardiovascular DAM Balloted       May 2008
Tuberculosis DAM Balloted          Sept 2008
Clinical Trial Registration        May 2010
Vital Records                      May 2010
Emergency Services DAM             Sept 2010
5 Tooling

Below we describe some available software that has been used within the working group. These are suggestions:

5.1 Clinical Experts Tooling

Prior to modeling the content the information needs to be vetted with the clinical groups. Groups have used excel documents and mind maps to help them define the clinical domain. Free mind is a free mind mapping tool to help clinical experts map out their domain prior to modeling it in the class model.

http://freemind.en.softonic.com/

5.2 Modeling Tooling

There are various tools used to model DAMs.

Enterprise Architect is the tool used for most CIC projects because of the inexpensive license. You can download a read only tool for free.

www.sparxsystems.com.au
6 Domain Analysis Model Constituents

An HL7 Domain Analysis Model (DAM) is not simply an information model. The DAM may include multiple components as shown in the diagram and discussed below. This section discusses the component models or structures of a DAM, and the relationships between them.

It is important to note there is no requirement that a particular DAM has to include all the model constituents. The choice of which of these constituents are needed will be made by the work group developing the model, based on their specific circumstances. However, when a particular constituent is created, its form needs to follow the suggestions within this document. In addition, when multiple DAM constituents are created, e.g., class model & activity model, it is important to address the relationships between the different types of model in order that the DAM as a whole be internally consistent.

At the same time, it is possible that some model constituents will have been developed outside of HL7, and have been brought to HL7 in order to inform creation of HL7 specifications. This will be most likely for data element lists, story boards, and collections of use cases. When this happens, the project team working on the DAM needs to consider the implications of the need to maintain consistency across the model, in particular in those cases where reviewers only focus on a portion of the DAM.

The diagram suggests some of the relationships between these model elements:

![Diagram of DAM Components]

Figure 2. DAM Components

However, it is important to note that the diagram could also be drawn as a spider web, linking each box to each of the others.
So far, clinical data model development has taken the creation of a class model as the primary focus. In a number of cases, the class model is based on a preexisting form or list of data elements representing material that stakeholders wish to bring into HL7. In most cases, functional requirements are also documented using storyboards or use case diagrams. These requirements are generally further elaborated using activity models.

When a DAM is constructed, the development team will need to determine which DAM constituents are to be constructed. (This topic is under discussion within the M&M and ARB work groups.) At this point, there is general agreement that, if an HL7 data exchange specification is to be produced, creation of the class model is required. This is also the case for submission of data elements to caDSR. However, the subject of developing DAMs to support creation of functional models for EHRs or for other application types still needs to be addressed.

It is important to stress that the DAM is a single model, not a collection of independent ones. Therefore, if multiple model constituents are created, it is essential to ensure consistency between them. This is a key issue for model integrity and completeness. The more strictly the contents of the individual constituents are aligned, the more powerful and useful the model will be when the time comes to create specifications. This is a topic that could addressed at great length. However, only the most critical and best understood interdependencies will be discussed. See the relevant sections within the discussion of the individual constituents below.

The notion of business rules needs more discussion.

“A Business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business. Business rules describe the operations, definitions and constraints that apply to an organization. Business rules can apply to people, processes, corporate behavior and computing systems in an organization, and are put in place to help the organization achieve its goals.”

It is clear that, in order to achieve interoperability in particular functional circumstances, in particular to fully address organizational dynamics, business rules need to be specified along with data definitions. At the same time, the subject is extremely broad. As we gain more experience with DAM construction, more specific guidelines will be developed in this area.

### 6.1 Data Elements

The DAM may contain a list of data elements. In many cases, this list will represent the starting point for the modeling effort. It may be a list of elements created by an

---

6 Wikipedia

7 In discussions leading up to this document, we have had questions about the status of the data element list. Is it part of the DAM or a separate thing? What are the requirements for keeping the data element list and the class model synchronized?
organization that is interested in the data supported within a particular domain. In other cases, a body of subject matter experts may have defined a list of data elements that are important within a defined context.

The data element list plays rather different roles depending on whether it is a pre-existing list of elements that is being brought into the HL7 environment, or whether it is being created as part of the DAM exercise. It can provide a starting point for the modeling effort. It can also provide a representation of the DAM content that is clearly accessible to subject matter experts.

### 6.1.1 Relationships

The following list of relationships with other DAM constituents is important:

- The set of included storyboards may define the scope within which the list of data elements is relevant and perhaps definitive.
- The content of the activity model may define the work flow during which the data within the list is created, modified, and used.
- The attributes within the class model should correspond directly with the contents of the data element list. At the attribute/data element level, there should be an explicit pointer from one to the other. Since the content of the data element list and the set of model attributes essentially represent the same information, consistency between the two elements is particularly essential. In practice, this means it is very advisable to find ways of deriving the data element list directly from the class model.

### 6.2 Story Boards

A storyboard is a tool for requirements definition and for indicating the scope of the DAM. It provides a narrative description of an illustrative sequence of events that is relevant for the model. The storyboards provide a reference to the work flow supported by the model, and to the information needed to support that work flow. A DAM should contain the set of storyboards that seem appropriate for demonstrating the model’s scope and content.

A storyboard provides an example of the functions and of the information that are relevant to the overall effort. It is an informal representation of the entire subject matter, and is expected to communicate the essence of the DAM domain to a broad audience.

### 6.2.1 Relationships

The following list of relationships with other DAM constituents is important:

- The storyboards indicate the scope of the model supported by the list of data elements. Critical data within the list is expected to appear in the storyboards.
- The storyboards indicate the scope around the relevant content supported by the classes and attributes of the class model. Critical data within the class model is expected to trace back to the storyboards. In particular, the nouns or names that
appear in the storyboard can be expected to turn up as classes or attributes of the class model.

- The storyboards indicate the scope of the workflow supported by the activity model. The key aspects of that workflow are expected to appear in the storyboards. Critical data within the activity model is expected to trace back to the storyboards. In particular, the verbs that appear in the storyboard can be expected to turn up as activities within the model.
- The storyboards indicate the range of discrete activities to be supported by the use case model.

### 6.3 Use Cases

A use case defines functions that need to be performed within the context of the DAM, by indicating the specific sequence of activities followed by a system in order to achieve an observable useful result.

The use cases in the model express the requirements of the system in a more rigorous way than is shown by the storyboard(s). In particular, a use case shows the relationship between actors, system activities, and the value provided by the system.

#### 6.3.1 Relationships

The following list of relationships with other DAM constituents is important:

- The storyboards define the scope within which use cases are defined. They also provide examples of the kinds of outcomes that the use cases are designed to accomplish.
- Consideration should be given to ensuring that swim lanes in the activity model correspond to the actors within the use case model.

### 6.4 Activities

The activity model provides a description of the sequence of activities needed to support the workflow of the DAM.

The activity model shows how the different activities that are significant within the model are related, and it indicates which parties (actors) are responsible for an activity. Given that it indicates the dependencies between related activities performed by different parties, it shows where information needs to be exchanged and indicates the behavioral underpinnings for new HL7 specifications.

#### 6.4.1 Relationships

The following list of relationships with other DAM constituents is important:
The documented storyboards indicate the scope within which the activities are modeled.

The information model specifies the data that is generated or used within the activities. Data objects that are defined within the activity model need to tie in with (be mapped to) the classes and attributes of the information model.

Consideration should be given to ensuring that swim lanes in the activity model correspond to the actors within the use case model.

Given that the interaction model, is essentially a focused subset of the activity model, roles within the interaction model need to correspond to swim lanes in the activity model. The data objects representing information passed between one party and another need to directly appear within the interaction model.

### 6.5 Classes, Attributes

The class model offers a more precise view of the information that is relevant than the list of data elements, since it (a) addresses the classes/entities that attributes are a characteristic of, and (b) provides information on the semantics and cardinality of the relationship between the data elements. In addition, given that the HL7 specifications are based on the Reference Information Model, it presents the information in a way that it can more easily be expressed as an HL7 standard.

The class model systematically includes and organizes the information content of the model. It provides the primary elements that will be included in an HL7 message or document specification. For service specifications, it contains, once RIM mappings have been completed, the information managed by the service.

#### 6.5.1 Relationships

The following list of relationships with other DAM constituents is important:

- The content of the information model must directly correspond to the list of data elements within the data element list.
- The scope of the information model is defined within the included set of storyboards.
- The work flow within the activity model illustrates the process by which data within the information model is created, retrieved, updated and/or deleted.
- The set of state transitions presented within the state model relates to a class, or multiple classes, defined within the information model.

### 6.6 State Machines

A model that specifies the sequence of states an object goes through during its lifetime in
response to events. The state model provides a focus on the behavior of individual, especially important, classes within the information model.

6.6.1 Relationships

The following list of relationships with other DAM constituents is important:

• The objects whose states are analyzed will correspond to a class within the information model.

6.7 Interactions

A model that clearly expresses the data exchanges required to support the activity pattern that has been defined. The need for an interaction is clearly apparent when an activity managed by a given party requires information generated in an activity managed by another party.

6.7.1 Relationships

The following list of relationships with other DAM constituents is important:

• The data exchanges exposed within this component are extracted from the activity model, and need to be consistent with it.

---

8 (At the current time none of the CIC inspired domain analysis models have developed state diagrams. It is likely that these diagrams are primarily relevant for further analyzing process. HL7 models have been mostly focused on discovering data requirements for subsequent mapping to the HL7 RIM. In this case, the state machines developed within the RIM will be used.)
7 A Walkthrough of the Model Constituents

This section provides more in depth discussion of the different constituents of a DAM, and offers help in interpreting the diagrams used for most of these constituents. The goals is to make these models easier to read for reviewers, and to offer some guidance to those who construct the models.

You will note that the diagrams shown here all are based on a single model – a representation of the process by which HL7 organizes specification development. The reader should look at the diagrams to get insights on how these diagrams are constructed, and on how the different diagrams within a single model offer different perspectives on a single construct. Please do not consider this material as an authoritative picture of how HL7 does its business. While we have tried not to introduce any errors, the material here is incomplete at best.

7.1 Data Elements

This section provides more detail on the data element list. It discusses the elements of the list, and the process for managing it. Note the ISO 1179 Metadata Registry Standard contains additional relevant detail on this topic.

The list of data elements must contain data element names and descriptions. It may contain an indication of element data types, and, for coded elements, a listing of allowable code values. There is no diagrammatic representation for the data element list.

7.1.1 Data Element

The data element is the primary component in the list. A data element is a “unit of data for which the definition, identification, representation and permissible values are specified by means of a set of attributes.”

However, in order to achieve more complete definition, the component also includes a listing of valid data types and a list of valid values for coded elements.

7.1.2 Name

A text name for the data element. It is expected that the conventions for the name will be set by the stakeholder group, so no formal naming rules are suggested

Note: if the list of data elements can be derived from the attributes of the class model, then name is likely to be the only property of the data element list that needs definition.

---


Also, in Wikipedia: “In metadata, the term data element is an atomic unit of data that has precise meaning or precise semantics….In the areas of databases and data systems more generally a data element is a concept forming part of a data model. As an element of data representation, a collection of data elements forms a data structure
The individual data element will pick up its description, data type, and associated listing of allowable values from the content of the class model.

### 7.1.2.1 Description
A text description of the element. The description should be concise, and should clearly indicate the meaning of the element.

### 7.1.2.2 Data type
An indication of the type of value that is valid as an instance of the data element. E.g., text, date. Definition of a data type for a data element is a critical aspect of the data element’s definition. However, if the DAM is constructed so that each data element is directly tied to an attribute in the class model – which is highly recommended – then the type does not have to be specified since it will already be known based on the attribute.

Since the DAM is an HL7 developed item, and since it most likely will be used to develop HL7 messages, documents or services, it is a good practice to rely on HL7’s already developed data type definitions. This eases the task of the DAM developer by relying on a pre-existing and well documented set of data types.\(^\text{10}\) It also eases the down the road transition to an HL7 specification. If any types are to be used that are not already defined within the HL7 definition, they need to be defined in a data types section within this component. The level of definition should be sufficient to support later mapping to one of the HL7 data types.

### 7.1.2.3 caDSR Considerations
Many clinically based DAM projects are developing lists of data elements that will be imported into the National Cancer Institute’s (NCI) Cancer Data Standards Registry and Repository (caDSR).

_A discussion of the requirements imposed when the data elements from the DAM are to be imported into caDSR will be included._

### 7.1.3 Lists of allowable values
If a data element is coded, if it draws its value from a constrained list, either the list of valid values for that data element should be provided or a reference to the appropriate code system (aka vocabulary, terminology, ontology).

---

\(^{10}\) Currently, Release 2 of the HL7 data type definitions should be used. The reader should refer to Data Types – Abstract Specification, to be found in the January 2011 ballot package.
7.2 Storyboards

This section provides more detail on the storyboards. It discusses the elements of this DAM component, and the process for managing it. There is no diagrammatic representation for the storyboards.

7.2.1 Story Board

The model may contain one or many storyboards, based on the needs and inclination of the modelers. A storyboard is defined as:

“A narrative of relevant events that serves to communicate a requirement or requirements of a domain to stakeholders, or that illustrates in a concrete fashion a significant activity or activities that lie within the scope of the model domain.” 11

An individual storyboard must contain a storyboard name, and the storyboard content. The storyboard content is a narrative that provides an example of the salient features of the domain.

7.2.1.1 Name

The storyboard name is a descriptive title for the storyboard.

7.2.1.2 Content

The content includes text which describes an illustrative situation within the scope of the DAM. It will normally be based on characters with stereotypical names, e.g., Doug Doctor, describe an exemplary activity or activities, and contain references to information that is important for the work flow being described.

7.2.2 Storyboard Example

Here is an example to show what a storyboard can look like:

Storyboard: Jonah Johns approaches the Home Affairs WG with a proposal to develop a specification to capture the information on a household shopping list so that a record of desired purchases can be sent to retail outlets. The WG agrees such a specification is needed. It forwards a project scope statement to the Technical Steering Committee, and starts to work. With an approved statement in hand, the committee develops a draft specification which is submitted for ballot. After successfully addressing all the comments received from balloters, the new specification is approved.

11 Loosely based on the description provided by the HL7 V3 Glossary.
7.3 Use Cases

This section provides more detail on a use case. It discusses the elements of a use case diagram, and the process for managing it. A use case diagram consists of use cases, the relationships between use cases, and the actors that participate in the use case. The use case itself has a name and a description that includes the relevant sequence of events.

7.3.1 Use Case Diagram

A use case diagram shows the responsibilities of a system and the roles that are involved in interacting with the system to achieve a result. The formal definition notes:

“A use case diagram shows a set of use cases and actors (a special kind of class) and their relationships. You apply use case diagrams to illustrate the static use case view of a system. Use case diagrams are especially important in organizing and modeling the behaviors of a system.”

A use case describes a set of action sequences – typically a normal sequence and the significant variants – that are performed by a system and that yield an outcome that provides value to an actor.

A use case can be either composite or leaf. If it is a composite, it is an aggregate of one or more use cases. A leaf use case must have an associated primary participant actor. If desired, the actors for leaf use cases may be shown on a composite use case.

The construction of use cases is a good way to start a DAM, since use case diagrams provide a good way to define the scope of the model. When the functional orientation of the model is important (almost always), it will then be useful to amplify and explain the content of the use case through development of an activity model – this becomes almost a graphic depiction of the storyboard.

---

7.3.2 Diagram Example

The diagram below shows a toy use case diagram. The goal is to expose the different components of the model in a familiar case. Please note that the model, while having familiar components, is not warranted to be accurate, that is not its purpose.

![Sample Use Case Diagram](image)

Figure 3. Sample Use Case Diagram

A use case model includes use cases, actors, and the relationships between them. The primary goal of the diagram is to show, at a high level, the scope of a project, the major activities within that scope, and the types of party who are involved. In this diagram, the actor participation with the arrow head indicates the party which initiates the use case.

7.3.3 Use Case

A use case describes a process that leads to an outcome providing value to an actor. On the diagram, the use case appears as an oval.

![Use Case](image)

Figure 4: Use Case

Use cases are documented with:

- **Name**: A descriptive name for the use case.
- **Description**: The use case description describes the action sequence or sequences that make up the use case. It must include the normal sequence at least. It may also include one or more variant sequences. For each sequence, its outcome should be described, and the series of steps leading up to it should be included as well. The description does not appear on the diagram.
7.3.4 Actor

An actor indicates a role (or, in some cases, a participation) played by an entity – most typically a person or organization (it could also be a device) – that takes part and/or gets value from the use case. On the diagram, the actor appears as the stick figure of a person.

![Figure 5: Actor](image)

Actors are documented with:

- **Name**: A descriptive name for the actor.
- **Description**: A text description of the actor. The description does not appear on the diagram.

7.3.5 Associative Elements

7.3.5.1 Actor Participation

This is an indication of the actor’s participation in a use case. It is important to note whether the actor is the primary participant or a secondary participant in the use case. The primary participant is the actor that initiates the use case. The primary actor of a leaf level use case is the actor that initiates the use case. On the diagram, the participation appears as a line drawn between the actor and the use case:

![Figure 6: Actor Primary Participation](image)

![Figure 7: Actor Secondary Participation](image)
The primary participation is indicated by providing an arrowhead where the association connects with the use case. A leaf level use case has one and only one primary participant.

### 7.3.6 Use Case Association

It is an indication of the relationship between a pair of use cases. A use case association is a relationship between two use cases; a source and target use case. Three kinds of relationships between use cases have been identified:

**Generalization:** A generalization association is used when the source use case is a special case of the more general target use case. In a diagram, it appears as –

![Figure 8: Generalization Association](image1)

**Include:** An include association is used when the target use case is an unconditionally executed set of actions that form a meaningful sub-component of a larger source use case. In a diagram, it appears as –

![Figure 9: Include Association](image2)

---

13 Within the Enterprise Architect tool, the arrowhead is provided by adding navigability to the association between the use case and the actor.
**Extend**: An extend association is used when the source use case is a conditionally executed set of actions that form a meaningful sub-component of a larger target use case. In a diagram, it appears as –

![Extend Association Diagram](image)

In the examples shown here:

- The process of creating a messaging specification is a special case of that of creating an HL7 specification. There are other kinds of specifications that could be created, e.g., document definition, functional model, service definition.
- Creation of any and all HL7 specifications has to include the development of a project scope statement, and its review by the Technical Steering Committee.
- Some HL7 specifications are balloted, others remain internal or informative.

### 7.4 Activity Model

An activity diagram is used to show the different activities that need to be carried out to accomplish the goals of a system (or higher level activity). It also shows the organization and sequencing of those activities. More formally, it is defined as:

“A diagram that shows the flow from activity to activity; activity diagrams address the dynamic view of a system. A special case of a state diagram in which all or most of the states are activity states and in which all or most of the transitions are triggered by completion of the activities in the source states.” (The Unified Modeling Language User Guide, P. 457)

Within the clinical models, the activity diagram shows the various activities that are carried out to provide a diagnosis, or to treat a disease.

Activities within the model must be named and have descriptions. The different roles that carry out activities are identified as swim lanes. The activity diagram illustrates the sequence of activities and the points of decision at which alternate work flows may be
followed.
One question that is raised is whether there is a difference between an activity model and the traditional flow diagram. Essentially, the activity model covers the same ground as a flow diagram, however, UML adds features that allow for a more precise representation. We do expect that clinical groups will come up with the flow of the patient. The model developer needs to understand the content of the activity being examined.

Creation of the activity model is a central feature of DAM construction. When the activity model is constructed, it is best for its content to be integrated with the content of the other DAM constituents:

- The actors which appear in the use case, are represented as swim lanes.
- Data objects in the activity model link to the content of the information model.

**Diagram Example**

Figure 11: Sample Activity Diagram
An activity model includes activities, and it shows the control flow between these activities. It may also show object flows, and document the triggering or recognition of events. The activities may be organized into swim lanes. These notions are further described below.

### 7.4.1 Activity

An activity describes a task that needs to be done. To put it another way, an activity “is an ongoing non-anatomic execution within a state machine” (The Unified Modeling Language User Guide, P. 259).

On a diagram, an activity appears as a rectangle with rounded edges.

![Figure 12: Activity](image)

Activities are documented with:

- **Name**: A text name for the activity. Note, an activity can either be a composite or a leaf in the structure. There is a naming convention that is used to distinguish the two. A composite activity will have a name that is formed by a noun followed by a very, e.g., Manage Procedure, while a leaf activity will have a name that is formed using a verb followed by a noun, e.g., administer anesthesia.

- **Description**: A text description of the activity. The description should characterize the activity, and, if it is not a simple unitary thing, should indicate the various steps that make it up. It is not shown on a diagram. Note, the name of the party which is responsible for the activity, the actor, is not shown in the use case, it can be grasped from the swim lane the activity sits in.

- **Guards**: If there is more than one control flow exiting an activity, the Guard, i.e., “SUBSTANTIVE CHANGE” shown above, indicates the conditions under which that particular control flow is used to transition to the next activity.

### 7.4.2 Swim Lane

Swim lanes make it possible to partition the activities into groups; each group represents the organization (actor) responsible for those activities.

In Enterprise Architect, a swim lane can be linked to an instance classifier – this allows you to trace swim lane back to an actor in a use case.

On a diagram, swim lanes are shown by vertically partitioning the diagram into different areas, each is a swim lane.
Swim lanes are documented with:

- **Name**: A text name for the swim lane. It identifies the actor responsible for the included activities.

### 7.4.3 Information Object

On occasion, within an activity model, we need to represent an exchange between parties, in other words, across swim lanes. The information object represents data that needs to flow from one party to another in order to accomplish the use case being modeled. Given that HL7 specifications are largely oriented around information exchange, the information objects within the activity diagrams are especially interesting and important.

Information objects are documented with:

- **Name**: A text name for the information object. It provides some information with regard to the content of the contained information. Later, it will be important to make clear how the content of the object is represented within the information model.

### 7.4.4 Associative Elements

The activities within the model are associated by directed associations known as flows. A flow is a transition inbound to or outbound from an activity. A flow may be a control flow or an information flow.

#### 7.4.4.1 Control Flow

A control flow indicates the link between two activities. It always has both a source and destination activity.

A control flow indicates the sequencing of activities, and of decisions. When you review an activity diagram, it is common to trace the sequence of activities, and the consequences of decisions by following the control flows.

On a diagram, control flows are shown as arrows, with the arrowhead connecting to the destination activity, and the arrow foot connecting to the originating activity.
A control flow may have a guard (a predicate statement which can be evaluated as true or false). A guarded control flow can only be traversed if the associated guard predicate statement is true.

### 7.4.4.2 Information Flow

An information flow is a connection between an activity and an information flow object. A source activity’s outbound information flow always has a destination information flow object. A destination activity’s inbound information flow always has a source information flow object.

The information flow has text that indicates whether it is an outbound flow – from an activity, or an inbound flow – from an activity.

### 7.4.5 Control Elements

These are items included in the diagram to improve its overall coherence.

#### 7.4.5.1 Fork/Join

In some cases, it is useful to provide some control over the sequencing of control flows. In particular, to indicate that multiple control flows are either generated or processed at the same time. (This structure is also known as a synchronization bar.) Forks are used to split an outgoing flow into multiple flows leading to different activities. The point of the structure is to note that each receiving activity will be initiated as a result of a single outcome from the originating activity. Conversely, a join merges multiple flows into a single flow leading to a single activity. The goal here is to make it clear when that the receiving activity starts it will make use of the result of all of the control flows from activities that took place.

On a diagram a fork/join is indicated by a heavy horizontal or vertical bar to which control flows are linked.
Fork/Joins have no particular associated documentation.

### 7.4.5.2 Process Initiation

Any activity needs to have a beginning. For the activity modeled within an activity diagram, this is signified by a process initiation. This is shown on a diagram as a filled in circle.

![Figure 18: Process Initiation](image)

### 7.4.5.3 Process Termination

The termination of the activity modeled by the diagram is shown by a process termination symbol. Sometimes it may have multiple ways of terminating. This is shown on the diagram as an empty circle with a dot in the middle.

![Figure 19: Process Termination](image)

An activity diagram may also have a symbol known as a “flow final”. It resembles the Activity Final except it has an X through it. It is used to indicate an intermediate termination to the activity flow.

### 7.5 Class Model

This section provides more detail on the class model. It discusses the elements of the model, and the process for managing it. The class model consists of attributes organized into classes, which are linked by associations, and categorized within folders. (Note, the content of the model is described extensively elsewhere.)
A class model is used to organize and define the static information that is relevant for a given system or activity. The formal definition states:

“A diagram that shows a set of classes, interfaces, and collaborations and their relationships; class diagrams address the static design view of a system.” 14

It is important to note that there are different perspectives from which a class model may be developed. One perspective is that of a conceptual model. In such a model

“you draw a diagram that represents the concepts in the domain under study. These concepts will naturally relate to the classes that implement them, but there is often no direct mapping. Indeed, a conceptual model should be drawn with little or no regard for the software that might implement it, so it can be considered language independent.”15

A conceptual model can be contrasted to a specifications model that defines the interfaces for software that will implement a system, or to an implementation model that shows the design of an object oriented software application. It is important to keep in mind that the clinical domain analysis models are conceptual models.

7.5.1 Diagram Example

The diagram below shows a toy class model. The goal is to expose the different components of the model in a familiar case (Please note that the model, while having familiar components, is not warranted to be accurate, that is not its purpose. In fact, it is not at all complete.)

14 The Unified Modeling Language User Guide, P. 259
15 UML Distilled, Page 55
It is important to note that this data structure – HL7 Specification – also appears within the activity model.

7.5.2 Package

A package is used to organize a model into subsections. Within the clinical domain analysis models, we normally provide a single diagram for each package, although a given diagram may contain classes from different packages. More formally, a package is defined as: “A general-purpose mechanism for organizing elements into groups.” (The Unified Modeling Language User Guide, P. 464).

Within a class diagram, the name of a class may be preceded by the name of its package, with “::” separating the two. Note that, within Enterprise Architect, both diagrams and classes may be placed in a specified package. When a class from another package is shown in a diagram, the package designation is included, but when both class and diagram are within the same package this label is omitted.

Packages are documented with a symbol resembling a file folder. Within Enterprise Architect, when you include a package on a diagram, the names of the included classes are listed within the package symbol.
Packages are documented with:

- A text name for the package\(^{16}\).

- Description: It may be useful to provide a description of the package. The
description ought to characterize the package, and provide the criteria for
including classes in a particular package. The description is not shown on
diagrams.

### 7.5.3 Class

A class is used to indicate some thing or concept that is important within the context of
the problem space being modeled. More formally:

“A class is an abstraction of things or concepts that are subjects of interest in
a given application domain. All things or concepts subsumed under a class
have the same properties and are subject to and conform to the same rules.
Classes are the people, places, roles, things, and events about which
information is kept.” (HL7 Version 3 Ballot Package, Version 3 Guide.)

Classes within the model are used to indicate the kinds of objects that are relevant. In
some cases, a class will be created specifically to highlight the relationship between two
concepts or things. For example, an address class may be created to show how a person
can have multiple addresses.

Within a class diagram, classes are represented using a rectangle with two horizontal
lines through it. The lines divide the class into three areas. The topmost area contains
the name of the class, the middle one has the names of the attributes, and the bottom one
contains methods or operations that can be performed on the class. Note, we have not
documented methods within the clinical domain analysis models, and, as a result, the
Enterprise Architect tool is not showing the lower section of the class that would indicate
its methods.

---

\(^{16}\) Refer to the BRIDG Information Model Style Guide for a discussion on naming conventions.
Classes are documented with:

- **Name**: A text name for the class\(^{17}\).
- **Description**: A text description that provides information about the class\(^{18}\).
- **Attribute List**: A listing of the attributes belonging to the class.

### 7.5.4 Attribute

An attribute describes a particular property of a class, it indicates a particular item of information that will be valued for an instance of the class\(^{19}\). Formally an attribute is defined as: “a named property of a classifier that describes a range of values that instances of the property may hold” (The Unified Modeling Language User Guide, P. 458).

Within a class diagram, attribute names are shown within the class they belong to. The attributes are included in the middle section of the class representation as text entries that show the attribute name, and the assigned data type.

Attributes are documented with:

- **Name**: A text name for the attribute
- **Description**: A text description that provides information about the attribute\(^{20}\) and provides relevant other information.
- **Data type**: A value that indicates the structural characteristics of the attribute. E.g., text, integer. The class model must include a list of valid data types for the model. It is recommended that data type values be drawn from the HL7 Abstract Data Type specification.

---

\(^{17}\) Refer to the BRIDG Information Model Style Guide for a discussion on naming conventions.

\(^{18}\) Refer to the BRIDG Information Model Style Guide for a discussion on the characteristics of a good description for a class.

\(^{19}\) Refer to the BRIDG Information Model Style Guide for a discussion on naming conventions.

\(^{20}\) Refer to the BRIDG Information Model Style Guide for a discussion on the characteristics of a good description for an attribute.
- **Multiplicity**: An indication of whether or not an attribute may repeat. This feature is supported in Enterprise Architect and uses the same multiplicity (cardinality) indicators as are used for associations – see below for details. However, for models that will be imported into caDSR, this feature should not be used.\(^{21}\)

- **Vocabulary Designation**: A way of indicating the list of allowable values (vocabulary) that an instance value is drawn from. This notion is only relevant for coded attributes. It is not shown on diagrams. In the clinical Domain Analysis models, we are implementing this concept using the Tagged Value feature supported by the Enterprise Architect tool.

Currently there is consideration of including a “data element mapping” or “alternate attribute name” as a property of the attribute. This would be an indication of a source that the attribute is associated with, and would show the name of the associated attribute in the data element list. In the Tuberculosis Domain Analysis Model, this constraint documents the relationship between model attributes and the spreadsheet that was developed by the Tuberculosis Stakeholders Group. This feature is still under discussion, and it is not clear how to document it so that it will be included in the model import into caDSR.

One approach is to only include those enumerations that are needed to support the structure of the model.

### 7.5.5 Data Type Specification

The list of valid data types is used to provide context for the data types assigned to attributes within the model. When a DAM is mapped to the HL7 RIM to support development of HL7 specifications, it will be important to show how the data types used in the DAM relate to those more widely used in HL7. In fact, the data type specification that HL7 has produced is broadly based and used beyond the boundaries of HL7. The HL7 Version 3 Data Types Release 2 specification is being used in the BRIDG model, and those data types are supported by NCI’s caDSR. Therefore, we recommend that RAM developers make use of the data types already developed by HL7. If additional or different types are used, these must be documented within the body of DAM class model.

This list is required for the class model component of the DAM. However, if a set of HL7 data types is used, a reference to the relevant documentation can be substituted.\(^{22}\)

A data type specification needs to include the following:

- **Name**: A text name for the data type.
- **Description**: A text description of the data type.

\(^{22}\) Grahame Grieve has produced a UML data types EA model which can be imported.
- **Structure Definition**: If the data type is a composite type; if it includes distinct parts, the type used for each must be defined in the data type specification.

### 7.5.6 Vocabulary Constraint

The notion that there is a valid range of values for instances is important, and this is shown in two ways. The first is through the data type which provides a general constraint on values, e.g., integers only, dates or date/time representations. The second way is the use of vocabulary constraints. That is through defining a set of values or a vocabulary that values of the attribute must be drawn from. E.g., gender code, diagnosis code.

HL7 vocabulary principles include the notion of “coding strength”. This is a property that is assigned to a coded element to indicate whether or not a code set can be expanded to meet local implementation needs. That is, is it allowable to add concepts that go beyond the list already provided. So far, within the clinical DAMs already produced, this notion has been addressed by adding an additional data element to allow a text entry of items that are not included within the existing code set.

Within clinical domain analysis models, we are linking each coded attribute with a class in a special package named **Value Sets**. Each class in this package represents an allowable set of codes, and each valid code is represented as an attribute within the class. We use the stereotype “enumeration” to indicate this is a special type of class. This usage follows the development style mandated by the Cancer Data Standards Repository (caDSR).

*The discussion of how to meet caDSR requirements may need to change based on recent suggestions that have been made to NCI.*

### 7.5.7 Associative Elements

These are elements of the model that define the relationship between model elements.

Note, within a class model, all classes must be contained within at least one folder. In addition, each attribute is contained within a class.

There are three types of associative element that are discussed below:

- **Association**
- **Generalization**
- **Aggregation**

The following characteristics can be asserted for an associative element.

- **Multiplicity**: A symbolic or numeric indication of the number of class instances at one end of the association that can be associated with a single instance on the other end. Note, when a recursive association is being created – one that links one instance of a class with another instance of the same class – the association must be optional for both instances. Otherwise, a boundless hierarchical structure is implied.

- **Assertion**: The association needs to include text statements of the multiplicity
that is being specified between the classes that are being linked. The statement should follow the pattern: “Class A is associated with X to Y instances of class B. Class B is associated with X to Y instances of class B. “X” and “Y” may be integers, but usually take the values “zero”, “one”, “many”.

- **Label**: The label for an association is a text tag that asserts the meaning of the relationship between the two classes.

- **Role Name**: Role names are attached to the ends of the relationships – where they indicate the classes that are being linked. A role name is used as needed to indicate the role played by the class in the context of the relationship.

For class models that are to be imported into caDSR, the role naming standards of caDSR need to be recognized. In such models, if there is at most one instance of the class in the association – multiplicity is 0..1 or 1..1, then the role name is simply the name of the linked class or the name of the class and an associated role qualifier. However, if there can be more than one instance of the class, - multiplicity is 0..* or 1..*, then the role name has to end with the word “collection”. For these models, all relationships have to include role names at each end of the relationship.

### 7.5.7.1 Association

An association captures the relationships between instances of classes.

Within a class diagram, associations are show as a line connecting two classes:

![Association Diagram](image.png)

**Figure 24: Association**

Associations are documented with:

- **Multiplicities (aka cardinality)**: These show the ways in which instances can participate in relationships with each other. Multiplicity is indicated by a symbol on each end of the association, where it joins to an associated class. It shows whether any valid instance of a class has to have an association, and whether there can be more than one. The multiplicity of an association for a class is evaluated by looking at the multiplicity on the far end of the association. For example, in the snippet above, a household has zero to many automobiles. An automobile must be associated with a single household. Multiplicity is shown on diagrams. The multiplicity symbols should be read as followed:
  - 1 (aka 1..1): one and only one. All instances of the class must have the association.
  - 0..1: zero to one. No instance of the class can have more than one of the association, but it may not have any.
  - 1..*: one to may. All instances of the class must have at least one of the associations, but it may have more.
0..*: zero to many. An instance of the class may have as many or few associations as desired

- **Role Names**: Generically, within the UML, role is associated with the link between an association and its class (so there are two of them). The formal description states that a role is “the behavior of an entity participating in a particular context.”

- **Association Label**: It provides a description of the semantic content of the association. That is to say, it characterizes the reason for identifying this association between two classes. On the diagram, it appears as a text tag centrally located along the association.

### 7.5.7.2 Generalization Relationship

The generalization relationship (aka generalization/specialization) is used to indicate that one class is a more specialized version of another. Conceptually, it indicates that any instance of the specialized class is, by definition, an instance of its generalization. It is relevant to note that, if you consider an instance of the specialized class, e.g. single person household, it will have the attributes and associations proper to its generalization as well as those shown for its class.

Within a class diagram, generalizations are show as a line with an arrowhead that points to the more general class.

![Generalization](image)

There is no particular documentation associated with a generalization.

### 7.5.7.3 Aggregation Relationship

The aggregation association captures the relationship between a whole and a related part. It is essentially a beefed up association with some additional semantics. It indicates that

---


24. However, in the clinical domain analysis models, role names echo the names of the associated class, and, if the association allows multiple instances of the class the class name has “collection” as a suffix. This usage follows the development style mandated by the Cancer Data Standards Repository (caDSR).
instances of one class are contained within the instances of the associated class.\textsuperscript{25} Within a class diagram, aggregations are shown as a line with an empty diamond-shaped head that points to the class representing the whole, or containing object.

![Figure 26: Aggregation](image)

Aggregation Relationships are documented with:

- **Multiplicities**: These operate the same way as do multiplicities for associations, and are shown on diagrams.

### 7.6 State Model

This section provides more detail on the state model(s). It discusses the elements of the model, and the process for managing it. The model includes a collection of described states, and the possible transitions between states. A state model is always based on a class within the class model. The state machine diagram should include the name of the state owning class.

Note, current DAMS do not include any state models. However, we expect, going forward, that these will be relevant. This will be especially true as DAMs starts to be more focused on behavioral modeling.

#### 7.6.1 Example Diagram

The diagram below shows a toy state diagram. The goal is to expose the different components of the model in a familiar case. Please note that the model, while having familiar components, is not warranted to be accurate, that is not its purpose.

\textsuperscript{25} Note, within the UML, you will also see the notion of composition. That is a more constrained form of aggregation in which the contained item cannot exist independently of its container.
These are discussed below in more detail.

### 7.6.2 State
A state is defined as” a condition or situation during the life of an object during which it satisfies some condition, performs some activity, or waits for some event.”\textsuperscript{26} A state is also described as a stage in the behavioral pattern of class. States may be nested; which implies that a class can be in multiple states concurrently - the nested state and the encapsulating state.

A state is documented with:
- **Name**: A text name for the state
- **Description**: A text description of the state.

### 7.6.3 State Transition
A transition is a progression from one state to another. It has a start and end state. An initial state transition can be from the null state (the default initial state for all

\textsuperscript{26} The Unified Modeling Language User’s Guide, Page 486.
A state machine may also include a final state. Only inbound transitions are allowed to the final state. Inbound and outbound transitions to states containing nested states imply a transition involving any of the nested states.

![In Reconciliation to Approved](image)

**Figure 29: State Transition**

A state transition is documented with:

- **Name**: A text name for the state transition.

### 7.6.4 Initiation

In a state model, it is relevant to show the state or states a class can take on when it is created or instantiated. Such states are shown by including a transition from the initiation point. Initiation is shown on a diagram as a filled in circle.

![Initial to Under Development](image)

**Figure 30: Initiation**

### 7.6.5 Termination

It is also relevant to show from which states an instance may be terminated. This is shown by including a transition to a terminal point. Termination is shown on the diagram as an empty circle with a dot in the middle. The flow final convention seen in the activity diagram can also be used.

![Approved to Final](image)

**Figure 31: Termination**

### 7.7 Interactions

This section provides more detail on the interactions model(s). This model is constructed to highlight the interchanges between the parties involved within the scope of the DAM as defined within the use case model. The UML provides two ways to model interactions: sequence diagrams and communication diagram. We have chosen to use the second because it is more intuitive. However, sequence diagrams are better suited for
showing the order in which things happen.

### 7.7.1 Example Diagram
The diagram below shows an example communication diagram. The goal is to highlight the data exchanges between the different parties identified in the use cases, and addressed in the activity diagrams.

![Communications Diagram](image)

Figure 32: Communications Diagram

The content of the diagram is further discussed below.

### 7.7.2 Actor
An actor – just as in the use cases - indicates a role (or, in some cases, a participation) played by an entity – most typically a person or organization (it could also be a device) – that takes part and/or gets value from the use case. On the diagram, the actor appears as the stick figure of a person.

![Actor](image)

Figure 33: Actor

Actors are documented with:

- **Name**: A descriptive name for the actor.
- **Description**: A text description of the actor. The description does not appear on the diagram.
7.7.3 Interaction

Interactions are documented with:

- **Name**: A descriptive name for the interaction. This will have already appeared within the Activity Diagram.
8 Requirements Gathering Process & Validation (Vetting)

8.1.1 Define scope
It is extremely important to define the purpose and mission of the DAM. Decide on the therapeutic area, sub specialty or function to help determine scope. It is recommended to develop a project scope statement to facilitate keeping the project on track.
Example Scope: Oncology- Breast Cancer Diagnosis

8.1.2 Identify uses
Identifying the uses of the model will define the scope of the domain. Facilitate a discussion with the stakeholders concerning their problem space and what issues need resolution. Use cases or story scenarios will also help explain and justify the project.
Example Uses:
• Create Electronic Triggers for Potential TB cases from ER to State or County public health offices.
• Development of a preoperative surgical checklist to be used in the UK

8.1.3 Identify key stakeholders
The stakeholders are most likely the organizations funding the projects but it is important to include a broader community of users, regulators, developers and standards development organizations. Representatives from these various groups can not only provide rich information but will most likely provide support and will use the product.

8.1.4 Gather existing standards or obtain current data elements
There are multiple methods in developing data elements for DAM creation with in the CIC working group. Leveraging existing information by gathering definitions from therapeutic guidelines, forms, published papers, national databases and current standards is recommended. Keep track of the source in which the information is obtained. There may be multiple definitions for one data element in which the stakeholders can decide on which definition is the preferred or if there is no definition one can be created by the stakeholders.

8.1.5 Project communication
Communication of the project status and activities is important to keep stakeholders engaged and the working group aware of important milestones and progress.
Methods of communication:
• HL7 CIC Working Group Wiki – Meeting materials and meeting minutes can be posted on the Wiki site
• Regular meetings with Stakeholders – an automatic conference call can be set up through HL7 teleconference center
• The project team facilitator should provide status updates to the broader CIC working group during the CIC monthly meetings and Working Group Meetings.
• Communication with stakeholders can be conducted through regular emails and web/teleconferences

8.1.6 Feedback mechanisms

Obtaining feedback throughout the development process is important to facilitate engagement from the stakeholder community and to ensure that the project is in alignment with CIC initiatives and processes.

Feedback can be obtained from the larger CIC working group and other HL7 working groups that have an interest in the project. It is encouraged that status and/or content presentations are made during the Patient Care Joint working group mtg (attended by PHER, CIC, EC & CH) at least once during the project.

All stakeholders may not be members of HL7 but may be critical to the success of the standard. Identify methods to gather feedback from those organizations and individuals prior to or concurrently with the HL7 balloting process.

8.1.7 Balloting

Currently DAMs are balloted as Informative.

Non-HL7 members can vote on the ballot. Please notify CIC co-chairs if you have individuals or organizations that are not HL7 members but want to vote on the standard.

Assisting with ballots- CIC has a publishing facilitator that can assist with the balloting process.

The project facilitator will work with the publishing facilitator to ensure the proper documents are posted for ballot.

Reconciliation of the comments can take place prior to the work group meeting or during the working group meeting. The project team may request that a CIC Co-chair help facilitate the reconciliation process and document decisions made. Official voting can take place if there is a quorum present in compliance with the working group decision making process.
9 Future Next Steps

• Mapping to the RIM as part of HL7 Specifications
• Loading into NCI’s Meta Data Repository and Registry
10 References:


CDISC, FDA, HL7, NCI; BRIDG Domain Analysis Static Model Style Guide, Release 001, October 25 2009

Enterprise Architect; Help contents for Version 7; Sparx Systems 2007


Fowler, Martin; Scott, Kendall; UML Distilled – Applying the Standard Object Modeling Language; Addison-Wesley 1997.

National Cancer Institute, Center for Bioinformatics; caCORE Software Development Kit 3.2.1, 2007

Notes: