**AIM 4.0 Model: Changes And Use Cases**

The following is a summary of changes since AIM version 3, revision 11. As of this writing, the current AIM version is 4, revision 23.

The most significant changes from the AIM 3.0 model to the 4.0 model are: a mandate from caBIG® to use ISO 21090 data types and annotation of findings using AIM Statements. An AIM statement describes a finding found on an image or series of images. AIM Statement represents a relationship between two AIM entities in the AIM 4.0 model: subject and object.

**Why AIM Statement?**

The AIM 3.0 UML model reflects relationships between classes using containment of one class in another, and inheritance such as IS-A relationships. The expressive power of the model was limited by these two types of relationships. No other types of relationships in the AIM 3.0 model are possible, and not all necessary relationships are present. For instance, there is no direct relationship between instances of AnatomicEntity class, e.g. right upper lobe of lung, and ImagingObservation class, e.g. mass. Such classes can be indirectly linked to each other only via Annotation class.

The desire to improve the expressiveness of the AIM model and specific use cases prompted us to create a flexible model of AIM Statements. Use cases that led us to change the AIM model from containment associations approach to explicitly stated relationships between two classes are as follows.

A. Justin Kirby at the NCI's Cancer Imaging Program has a use case for storing information from a mammography case report form (CRF). The CRF has "Associated Findings" or imaging observation characteristics that are associated with the entire breast and are not specific to a mass. Dr. David Channin at Guthrie Clinic also has a similar use case.

B. Dr. Lior Weizman, a research fellow working with Dr. Daniel Rubin, wants to associate calculation results with a DICOM segmentation object.

C. Several AIM users want to capture imaging observations and calculations related directly to image markup.

D. A user wants to capture a measurement of a liver volume from a CT scan to facilitate clinical assessment of liver disorders, to improve decision making in liver transplant surgery and to avoid donor-recipient graft mismatch.

Adding an association relationship between AnatomicEntity and ImagingObservationCharacteristic resolved the CRF issue. Adding an association between segmentation and calculation satisfied Dr. Weizman's comment. Applying the same approach could satisfy use cases C and D.

As AIM model is used by an increasing number of users, there will be additional requests to add new relationships between existing classes or to create new classes to store other important information related to AIM. Managing and rearranging associations with classes will be too complex to manage without the AIM Statement class and its subclasses.

**Summary of changes between AIM 3.0 to 4.0 models**

AIM 4.0 has fifty six new classes. We renamed twelve classes and deleted four classes. This document describes changes from AIM 3.0 to AIM 4.0 foundation information model.

1. AIM 4.0 uses ISO 21090 data types.

AIM 3.0 Data Type | ISO 21090

---------------------------------------------------

boolean | BL

CalculationResultIdentifier | Not Applicable

ComparisonOperators | Not Applicable

Date | TS

Double | REAL

Integer | INT or II

String | ST or Uid

Four attributes in AIM are now mapped to a single ISO 21090 CD data type.

AIM 3.0 Data Type | ISO 21090

---------------------------------------------------

codeValue | CD

codeMeaning | CD

codingSchemeDesignator | CD

codingSchemeVersion | CD

2. The new classes are described as follows.

2.1 AnnotationCollection

AIM 3.0 model does not support a collection concept. In AIM 3.0, each AIM annotation is stored as a single AIM XML document or AIM DICOM SR. A typical imaging study generates more than one AIM annotation. Managing AIM annotations of the same study becomes an extra activity that AIM implementers have to deal with. AIM 4.0 model has adopted the ability to store the same type of related AIM annotations as a single source. AIM has two types of annotations, image annotation and annotation of annotation. Image annotations annotate images. Annotation of annotations annotate other annotations, both image annotations and annotation of annotations. AIM 4.0, therefore, has a mechanism to manage collections of the same type of AIM annotations.

AnnotationCollection abstract class is the parent of ImageAnnotationCollection and AnnotationofAnnotationCollection. It provides the general concept that AIM may contain one or more instances of the same type. It associates with two optional classes used to capture information about a person, software, or software manufacturer that is generating AIM instances.

2.2 ImageAnnotationCollection

This class is one of two root classes in the model. It inherits all AnnotationCollection properties. This class signifies that all members of a collection are of type ImageAnnotaton.

2.3 AnnotationofAnnotationCollection

This class is the second root class in the model. It inherits allAnnotationCollection properties. This class signifies that all members of a collection are of type AnnotatonOfAnnotation.

AnnotationOfAnnotationStatement can be used to create statements about image annotations and annotation of annotations.

2.4 AnnotationStatement

Relationships in AIM 3.0 model are represented by inheritance or IS-A relationships and association relationships. These two expressions do not always precisely represent what AIM users want to state in annotations. AIM 4.0 introduces subject-predicate-object statement constructs, called AIM Statements, to precisely define relationships between two entities: a subject and an object. A subject and an object can come from a set of particular classes in the AIM model. The introduction of AIM statements requires structural and class name changes from the AIM 3.0 model. The following classes in AIM 3.0 model were renamed and can be used as a subjects and objects in AIM 4.0.

- Annotation was renamed to AnnotationEntity.

- AnnatomicEntity was renamed to ImagingPhysicalEntity.

- ImagingObservation was renamed to ImagingObservationEntity.

- Inference was renamed to InferenceEntity.

- AnnotationRole was renamed to AnnotationRoleEntity.

- GeometricShape was renamed to GeometricShapeEntity.

- TextAnnotation was renamed to TextAnnotationEnitty.

- ImageReference was renamed to ImageReferenceEntity.

- DICOMImageReference was renamed to DICOMImageReferenceEntity.

- WebImageReference was renamed to UriImageReferenceEntity.

- Calculation was renamed to CalculationEntity.

The classes listed above are derived from the Entity class. Prior relationships between many classes in the AIM 3.0 model have been deleted, see section 4 below for further information. Instead, an AIM Statement can be created by using the classes listed above as a subject and object of an AIM Statement. The naming convention used to create an AIM statement is a concatenation between subject, predicate and object. The current predicates in AIM are as follows.

- Excludes

- Has

- IsComparedWith

- IsCompriseOf

- IsFoundIn

- IsIdentifiedBy

- References

- Uses

AnnotationStatement class is the parent of ImageAnnotationStatement and AnnotationOfAnnotationStatement classes. It represents a general concept about a statement used to describe something found and to be addressed on an image or the same thing on images in a series. A statement concept expresses the most granular amount of information an AIM annotation can have. AIM annotation presents its content in a collection of semantic statements. Statements describe things found, measured and/or graphically annotated on an image. There are three types of statements: AnnotationStatement, AnnotationOfAnnotationStatement and ImageAnnotationStatement.

The AnnotationStatement class has seven different subtypes of annotation statements that can be applied to both AnnotationOfAnnotationStatement and ImageAnnotationStatement. The classes derived from AnnotationStatementcan are seen in the AIM 4.0 UML diagram section called "AnnotationStatement (common to both types of annotations)".

A statement class represents relationships via the use of the predicates listed above, inserted between subject and object. A subject and object associations can have a link between ImagingPhysicalEntity, ImagingObservationEntity, InferenceEntity, ReferencedAnnotationEntity, GeometricShapeEntity, TextAnnotationEntity, UriImageReferenceEntity, SegmentationEntity, DICOMImageReferenceEntity, ImageAnnotation or AnnotationOfAnnotation. Not all combinations between these classes are valid AIM statements. See the AnnotationStatement, AnnotationOfAnnotationStatement and ImageAnnotationStatement sections in the AIM 4.0 UML diagram for all valid AIM statements.

2.5 ImageAnnotationStatement

This class is a parent class of AIM statements that can only be applied to ImageAnnotation.

2.6 AnnotationOfAnnotationStatement

This class is a parent class of AIM statements that can only be applied to AnnotationOfAnnotation.

2.7 Entity

Entity abstract class represents the existence of a thing, concept, observation, calculation, measurement and graphical drawing in AIM. It is a parent class of all entities that subjects and objects of AIM Statements.

2.8 MarkupEntity

This abstract class represents graphical drawing and textual description that can be placed on an image.

2.9 CompactCalculaitonResult

The result of a calculation captured in a string format. The type attribute of the base class defines what kind of data format the string is captured as.

A string value of a calculation and its type are determined by the type attribute in this class, which inherits from the CalculationResult class. A type can be an array, binary, histogram, matrix, scalar, URI or vector. An encoding method is applied to the content of the value attribute. A compression method attribute can be used to define compression algorithm of the content of the value attribute in order to reduce the size of the value attribute.

2.10 ExtendedCalculationResult

This class stores a result of a calculation explicitly with the precise location of each element in the result. It supports sparse matrix type results.

2.11 Algorithm

The class provides well-defined instructions for arriving at results.

2.12 Parameter

The class represents a set of structured data to be used in the calculation or computation of an algorithm.

2.13 AdjudicationObservation

An observation is made about the comparison between two or more performers’ clinical results (i.e., “adjudication”). It describes a specific time point or all previous time points as well as those explicitly called out.

2.14 AuditTrail

It is used to capture any activity, in coded terms, that requires an entry in the audit trail log, including general and time point lesion observations, time point observations, etc.

2.15 SegmentationEntity

This is an abstract class representing the result of a segmentation process where digital images are segmented into a set of pixels. These pixels are stored in a separate file from the original images. Segmentation typically represents a region of interest.

**Classes derived from Annotation Statement -- common to both ImageAnnotation and AnnotationOfAnnotation:**

2.16 CalculationEntityReferencesCalculationEntityStatement

A calculation result can reference another calculation result without using its referenced calculation outcome.

A use case:

A user wants to store measurement results of left and right ventricular parameters from a cardiac MRI study.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create a calculation result to store left ventricular measurement.
4. Create a calculation result to store right ventricular measurement.
5. Create CalculationEntityReferencesCalculationEntityStatement. It links left and right ventricular measurements.

2.17 CalculationEntityUsesCalculationEntityStatement

A calculation result can use another calculation result for its own computation purposes.

A use case:

A user wants to store mean and standard deviation measurement results from a region of interest in a CT study.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create a markup, which is a region of interest of type polygon.
4. Create a calculation result to store mean value.
5. Create a calculation result to store standard deviation (SD) result. Note that the mean value is used to calculate SD.
6. Create CalculationEntityUsesCalculationEntityStatement. It uses SD calculation result as a subject and mean calculation result as an object of the statement.

2.18 ImagingObservationEntityHasCalculationEntityStatement

An image observation can have a calculation result associated with it.

A use case:

A user wants to measure a mass found on an image.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create an imaging observation containing mass, RID3874, RadLex.
4. Create a makeup of type line measurement. The line measurement has a length as a result.
5. Create a calculation result to store the length of the mass.
6. Create ImagingObservationEntityHasCalculationEntityStatement to link between the imaging observation (subject) and calculation results (object).

2.19 ImagingObservationEntityIsFoundInImagingPhysicalEntityStatement

An image observation can be found in an imaging physical entity.

A use case:

A user wants to state that a mass is found on the left upper lobe of the lung.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create an imaging observation containing mass, RID3874, RadLex.
4. Create an imaging physical entity containing left upper lobe, RID1327, RadLex.
5. Create ImagingObservationEntityIsFoundInImagingPhysicalEntityStatement to link the imaging observation (subject) and imaging physical entity (object).

2.20 ImagingPhysicalEntityHasCalculationEntityStatement

An imaging physical entity can have a calculation result.

A use case:

A user wants to store a diagnostic measurement of liver size.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create an imaging physical entity containing liver, RID58, RadLex.
4. Create a calculation result to store the size of liver.
5. Create ImagingPhysicalEntityHasCalculationEntityStatement to link imaging physical entity (subject) with calculation result (object).

2.21 ImagingPhysicalEntityHasImagingObservationEntityStatement

An image physical entity can have an imaging observation.

A use case:

A user wants to state that there is a mass in the left upper lobe of a patient.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create an imaging physical entity containing left upper lobe, RID1327, RadLex.
4. Create an imaging observation containing mass, RID3874, RadLex.
5. Create ImagingPhysicalEntityHasImagingObservationEntityStatement to link imaging physical entity (subject) with imaging observation (object).

2.22 ImagingPhysicalEntityHasImagingObservationCharacteristicStatement

An imaging physical entity has an imaging observation characteristic to describe associated findings with the imaging study or body part. This characteristic is not directly associated with an imaging observation, e.g. mass in the right upper lobe lung.

A use case:

A user wants to associate imaging observation characteristics with the entire breast.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create an imaging physical entity containing breast, RID28749, RadLex.
4. Create required number of imaging observation characteristics.
5. Create an ImagingPhysicalEntityHasImagingObservationCharacteristicStatement to link imaging physical entity (subject) with each imaging observation characteristic (object) from step 4.

**Classes derived from AnnotationOfAnnotationStatement:**

2.23 AnnotationOfAnnotationHasAnnotationRoleEntityStatement

A given instance of type AnnotationOfAnnotation can have an assigned role. Examples of roles can be baseline, follow-up, referenced case, etc. They are captured in the model as coded terms in AnnotationRoleEntity class. Some of these roles have been defined in DICOM standard part 16, Content Mapping Resource. For example:

Baseline Category:

DCM 112074 Target Lesion at Baseline

DCM 112075 Non-Target Lesion at Baseline

DCM 112076 Non-Lesion at Baseline

A use case:

A user wants to assign a baseline role to annotation of annotation consisting of image annotations of non-target lesion.

Assumption:

1. Image annotations for non-target lesion were created earlier.

Working with AIM:

1. Create annotation of annotation instance.
2. Create annotation role entity containing, Baseline Category, 112016, DCM.
3. The annotation of annotation has a statement referencing image annotations with non-target lesion.
4. Create AnnotationOfAnnotationHasAnnotationRoleEntityStatement to link the annotation of annotation (subject) to the annotation role entity (object).

*2.24 AnnotationOfAnnotationHasCalculationEntityStatement*

An instance of AnnotationOfAnnotation can have one or more calculation results. The instance can reference one calculation at a time in a statement. For example, if there are three calculation results, there must be three AnnotationOfAnnotationHasCalculationEntityStatements.

A use case:

An adjudicator wants to compute an average size of the mass from the three markups created by three readers.

Assumption:

1. Three image annotations for target lesion were created earlier from three different readers.
2. There is a system capable of reading and extracting information from the three annotations for further computational and manipulation purposes.

Working with AIM:

1. Create annotation of annotation instance.
2. Create three AnnotationOfAnnotationHasImageAnnotationStatement statements linking annotation of annotation (subjects) to image annotations (objects).
3. Extract the mass size from each image annotation and calculate an average.
4. Create CalculationEntity and store the average result calculation and other required information.
5. Create AnnotationOfAnnotationHasCalculationEntityStatement to link the annotation of annotation (subject) to the calculation entity (object).

*2.25 AnnotationOfAnnotationHasImageAnnotationStatement*

An instance of AnnotationOfAnnotation can reference existing image annotations. Results from image annotations can be used for further analysis, computation, comparison, reference, etc.

A use case:

An adjudicator wants to annotate a study read by three different readers.

Assumption:

1. Three image annotations for target lesion were created earlier from three different readers.
2. There is a system capable of reading and extracting information from the three annotations for further computational and manipulation purposes.

Working with AIM:

1. Create annotation of annotation instance.
2. Create three AnnotationOfAnnotationHasImageAnnotationStatements to link the annotation of annotation (subject) with each instance of image annotation (object).
3. The adjudicator may want to create additional imaging physical entity statements and imaging observation entity statements, etc.

*2.26 AnnotationOfAnnotationHasImagingObservationEntityStatement*

An instance of AnnotationOfAnnotation can have one or more imaging observations associated with the instance. AnnotationOfAnnotationHasImagingObservationEntityStatement expresses a relationship between the instance of AnnotationOfAnnotation and an imaging observation.

A use case:

An adjudicator wants to annotate imaging observations on a study read by three different readers.

Assumption:

1. Three image annotations for target lesion were created earlier from three different readers.
2. There is a system capable of reading and extracting information from the three annotations for further computational and manipulation purposes.

Working with AIM:

1. Create annotation of annotation instance.
2. Create three AnnotationOfAnnotationHasImageAnnotationStatement statements linking annotation of annotation (subject) to image annotations (objects).
3. Create ImagingObservationEntity and store related question and imaging observation result as well as optional information, see ImagingObservationEntity.
4. Create AnnotationOfAnnotationHasImagingObservationEntityStatement to link annotation of annotation (subject) to the imaging observation entity (object).

*2.27 AnnotationOfAnnotationHasImagingPhysicalEntityStatement*

An instance of AnnotationOfAnnotation can reference an imaging physical entity, an anatomical part or a physical object that can be identified on an image.

A use case:

An adjudicator wants to annotate imaging physical entity on a study read by three different readers.

Assumption:

1. Three image annotations for target lesion were created earlier from three different readers.
2. There is a system capable of reading and extracting information from the three annotations for further computational and manipulation purposes.

Working with AIM:

1. Create annotation of annotation instance.
2. Create three AnnotationOfAnnotationHasImageAnnotationStatement statements linking annotation of annotation (subject) to image annotations (objects).
3. Create ImagingphysicalEntity and store related question and imaging physical entity result as well as optional information, see ImagingPhysicalEntity.
4. Create AnnotationOfAnnotationHasImagingPhysicalEntityStatement to link annotation of annotation (subject) to the imaging physical entity (object).

*2.28 AnnotationOfAnnotationHasInferenceEntityStatement*

An instance of AnnotationOfAnnotation can have a conclusion derived by interpreting images and/or other supplemental information related to the images. The conclusion is stored in InferenceEntity.

A use case:

An adjudicator wants to provide a medical conclusion to a study read by three different readers.

Assumption:

1. Three image annotations for target lesion were created earlier from three different readers.
2. There is a system capable of reading and extracting information from the three annotations for further computational and manipulation purposes.

Working with AIM:

1. Create annotation of annotation instance.
2. Create three AnnotationOfAnnotationHasImageAnnotationStatement statements linking annotation of annotation (subject) to image annotations (object).
3. Create InferenceEntity and store related question and medical conclusion, see InferenceEntity.
4. Create AnnotationOfAnnotationHasInferenceEntityStatement to link annotation of annotation (subject) to the inference entity (object).

*2.29 AnnotationOfAnnotationIsComparedWithAnnotationOfAnnotationStatement*

An instance of AnnotationOfAnnotation can be used for comparison with another instance of AnnotationOfAnnotation. AIM users can further create CalculationEntityIsComparedWithCalculationEntityStatement to compare a calculation result from the subject AnnotationOfAnnotation instance with a calculation result from the object AnnotationOfAnnotation.

A use case:

An adjudicator wants to compare annotation of annotations’ calculation results from two different time points, baseline and first follow-up.

Assumption:

1. There are two annotation of annotations from two time points, baseline and the first follow-up.

2. There is a system capable of reading and extracting information from the annotations for further computational and manipulation purposes.

Working with AIM:

1. Create annotation of annotation instance.

2. Create two AnnotationOfAnnotationHasAnnotationOfAnnotationStatement statements linking newly created annotation of annotation (subject) to the two existing annotation of annotations (objects).

3. Create AnnotationOfAnnotationIsComparedWithAnnotationOfAnnotationStatement to link baseline (subject) and follow-up (object) instances.

4. Extract a calculation result from baseline and the first follow-up annotation of annotation instances.

5. Compare results.

6. Create and store calculation entity if comparing the difference between two sizes.

7. Create inference entity to store the result, e.g. smaller or larger what the question "Is the size getting larger or smaller?"

8. Create AnnotationOfAnnotationHasCalculationEntityStatement to associate the annotation of annotation (subject) with calculation entity (object).

9. Create AnnotationOfAnnotationHasInferenceEntityStatement to associate the annotation of annotation (subject) with inference entity (object).

*2.30 AnnotationOfAnnotationIsComparedWithImageAnnotationStatement*

An instance of AnnotationOfAnnotation can be used for comparison with an instance of ImageAnnotation. AIM users can further create CalculationEntityIsComparedWithCalculationEntityStatement to compare a calculation result from the subject AnnotationOfAnnotation instance with a calculation result from the object ImageAnnotation.

A use case:

An adjudicator wants to compare an annotation of annotation calculation result from baseline and image annotation created by a reader from first follow-up.

Assumption:

1. There is an annotation of annotation from the first time point and an image annotation from the first follow-up.
2. There is a system capable of reading and extracting information from the annotations for further computational and manipulation purposes.

Working with AIM:

1. Create annotation of annotation instance.
2. Create AnnotationOfAnnotationHasAnnotationOfAnnotationStatement linking annotation of annotation (subject) to the baseline annotation of annotation (object).
3. Create AnnotationOfAnnotationHasImageAnnotationStatement linking annotation of annotation (subject) to the first follow-up image annotation (object).
4. Create AnnotationOfAnnotationIsComparedWithImageAnnotationStatement to link baseline annotation of annotation (subject) with the follow-up image annotation (object).
5. Extract a calculation result from baseline annotation of annotation and first follow-up image annotation.
6. Compare results.
7. Create and store calculation entity if comparing the difference between two sizes.
8. Create inference entity to store the result, e.g. a result of “smaller” or “larger” when the question is, "Is the size getting larger or smaller?"
9. Create AnnotationOfAnnotationHasCalculationEntityStatement to associate the annotation of annotation (subject) with calculation entity (object).
10. Create AnnotationOfAnnotationHasInferenceEntityStatement to associate the annotation of annotation (subject) with inference entity (object).

*2.31 CalculationEntityIsComparedWithCalculationEntityStatement*

When an AIM user wants to compare two calculation results, the user can use CalculationEntityIsComparedWithCalculationEntityStatement to identify a subject and object of calculation results.

This statement should not exist alone. There should be a statement such as AnnotationOfAnnotationIsComparedWithAnnotationOfAnnotationStatement or AnnotationOfAnnotationIsComparedWithImageAnnotationStatement existing alongside the CalculationEntityIsComparedWithCalculationEntityStatement.

A use case:

An adjudicator wants to compare an annotation of annotation calculation result from the baseline and an image annotation created by a reader from the first follow-up.

Assumption:

1. There is an annotation of annotation from the first time point and image annotation from the first follow-up.
2. There is a system capable of reading and extracting information from the annotations for further computational and manipulation purposes.

Working with AIM:

1. Create annotation of annotation instance.
2. Create AnnotationOfAnnotationHasAnnotationOfAnnotationStatement linking annotation of annotation (subject) to the baseline annotation of annotation (object).
3. Create AnnotationOfAnnotationHasImageAnnotationStatement linking annotation of annotation to the first follow-up image annotation (object).
4. Create AnnotationOfAnnotationIsComparedWithImageAnnotationStatement to link baseline annotation of annotation (subject) with the follow-up image annotation (object).
5. Create CalculationEntityIsComparedWithCalculationEntityStatement to link the calculation from the baseline annotation of annotation (subject) with the calculation from the image annotation (object).
6. Extract a calculation results from the baseline annotation of annotation and the first follow-up image annotation.
7. Compare the results.
8. Create and store calculation entity if comparing the difference between two sizes.
9. Create inference entity to store the result, e.g. smaller or larger what the question "Is the size getting larger or smaller?"
10. Create AnnotationOfAnnotationHasCalculationEntityStatement to associate the annotation of annotation (subject) with calculation entity (object).
11. Create AnnotationOfAnnotationHasInferenceEntityStatement to associate the annotation of annotation (subject) with inference entity (object).

*2.32 ImageAnnotationHasAnnotationRoleEntityStatement*

This class is used to assign an annotation role to an image annotation. For example, an image annotation can be a baseline in one study. But, it also can be a follow-up in another study.

A use case:

A user wants to assign a baseline role to image annotation of non-target lesion.

Assumption:

1. An image annotation for non-target lesion was created earlier.

Working with AIM:

1. Create annotation of annotation instance.
2. Create annotation role entity containing, Baseline Category, 112016, DCM.
3. Create AnnotationOfAnnotationHasImageAnnotationStatement to link annotation of annotation (subject) with image annotation (object).
4. Create ImageAnnotationHasAnnotationRoleEntityStatement to link the image annotation (subject) to the annotation role entity (object).

*2.33 ImageAnnotationIsComparedWithImageAnnotationStatement*

An instance of ImageAnnotaion is compared with another instance of ImageAnnotation. AIM users can further create CalculationEntityIsComparedWithCalculationEntityStatement to compare a calculation result from the subject ImageAnnotation instance with a calculation result from the object ImageAnnotation.

A use case:

An adjudicator wants to compare image annotation calculation results from two different time points, baseline and the first follow-up.

Assumption:

1. There are two image annotations from two time points, baseline and the first follow-up.
2. There is a system capable of reading and extracting information from the annotations for further computational and manipulation purposes.

Working with AIM:

1. Create annotation of annotation instance.
2. Create two AnnotationOfAnnotationHasImageAnnotationStatement annotation of annotation statements to link the annotation of annotation (subject) to each image annotation instances (objects).
3. Create ImageAnnotationIsComparedWithImageAnnotationStatement to link baseline (subject) and follow-up (object) instances.
4. Extract calculation results from baseline and the first follow-up image annotation instances.
5. Compare the results.
6. Create and store calculation entity if comparing the difference between two sizes.
7. Create inference entity to store the result, e.g. smaller or larger with the question like "Is the size getting larger or smaller?", "PRV-SIZE01" and "Private".
8. Create AnnotationOfAnnotationHasCalculationEntityStatement to associate the annotation of annotation (subject) with calculation entity (object).
9. Create AnnotationOfAnnotationHasInferenceEntityStatement to associate the annotation of annotation (subject) with inference entity (object).

*2.34 ImageAnnotationIsComparedWithAnnotationOfAnnotationStatement*

An instance of ImageAnnotation is compared with another instance of AnnotationOfAnnotation. AIM users can further create CalculationEntityIsComparedWithCalculationEntityStatement to compare a calculation result from the subject AnnotationOfAnnotation instance with a calculation result from the object ImageAnnotation.

A use case:

A user wants to compare an image annotation calculation result with annotation of annotation calculation result created by an adjudicator.

Assumption:

1. There is an image annotation with a calculation result.
2. There is an annotation of annotation with a calculation result.
3. There is a system capable of reading and extracting information from the annotations for further computational and manipulation purposes.

Working with AIM:

1. Create annotation of annotation instance.
2. Create AnnotationOfAnnotationHasAnnotationOfAnnotationStatement to link the annotation of annotation (subject) with the referenced annotation of annotation (object).
3. Create AnnotationOfAnnotationHasImageAnnotationStatement to link the annotation of annotation (subject) with the instance of image annotation (object).
4. Create ImageAnnotationIsComparedWithAnnotationOfAnnotationStatement to link the image annotation (subject) with the referenced annotation of annotation (object).
5. Extract calculation results from annotation of annotation and image annotation.
6. Compare results.
7. Create and store calculation entity if comparing the difference between two sizes.
8. Create inference entity to store the result, e.g. smaller or larger what the question "Is the size getting larger or smaller?"
9. Create AnnotationOfAnnotationHasCalculationEntityStatement to associate the annotation of annotation (subject) with calculation entity (object).
10. Create AnnotationOfAnnotationHasInferenceEntityStatement to associate the annotation of annotation (subject) with inference entity (object).

**Classes derived from ImageAnnotationStatement**

*2.35 DICOMImageReferenceEntityHasCalculationEntityStatement*

A DICOM image can have a calculation associated with the image.

A use case:

A user wants to store a calculation result associated with a DICOM image. The calculation comes from a region of interest (ROI) of a mass. The user uses a free hand drawing tool to outline the mass region.

Working with AIM:

1. Create an image annotation instance.
2. Create DICOMImageReferenceEntity containing image with a mass.
3. Create ImageAnnotationHasDICOMImageReferenceEntityStatement to link the image annotation (subject) to the DICOM image reference entity (object).
4. The user creates ROI of the mass.
5. Compute area of the mass.
6. Create CalculationEntity and store the ROI result.
7. Create DICOMImageReferenceEntityHasCalculationEntityStatement to link DICOM image (subject) with the calculation entity (object).

*2.36 DICOMImageReferenceEntityHasImagingObservationEntityStatement*

A DICOM image, captured in DICOMImageReferenceEntity, can be associated with an imaging observation, captured in ImagingObservationEntity, to describe an observation on the image.

A use case:

A user wants to store an imaging observation associated with a DICOM image.

Working with AIM:

1. Create an image annotation instance.
2. Create DICOMImageReferenceEntity containing image with a mass.
3. Create ImageAnnotationHasDICOMImageReferenceEntityStatement to link the image annotation (subject) to the DICOM image reference entity (object).
4. The user creates ROI of the mass.
5. The user provides imaging observation with values mass, RID3874, RadLex.
6. Create ImagingObservationEntity and store the value above.
7. Create DICOMImageReferenceEntityHasImagingObservationEntityStatement to link DICOM image (subject) with the imaging observation entity (object).

*2.37 DICOMImageReferenceEntityHasImagingPhysicalEntity*

A DICOM image, captured in DICOMImageReferenceEntity, can associate with an image physical entity, captured in ImagingPhysicalEntity, to describe an observation on the image.

A use case:

A user wants to store an imaging physical entity associated with a DICOM image.

Working with AIM:

1. Create an image annotation instance.
2. Create DICOMImageReferenceEntity containing image with a mass.
3. Create ImageAnnotationHasDICOMImageReferenceEntityStatement to link the image annotation (subject) to the DICOM image reference entity (object).
4. The user creates ROI of the mass.
5. The user provides imaging physical entity with values left upper lobe, RID1327, RadLex.
6. Create ImagingPhysicalEntity and store the value above.
7. Create DICOMImageReferenceEntityHasImagingPhysicalEntity to link DICOM image (subject) with the imaging physical entity (object).

*2.38 DICOMSegmentationEntityHasImagingObservationEntityStatement*

A DICOM segmentation object can have an imaging observation to further describe the segmentation object.

A use case:

A user wants to associate a DICOM Segmentation with an imaging observation.

Assumption:

1. There is a system capable of creating a DICOM segmentation object based on a user ROI for a set of images in a series.

Working with AIM:

1. Create an image annotation instance.
2. Create DICOMImageReferenceEntity containing images with a mass.
3. Create ImageAnnotationHasDICOMImageReferenceEntityStatement to link the image annotation (subject) to the DICOM image reference entity (object).
4. The user designates ROI of the mass.
5. The system creates DICOM segmentation object.
6. Create DICOMSegmentationEntity with required information from step 5.
7. The user provides imaging observation with values mass, RID3874, RadLex.
8. Create ImagingObservationEntity and store the value above.
9. Create DICOMSegmentationEntityHasImagingObservationEntityStatement to link DICOM segmentation entity (subject) with the imaging observation entity (object).

*2.39 GeometricShapeEntityExcludesGeometricShapeEntityStatement*

A graphical drawing captured as GeometricShapeEntity can contain another graphical drawing that is used to subtract from the first graphical drawing. An example of how this statement is used is a donut shaped drawing where the total area of the enclosed object is subtracted from the area of the enclosing object.

A use case:

A user wants to exclude a region from another region.

Working with AIM:

1. Create an image annotation instance.
2. Create a markup of an area of enclosed a mass.
3. Create GeometricShapeEntity to store information in step 2.
4. Calculate area from step 3.
5. Create a markup of an area that the user wants to subtract from the enclosed mass.
6. Create GeometricShapeEntity to store information in step 5.
7. Calculate area from step 5 (optional).
8. Subtract 5 from 4 and store a calculation result in CaculcationEntity (optional).
9. Create GeometricShapeEntityExcludesGeometricShapeEntityStatement to link the first markup (subject) to the second markup (object).

*2.40 GeometricShapeEntityIsComprisedOfGeometricShapeEntityStatement*

Two or more graphical drawings captured as GeometricShapeEntity instances have a direct relationship with other graphical drawings when these drawings are placed on the same physical entity or thing.

A use case:

A user draws two different markups on the same lesion. This lesion is the same thing that appears on the same image.

Working with AIM:

1. Create an image annotation instance.
2. Create a markup of the first area of a mass.
3. Create GeometricShapeEntity to store information in step 2.
4. Calculate area from step 3 (optional).
5. Create a markup of the second area of the same mass.
6. Create GeometricShapeEntity to store information in step 5.
7. Calculate area from step 5.
8. Create GeometricShapeEntityIsComprisedOfGeometricShapeEntityStatement to link the first markup (subject) to the second markup (object).

*2.41 ImageAnnotationHasCalculationEntityStatement*

An image annotation can have a calculation result associated with it.

A use case:

A user wants to compute and store a size of the mass from an image.

Working with AIM:

1. Create image annotation instance.
2. Create a markup to measure the size of a mass.
3. Extract the mass size.
4. Create CalculationEntity and store the result and other required information from step 3.
5. Create ImageAnnotationHasCalculationEntityStatement to link the image annotation (subject) to the calculation entity (object).

*2.42 ImageAnnotationHasChildImageAnnotationStatement*

An image annotation can have a child image annotation. This represents a hierarchical structure between images. A child image implicitly has a parent image. A use case of this type of statement is a diagnostic mammogram image that has another magnification image associated with the diagnostic image. Another use case is a whole slide image with 10X magnification factor that may have a section of the image magnified and stored as a separate image. Both images can have a parent-child relationship.

A use case:

A user wants to link a mammography image with a corresponding magnification image.

Assumption:

1. There is a DICOMImageReferenceEntity containing original image.

Working with AIM:

1. Create image annotation instance.
2. User views a left mediolateral oblique (MLO)
3. User creates a magnification view of ROI and save it as a DICOM secondary captured image.
4. Create DICOMImageReferenceEntity containing secondary captured image.
5. Create ImageAnnotationHasChildImageAnnotationStatement to link the image annotation (subject) to an instance of DICOMImageReferenceEntity containing secondary captured image (object).

*2.43 ImageAnnotationHasDICOMImageReferenceEntityStatement*

An instance of image annotation can have DICOM images of the same series.

A use case:

A user views a set of CT axial series.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances using DICOMImageReferenceEntity.
3. Create ImageAnnotationHasDICOMImageReferenceEntityStatement to link image annotation (subject) and DICOM image reference (object).

*2.44 ImageAnnotationHasDICOMSegmentationEntityStatement*

An image annotation can have one or more DICOM segmentation objects. An ImageAnnotationHasDICOMSegmentationEntityStatement represents a direct relationship between an image annotation and DICOM segmentation entity. If the image annotation has three DICOM segmentation entities, there needs to be three ImageAnnotationHasDICOMSegmentationEntityStatements.

A use case:

A user found two nodules in a CT axial series. The user used a workstation to create two DICOM segmentation objects. The user annotated and stored the findings in AIM format with segmentation information.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. An image annotation has the set of CT images in the series. We need to create a statement that associates image annotation with DICOM images.

\*\* Use : ImageAnnotationHasDICOMImageReferenceEntityStatement

\*\* The statement has the image annotation instance as subject and DICOM image reference instance as object.

1. Create two DICOM segmentation entity instances. Each instance references DICOM segmentation object from step 2.
2. Create two statements of type ImageAnnotationHasDICOMSegmentationEntityStatement. Each statement has the same image annotation (subject) but different DICOM segmentation entity (object).

*2.45 ImageAnnotationHasGeometricShapeEntityStatement*

An instance of image annotation may have one or more graphical drawings associated with it.

ImageAnnotationHasGeometricShapeEntityStatement represents a direct relationship between an instance of image annotation and a drawing. If you have two drawings on an image, you will need to create two statements.

A use case:

A user outlines of a mass on an image from a set of CT axial series.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances using DICOMImageReferenceEntity.
3. Create a markup of a mass.
4. Create GeometricShapeEntity to store information from step 3.
5. Create ImageAnnotationHasGeometricShapeEntityStatement to link the image annotation (subject) with the geometric shape entity (object).

*2.46 ImageAnnotationHasImagingObservationEntityStatement*

An image annotation can have an imaging observation associated with it.

A use case:

A user wants to state that a mass is found on an image.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create an imaging observation containing mass, RID3874, RadLex.
4. Create ImageAnnotationHasImagingObservationEntityStatement to link the imaging annotation (subject) with imaging observation (object).

*2.47 ImageAnnotationHasImagingPhysicalEntityStatement*

An image annotation can have an imaging physical entity associated with it.

A use case:

A user wants to label left upper lobe of lung on an image.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create an imaging physical entity containing left upper lobe, RID1327, RadLex.
4. Create ImageAnnotationHasImagingPhysicalEntityStatement to link the imaging annotation (subject) and imaging physical entity (object).

*2.48 ImageAnnotationHasInferenceEntityStatement*

An image annotation can have a conclusion derived by interpreting images and/or other supplemental information related to the images.

A use case:

A user wants to provide medical conclusion for a study.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create InferenceEntity and store related question and medical conclusion, see InferenceEntity.
4. Create ImageAnnotationHasInferenceEntityStatement to link image annotation (subject) to the inference entity (object).

*2.49 ImageAnnotationHasTextAnnotationEntityStatement*

An image annotation can have a text annotation associated with it.

A use case:

A user creates a text markup on an image from a set of CT axial series.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances using DICOMImageReferenceEntity.
3. Create a text markup on a mass.
4. Create TextAnnotationEntity to store information in step 3.
5. Create ImageAnnotationHasTextAnnotationEntityStatement to link the image annotation (subject) with the text annotation entity (object).

*2.50 ImageAnnotationHasUriImageReferenceEntityStatement*

An image annotation can have a URI reference that directs to the actual storage location of the image used in the image annotation.

A use case:

A user views a set of CT axial series from URI sources.

Working with AIM:

1. Create image annotation instance.

2. Create UriImageReferenceEntity for each image in the CT series.

3. Create ImageAnnotationHasUriImageReferenceEntityStatements to link the image annotation (subject) with each UriImageReferenceEntity (object).

*2.51 ImagingObservationEntityIsIdentifiedByGeometricShapeEntityStatement*

An imaging observation can be identified by a graphical drawing.

A use case:

A user wants to create a markup for a mass found on an image.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create an imaging observation containing mass, RID3874, RadLex.
4. Create a markup of type line measurement. The line measurement has a length as a result.
5. Create geometric shape entity based on information from step 4.
6. Create ImagingObservationEntityIsIdentifiedByGeometricShapeEntityStatement to link between the imaging observation (subject) and geometric shape entity (object).

*2.52 ImagingObservationEntityIsIdentifiedByTextAnnotationEntityStatement*

An imaging observation can be identified by a text annotation.

A use case:

A user wants to create a text markup for a mass found on an image.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances.
3. Create an imaging observation containing mass, RID3874, RadLex.
4. Create a text markup on a mass.
5. Create text annotation entity based on information from step 4.
6. Create ImagingObservationEntityIsIdentifiedByTextAnnotationEntityStatement to link the imaging observation (subject) with text annotation entity (object).

*2.53 ImagingPhysicalEntityHasGeometricShapeEntityStatement*

An image physical entity can have a graphical drawing.

A use case:

A user creates a markup to label imaging physical entity from a set of CT axial series.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances using DICOMImageReferenceEntity.
3. Create an imaging physical entity containing left upper lobe, RID1327, RadLex.
4. Create a markup on a mass.
5. Create GeometricShapeEntity to store information in step 4.
6. Create ImagingPhysicalEntityHasGeometricShapeEntityStatement to link the imaging physical entity (subject) with the geometric shape entity (object).

*2.54 ImagingPhysicalEntityHasTextAnnotationEntityStatement*

An image physical entity can have a text annotation.

A use case:

A user creates a text markup to label imaging physical entity from a set of CT axial series.

Working with AIM:

1. Create image annotation instance.
2. Create DICOM image reference instance and its associated image study, image series and image instances using DICOMImageReferenceEntity.
3. Create an imaging physical entity containing left upper lobe, RID1327, RadLex.
4. Create a text markup on a mass.
5. Create TextAnnotationEntity to store information from step 4.
6. Create ImagingPhysicalEntityHasTextAnnotationEntityStatement to link the imaging physical entity (subject) with the text annotation entity (object).

*2.55 UriImageReferenceEntityHasImagingObservationEntityStatement*

A URI image reference can have an imaging observation. It is used in conjunction with UriImageReferenceHasImagingPhysicalEntityStatement.

A use case:

A user wants to state that a mass is found on an image.

Working with AIM:

1. Create image annotation instance.
2. Create UriImageReferenceEntity instance with the image location.
3. Create an imaging observation containing mass, RID3874, RadLex.
4. Create UriImageReferenceEntityHasImagingObservationEntityStatement to link the UriImageReferenceEntity (subject) with imaging observation (object).

*2.56 UriImageReferenceEntityHasImagingPhysicalEntityStatement*

An URI image reference can have an imaging physical entity associate with it.

A use case:

A user wants to label left upper lobe of lung on an image.

Working with AIM:

1. Create image annotation instance.
2. Create UriImageReferenceEntity instance with the image location.
3. Create an imaging physical entity containing left upper lobe, RID1327, RadLex.
4. Create UriImageReferenceEntityHasImagingPhysicalEntityStatement to link the UriImageReferenceEntity (subject) with the imaging physical entity (object).

*2.57 UriImageReferenceEntityHasCalculationEntityStatement*

A URI image reference can have a calculation associated with it.

A use case:

A user wants to compute and store a size of the mass from an image.

Working with AIM:

1. Create image annotation instance.
2. Create UriImageReferenceEntity instance with the image location.
3. Create a markup to measure the size of a mass.
4. Create CalculationEntity and store the size of the mass.
5. Create UriImageReferenceEntityHasCalculationEntityStatement to link the UriImageReferenceEntity (subject) with the calculation entity (object).

**3. We renamed the following classes.**

3.1 AnatomicEntity was renamed to ImagingPhysicalEntity.

ImagingPhysicalEntity is an entity or object that can be identified on an image. For medical imaging, it may represent anatomical location of an organ or body structure. Terms from controlled vocabulary such as RadLex, SNOMED CT, DCIOM, etc. are used to record the type of entity.

3.2 AnatomicEntityCharacteristic was renamed to ImagingPhysicalEntityCharacteristic.

ImagingPhysicalEntityCharacteristic are characteristics of imaging physical entities. These are in contradistinction to ImagingObservationCharacteristic. So, for example in a medical area, "dilated" might be an imaging physical entity characteristic of the "colon" imaging physical entity.

3.3 ImagingObservation was renamed to ImagingObservationEntity.

An ImagingObservationEntity is the description of things that are seen in an image. "Mass", "Pleural Effusion", "Foreign Body", and "Artifact", are all examples of imaging observation entity.

3.4 Inference was renamed to InferenceEntity.

A conclusion derived by interpreting images and/or other supplemental information related to the images such as medical history, geographic history, etc.

3.5 AnnotationRole was renamed to AnnotationRoleEntity.

AnnotationRoleEntity describes the role of referenced annotation.

3.6 ImageReference was renamed to ImageReferenceEntity.

ImageReference is an abstract class that references the image which is being annotated.

3.7 WebImageReference was renamed to UriImageReferenceEntity.

UriImageReferenceEntity is a source image for the annotation. It can be accessed via Intranet, Internet, local computer and/or file sharing systems.

3.8 DICOMImageReference was renamed to DICOMImageReferenceEntity.

DICOMImageReferenceEntity is a source image for the annotation.

3.9 Annotation was renamed to AnnotationEntity.

AnnotationEntity captures information that can be described, measured, calculated and drawn on images either by a human or machine observer.

3.10 TextAnnotation was renamed to TextAnnotationEntity.

TextAnnotationEntity represents the text and the markup of text intended to be rendered on the image.

3.11 GeometricShape was renamed to GeometricShapeEntity.

GeometricShapeEntity is the shape of a region of interest (ROI).

3.12 Segmentation was renamed to DICOMSegmentation.

DICOMSegmentation is a multi-frame image representing a classification of pixels in one or more referenced images. Segmentations are either binary or fractional. See DICOM part 3 Segmentation IOD for further information.

**4. Four classes were deleted since AIM statements and new classes cover the concepts represented by the deleted classes.**

4.1 ReferencedGeometricShape is represented by ImagingObservationEntityIsIdentifiedByGeometricShapeEntityStatement.

4.2 AimStatus is represented by AuditTrail.

4.3 ReferencedAnnotation is represented by AnnotationOfAnnotationHasAnnotationRoleEntityStatement and ImageAnnotationHasAnnotationRoleEntityStatement.

4.4 ReferencedCalculation is represented by CalculationEntityReferencesCalculationEntityStatement.

**5. New attributes have been added to the following classes.**

5.1 Inference

- questionTypeCode is used to collect coded entry data that describes the question being asked that is related to the typeCode attribute.

- isPresent, a boolean value indicating whether or not an inference exists in the observed images.

- label, a human readable description of the Inference.

5.2 GeometricShape

- label is a human readable description of the geometric shape.

- description is a free text about the geometric shape, not intended for rendering.

5.3 ImageSeries

- modality is the equipment used to acquire images of subjects or things, such as human and animal bodies.

5.4 Calculation

- questionTypeCode is used to collect coded entry data that describes the question being asked that is related to the typeCode attribute.

- imageIdentifierReference is a number that refers to an existing image identifier from ImageReference.

5.5 ImageStudy

- procedureDescription is information about the procedure being performed on a subject.

5.6 Scale

- type represents different types of scales that are Nominal, Ordinal or Ratio.

5.7 AnnotationRole

- questionTypeCode is used to collect coded entry data that describes the question being asked that is related to the typeCode attribute.

5.8 ImageReference

- imageIdentifier is a unique number within an ImageAnnotation object that uniquely identifies this object.

5.9 ImagingObservation

- imageIdentifierReference is a number that refers to an existing image identifier from ImageReference.

5.10 Annotation

- templateUid is a UID that references to an AIM template used to capture semantic meaning of pixel data, markup and calculation.

5.11 CalculationResult

- dataType is a coded entry data used to describe or capture a type of parameter. Coded data type can be a primitive programming data type such as integer, double, etc. as well as other data types such as URI.

**6. Attribute name change.**

6.1 typeCode attribute replaces a set of coded term attributes as a single attribute in a class. The replaced attributes include:

* codeValue
* codeMeaning
* codingSchemeDesignator
* codingSchemeVersion

These attributes are mapped to a single CD ISO 21090 data type. The effected AIM classes include:

* AnnotationEntity (formerly known as Annotation)
* AnnotationRoleEntity (formerly known as AnnotationRole)
* CalculationEntity (formerly known as Calculation)
* ImagingPhysicalEntity (formerly known as AnatomicEntity)
* ImagingPhysicalEntityCharacteristic (formerly known as AnatomicEntityCharacteristic)
* ImagingObservationEntity (formerly known as ImagingObservation)
* ImagingObservationEntityCharacteristic (formerly known as ImagingObservationCharacteristic)
* InferenceEntity (formerly known as Inference)
* NonQuantifiable