J Digit Imaging (2014) 27:692–701
DOI 10.1007/s10278-014-9710-3

The National Cancer Informatics Program (NCIP) Annotation and Image Markup (AIM) Foundation Model

Pattanasak Mongkolwat · Vladimir Kleper · Skip Talbot · Daniel Rubin

Published online: 17 June 2014

Abstract Knowledge contained within in vivo imaging annotated by human experts or computer programs is typically stored as unstructured text and separated from other associated information. The National Cancer Informatics Program (NCIP) Annotation and Image Markup (AIM) Foundation information model is an evolution of the National Institute of Health's (NIH) National Cancer Institute's (NCI) Cancer Bioinformatics Grid (caBIG®) AIM model. The model applies to various image types created by various techniques and disciplines. It has evolved in response to the feedback and changing demands from the imaging community at NCI. The foundation model serves as a base for other imaging disciplines that want to extend the type of information the model collects. The model captures physical entities and their characteristics, imaging observation entities and their characteristics, markups (two- and three-dimensional), AIM statements, calculations, image source, inferences, annotation role, task context or workflow, audit trail, AIM creator details, equipment used to create AIM instances, subject demographics, and adjudication observations. An AIM instance can be stored as a Digital Imaging and Communications in Medicine (DICOM) structured reporting (SR) object or Extensible Markup Language (XML) document for further processing and analysis. An AIM instance consists of one or more annotations and associated markups of a single finding along with other ancillary information in the AIM model. An annotation describes information about the meaning of pixel data in an image. A markup is a graphical drawing placed on the image that depicts a region of interest. This paper describes fundamental AIM concepts and how to use and extend AIM for various imaging disciplines.

Keywords Big data · Image annotation · Imaging informatics · Image markup · Information resources

Introduction

Images created by various imaging modalities, techniques, and processes (e.g., bioluminescence imaging, computed tomography (CT), digital camera, fluorescence imaging, intravital microscopy, magnetic resonance imaging (MRI), molecular imaging, optical imaging, positron emission tomography (PET), single-photon emission computed tomography (SPECT), tomographic imaging, ultrasound, virtual microscopy, etc.) contain vast amounts of information encoded in the pixel data. Image data can be stored in a variety of image file formats classified by format types such as raster (e.g., JPEG, JPEG 2000, TIFF, GIF, BMP, and Portable Network Graphics (PNG)), vector (e.g., CGM and SVG), stereo (JPS and MPO), and compound (raster and vector) formats [1]. Depending on the file format used to store image data, the majority of available metadata or data about pixel data describes the physical natures of the pixel data. Image metadata may include resolution and aspect ratio, color space, thumbnail (a reduced image), lighting and exposure conditions, compression method, geolocation, equipment and correction techniques used to generate the image, private manufacturing data, bytes order, and so on. None of this information describes what is in the image or what the image is. Automatic image annotation [2–5] or tagging is an active research area of computer vision and pattern recognition, machine learning, and content-based image retrieval in computer science. It focuses on identifying the content of a raster image and

P. Mongkolwat (✉) · V. Kleper · S. Talbot
Department of Radiology, Northwestern University, 737 N. Michigan Ave, Suite 1600, Chicago, IL 60611, USA
e-mail: p-mongkolwat@northwestern.edu

D. Rubin
Department of Radiology, Stanford University, Stanford, CA 94305, USA

assigning a limited number of unstructured semantic labels or text keywords to an image. These keywords (e.g., people, car, scenery, painting, and sport) simply identify known real-world objects and concepts contained in the image and are typically stored in a database. They do not provide informative relationships between objects and derived findings found in an image in the form of structured information that can be systematically processed and analyzed using a computer program.

One of the missions of National Cancer Informatics Program (NCIP) is to provide a standard imaging informatics infrastructure and supporting tools for creating, communicating, and sharing data and research results between imaging research participants. Imaging is critical to cancer research. It lies at a unique juncture in the translational spectrum between clinical practice and research activity. Image and image annotation information obtained in cancer clinical trial research is typically collected in clinical settings using commercial information systems. Image annotations need to be available in a standard format that is at the same time semantically and syntactically interoperable with the infrastructure of NCIP while supporting widespread clinical health-care standards, such as Digital Imaging and Communications in Medicine (DICOM) [6] and Health Level 7 (HL7) [7].

Modern medical imaging techniques generate data in large volumes, which in turn require computer program analyses. A major challenge associated with "big data" is the lack of structured metadata for collecting image annotation information. NCIP Annotation and Image Markup (AIM) captures the semantic meaning of pixel data with user-generated graphical drawings placed on the image and calculations that may or may not be directly related to the drawings, along with associated image identification and supplemental information, thus making AIM an imaging semantic infrastructure for various types of images. The model may be extended to collect information currently not available in the model. The AIM project provides both an information model based on Unified Modeling Language (UML) class diagram, an AIM Extensible Markup Language (XML) schema, and a set of software products to create and view AIM image annotations.

Understanding the Problems

Humans or computer programs can extract observational and computational descriptions of image features from pixel data in the form of annotations. This information is typically captured as free text in a report stored in an information system. A markup or graphical drawing is stored in an imaging system such as a picture archiving and communication system (PACS) that is separate from the text report. A markup can be stored as a DICOM presentation state object, a DICOM structured reporting (SR) [8, 9] object, or in a proprietary format. A measurement that may or may not be directly related to a markup is stored as text in a report.

Imaging observations recorded as free text are difficult to relate to their corresponding image locations. Free text is also very difficult to process and query accurately even by the state-of-the-art natural language processing programs such as caTIES [10] and MetaMap [11]. Here are examples of free text observations and queries:

1. There is a nodule in the upper lobe of the left lung measuring 25 mm×30 mm, best seen on series 5, image 51.
2. Find all patients that have one or more nodules that are larger than 2 cm×2 cm.
3. Find all patients that have a cancerous lung nodule or nodules decreasing in volume by 50 % after radiation and/or chemotherapy treatment.

Another major challenge for communicating imaging observations has been the lack of an agreed-upon information model, syntax, and semantics to describe and record annotation and markup in a uniform data collection scheme using well-defined controlled terminology such as National Cancer Institute (NCI) Thesaurus [12], RadLex® [13], Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT©) [14], DICOM, or user-defined terminology. In addition, there has been no standard storage format for annotation and markup. This presents a unique problem with quality control standards because image findings are typically stored in unstructured text formats or private database schemas and custom or nonstandard lexicons. The AIM project [15–18] objective is to provide a standardized semantically interoperable information model of image pixel meaning and to provide standardized storage formats such as DICOM, XML, and HL7 Clinical Document Architecture (CDA).

Basic Annotation and Image Markup

This section provides an introduction to the concepts behind and relation between annotation and image markup. An annotation is explanatory or descriptive information that is related to referenced images. An image markup is a graphical symbol associated with an image. An image annotation captures both image description and markup, capturing the meaning of image pixel data in a single information source. The following two sets of figures illustrate customary ways of using annotation and image markup.

Figure 1 depicts how an image may contain an annotation and an image markup. Image descriptions are captured as coded terms. A coded term consists of three mandatory values: code, code meaning, and coding scheme designator (information source such as RadLex or SRT for SNOMED CT®), e.g., RID1301, Lung, RadLex. Figure 1a shows an

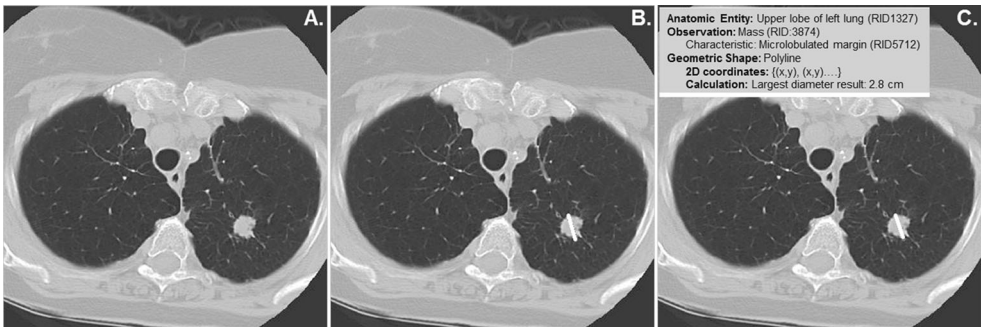

Fig. 1 A sequence of images depicts an image, the image with markup, and the image with markup and coded descriptions

image without any information, Fig. 1b illustrates an image with a markup, and Fig. 1c depicts an image with a markup as well as annotation information captured as coded descriptions of an imaging observation and its characteristics.

Figure 2 displays a set of images containing multiple markups and time points. Figure 2a was acquired in 2001 and has two target (T) lesions: T1 and T2. Figure 2b was a follow-up examination of Fig. 2a, which has two image annotations of the same target lesions, T1 and T2. Each target lesion is captured as an image annotation. Thus, there are four image annotations from the two time points.

To support further image analyses, annotation of annotation is used to annotate one or more image annotations or annotation of annotations for comparison, evaluation, and reference purposes. An annotation of annotation can be created to compare measurements between T1 and T2 from the two time points.

Material and Methods

The AIM project has been developed using Agile methodology [19] and iterative and incremental development [20]. The AIM Foundation model responds to the continued changing demands and new use cases from AIM users and imaging subject-matter experts (SMEs). Figure 3 depicts an overall process for creating AIM models and libraries. Use cases and data collection requirements must be submitted to the NCI JIRA (Altassian, San Francisco, CA) issue tracking system so that they can be evaluated and considered for inclusion in the AIM Foundation model [21]. The foundation model was created in UML using the Enterprise Architect (EA) (Sparx Systems, Creswick, Victoria, Australia) software tool. The model went through 47 iterations with SMEs and three public reviews. The model was curated using the Semantic Integration Workbench (SIW) [22] tool to ensure that all classes and attributes in AIM models conform to and are interoperable with the common data elements (CDEs) standard.

The AIM Foundation model was submitted to the NCI Enterprise Vocabulary Services (EVS) curation team to review the terminology used in the models. The EVS team approved the AIM model, allowing the AIM team to submit it for inclusion in the NCI Cancer Data Standards Registry and Repository (caDSR) [23]. The repository is a centralized source for storing CDEs. Each data element is a consolidation of similar terms and concepts into a single standardized name and definition, valid values, and unique identification for reuse purposes. Using caDSR to create new data elements eliminates ambiguity of terms or concepts used across biomedical research data and organizations. NCIP maintains the approved AIM models on NCI caDSR. Earlier versions of the AIM model have been retired and removed from the caDSR for compatibility reasons. This ensures that the latest values of certain elements in the model are used, such as enumerated

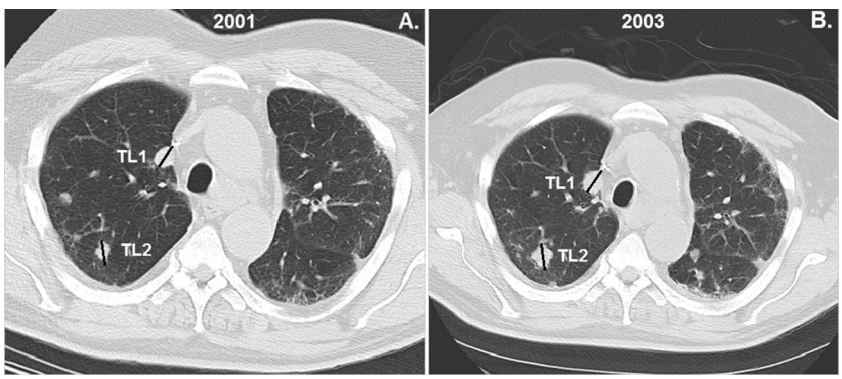

Fig. 2 **a**, **b** Two target lesions on different time points: 2001 and 2003. Measurement results captured as AA from **a** and **b** can be used for comparison purposes

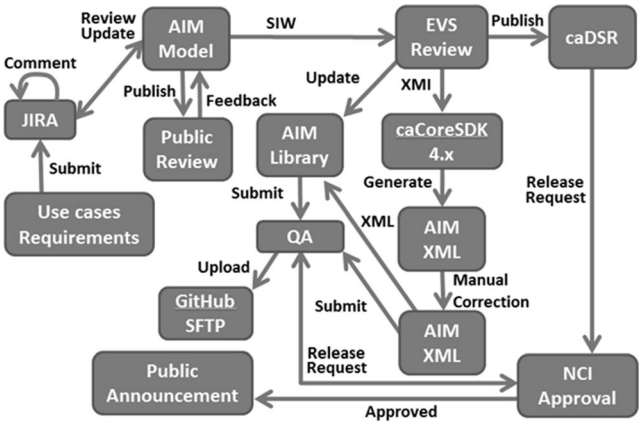

Fig. 3 The AIM creation process

value domains which are often used as annotation question responses and are frequently updated. The active AIM models can be viewed from CDE Browser [24] and the UML Model Browser [25].

Once the EVS team reviewed the AIM Foundation model, the AIM team created the AIM XML schema using a three-step process. First, an AIM XML Metadata Interchange (XMI) file was exported from the EA UML tool. Second, the XMI file was imported to the caCORE Software Development Kit (SDK) version 4.1 [26], which generated an XML schema. Finally, this schema was modified manually with Altova XML Spy 2013 (Altova, Beverly, MA) to compensate for limitations of the caCORE SDK.

Once the AIM team created the AIM XML schema, they updated the AIM C++ programming library using the Standard Template Library (Silicon Graphics, Sunnyvale, CA), DICOM DCMTK library version 3.6.x (OFFIS E.V., Oldenburg, Germany), and the Apache Xerces XML library version 3.1. The AIM 4.x C++ library is available for Linux, Mac OS X, and Windows operating systems using CMake 2.8.x (Kitware, Clifton Park, NY). The AIM library provides a ready-for-use programming tool for creating and reading AIM instances as DICOM SR objects and XML documents. The library implements the AIM model. It provides mutator methods (set and get methods) for every attribute in all AIM classes. The AIM toolkit consists of the library and a program called ANIVATR, which is a referenced implementation of the AIM library. ANIVATR is used to translate between AIM's DICOM SR and XML file formats. Using the AIM library will streamline software development process and allow AIM software implementers to focus on providing annotation features and functionality in their software products. A translator program to translate between AIM XML 4.x and HL7 CDA documents is planned for future development. A status update can be found on the AIM NCI JIRA [21] site.

AIM Concepts

Image annotation information is collected in both clinical settings of commercial information systems and through imaging research tools. This diversity of systems has led to the development of a variety of proprietary and standard information models and storage formats. Having an agreed-upon single standard format to store image annotations enables to focus on providing rich annotation features and functions, easier data exchange and storage, and streamlined software development.

The AIM model captures image annotation, markup, and other information relevant to images. It explicitly defines what kind of information the model can collect. The following describes major concepts and terms used in the AIM context.

- An image annotation can be explanatory or descriptive information, generated by humans or computer programs, directly related to the content of referenced images. It describes information about the meaning of pixel information in images captured as coded terminology supplied by lexicons such as SNOMED CT®, RadLex®, LOINC, or user-defined terms. Annotations become a collection of image semantic content that can be used for data mining purposes.
- An image markup is a graphical drawing or textual description placed on an image. Markups can be used to depict textual information and region of interest visually laid over an image.
- An *Image Annotation* (IA) captures information about the meaning of pixel information in images. It is used to annotate an image or images. It describes a single thing found such as a nodule and, together with markup information, refers to one or multiple images of the same thing. If an image has three different nodules, there are three IAs.
- The *Annotation-of-Annotation* (AA) annotates other AIM annotations (both IA and AA) for grouping (same type of disease from multiple patients), comparison (different time points of the same lesion), and reference purposes.
- An AIM annotation collection may contain either IA or AA instances. An AIM annotation collection is stored as a single file and is either an AIM DICOM SR object or an AIM XML document. In older AIM models, only a single IA or AA was allowed per AIM document instance. For example, in the past, when annotating pathology images, the process of annotating each image could generate hundreds of thousands of AIM files by an automated annotation program. It became very complicated to manage and process a large number of files. The collection concept introduced in the foundation model allows for simpler AIM file management and better logical organization of annotations.

- An annotation has a type captured as a coded term that includes code, description, and code source. The type is used to identify the annotation, e.g., IA (112041, target lesion complete response, DCM) and AA (C96632, sum of longest diameter, NCIt).
- The older AIM models do not store a question as a coded term that is associated with an answer. A question is needed to accompany an answer as the question indicates a reason to collect the answer.
- A calculation result may or may not be directly related to image markups, including other computational methods that provide more meaningful information about images.
- Image references are used to uniquely identify the original images from an imaging archive.
- An AIM statement describes a relationship between two entity type classes in the AIM Foundation model.

An AIM annotation is a collection of associated image annotations and markups. It can be of type IA or AA, which are the only annotation types that can be created. An AIM annotation collection may contain either IA or AA. Each collection may contain one or more IA or AA instances. IA is for annotating images and, among other data, contains annotation markups and image references. AA is for annotating existing annotations. An instance of AA contains neither an image markup nor image reference, while an IA instance contains both. AA is used to annotate IA as well as AA.

The AIM Model

The AIM project has evolved based on new requirements, real-world usage, and user feedback. New requirements were collected to create the AIM Foundation model. Those new requirements are as follows: (1) use the ISO 21090 data type, (2) store more than one image annotation or annotation of annotation in a single AIM instance as an AIM DICOM SR object or AIM XML document, (3) explicitly declare association relationships between AIM entity type classes via AIM statements, (4) store compact calculation results, (5) add the ability to support clinical and research workflow activities via task context, (6) support three-dimensional markup, (7) store a question as a coded term in an AIM class that captures an answer, (8) store a unique reference for other types of DICOM objects, (9) improve storage of references for the image reference class, (10) store DICOM general image and image plane information, (11) store a unique reference for DICOM segmentation, and (12) store information about adjudication observations. The AIM Foundation model was created in response to the requirements. The model is extensible, allowing other image researchers to capture information not available in the foundation model.

A UML class diagram describing the AIM foundation model can be downloaded from the NCI Wiki [27]. It was developed based on the earlier AIM version 3 revision 11. Free EA UML viewer software can be used to view the AIM Foundation UML class diagram. The diagram is also available in PNG file format. The model explicitly describes how image annotation semantic contents can be collected and stored. Class names in the model were intentionally created to inform readers of what information is stored in the classes. Each attribute in a class collects data to support the overall class objective. A relationship between classes is an indication of how one class works with other classes to form and capture a larger concept and information. In Fig. 4, classes are divided into six groups as follows: AIM statement (A), markup (B), image reference (C), calculation (D), and image semantic content (E). Classes that do not pertain to previously mentioned groups are in the general information group (F). This group contains classes that store information about inference or conclusion, the equipment or workstation used to create the AIM annotations, the user who creates AIM annotations, patient identification, reference to DICOM segmentation, annotation role, audit trail, adjudication observation, and workflow activity. The following sections describe the AIM classes in each major group.

AIM Statement

An AIM statement, shown in Fig. 4(a), has a subject-predicate-object construct precisely defining the relationship between a subject and an object class. The naming convention used to create an AIM statement is a concatenation of subject, predicate, and object. A predicate defines a relationship between a subject and an object. Subject and object information comes from entity classes in AIM models. The current predicates in AIM are excludes, has, is compared with, is comprised of, is found in, is identified by, references, and uses. AIM statements describe a thing found, compared, measured, addressed, and graphically annotated on an image or images from the same series in an imaging study. An AIM statement expresses the most granular amount of information an AIM annotation can have at varying degrees and thus is highly extensible to meet future use cases of other imaging domains. A statement belongs to one of the three logical groups.

The *AnnotationStatement* group can be used for IA and AA. The *AnnotationOfAnnotationStatement* group represents a group of statements used to describe AA relationships between a subject and an object class entity that do not have a direct reference to the image or images. The *ImageAnnotationStatement* group represents a group of statements used to describe IA relationships that have direct reference to the image or images.

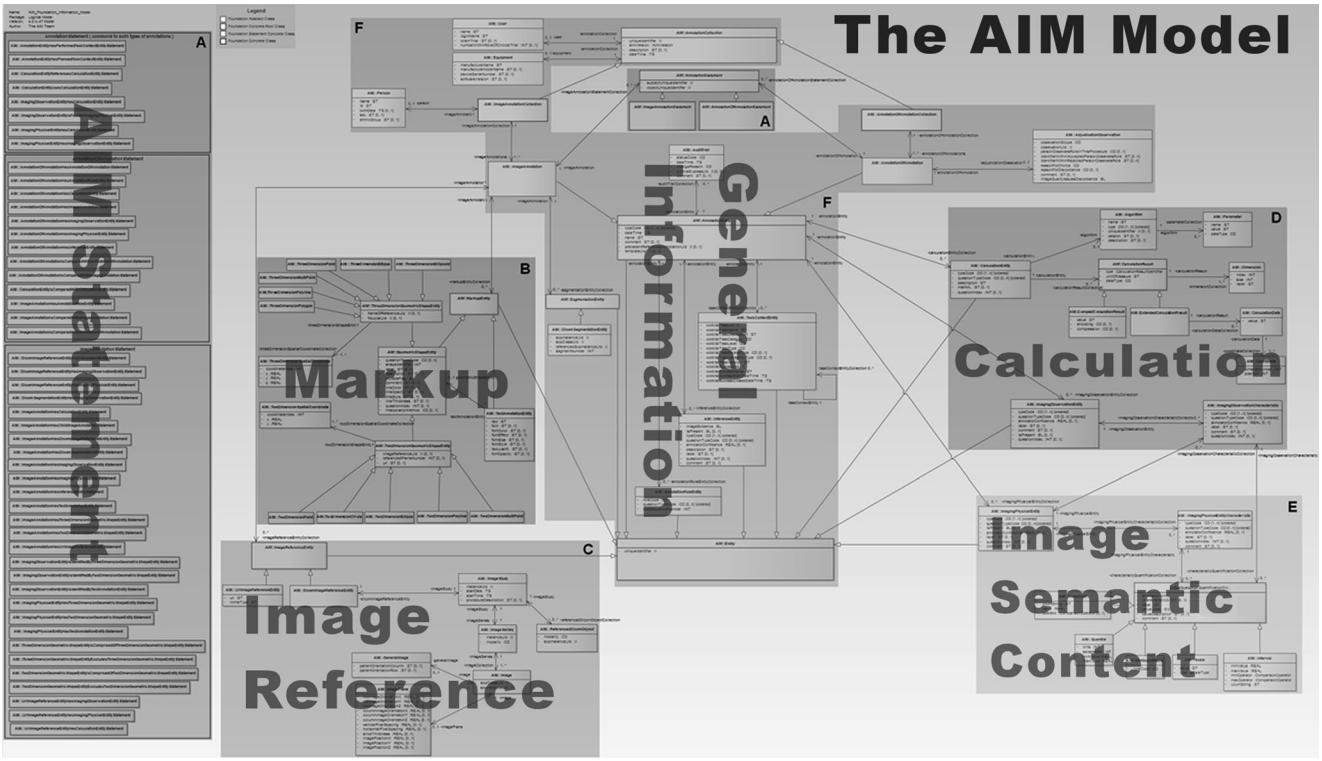

Fig. 4 AIM Foundation UML diagram

Markup

The markup group, shown in Fig. 4(b), captures textual information and graphical representation as DICOM SR value types SCOORD and SCOORD3D for two- and three-dimensional spatial coordinates, respectively. The available graphic types for two dimensions, store *x* and *y* coordinates, are point, multipoint, polyline, circle, and ellipse [6]. The *TwoDimensionGeometricShapeEntity* class has *two-dimensional coordinates*: the service-object pair (SOP) instance unique identifier (UID) of the image that contains the pixel and the frame number within the referenced SOP instance to which the reference applies. The first frame must be denoted as frame number 1. In the case of a multiframe image, the frame number from the DICOM header is used.

The available graphic types for three dimensions, store *x*, *y*, and *z* coordinates, are point, multipoint, polyline, polygon, ellipse, and ellipsoid [6]. The *ThreeDimensionGeometricShapeEntity* class has *ThreeDimensionSpatialCoordinate* as well as the frame of reference UID for a series.

The *TextAnnotationEntity* class captures a short description or label and its coordinates as SCOORD or SCOORD3D graphic type, resulting in *TwoDimensionMultiPoint* or *ThreeDimensionMultiPoint*, respectively. A multipoint implementation is expected to have no more than two coordinates that can be represented as an arrow connecting a text to a point on an image. Only the IA class can have markups.

Image Reference

The image reference group, shown in Fig. 4(c), represents an image or collection of images being annotated. The two possible types of references are DICOM and uniform resource identifier (URI) or web image references. *DICOMImageReferenceEntity* associates with other classes that follow the DICOM information model's study-series-image hierarchy. The *ImageStudy* class has study instance UID, start date and start time, and procedure description. ImageStudy may have zero or more references to DICOM objects via the *ReferencedDicomObject* class. The *ImageSeries* class has a series instance UID. The *Image* class has an SOP class UID and SOP instance UID. The Image class may have associations with *GeneralImage* and *ImagePlane*; these classes came from the DICOM module general image and image plane, respectively. They are used to store frequently used DICOM tags such as patient orientation, pixel spacing, and image position. The second image reference type is *WebImageReference*, which contains a URI to an image.

Calculation

The calculation group, shown in Fig. 4(d), represents the calculation results of an AIM annotation. Calculation results may or may not be directly associated with graphical symbols or markups. For example, given an image with a single line markup measurement of a nodule, a calculation result is the

length of the line in millimeters. As another example, an image has an arrow pointing to a specific location and two concentric circles, with an area measurement of the larger circle minus the smaller circle. The computation result is based on the independent calculation of each circle. The *CalculationEntity* class collects information about a calculation performed directly to an image or images. It defines a type of calculation such as area, height, radius, and volume of ellipsoid from the Unified Code for Units for Measure (UCUM), as controlled terminology that can be captured in code value, code meaning, and coding scheme designator in a single attribute called *TypeCode*. It also captures MathML as a string attribute within the class. A *QuestionTypeCode* attribute captures a reason why a calculation is performed. A textual description can be stored in the *description* attribute. The *CalculationResult* abstract class contains the type of result (e.g., binary, scalar, vector, histogram, array, histogram, or matrix), unit of measurement, and coded data type (a primitive data type such as integer and double as well as other data types such as URI). A *calculation result* can be stored as a compact or extended result, *CompactCalculationResult* and *ExtendedCalculation*, respectively. The result of a calculation is captured in string format in a value attribute that can hold a value of array, binary, histogram, matrix, scalar, and vector. Encoding is a method applied to the content of the value attribute. Compression is a method used to compress the content in a value attribute. The *Dimension* class stores how many dimensions a calculation result has; e.g., a scalar result would have one dimension and a two-by-two matrix would have two dimensions. An extended calculation stores the result of a calculation individually with the precise location of each element in the result. The *Data* class is used to store result value. The *Coordinate* class identifies location within a dimension for the Data class. A *calculation* may have a relationship with a markup or a collection of markups, other calculations, imaging observation, and imaging physical entity. These types of relationships can be captured as AIM statements.

Image Semantic Content

The classes in the image semantic content group, shown in Fig. 4(e), are used to gather image findings or interpretations of images. The *ImagingPhysicalEntity* class stores an anatomical location as a coded term (e.g., RID2662, femur, RadLex) from a recognized controlled vocabulary such as RadLex®, SNOMED CT®, unified medical language system (UMLS), and others. As mentioned in the "AIM Concepts" section, a question associated with the anatomical location can be stored in this class as a coded term, e.g., "PRI9724, What is the anatomical location?, 99Private." The *ImagingPhysicalEntityCharacteristic* further describes ImagingPhysicalEntity such as "RID6296, bone forming characteristic, RadLex." A question associated with this characteristic could be "PRI3754, What is the characteristic of the femur?, 99Private." The *ImagingObservationEntity* class is the description of things that are seen in an image. "Mass," "radiographic evidence of pleural effusion," "foreign body," and "artifact" are all examples of ImagingObservationEntity. The *ImagingObservationEntityCharacteristic* class contains descriptors of the ImagingObservationEntity class such as "dense," "heterogeneous," "hypoechoic," and "spiculated." Both ImagingPhysicalEntityCharacteristic and ImagingObservationEntityCharacteristic may have *CharacteristicQuantification* associated with them. A quantification can be a numerical value, an interval (e.g., 34–67 %), a scale (e.g., 1, none; 2, mild), a quantile (e.g., 1 (1–50), 2 (51–100)), and a nonquantifiable (e.g., none, mild, mark).

General Information Group

Classes in this group, shown in Fig. 4(f), do not provide a larger concept. We begin with the *AnnotationCollection* class, which is an abstract concept of a container that collects one type of annotations of type IA or AA (see Table 1). The class has an UID used to identify a collection, a version of the AIM model used for collecting annotation data, creation date and time, and description about the AIM annotation contained in the collection. AnnotationCollection is the parent of *ImageAnnotationCollection* and *AnnotationofAnnotationCollection*. AnnotationCollection associates with two optional classes used to capture the information about the person or computer program that created the AIM instances and information about the manufacturer that created the software used to create AIM instances. The *User* class represents a person or a computing resource that creates an AIM annotation. The *Equipment* class provides information about the system used to create AIM annotations. ImageAnnotationCollection stores instances of IA. It associates with the *Person* class that contains basic patient demographic information. The Person class represents both human and animal. Annotation-of-AnnotationCollection stores instances of AA, which can be used to annotate IA and AA. Both IA and AA have a collection of AIM statements. As described earlier, an AIM statement contains two UIDs for subject (which has an UID of an entity) and object (which has an UID of another entity) that participate in the statement.

Next, the *Entity* class is an abstract class that represents the existence of a thing, concept, observation, calculation, measurement, and graphical drawing in AIM. A class derived from the Entity class can be used in AIM statements. The *AnnotationEntity* class is an abstract-based class for the IA and AA classes. The AnnotationEntity class captures name, a general description of the AIM annotation, type of annotation via controlled terminology, creation date and time, a reference to the AIM template used to create an annotation, a reference

Table 1 Examples of codes used for IA and AA

AIM annotation class	Code	Code meaning	Source
ImageAnnotation	112041	Target lesion complete response	DCM (DICOM)
ImageAnnotation	112042	Target lesion partial response	DCM
ImageAnnotation	112043	Target lesion progressive disease	DCM
ImageAnnotation	112044	Target lesion stable disease	DCM
ImageAnnotation	112045	Nontarget lesion complete response	DCM
AnnotationOfAnnotation	C96632	Sum of longest diameters	NCIt
AnnotationOfAnnotation	C96633	Sum of longest perpendiculars	NCIt
AnnotationOfAnnotation	C96635	Sum of volumes	NCIt
AnnotationOfAnnotation	C112371	Percent change from baseline in sum of longest diameter	NCIt

to a previously related AIM annotation, and AIM annotation UID.

Both IA and AA have *AnnotationRoleEntity*, *AuditTrail*, *InferenceEntity*, and *TaskContextEntity* classes. The IA class associates with the *SegmentationEntity* abstract class. The model currently supports DICOM segmentation via *DicomSegmentationEntity*. DICOM segmentations are either binary or fractional. The DicomSegmentationEntity class represents a multiframe image that contains a classification of pixels in one or more referenced images. The class contains a DICOM SOP class UID that defines the type of segmentation, references to its own instance UID, and the referenced instance UID of the image to which the segmentation is applied. It also includes the identification number of the segment that must be unique within the segmentation instance in which it is created. The AnnotationRoleEntity describes the role of an annotation. Each instance can have a role associated with it, e.g., a baseline case. The InferenceEntity class provides a conclusion derived through interpreting an imaging study and/or medical history. The AuditTrail class captures the status of an annotation instance using a coded term and contains identifying and descriptive attributes of the reading session and the reading subtask that results in clinical or trial findings. The class consists of the overall task and the specific subtask. A task represents a unit of overall work. It may have one or more subtasks.

Extending AIM

The AIM Foundation model was created as a base model that can be extended to serve image annotations from various disciplines. As such, this section provides a set of guidelines to extend the AIM Foundation model. AIM users may extend the foundation model to fit their needs by following these criteria:

1. Use ISO 21090 data type for every attribute in a class. Using ISO 21090 data type provides a good foundation to convert AIM information to HL7 CDA.
2. AIM markup is modeled after DICOM 2D and 3D spatial coordinate geometries [24]. New additional markups must be able to be stored in DICOM SR format.
3. A new class must be able to be mapped to DICOM SR data elements. More details about AIM DICOM SR templates are on the NCI Wiki [27]. It is important for a user to understand DICOM SR and how to modify a DICOM SR template properly because new classes added to the model will need to be stored in DICOM format.
4. Naming conventions are used to identify and distinguish classes from their association names.
   a. A class name must explicitly describe what information the class collects. In class names, only the first character of each word in the name is capitalized, even if the name contains an acronym, e.g., DICOM should be Dicom. An example of an existing AIM class is ReferencedDicomObject.
   b. An association name is distinguished from its source or target class by making the first character of the name lower case. For example, *ImageStudy* has an association name of "imageStudy".
   c. A source class may contain 0…* or 1…* (zero-to-many or one-to-many) associations to a target class. Since there could be many instances of the target class contained within the source class, this collection of objects is distinguished by appending "collection" to the target association name. For example, an ImageStudy with 1…* ReferencedDicomObjects would have a target association name of "referencedDicomObjectCollection."
5. The name of any class extending from the Entity class must end with the word "Entity," except *ImageAnnotation* and *AnnotationOfAnnotation*. A class derived from the Entity class represents the existence of a thing or concept that is not currently captured in the AIM Foundation model. A new entity class may also be included if a user wants to explicitly express what information the new model will be able to store.

6. When constructing a new AIM statement, one must adhere to the following rules: A class that can be a subject and an object of an AIM statement must extend or inherit from the Entity class. A predicate must be selected from a list of existing predicates from the "AIM Statement" section of this article. You may use your own predicates. The name of an AIM statement shall be a concatenation of the name of a subject class, predicate, and the name of the object class.

By extending the AIM Foundation model, AIM users will be able to store additional information that has not been covered in the current model. Any additions to be included in the official NCI AIM model can be submitted directly to AIM JIRA at NCI [21]. AIM principles of governance can be found at [27].

AIM 4.0: an AIM Foundation Extension

AIM version 4.0 is an extension of the AIM Foundation model. It has nine additional classes supporting lesion tracking derived during image-based clinical trials [9, 28]. *LesionObservationEntity* is an abstract class and stores observations made about lesions in both clinical trial and day-to-day clinical treatments. *GeneralLesionObservationEntity* contains general observations made about lesions in clinical trial results and day-to-day clinical treatments that are not specific to a point in time. *TimePointLesionObservationEntity* contains observations made about lesions in day-to-day clinical interpretations and clinical trial results at a specific time point. It also includes "lesions" that are created for the purpose of calibrating scanned film or other secondary capture images.

Discussion

Before an existence of the AIM solution, image annotation descriptions, graphical drawings of region of interests (ROIs) and measurements were collected as textual descriptions and separated from ROIs and measurements that may be stored as DICOM key image objects or proprietary formats. The process of extracting this type of information requires trained personnel to manually review, identify, and record wanted data. It becomes a slow and cumbersome endeavor. The AIM project provides an information model that collects annotation data in a structured format that can be searched using a computer program and a set of programming and end-user tools for defining collected data parameters, facilitating machine-supervised data collection, and presenting results in a standardized form.

The AIM model is complex and comprehensive out of necessity. NCIP provides and continues to support and update the AIM model and the AIM C++ library available on Linux, Mac OS X, and Windows platforms. The model and the library are designed for image researchers and software implementers to apply or extend the model and the library for their own use. However, most users want to utilize AIM capabilities without getting involved with intricacy of the model or implementing AIM on an imaging workstation. In a typical scenario, imaging workstation users want to focus on interpreting imaging studies and record image findings in an efficient manner.

NCIP provides a suite of complementary software products to increase AIM adoption and demonstrate the usefulness of AIM. An AIM template was introduced to assist imaging experts in defining a set of well-defined questions and answer choices to each question to facilitate collecting information for a particular purpose. The AIM Enterprise Service (AES) [21] project consists of the AIM Template Service (ATS) [29] and AIM Template Builder (ATB) [30]. ATS is an Internet application created for storing, downloading, sharing, and searching AIM templates. ATS also has a set of web services allowing an application to query, retrieve, and upload AIM templates. To facilitate the creation of constrained AIM annotations, ATB provides a Java-based application for creating and storing AIM templates on a local computer and/or ATS. AIM Workstation 4.x [27] is a reference implementation of the AIM 4.0 model on an open source ClearCanvas workstation. It can import and use AIM templates created by ATB. It is capable of creating AIM annotations and storing them as AIM XML documents or AIM DICOM SR objects on a local computer and remote storage such as AIM data service and PACS, respectively.

The AIM project and its related software products have been used in several projects at the Cancer Imaging Program (CIP) at NCI to support The Cancer Genome Atlas (TCGA) research groups in the breast, brain, ovarian, and renal areas. The CIP staff worked with imaging specialists to develop feature sets that describe imaging observation of things that can be found on a particular type of imaging study. Each feature set has defined questions with possible answer choices for each question. Example images for each feature were identified and collected for reference purposes. A feature set was used to build an AIM template using ATB. Created templates were stored on the ATS. The AIM on ClearCanvas workstation could retrieve AIM templates from the ATS. Resulting AIM annotations were stored on an AIM Data Service. Before a template was used by imaging interpreters from each research group, the CIP staff extracted all the answers and performed statistical analysis of inter-reader agreement on the individual features. The staff consulted imaging specialists to refine wording, definitions, and answer choices and to clarify any problems. Once an agreement was reached, the template and the workstation were used to create AIM annotations. The CIP staff assessed the TCGA data and

compared the AIM data with other data types such as clinical outcomes, genomics, computer generated feature extraction, etc.

AIM is also used at the Center for Neuroscience and Regenerative Medicine at National Institute of Health (NIH) and Uniformed Services University of the Health Sciences (USUHS) for traumatic brain injury (TBI) research project. The National Institute of Neurological Disorders and Stroke (NINDS) has established CDEs as standards to characterize findings and outcomes in patients with TBI. Use of CDEs is intended to harmonize data collected from clinical research studies across the TBI research community to facilitate multicenter research. Major components of the TBI CDEs are those related to imaging findings, but the CDEs do not take into account the spatial nature of imaging findings. TBI AIM templates were created using the ATB with the corresponding CDEs. The basic template probes the simple presence or absence of imaging abnormalities. The descriptive template includes a simple grading scale to provide a sense of the extent and location of TBI lesions. The advanced template involves in-depth measurements and interpretation.

Conclusion

AIM defines a standard information model for annotation descriptions, markups (graphical drawings), and computational results of image features using well-defined medical lexicons such as RadLex®, UMLS, SNOMED CT®, and others. This article explains how AIM is organized and what kind of information it captures. Annotations stored in the AIM format can be queried to find images containing similar image content and to correlate human imaging observations with other biomedical data.

The AIM model continues to evolve according to new requirements from AIM users. Its supporting software library and software products are also updated based on a new AIM model and new software feature requests. Current AIM information can be found on the AIM NCI Wiki [23] and AIM JIRA [16].

Acknowledgments Research reported in this publication was partially supported by the National Cancer Institute of the National Institutes of Health under award number 12XS577. The authors would like to acknowledge significant feedback from Dr. David Channin at Guthrie Clinic, Justin Kirby at CIP, and Dr. Dzung Pham and Dr. John Butman at NIH/USUHS.

References

1. Image file formats. http://en.wikipedia.org/wiki/Image_file_formats, accessed October 30, 2013
2. Smeulders AWM, Worring M, Santini S, Gupta A, Jain R: Content-Based Image Retrieval at the End of the Early Years. IEEE Trans Pattern Anal Mach Intell 22(12):1349–1380, 2000
3. Carneiro G, Chan AB, Moreno PJ, Vasconcelos N: "Supervised Learning of Semantic Classes for Image Annotation and Retrieval", IEEE Trans Pattern Anal Mach Learning, 29:3 March 2007
4. Li J, Wang JZ: Real-time Computerized Annotation of Pictures. IEEE Trans Pattern Anal Mach Intell 30(6):985–1002, 2008
5. Weston J, Bengio S, Usunier N: "Machine Learning, Large Scale Image Annotation: Learning to Rank with Joint Word-Image Embeddings", 81:(1) 21-35, 2010
6. Digital Imaging and Communications in Medicine (DICOM). http://medical.nema.org/standard.html, accessed October 30, 2013
7. HL7. http://www.hl7.org/, accessed October 30, 2013
8. Clunie DA: DICOM Structured Reporting. PixelMed Publishing, Bangor, 2000
9. Clunie DA: DICOM Structured Reporting and Cancer Clinical Trials Results. Cancer Informat 4:33–56, 2007
10. caTIES. http://caties.cabig.upmc.edu, accessed October 30, 2013
11. MetaMap. http://metamap.nlm.nih.gov, accessed October 30, 2013
12. NCI Thesaurus. http://ncit.nci.nih.gov/ncitbrowser, accessed October 30, 2013
13. RadLex®. http://www.radlex.org, accessed October 30, 2013
14. SNOMED CT©. http://www.ihtsdo.org/snomed-ct, accessed October 30, 2013
15. Channin DS, Mongkolwat P, Kleper V, et al: The caBIG™ Annotation and Image Markup Project. J Digital Imaging 23:2, April, 2010
16. Annotation and Image Markup Version 3 Project: Requirements, Design, Implementation and Usage; October 2010, https://ncisvn.nci.nih.gov/svn/files/trunk/aim/aim/AIMToolkit3.0.2/AIMToolkit_v3.0.2_rv11.rar, accessed October 30, 2013
17. Rubin DL, Mongkolwat P, Kleper V, Supekar K, Channin DS: "Medical Imaging on the Semantic Web: Annotation and Image Markup", Association for the Advancement of Artificial Intelligence, 2008. Spring Symposium Series, Stanford, 2008
18. Channin DS, Mongkolwat P, Kleper V, Rubin DL: The Annotation and Image Mark-up Project. Radiology 253:590–592, 2009
19. Agile. http://agilemanifesto.org, accessed October 30, 2013
20. Larman C, Basili VR: Iterative and Incremental Development: A Brief History. IEEE Comput 36(6):47–56, 2003
21. AIM JIRA. https://tracker.nci.nih.gov/browse/AIM, accessed October 30, 2013
22. Semantic Integration Workbench. https://cabig.nci.nih.gov/community/tools/SIW, accessed October 30, 2013
23. caDSR. https://cabig.nci.nih.gov/community/concepts/caDSR, accessed October 30, 2013
24. CDE Browser. https://cdebrowser.nci.nih.gov/CDEBrowser, accessed October 30, 2013
25. UML Model Browser. http://umlmodelbrowser.nci.nih.gov/umlmodelbrowser, accessed October 30, 2013
26. caCORE. https://wiki.nci.nih.gov/x/BIAe, accessed October 30, 2013
27. AIM on the NCI Wiki. https://wiki.nci.nih.gov/x/z4X3Ag, accessed October 30, 2013
28. Clunie DA: Clinical Trials Results Reporting, http://www.dclunie.com, October 30, 2013
29. AIM Template Service (ATS). https://wiki.nci.nih.gov/x/ewLgB, accessed October 30, 2013
30. Mongkolwat P, Channin DS, Kleper V, Rubin D: An Open Source and Open Access caBIG® Annotation and Image Markup (AIM) Template Builder. Radiographics 32(4):1223–1232, 2012