The caBIG® Annotation and Image Markup 4.0 Assessment Report and Development Plan

Development Team(s)		
Development	***Technical Documentation***	***Program Management***
Pattanasak Mongkolwat Vladimir Kleper	Pattanasak Mongkolwat Carolyn Klinger (Technical Editor)	Peter Yan

Contacts and Support	
Pattanasak Mongkolwat, Ph.D. p-mongkolwat@northwestern.edu	NA
NA	NA

1. Scope and Purpose

The scope of this document is to provide an initial assessment and planning for the development of the caBIG® Annotation and Image Markup (AIM) version 4.0 based on AIM version 3.0 [1]. The AIM 4.0 information model must harmonize with the BRIDG and LS DAM model. The AIM 3.0 model covers information required by Radiology and Oncology. The AIM 4.0 model will cover additional requirements from Oncology and Pathology.

The AIM 3.0 toolkit consists of the AIM C++ library and an example of a referenced implementation tool called ANIVATR. The implementation is an executable program running on Microsoft Windows XP or greater. It is used to convert AIM DICOM SR objects to AIM XML documents and vice versa. The new AIM 4.0 toolkit library covers the information model related to Radiology and Oncology but not Pathology. The implementation of C++ library for Pathology will be done in the next phase of development.

2. Overview

The AIM information model is used to express and capture image annotation and markup information relevant to images. An annotation can be explanatory or descriptive information, generated by humans or machines, that directly relates to the content of a referenced image or images. It describes information regarding the meaning of pixel information in images. An image markup includes the graphical symbols or textual descriptions associated with an image. Markups can be used to depict textual information and regions-of-interest (graphical drawing) visually along side of, or more typically when overlaid upon, an image. Information from annotations and markups are used to populate the AIM instance via the AIM library for the purpose of generating AIM DICOM Structure Report (SR) [4] objects and AIM native XML documents.

The AIM programming library is used to create, validate, and transform an AIM instance between AIM XML documents and AIM DICOM SR objects. An AIM XML schema is also provided with the toolkit. It can be used to guide and validate creation of AIM XML documents. Implementers may choose to generate AIM instances using XML programming.

2.1. Technology Stack

Annotation and Image Markup	A UML Information Model
Digital Imaging and Communications in Medicine	An Information Model for Medical Related Images
Standard C++	Microsoft Visual Studio 2008
XMI	Enterprise Architect 4.5

2.2. Background Knowledge and Assumption

A reader of this document is assumed to have a good knowledge about AIM 3.0 information model [1], UML, DICOM, and C++. The AIM toolkit was created as a partner to the AIM model to provide application developers a faster way to adopt AIM in their application. The toolkit consists of AIM C++ programming library and a referenced implementation to show how the library can be used.

The AIM library can be used to generate AIM DICOM SR objects and AIM XML documents.

2.3. AIM Model

2.3.1. Overview

The AIM information model is described using an UML class diagram. The model is used to express and capture image annotation and markup information relevant to images. It is very important that the reader of this document fully understands the AIM 3.0 model.

The AIM 3.0 information model consists of 45 classes and 248 attributes. Most of the AIM classes are optional. The three classes required for a valid AIM instance are ImageAnnotation, Person, and ImageReference. The AIM 3.0 revision 11

does not conform to ISO 21090 data types because caGRID SDK did not support these data types at the time that the AIM 3.0 model was completed.

2.4. DICOM Structure Reporting

2.4.1. SR Document Information Entity

A DICOM SR Information Entity describes the attributes, semantic context, and requirements of a valid DICOM SR document. A DICOM SR object belongs to a series within a single study. The AIM 3.0 information model is being captured as DICOM comprehensive SR.

2.4.2. Missing Information from DICOM SR

A DICOM SR document typically does not store presentation information. AIM 3.0 has two classes that contain presentation information, TextAnnotation and GeometricShape. TextAnnotation includes the font, fontColor, fontEffect, fontSize, fontStyle, fontOpacity and textJustify attributes. GeometricShape includes the lineColor, lineOpacity, lineStyle and lineThickness attributes. These attributes are not currently being stored in AIM DICOM SR. DICOM considers this information as presentation state information. Therefore, presentation information is omitted when an AIM XML document is converted to AIM DICOM SR.

2.4.3. AIM DICOM SR Template

The DICOM SR template for AIM was created in order to capture AIM information in DICOM SR Comprehensive format. The template is explained in detail in AIM_4.0_DICOM_Harmonization.doc.

2.5. AIM XML Schema

2.5.1. AIM 4.0XML Schemas

Two XML schemas are generated from the final AIM 4.0 UML model. The first schema contains bi-directional associations present in the model and is used by the AIM 4.0 data service. The second schema contains only single-directional associations and is used to generate and validate AIM 4.0 XML documents.

3. AIM 4.0 Feature Requests

This section provides new feature requests for AIM 4.0 proposed by AIM adopters. Note that not all feature requests listed in table 3.1.1-1 to 3.1.1-3 will be incorporated in the AIM 4.0 model. The AIM team reserves the right to selectively implement the feature requests based on consultations and recommendations with the members of AIM team, DICOM WG-18, BRIDG, and LS DAM.

3.1.1. AIM 4.0 Core Model Requirements

Table 3.1.1-1 depicts requirements for the core AIM model.

Requirement Number	Description	AIM 4.0
R1.	**Annotation Collection**	
R1.1.	An AIM instance shall be able to contain one or more ImageAnnotation or AnnotationOfAnnotation instances.	Yes
R2.	**Geometric Shape**	
R2.1.	Add label and description attribute to GeometricShape class	Yes
R3.	**Inference**	
R3.1.	Add label and description attribute to Inference class	Yes
R4.	**Storing Arbitrarily DICOM Tags**	
R4.1.	Slice thickness (0018,0050)	
R4.2.	Radiation exposure (0018,1152) or (0018,1153)	
R4.3.	Radiation dosage	
R4.4.	Reconstruction parameters or kernel	
R4.5.	kVp (0018,0060)	
R4.6.	mAs	
R4.7.	Pitch	
R4.8.	Collimation	
R4.9.	Density (avg. HU)	
R4.10.	Nodule orientation and location - Anatomic entity can be use to capture "location " - Nodule orientation (?)	
R4.11.	Shape (lobulated, speculated, ellipsoid, spherical, irregular) - Imaging Observation Characteristic, for example: Imaging Observation = mass Imaging Observation Characteristic = spherical	
R4.12.	Attachment (attached/non-attached to vasculature) - Anatomic Entity Characteristic	
R4.13.	Sizing technique	
R4.13.1.	1D (RECIST-like)	
R4.13.2.	2D (RECIST-like)	

Table 3.1.1-1. Core Requirements

3.1.2. AIM 4.0 Radiology Requirements

Table 3.1.1-2 depicts requirements for Radiology that shall be captured in the AIM model.

Requirement Number	Description	AIM 4.0
R5.	**Capture Question**	
R5.1.	Capture a question as a coded term in Anatomic Entity, Anatomic Entity Characteristic, Imaging Observation, and Imaging Observation Characteristic	Yes
R6.	**Segmentation**	

Requirement Number	Description	AIM 4.0
R6.1.	Associate with CharacteristicQuantification	
R6.2.	Type shall be captured as a code value, code meaning, and coding scheme designator with coding scheme version as an option.	

Table 3.1.1-2. Radiology Requirements

3.1.3. AIM 4.0 Oncology Requirements

Table 3.1.1-3 depicts requirements for Oncology that shall be included in the AIM model.

Requirement Number	Description
R7.	**Lesion Tracking (AIM 4.0 modeling)**
R7.1.	Numbering (identification)
R7.2.	Typing
R7.3.	Separating
R7.4.	Merging
R7.5.	Annotation of a specific portion of a lesion
R7.6.	Temporal relationship between lesions
R8.	**Terminology (Documentation)**
R8.1.	Identify terminologies required for Oncology
R8.2.	Work with EVS team
R9.	**RECIST Use Cases (Documentation)**
R9.1.	AIM Annotation of Annotation
R10.	**Clinical Trials Results Reporting**
R10.1.	Combine the unpublished David Clunie's DICOM Supplement Clinical Trials Results Reporting. See "supnnn_01_ClinicalTrialsResults.doc"

Table 3.1.1-3. Oncology Requirements

3.1.4. AIM 4.0 Pathology Requirements

The Pathology module was assigned to the Emory team under the supervision of the principle investigator at Northwestern University. The Emory team shall establish a set of requirements for Pathology. The requirements will be evaluated and compared against DICOM Pathology standard,

Once the initial AIM 4.0 model is ready, approximately in early February 2011, the model will be distributed to the Emory team, Dr. Lawrence Tarbox, and the DICOM working group 18 (WG-18). During this time, the AIM team at Northwestern, with assistance from Dr. Lawrence Tarbox and WG-18 for DICOM Pathology, will assist the Emory team with any questions related to AIM 4.0 and DICOM Pathology.

4. Gap Analysis between AIM 3.0 and AIM 4.0

4.1. Initial Assessment

The new AIM 4.0 model must be harmonized with the BRIDG and LS DAM models. The two models have been created based on UML Domain Analysis Models (DAM). A DAM is used to capture and describe required information, information collection, information flow, and related activities among different entities in the model. BRIDG and LS DAM have adopted the ISO 21090 standard data types that were defined for healthcare information exchange and based on HL7 data types.

The AIM 3.0 model does not use ISO 21090 standard data types because caCORE SDK 3.x and 4.1 did not support usage of those data types when AIM 3.0 was being developed. The caCORE SDK 4.4, which will be released soon, will support ISO 21090. caCORE SDK is used to generate the XML schema for AIM models.

4.1.1. AIM 3.0 with 21090 Data Type

Table 4.1.1-1 illustrates data type mapping between AIM 3.0 and ISO 21090.

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

Table 4.1.1-1. Data Type Mapping

Table 4.1.1-2 depicts four coding attributes that are combined to create the CD data type in ISO 21090.

AIM 3.0 Data Type	**ISO 21090**
codeMeaning	CD
codingSchemeDesignator	CD
codingSchemeVersion	CD
codeValue	CD

Table 4.1.1-2. AIM Coding Attributes Mapping to ISO 21090

AIM 3.0, revision 11 was used to create a new version of AIM 3.0 with ISO 21090 data types. It was created as a base model for the development of the AIM 4.0 model.

4.2. AIM 3.0 Library and Classes

Table 4.1.2-1 depicts how each ISO 21090 data type is mapped to a C++ data type.

ISO 21090 Data Type	**C++ Data Type**
BL	boolean
CD	C++ class for code meaning
II	Integer
INT	Integer
REAL	Double
ST	String

TS	_strdate and _strtime
Uid	String

Table 4.1.2-1. C++ and ISO 21090 Data Type Mapping

The AIM 3.0 library consists of 45 classes and 248 attributes. Based on the AIM 3.0 model with ISO 21090 data types, all attributes in AIM classes have their data type changed; see Table 4.1.2-2. Therefore, the AIM 4.0 library will have to be reimplemented.

AIM 3.0 Class	**ISO 21090**
AimStatus	Yes
AnatomicEntity	Yes
AnatomicEntityCharacteristic	Yes
Annotation	Yes
AnnotationOfAnnotation	Not Applicable
AnnotationRole	Yes
Calculation	Yes
CalculationData	Yes
CalculationResult	Yes
CharacteristicQuantification	Yes
Circle	Not Applicable
Coordinate	Yes
DICOMImageReference	Not Applicable
Dimension	Yes
Ellipse	Not Applicable
Equipment	Yes
GeometricShape	Yes
Image	Yes
ImageAnnotation	Not Applicable
ImageReference	Yes
ImageSeries	Yes
ImageStudy	Yes
ImagingObservation	Yes
ImagingObservationCharacteristic	Yes
Inference	Yes
Interval	Yes
MultiPoint	Not Applicable
NonQuantifiable	Yes
Numerical	Yes
Person	Yes
Point	Not Applicable
Polyline	Not Applicable
PresentationState	Yes
Quantile	Yes
ReferencedAnnotation	Yes
ReferencedCalculation	Yes
ReferencedGeometricShape	Yes
Scale	Yes
Segmentation	Yes
SpatialCoordinate	Yes
TextAnnotation	Yes
ThreeDimensionSpatialCoordinate	Yes
TwoDimensionSpatialCoordinate	Yes

User	Yes
WebImageReference	Yes

Table 4.1.2-2. Classes in AIM Library

5. Project Planning and Development

This section provides information about AIM UML modeling and AIM C++ library development, harmonization, and integration with the Pathology module from Emory. A project plan was developed to further detail and coordinate all tasks required to accomplish the project. The Microsoft Project plan is in AIM_FY11_v9.mpp.

5.1. AIM 4.0 Modeling

The new requirements in chapter 3 must be mapped to attributes and classes in the AIM 4.0 model.

5.2. AIM 4.0 Harmonization with BRIDG, LS DAM and DICOM Standard

Once the new AIM 4.0 model is complete, it will be distributed to the points of contact for BRIDG and LS DAM, namely Steve Sandberg and Lisa Schick. We had an initial meeting and general consultation with Steve and Lisa. The length of the harmonization is between one to six months.

The AIM team will work closely with the DICOM WG-18 committee members to harmonize the AIM 4.0 model with the DICOM standard.

5.3. AIM 4.0 Integration with Pathology

The AIM team, along with DICOM WG-18, will provide technical and consulting support to the Emory team to accomplish a Pathology information model.

5.4. AIM 4.0 C++ Library and Referenced Implementation

After the AIM 4.0 model has been finalized, it will be used to develop the AIM C++ library.

Appendix A References

No.	Artifact	Type	Link
1	02_03_AIM_Project_Report.doc	Microsoft Word	https://gforge.nci.nih.gov/frs/download.php/9112/AIM_v3.0.1_rv11.rar
2	BRIDG	URLs	http://www.bridgmodel.org/ http://gforge.nci.nih.gov/frs/?group_id=342
3	LS DAM	Wiki	https://wiki.nci.nih.gov/display/LS/Life+Sciences+Domain+Analysis+Model+%28LS+DAM
4	DICOM	FTP	ftp://medical.nema.org/medical/dicom

Appendix B Glossary

Term	Definition
AIM	Annotation and Image Markup
API	Application Programming Interface
caAERS	Cancer Adverse Events Reporting System
DICOM	Digital Imaging and Communications in Medicine
GUI	Graphical User Interface
XML	eXtended Markup Language
UID	Unique Identifier
UML	Unified Modeling Language