

# Java Application Programming Interface (API) for Annotation Imaging Markup (AIM)

Hakan Bulu<sup>1</sup>, Daniel L. Rubin<sup>2</sup>

<sup>1</sup>Dokuz Eylul University, Department of Computer Engineering, Izmir, Turkey

<sup>2</sup>Stanford University, Department of Radiology and Center for Biomedical Informatics Research, Stanford, U.S.

## 1. Introduction

### *About AIM*

Information Sciences in Imaging at Stanford (ISIS) Laboratory of Stanford University is currently funded by the National Cancer Institute of the National Institutes of Health to develop techniques and tools to enable extracting quantitative and semantic information from radiology images in a standard format called Annotation and Image Markup (AIM).

The goal of the AIM project is to develop a mechanism, for modeling, capturing, and serializing image annotation and markup data that can be adopted as a standard by the medical imaging community. The AIM project produces both human and machine-readable artifacts. One of the most important points about the AIM is storing the annotations as XML (Extensible Markup Language) format called as “AIM XML”. Additional applications transform the AIM XML into DICOM-SR, HL7-CDA XML and AIM ontology instances. The AIM model is shown in Figure 1. For more information about the AIM project; [1], [2], [3].

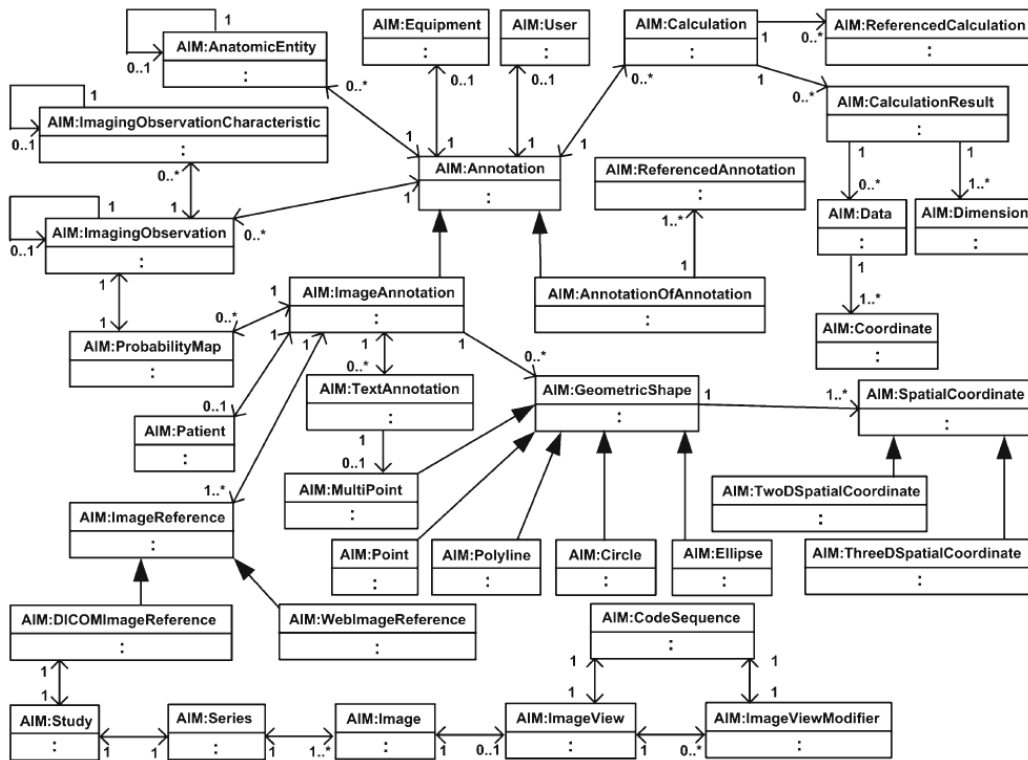


Figure 1 - AIM UML Class Diagram

*Simplify the usage of AIM structure.*

Structure of the AIM is not small and understanding this structure can be difficult when you become familiar. Also, if any developer/researcher wants to use AIM, they have to know how to parse and write XML files. In other words they have to know XML technologies. Because of that, to simplify the usage of AIM Structure for Java programmers, we have developed an Application Programming Interface (API) called as “AIM API”. The AIM API [4] was developed in Java programming language by using NetBeans IDE 7.0.1 and it creates an object model, using Java classes for each class in the AIM Schema. The class hierarchy of the API closely follows the AIM Schema. Each class in the object model provides Set and Get methods for every attribute in the corresponding AIM Schema class. You can download the source code of the API from [5].

*Perform operations (read and write) for AIM XML files, both on hard drive (local computer) and XML Database (remote server).*

By using the API, users can not only create new AIM XML files, but also edit any existing AIM XML files where the files can be current or older version of the AIM. The XML files can be saved to both local file system and remote XML database. Additionally, users are able to query one or set of AIM Annotations from the server by using predefined functions.

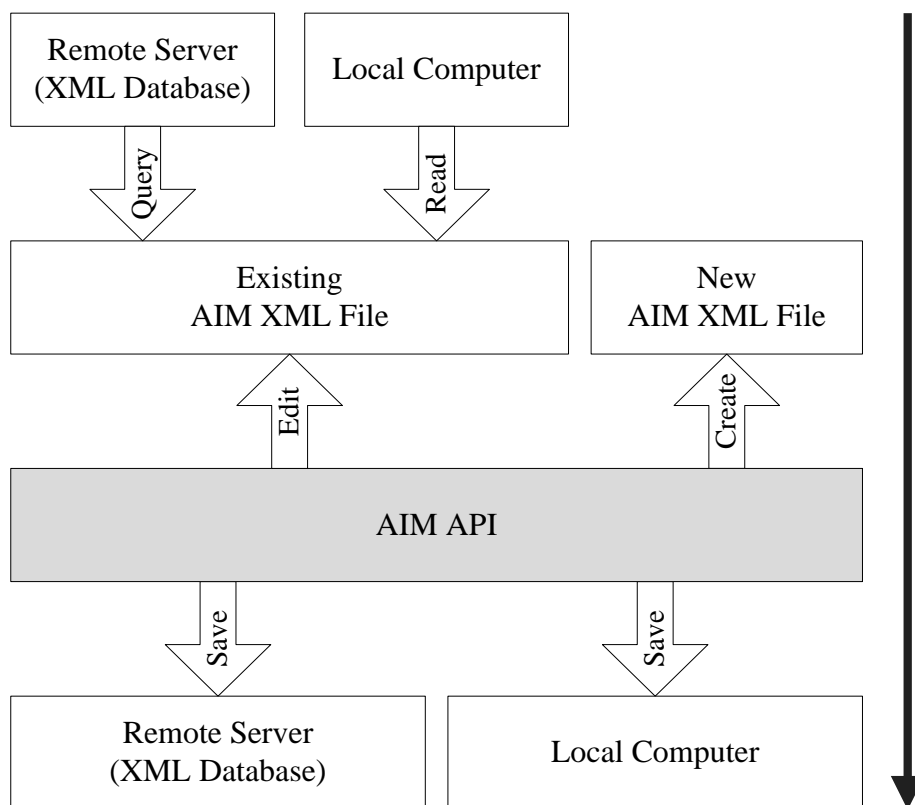


Figure xyz – General system overview of the API

*While the AIM version will be increased, there is no significant change for the usage of the API.*

Since the AIM project started to develop, the version of AIM XSD have been updated. So, reading/parsing any version AIM XML file is an important capability for the API. At this point

Instead of creating individual AIM API for each AIM XSD file, we have

So, the users must

To be able to analysis AIM annotations with Semantic Web Technologies, the annotations must be converted AIM Ontology Instances. So, I have updated the API with respect of this necessity. In following sections sample usage of the API is given to simulate the annotation of a lesion on the mammogram.

*Validate the existing and new AIM XML files based on the AIM XML Schema (XSD). In this way, reducing inconsistency between AIM XML files.*

Each AIM XML file has to be valid according to AIM XSD so validation process is very important.

## **2. Methods**

*Which programming language did we use?*

We use Java programming language to develop the AIM API.

*About XML and XSD?*

## **3. Results**

*Who will use the API?*

*Sample usage of the API*

## **4. Conclusion and Future Works**

*What did we do?*

*What did we obtain?*

*What will we do?*

## **5. Acknowledgements**

This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under project number 107E217 and National Cancer Institute (NCI) through the cancer Biomedical Informatics Grid (caBIG) Imaging Workspace subcontract from Booz-Allen & Hamilton, Inc. 85983CBS43.

## **6. References**

[1] Rubin DL, Mongkolwat P, Kleper V, Supekar K and Channin DS, Medical Imaging on the Semantic Web: Annotation and Image Markup. In: 2008 AAAI Spring Symposium Series, Semantic Scientific Knowledge Integration, Stanford University, 2008.

[2] Rubin DL, Supekar K, Mongkolwat P, Kleper V and Channin DS, Annotation and Image Markup: Accessing and Interoperating with the Semantic Content in Medical Imaging. IEEE Intelligent Systems 24(1): 57-65, 2009.

[3] Channin DS, Mongkolwat P, Kleper V, Sepukar K and Rubin DL, The caBIG Annotation and Image Markup Project. J Digit Imaging, 2009.

[4] AIM API Web Page, [http://www.stanford.edu/group/qil/cgi-bin/mediawiki/index.php/AIM\\_API](http://www.stanford.edu/group/qil/cgi-bin/mediawiki/index.php/AIM_API), 2011.

[5] AIM API Spurce Code, <http://sourceforge.net/projects/aimapi/>, 2011.