

Development of a Data Integration and Visualization Software for LIDC

Chao Zeng[1, 2]
[1]School of Geosciences and Info-physics, Central South University, Changsha, China
[2]College of Information Science and Technology, Shihezi University, Shihezi, China
Email: zengc_bme@csu.edu.cn

Hongli Lin[*1, 3] and Weisheng Wang[3]
[*1]School of Geosciences and Info-physics, Central South University, Changsha, China
[3]College of Information Science and Engineering, Hunan University, Changsha, China
Email: hllin@hnu.edu.cn

***Abstract*—With the increasingly using of LIDC data in lung cancer research, education, and clinical medicine areas, the needs for effectively accessing and visualizing the CT scans and annotations collected by LIDC is more and more underscored. In this paper, we introduce the data integration and visualization software for the LIDC data exploration. A XML-based data model is provided for storing the LIDC data to combine different kinds of information into a uniform format. The data integration component is developed based on technologies of Java Excel API and JAXP. A nodule viewer is developed to visualize DICOM images and the annotations (commonly known as nodules) using DCMTK. The software offers a solution to the challenge of lack of a uniform data format for the LIDC data and nodule viewer. This software has the potential application in the areas of lung cancer research, education and patient care.**

***Index Terms*—LIDC, XML, DICOM, Data Integration, Visualization**

I. Introduction

Lung cancer is the primary cause of cancer related deaths in the world [1]. This makes lung cancer a major frontline in the war against cancer. Early detection of potentially cancerous pulmonary nodules may be a way to improve patient's chance for survival [2]. CT is the most commonly used diagnosis technique for detecting small pulmonary nodules due to its sensitivity and capacity to represent/visualize a complete three-dimensional structure of the human chest [3]. However, each scan contains numerous sectional images that must be evaluated by a radiologist in a potentially fatiguing process. As a result, the performance of a human interpreter can be adversely affected. Therefore, it is an important task to develop computer aided detection (CAD) system that can aid radiologist workflow and potentially reduce false positive findings.

To further stimulate research and development activities in lung cancer CAD, Lung Image Database Consortium (LIDC) was formed under the aegis of the Cancer Imaging Program at NCI. It aims to develop an image database as a web-accessible international research resource for the development, training, and evaluation of the CAD system for lung cancer detection and diagnosis using CT. This database enables the correlation of performance of the CAD methods for detection and classification of lung nodules with spatial, temporal, and pathologic ground truth. To some degree it can hasten advancement of lung nodule CAD research by providing clinical images to investigators who might not have access to patient images and creating a reference database that will support the relative comparison of the different CAD system performance [4][5].

LIDC have collected and released a database of lung CT scans with annotations of nodule. Up to date, the LIDC database contained 399 patient cases including 85 cases from an earlier release [6]. This data is stored in anonymized DICOM format [7] that eliminates patient identification elements from the images, but preserves other data annotations such as dosage and reconstruction parameters as well as the make and model of the scanner for each study [8]. The image data is accompanied with an XML file including information from four experienced radiologists' reading. The XML details nodules smaller than 3mm in diameter, nodules larger than 3mm and non nodules [9]. In early 2009, the LIDC Nodule Size Report in EXCEL format with size estimate for all the pulmonary nodules with boundary markings was provided to augment the LIDC public image database [10]. The purpose [10] of the list is to provide a common size index for the selection of subsets of nodules with a given size range. The goal is to ensure that when multiple research groups use the same size-selected sub-range of nodules that they will be using the same set of nodules as each other. In early 2010, an EXCEL file with diagnostic data associated with the case was released. EXCEL details the diagnosis results and diagnosis methods at the patient level and the nodule level [9].

*Corresponding author: Hongli Lin, Email: hllin@hnu.edu.cn.
This work was partially supported by the National Natural Science Foundation of China under grant project No. 81201151, and the Science Foundation of Hunan Province, China under grant project No. 12JJ6061.

doi:10.4304/jsw.8.9.2297-2304

LIDC database provides an excellent resource for CAD algorithms and plays an important role in CAD research [11] [12]. The development of the LIDC has led to a large amount of research based on the image sets. In Meyer et al [13], the authors investigate the effects of radiologist agreement on the development of a "ground truth" and the subsequent impact on CAD performance. In Anthony et al. [19], the authors compare the volumetric sizes computed by their research system with those derived from the LIDC archive.

However, the effort of LIDC has concentrated on data collection and transmission and has left development of application and tools to the research community. Tools that allow users to access to and manipulate such data effectively are becoming essential.

Firstly, as LIDC database is a set of various kinds of files, it is difficult to access to data collected in LIDC database. For example, to acquire the total information of a nodule, users should obtain the nodule outline and characteristics from XML file, the size information from the LIDC Nodule Size Report (a EXCEL file), and the diagnosis results and methods from the EXCEL diagnosis data file respectively.

Secondly, because of the specific image annotation encoding, conventional DICOM image viewers such as Osirix [15], ClearCavas [16], and ImageJ [17] lack the capability to visualize nodules stored in LIDC XML format. As far as we know, no free tool has the ability to visualize nodules stored in the XML format provided by LIDC. Thus, it is necessary to provide a nodule viewer with the capability of visualizing nodules and images.

The goal of our work is to develop a data integration component and nodule viewer for LIDC data to allow researchers to access and visualize effectively data collected by LIDC.

II. Design Considerations

A single format to store image annotations will enable researchers who focus on providing rich accurate CAD algorithms to access to them easily and quickly. To provide a unified data view for researchers, both XML-based storage and free/commercial database solutions were considered. The portable and interchangeable XML format was chosen in view of the requirements of data exchange. Based on the original XML schema provided by LIDC, new XML schemas were designed to represent unambiguously the annotations and the LIDC cases index. Note that in this paper, a 'case' refers to a CT scan of a single patient.

Once the XML-based solution is determined, a data integration program should be provide for combining the information in XML format and the one in EXCEL format into a unified data model. How to parse XML file and EXCEL file must be considered. Due to the language familiarity of the development team, Java was chosen to implement this software. There are some different types of parsers, DOM (Document Object Model), SAX (Simple API for XML) and JDOM. DOM [18] parsers parse the XML document and create an object-oriented hierarchal representation of the document, which can be navigated during run-time. The structure is intuitive and easy to manipulate, but it is very resource demanding both in memory and CPU usage. SAX parsers [19] do not store any information. Instead they scan the information and call handler functions that are associated with specific instructions and tags in the XML document. The SAX parser is probably faster than the DOM parser and is not as resource intensive. The drawback is that the logic for handling the XML document instructions and tags is significantly more complicated and complex. JDOM provides a means of accessing XML document within a tree structure, and in that respect is somewhat similar to the DOM [20]. However, it was built specifically for Java and more intuitive in many ways to a Java developer than DOM. Though SAX, DOM and JDOM can meet all the requirements of handling XML document, taking into account the XML specification, we selected JDOM. We used JDOM to implement a component for reading, recording, updating annotations in an XML file.

We used JDOM to implement a component for reading, recording, updating annotations in an XML file.

Parsing EXCEL file is another issue in our work. We used the JExcel API [21] (the API technology for java developed by Khan) to interface with MS Excel format spreadsheets.

There are two approaches to visualizing DICOM images and annotation in XML format. One way is converting XML-based annotation into DICOM annotation as a data element of DICOM file, and then DICOM image and annotation can be visualized using traditional DICOM viewer. However, the reuse of DICOM image and annotation may be a challenge and the capability of customizing annotation visualization may be lost. The other approach is storing annotation as an external annotation file and implementing a customized nodule viewer to adapt the specific application. In view of the future usage of annotations, we selected the latter.

There are some SDKs used to build nodule viewer such as LEADTOOLS [22], DCMTK [23], and ITK [24]. We developed our nodule viewer based on DCMTK. DCMTK is a collection of libraries and applications implementing large parts the DICOM standard developed by the OFFIS computer science institute. It includes software for examining, constructing and converting DICOM image files, handling offline media, sending and receiving images over a network connection, as well as demonstrative image storage and work list servers. DCMTK is written in a mixture of ANSI C and C++. It comes in complete source code and is made available as "open source" software.

Based on DCMTK, we implement a nodule viewer for handling DICOM images and annotations. The nodule viewer features the capacity to read in and display CT imaging studies, basic image manipulation functions such as window/level, zoom, pan, cine view, extracting information from DICOM headers through data element tag, and personalized displaying annotation (nodule) on DICOM image.

III. System Description

A. Software Architecture

In modern software engineering, layer architecture [25] [26] is mostly used, in which the lower layer provides application programming interfaces (API) to the one upon it. This architecture makes the software convenient to be extended, developed and maintained again. Multi-layer software architecture was introduced into this software design. This software was structured in three layers: presentation layer, behavior layer and persistence layer. Fig.1 shows this software architecture.

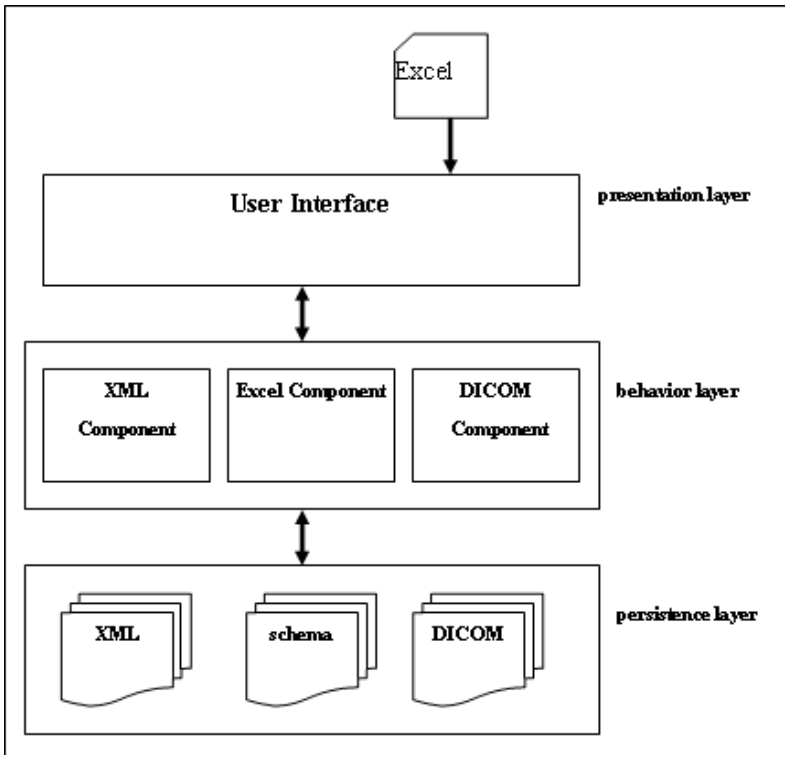


Figure 1. The software architecture.

The presentation layer includes the user interface developed using Java Swing. Basically, it provides user interface to interact with users. Users can click button, menu item and other visual component to invoke specific functions.

The behavior layer deals with DICOM image, XML file and Excel file. Tree components compose this layer: DICOM component, XML component and Excel component. The DICOM component wraps all the complicated functionality on DICOM involved in reading in CT imaging studies, basic image manipulation functions (window/level, zoom, pan, cine view), extracting information from DICOM headers and converting DICOM file into other types files (BMP, TIFF, JPEG). XML component encapsulates much of functionality on handling XML file including reading, writing, querying XML document. The Excel component wraps the functionality on dealing with EXCEL file.

The persistence layer provides storage data for upper layer. In this software, it contains a collection of XML document and DICOM document.

B. Persistence Layer

To store image annotations, LIDC developed a portable and interchangeable file format implemented in XML. The XML schema [6] was designed to represent unambiguously the results of all readers' review of each CT scan. Each XML file contains information as fellows: the CT being reviewed, the type of reading session, reader identification, results from the reading session, where a reading session consists of a set of annotations marked by a single reader. For each nodule larger than 3mm, both the nodule characteristics and the full volumetric boundary are reported, while for each nodule smaller than 3mm, only the approximate centroid point and the identification of the nodule are reported. For each non-nodule larger than 3mm, the data of it are similar to data recorded for nodule smaller than 3mm.

To provide a unified view of data for researcher, we combined the XML-based specification of annotations, the nodule size information in EXCEL, and the diagnosis information of the nodule into a uniform XML format. The XML schema was designed as shown in Fig.2.

To meet the needs to browse and retrieve the information about LIDC case and nodule, a XML file containing information about LIDC case as index was provided and the XML schema (Fig.3) was designed. The LIDC case XML file contains information about: 1) the case identification: a unique id, 2) the diagnosis result and methods of the case, 3) the DICOM image collection

(each DCIOM image contains the DICOM file path and name, the image SOP_UID identification), 4) the annotation XML file name of this case.

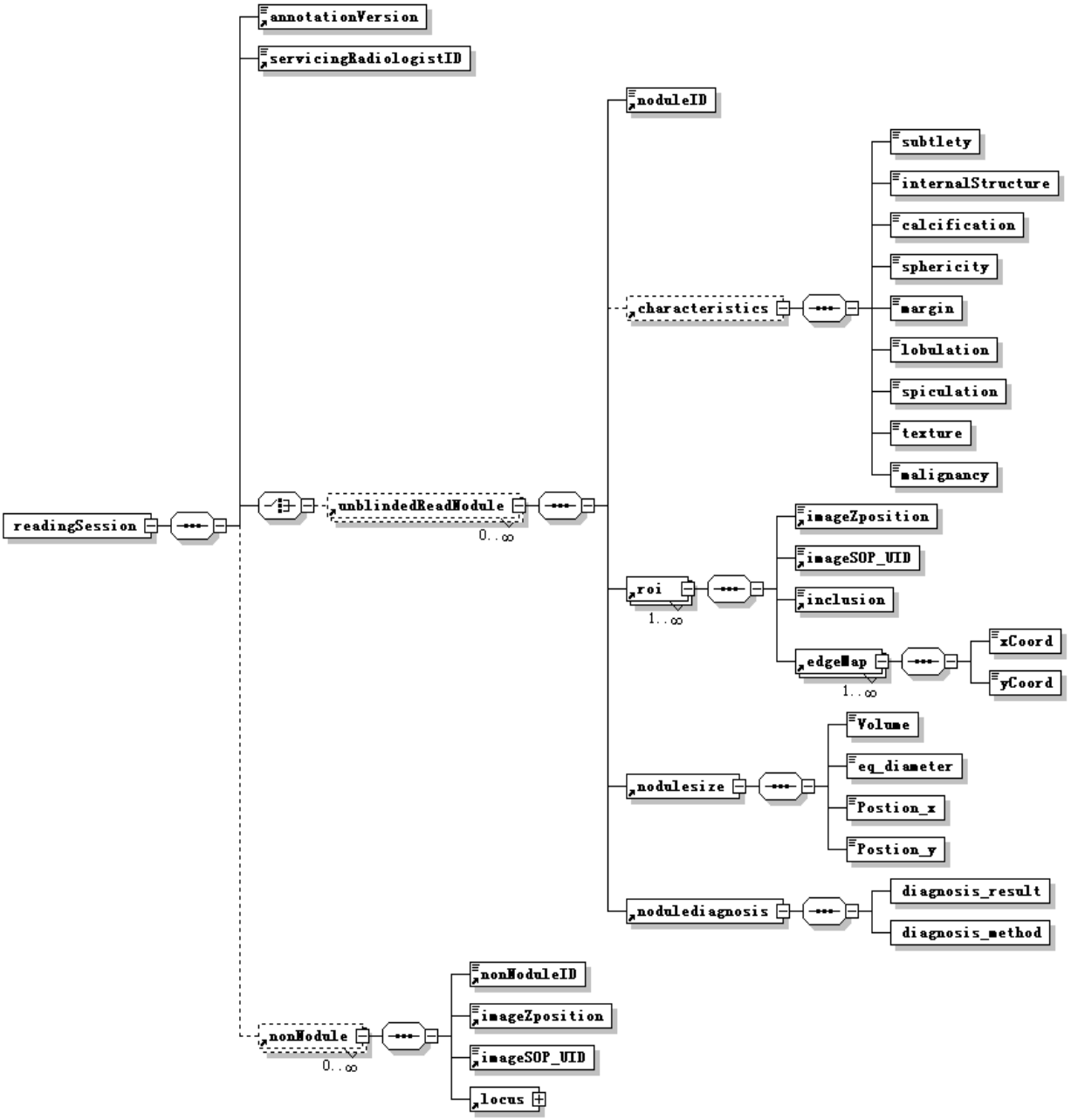


Figure 2. The annotation XML scheme.

C. Behavior Layer

The main components of the software are placed in the behavior layer, which deals with the requests coming from presentation layer and manipulates the data storing in persistence layer. XML component, DICOM component and Excel component compose this layer.

XML component encapsulates the functionality on manipulating XML document. To hide the tree structure from other component and presentation layer, XML component wraps major nodes with classes through mapping tag to class. Once XML component read in and parses XML document, an objects tree structure is created in memory, that is to say a node is an object. So, methods of the classes are used to manipulate the data in the tree. Each of the classes provides methods for insertion, deletion, update, and retrieval of information about its corresponding XML tags. This design that wraps classes around XML tags, makes it easy for XML component to be called by other components. In addition, maintenance of this program is easy.

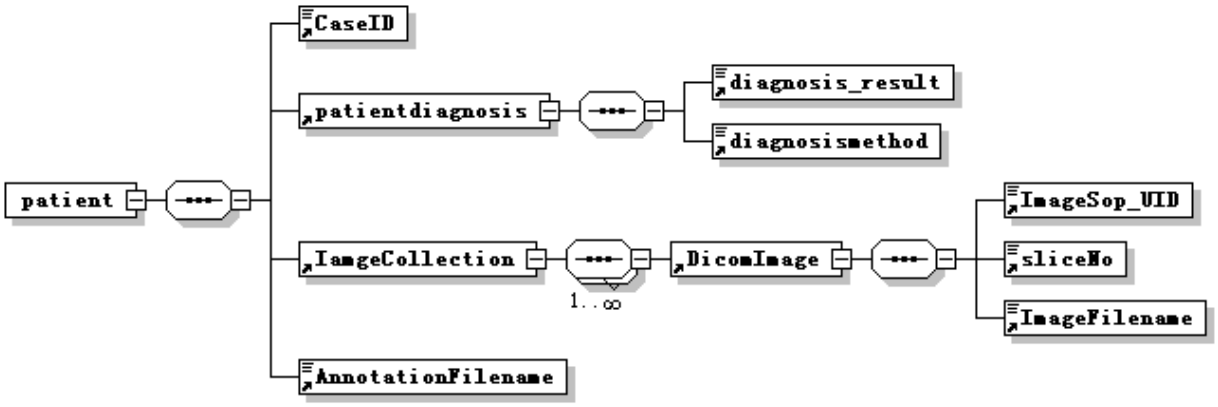


Figure 3. The case index XML scheme.

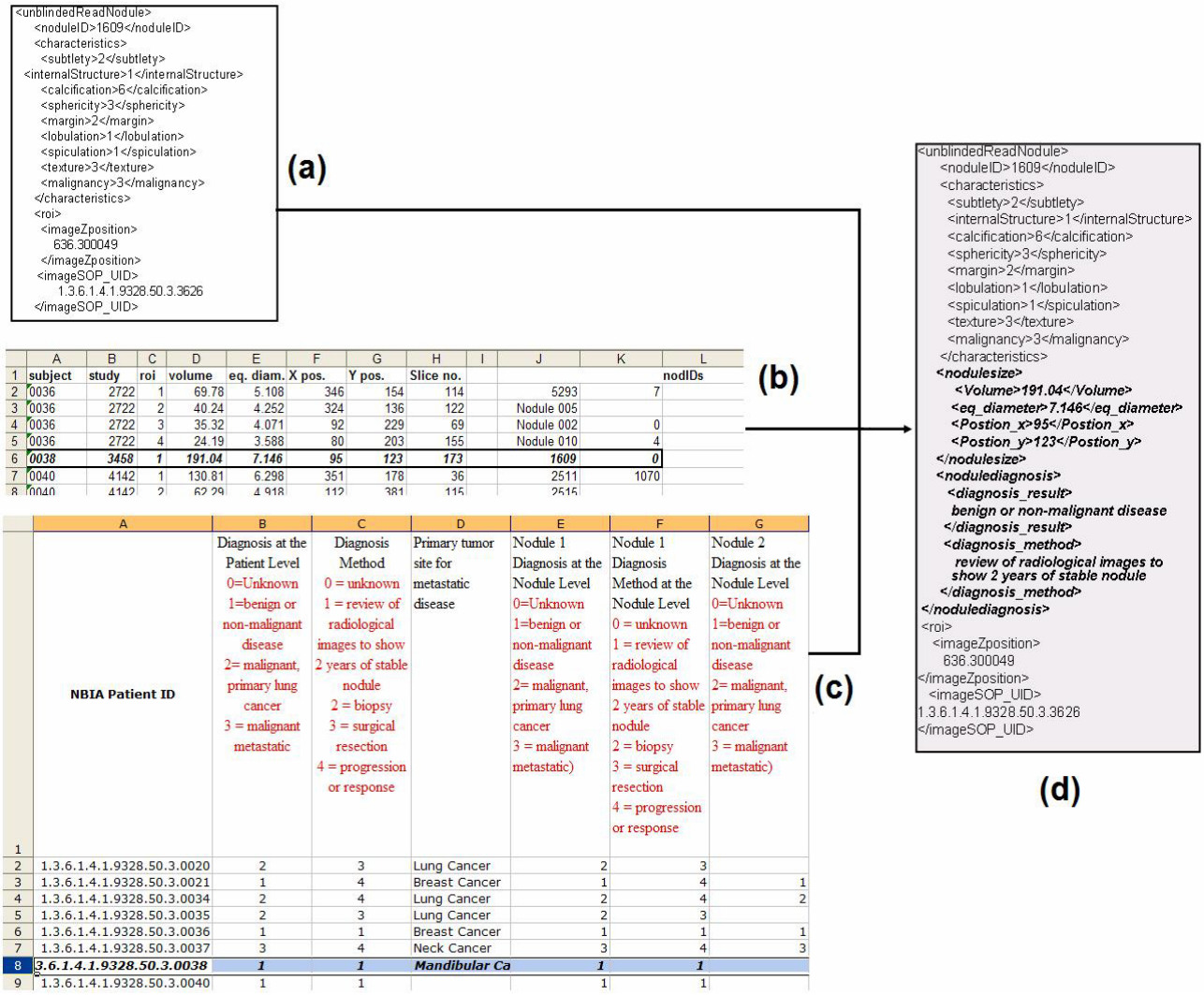


Figure 4. An example of annotation Integration: (a) a part of the annotation of nodule1609 in LIDC case 13614193285030038 before integration, (b) a part of the size data in EXCEL, in which the size information of nodule 1609 in LIDC 13614193285030038 is highlighted in bold and italic, (c) a part of the diagnosis data in EXCEL, of which the diagnosis information of LIDC case 13614193285030038 is highlighted in bold and italic too, (d) a part of the result annotation of nodule 1609 after integration, in which the newly added information marked in bold and italic.

Modifying the XML component to use a different XML parser that keeps the tree structure in memory can be done without affecting the code in other component. Access to information about XML tag of a given LIDC term is quick because the object instantiated from the wrapper class of the tag knows the exact location in the XML tree to update without having to traverse the tree from the root. XML component provides the capacity of creating, loading, update, and saving annotation in object form.

DICOM component provides a series of functionality on handling DICOM image. DICOM component is composed of two major modules: Dcmimgle and Dcmdata. The Dcmimgle module contains classes to access and render DICOM monochrome images, aiming to image process including zoom, clip, and window level/width of image. The Dcmdata module contains classes to manage DICOM data structures and files, for instance, getting the value of a specific data element.

Excel component is a simple class. It provides the ability to read in Excel file and to extract information about nodule size, nodule diagnosis, and patient diagnosis.

D. Presentation Layer

The main feature of presentation layer is its user interaction facility which makes it attractive for commercial and research usages and the capacity of showing information.

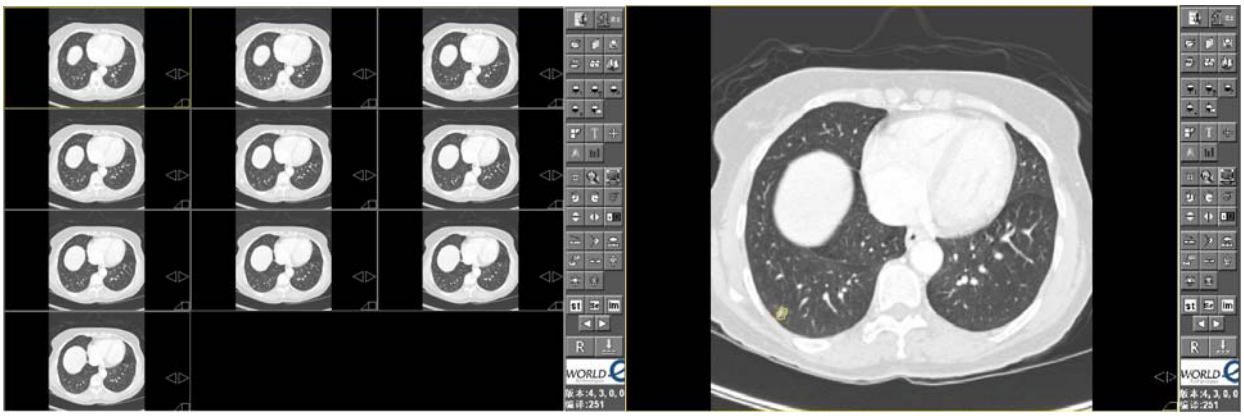
In this layer, a tradition UI engine is developed base on the requirements of the UI. This engine covers the base functionalities of a UI such as menus, toolbars, and routines image handling and drawing functionalities. It contains some specific UI components to meet the requirements of viewing and visualizing nodules and other annotations.

E. Data Integration

Once data model is developed, the data integration process should be called to build database. When integrating data, the main Java program starts by calling data integration programming and transfers the directory name containing a patient's CT scan and the annotation XML file, the nodule size report filename, and the EXCEL filename containing the diagnosis data for it as input parameters. After receiving the input parameters from the main Java program, the data integration programming gets all files in the directory and calls DICOM component for extracting some technical parameters from DICOM image, then EXCEL component is called to get the size, diagnosis information of nodules that belong to the patient and the patient's diagnosis data. In next step, the XML component is called to insert the size and diagnosis information of nodules from the excel component into the XML annotation file of the patient and create or modify the patients index XML file to insert the information about CT scan and the diagnosis at patient level. An example of integrating data is shown in Fig.4.

F. Nodule Viewer

A nodule viewer was developed for displaying DICOM image and annotation. This viewer can deal with DICOM image with the ability to load in DICOM image studies or a single DICOM image. Each image can be rotated, flipped or mirrored. Continuous or discrete zooms are available for image viewing. Each image can be viewed separately in a new window. In addition, each image can be zoomed and translated. At same time, the nodule viewer has unique capacity of displaying annotations encoded in specific LIDC XML format. Users can display DICOM image and annotation in a way in which they are interested. An example of DICOM image and annotation visualization is shown in Fig.5.



(a) (b)

Figure 5. An example of DICOM image and annotation visualization, on (a) only the DICOM images containing nodules larger than 3mm of LIDC case 13614193285030038 are visualized, on (b) the DICOM image and the boundaries of nodules on it are shown, the DICOM image in (b) is the one showing on row 2/ column 2 in (a).

IV. Discussion and Conclusion

We have developed this software for integrating different kinds of data into a uniform XML format and visualizing DICOM images and annotations. With this software, a repository of lung CT scans with annotations has been built. With nodule viewer and the database, CAD researchers will be able to store the results of CAD algorithms and compare visually them to ground truth stored in XML files. For medical students, residents, and fellows, this software is useful to learn to recognize the common lung abnormalities and then can see the characteristics described by experienced radiologists as well as the pathology diagnosis results. This software will be a useful tool in the future.

Several lessons were learned from the development of this software, and the most important point is that the component-based method made it easy to reuse the component in behavior layer in other relevant application. The DICOM component can be used in other application about DICOM image without changed.

In the future, additional features will be added to enhance nodule viewer as a useful tool for viewing DICOM image and annotation such as the ability to retrieve nodule from annotation file given some LIDC terms or an example image.

Acknowledgment

This work was partially supported by the National Natural Science Foundation of China under grant project No. 81201151, the Science Foundation of Hunan Province, China under grant project No. 12JJ6061, and the Research Plan on Science and Technology of Hunan Province, China under the grant project No. 2012SK3185.

References

[1] L. L. Humphrey, S. Teutsch, M. Johnson, “Lung cancer screening with sputum cytologic examination, chest radiography, and computed tomography: an update for the U.S. Preventive Services Task Force”, Annals of Internal Medicine, vol.140, pp.740-753, 2004.

[2] C. I. Henschke, D. P. Naidich, D. F. Yankelevitz, G. McGuinness, D. I. McCauley, J. P. Smith, D. Libby, M. Pasmantier, M. Vazquez, J. Koizumi, D. Flieder, N. Altorki, O. S. Miettinen, “Early lung cancer action project: Initial findings on repeat screenings”, Cancer, vol.92, no.1, pp.153-159, 2001.

[3] K. Suzuki, M. Kusumoto, S. Watanabe, R. Tsuchiya, H. Asamura, “Radiologic Classification of Small Adenocarcinoma of the Lung: Radiologic-Pathologic Correlation and Its Prognostic Impact”, The Annals of Thoracic Surgery, vol.81, no.2, pp.413-419, 2006.

[4] S. G. Armato III, G. McLennan, M. F. McNitt-Gray, C. R. Meyer, D. Yankelevitz, D. R. Aberle, C. I. Henschke, E. A. Hoffman, E. A. Kazerooni, H. MacMahon, A. P. Reeves, B. Y. Croft, L. P. Clarke, “Lung Image Database Consortium: Developing a Resource for the Medical Imaging Research Community”, Radiology, vol.232, pp.739-748, 2004.

[5] Hongli Lin, Zhencheng Chen, Weisheng Wang, "Pulmonary Nodule View System for the Lung Image Database Consortium (LIDC)", Academic Radiology, vol.18, no.9, pp.1181-1185, 2011. vol.18, no.9, pp.1181-1185, 2011.

[6] https://imaging.nci.nih.gov/ncia/.

[7] Peijiang Chen, “Study on Medical Image Processing Technologies Based on DICOM”, Journal of Computers, vol.7, no.10, pp.2354-2361, 2012.

[8] M. F. McNitt-Gray, S. G. Armato III, C. R. Meyer, “The Lung Image Database Consortium (LIDC) Data Collection Process for Nodule Detection and Annotation”, Academic Radiology, vol.14, no.12, pp.1464-1474, 2007.

[9] https://wiki.nci.nih.gov/display/CIP/LIDC.

[10] http://www.via.cornell.edu/lidc/.

[11] Stelmo Magalhães Barros Netto, Aristófanes Corrêa Silva, Rodolfo Acatauassú Nunes, Marcelo Gattassc, "Automatic segmentation of lung nodules with growing neural gas and support vector machine", Computers in Biology and Medicine, vol.42, no.11, pp.1110-1121, 2012.

[12] D. Cascio, R. Magro, F. Fauci, M. Iacomi, G. Raso, "Automatic detection of lung nodules in CT datasets based on stable 3D mass–spring models", Computers in Biology and Medicine, vol.42, no.11, pp.1098-1109, 2012.

[13] C. R. Meyer, T. D. Johnson, G. McLennan, D. R. Aberle, E. A. Kazerooni, H. MacMahon, B. F. Mullan, D. F. Yankelevitz, E. J. R. van Beek, S. G. Armato III, M. F. McNitt-Gray, A. P. Reeves, D. Gur, C. I. Henschke, E. A. Hoffman, P. H. Bland, G. Laderach, R. Pais, D. Qing, C. Piker, J. F. Guo, A. Starkey, D. Max, B. Y. Croft, L. P. Clarke, “Evaluation of Lung MDCT Nodule Annotation Across Radiologists and Methods”, Academic Radiology, vol.13, no.10, pp.1254-1265, 2006.

[14] A. P. Reeves, A. M. Biancardi, T. V. Apanasovich, C. R. Meyer, H. MacMahon, E. J. R. van Beek, E. A. Kazerooni, D. Yankelevitz, M. F. McNitt-Gray, G. McLennan, S. G. Armato III, C. I. Henschke, D. R. Aberle, B. Y. Croft, L. P. Clarke, “The Lung Image Database Consortium (LIDC): a comparison of different size metrics for pulmonary nodule measurements”, Academic Radiology, vol.14, no.12, pp.1475-1485, 2007.

[15] http://www.osirix-viewer.com/.

[16] http://www.clearcanvas.ca/.

[17] http://rsbweb.nih.gov.

[18] http://www.w3.org/DOM/.

[19] http://www.saxproject.org/.

[20] http://www.jdom.org/.

[21] http://www.andykhan.com/jexcelapi.

[22] http://www.leadtools.com/.

[23] http://dicom.offis.de/dcmtk.php.en.

[24] http://www.itk.org/.

[25] Y. Zhang, X. Liu, Z. Wang, L. Chen “A Service-Oriented Method for System-of-Systems Requirements Analysis and Architecture Design”, Journal of Software, vol.7, no.2, pp.358-365, 2012.

[26] D. Batory, S. O. Malley, “The Design and Implementation of Hierarchical Software Systems with Reusable Components”, ACM Transactions on Software Engineering and Methodology, vol.1, no.4 pp.355-398, 1992.

Chao Zeng was born in Yiyang, China in August, 1982. He received his B.S. from Xiangtan University in 2005. He is now a Ph.D. candidate in Central South University, Changsha, China. He is also a lecture in Shihezi University, Shihezi, China since 2010. His major field of study covers biomedical image and biomedical signal processing techniques.

Hongli Lin was born in Henan, China in July, 1973. She received her Ph.D degree from Central South University in 2011. She is now an associate professor in Hunan University. Her major field of study covers biomedical image and software engineering.

Weisheng Wang was born in Gansu, China in May, 1972. He received his Master degree from Central South University in 2008. He is now an assistant professor in Hunan University. His major field of study covers biomedical image and software engineering.