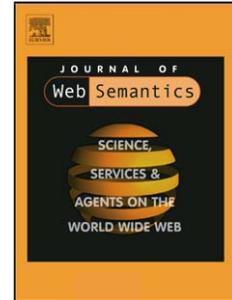


Accepted Manuscript

Title: Interactive Publication: The document as a research tool

Authors: George R. Thoma, Glenn Ford, Sameer Antani, Dina Demner-Fushman, Michael Chung, Matthew Simpson

PII: S1570-8268(10)00023-5
DOI: doi:10.1016/j.websem.2010.04.001
Reference: WEBSEM 191



To appear in: *Web Semantics: Science, Services and Agents on the World Wide Web*

Received date: 14-5-2009
Revised date: 29-3-2010
Accepted date: 1-4-2010

Please cite this article as: G.R. Thoma, G. Ford, S. Antani, D. Demner-Fushman, M. Chung, M. Simpson, Interactive Publication: The document as a research tool, *Web Semantics: Science, Services and Agents on the World Wide Web* (2008), doi:10.1016/j.websem.2010.04.001

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Interactive Publication: *The document as a research tool*

George R. Thoma, Glenn Ford, Sameer Antani, Dina Demner-Fushman, Michael Chung, Matthew Simpson
U.S. National Library of Medicine, Bethesda, Maryland

Abstract

The increasing prevalence of multimedia and research data generated by scientific work affords an opportunity to reformulate the idea of a scientific article from the traditional static document, or even one with links to supplemental material in remote databases, to a self-contained, multimedia-rich *interactive* publication. This paper describes our concept of such a document, and the design of tools for authoring (Forge) and visualization/analysis (Panorama). They are platform-independent applications written in Java, and developed in Eclipse¹ using its Rich Client Platform (RCP) framework. Both applications operate on PDF files with links to XML files that define the media type, location, and action to be performed. We also briefly cite the challenges posed by the potentially large size of interactive publications, the need for evaluating their value to improved comprehension and learning, and the need for their long-term preservation by the National Library of Medicine and other libraries.

Keywords: Interactive publication, multimedia, visualization tool, authoring tool, clinical images, DICOM, medical video

1. Introduction

Multimedia and large stores of research data are increasingly considered indispensable to the scientific publishing enterprise. While “multimedia documents” have been in existence for a decade or more, in common usage they refer to entities that consist of text that links to images and video clips, which usually reside in databases apart from the text. In contrast, our vision is a comprehensive, self-contained and platform-independent multimedia document with capability to connect with online material, with which a reader can interact intuitively and fluidly [1]. Such an “interactive publication” (IP) could contain many media objects: text, video, audio, bitmapped images, spreadsheets, presentation graphics, or animation sequences. These objects could be in different file formats: e.g., text in Microsoft[®] Word or PDF, spreadsheets in Microsoft Excel, video in Adobe[®] Flash Video or Quicktime[®] format, animations in Flash, clinical images following the DICOM² standard, 3D renderings of 2D image sequences, etc.

While using such a document, the reader should be able to: (a) view any of these objects on the screen, and importantly, *without losing context within the text*; (b) link from one object to another; (c) interact with the objects in the sense of exercising control over them (e.g., start and stop video); (d) reuse the media content for analysis and presentation (e.g., convert data tables to graphs, or perform statistical analyses).

Such a publication would have the conventional characteristics of a “document” in the sense of a completed work of an author presenting hypotheses, findings and conclusions, but also give a reader (or peer reviewer) the facility to check the underlying data and possibly derive alternative conclusions. In other words, the “document” becomes a research tool.

By creating such documents and providing tools for their usage, we propose to partner with publishers to disseminate these to a broad readership, and examine improvements in learning and comprehension over conventional (static) publications.

¹ <http://www.eclipse.org>

² DICOM: Digital Imaging and COmunications in Medicine standard. (<http://medical.nema.org>).

2. Interactive publications: desirable attributes

What makes a document an Interactive Publication? A core set of desirable attributes of such a document, as we envision it, appears in the list below. However, this list is a subset of a larger collection of desired features and functions, many of which can be found in modern document browsers and authoring tools and are thus ignored in this discussion. Examples of the latter are features inherent to document creation such as authoring, styling, layout controls, table creation, and referencing, among others that are commonly found in present-day tools. We consider the following attributes necessary for an interactive publication.

- Appearance
 - Paginated view of the document should be similar to that of a traditional article, implying the availability of a large variety of fonts, weights, styles, paragraphing, multi-column formatting, etc.
- Page transitions
 - Traditional use of keyboard keys (page up/down) and mouse (scroll bar) should be possible.
- In-page navigation
 - Traditional use of keyboard keys (cursor up/down/left/right) and mouse, as well as additional use of control keys (as shortcuts).
- Image browsing
 - Commonly used image formats, such as JPEG, PNG, TIFF, DICOM, should be natively supported.
 - It should be easy to encode some degree of interaction with these into the document model.
- Navigating to an embedded / linked media object
 - Mouse-click (or keyboard) activation of audio, video and other objects should be possible.
 - Embedded or linked media objects should be able to invoke appropriate viewers or players.
- Native support for interactivity
 - The document model should provide native support for adding interactivity to tabular data, images and other multimedia data.
 - The document model should allow authors to define metadata needed to control interactivity with multimedia data, e.g., start-frame and end-frame numbers for video, row-column selections in a table, etc. These metadata could enhance the reader's interaction with the document.
 - Data in specialized and proprietary formats should be viewable using appropriate supporting application software.
- Transmission
 - The document model should support a reader-controlled order of transmission for data intensive multimedia-rich documents for convenient usage.
- Embedding and linking of multimedia/interactive objects
 - The document model should support both embedding and linking of multimedia and other interactive data such as dynamic tables or active images.
- Document integrity and structure
 - It is imperative that the document be self-contained. That is, the multimedia components should exist within the document, and not simply exist in remote databases at, say, publishers' Web sites. This is important for several reasons, including the need for major libraries to preserve the scientific record, a difficult task if the contents of the document were scattered in remote locations.
 - The document model should support document integrity by closely linking the text document to the multimedia components. However, for a reader who might not be interested in downloading the datasets associated with the publication, a streaming media service should be available as an alternative.

Many of the desirable characteristics listed are found in present-day file formats, published standards, or recommendations. For example, it is possible to embed multimedia components within the Microsoft Office or Adobe Acrobat product families. However, the lack of a document framework that comprehensively addresses all of the desired characteristics provides impetus for our research. In the next section we present our approach to the development of tools for creating as well as viewing interactive publications.

3. Authoring and viewing tools

The literature on documents similar in characteristics to an IP [2,3,4,5] lacks information on reliable open-source and platform-independent authoring and reading tools with desirable capabilities, suggesting an opportunity for developing and freely distributing such tools. Our software consists of two applications; an authoring tool (Forge) and a viewer (Panorama). They are platform-independent applications written in Java, and developed in Eclipse³ using its Rich Client Platform (RCP) framework. A noteworthy benefit of the Eclipse RCP framework is that each component is natively a separate entity that can be detached from the application window. This is a useful feature in situations where, for example, multiple large analysis windows are to appear side-by-side without interference from other content, such as a multi-slice DICOM image view, or while comparing multiple high-resolution images.

Both Forge and Panorama operate on PDF files with links to XML files that define the *media type*, *location*, and *action to be performed*. An example XML file is shown in Figure 1 where the *media type* is a chart with two parameters (Polyp Diameter as the pattern and Colon Cancer as the score). The *media location* is the current folder with the file name “colon.csv” and the desired *interaction* is a receiver operating characteristic (ROC) analysis. Other media types, such as a CT study of the lung stored as DICOM image slices, may have other options specific to the initial view, or to a magnification of a particular slice in the study with appropriate window and level values, and to any image annotations pertinent to the topic being discussed in the article. All view options are specified in the XML file along with the definition of the media type and a path to its location. This path may also be a hyperlink to a remote site where the media may be situated.

3.1 Tool design considerations: Open source and standards vs. commercial software

Tools developed at the National Library of Medicine are in keeping with our philosophy of freely available, open-source software and/or standards. We also use in our development any pertinent software components that are similarly available. This approach avoids limitations introduced by commercial interests and allows newer technologies the protective environment needed during the critical gestation period. However, it is necessary to be careful when exploring new directions for well-established processes that may also be controlled by commercial interests. For example, in scientific publishing, publishers routinely provide PDF formatted files for client-downloadable versions of articles. Further, most authors prefer to use commercial authoring software such as Microsoft® Word to develop their documents. While it is possible to explore approaches where the IP is completely encoded in an open source standard, such as XML, which could then be converted to any alternate desirable format, e.g. PDF or HTML, it is challenging to do this when commercial products are widely used. Immediate introduction of such “extreme” shifts in accepted practices could introduce hurdles for the typical author or reader. Further, this could interfere with the evaluation of the science and tools. However, we intend to keep open source standards in sight and introduce them into our tools as appropriate.

INSERT FIGURE 1 HERE

³ <http://www.eclipse.org>

3.1 Forge: Authoring tool

Easily usable authoring tools are necessary for authors or publishers to conveniently create (and maintain) an interactive publication as it iterates through the peer review process. The author of an interactive publication would place media objects inline with the text. Forge is designed to streamline the process of adding interactive capability to the media objects by providing an easy and intuitive set of procedures, and where possible, through the use of wizards. The author must use the tool to connect the text document created in the conventional manner (with Microsoft Word[®] for example, and converted to PDF) to media objects; import (link) all media types of interest to placeholders in the document; provide intuitive navigation controls; and enable access to (appropriate) analysis and viewing tools. The media type definitions and desired actions are made possible through the use of XML-based files with a “.IP” extension in their names (.IP files, hereafter).

After the author has selected the PDF, Forge launches a wizard that will perform the following actions:

1. Check for existing defined .IP files. This step is skipped initially, since no .IP files exist. If the hyperlinks and .IP files are detected, the wizard will cycle through the hyperlinks to verify that .IP files are not out-of-sync and are assigned to the proper hyperlink.
2. Cycle through all hyperlinks for which .IP files do not exist, and prompt the user for:
 - a. Object Type –The author selects from a drop down box the appropriate object type (e.g., video, image, table, chart, etc.).
 - b. Open Object – The author is prompted to select the object file for linking.
 - c. Define Actions – The author is prompted with wizard screens asking for the type of action to be performed. For example, for a video, the author may want to create chapters or create playback settings to start at a specific position and run for a set period.
 - d. Wizard saves these settings to the .IP file as XML.
3. Preview Mode – Once the wizard has completed Step 2 the author may preview the IP and make any necessary adjustments.

The system is also designed to support a peer review process through its Preview Mode, with the additional ability for the reviewer to provide annotations. This helps maintain an auditable paperless review system. When revising an interactive publication, the author can iteratively repeat the processes discussed above to add or modify media objects. Modifications made during revisions can be tracked through .IP files that provide the capability to retain prior configurations. Additional work would only be required in specifying link actions for any newly added media objects. A Flash video of the Forge authoring tool is available at: <http://marg.nlm.nih.gov/IP/IPauthoring.html>.

3.2 Panorama: Viewing and analysis tool

The IP Viewer, Panorama, is configured to view PDF files and accepts Forge-authored IP. In case of an IP it processes the underlying .IP files that define the initial views to be presented to the reader (user).

INSERT FIGURE 2 HERE

Panorama’s capabilities are best-described using screenshots of the application, shown in Figures 2 through 5. In Figure 2, various sections of the application screen are labeled with green numbered circles and correspond to the following view panels:

1. **PDF View** – The PDF text in an IP is shown here, similar to a display in Adobe[®] Reader⁴. Clicking the active regions in the text document invokes the multimedia objects.

⁴ <http://www.adobe.com>

2. **Media View:** This is the interactive area for charts, tables, videos, DICOM images, Web browser, and other media types. The functionality available to a user depends on the media type. For example, video data has PAUSE, PLAY, and VOLUME, and CHAPTER SELECTION controls, while tabular data supports row selection, filtering, etc.
3. **Form View** – This window allows interaction with the data currently displayed in the Media View panel. The controls are dependent on the media type. In Figure 2, an ROC Curve is shown in the Media View panel corresponding to the graph in the article in the PDF View. The reader may select appropriate fields in the Form View to perform the ROC analysis.
4. **Recent View** – All previously viewed objects are listed as a tree, allowing the user to easily navigate between media objects. Double-clicking on any tree node will bring up the corresponding data in the Media View and Form View panels.

3.2.1 Built-in media viewers and analysis tools

Panorama comes equipped with viewers for a variety of media types, such as charts and graphs, tables, images, and video. It also has an embedded Web browser (defined by the operating system), and consequently can support extensions, such as Flash, JavaScript, etc. A Flash video of some of Panorama's viewing tools is available at: http://marg.nlm.nih.gov/IP/panorama_demo.mov.

INSERT FIGURE 3 HERE

Tables and Charts Viewers: The line graph charting capability of the Charts viewer is shown in Figure 2. Figure 3 shows a Table viewer that supports row/column selection, sorting, subset, and filtering functions. The data may be exported in comma separated values (CSV) format permitting further analysis and research in sophisticated statistical applications such as R⁵ or SAS⁶. Note, however, that Panorama includes built-in capability for computing lightweight measures and descriptive statistics with and without stratification. It allows a user to perform ROC, Logistic and linear regression analyses. Results may be expressed numerically or as chart overlays, as applicable.

INSERT FIGURE 4 HERE

Video Viewer: As shown in Figure 4, Panorama also supports video with or without chapter markers provided by authors. Video is played using the freely available Quicktime[®] for Java⁷ libraries from Apple Inc. The Form view in this context provides the capability to select video playback start and stop positions that may be different from author-provided chapter markers, and also the capability to save video-frame snapshots and to export portions of the video clip.

INSERT FIGURE 5 HERE

DICOM Viewer: In light of our special interest in the life sciences, Panorama includes the capability to view and manipulate multi-slice clinical images in the DICOM format and generates 3D renderings for this data. As shown in Figure 5, the DICOM viewer can be expanded to fill the application window. This permits each slice to be viewed in a format similar to most DICOM image viewers in routine use.

3.2.2 Third party extensions

While the current functionalities of Forge and Panorama are considerable, we encourage and anticipate third parties to incorporate additional functionalities. Building on the basic capabilities of Eclipse RCP,

⁵ <http://www.r-project.org>

⁶ <http://www.sas.com/>

⁷ <http://developer.apple.com/quicktime/qtjava>

third parties may develop plug-ins⁸ following the standards written by the Eclipse Foundation. Access method definitions and instructions on developing plug-ins will assist third-party developers in designing viewers or additional capabilities for the tools. The common codebase for Forge and Panorama will enable seamless integration of such modules into these applications, while providing a consistent look-and-feel. Use of XML definitions for the media types, actions, and location permits easier specification of these values particular to any plug-in viewer and would be specified by its developer.

4. Next steps

While our focus has mainly been in developing tools for interactive publications, we are planning to meet challenges that need to be addressed if such documents are to be widely produced and disseminated. The following outlines three such challenges.

Efficient downloading: The size of an interactive publication can be very large due to data-intensive media objects such as video, DICOM images or tables with tens of thousands of rows. Since the very concept of an IP is rooted in encouraging the inclusion of as much research data as necessary to promote understanding, one can conceive of documents ranging in size from tens to hundreds of megabytes. Such large sizes can pose a serious barrier to widespread dissemination and use.

Our solution has been to design an intelligent Download Manager utility to download the relatively compact textual portion of the document first (allowing the reader to start perusing the text), while the data-intensive media objects arrive in the background. Importantly, the reader can control the order in which these media are downloaded. For example, if a reader wishes to forego the introduction or methods section typically found in research articles in favor of a dynamic table further down in the article, the Download Manager will deliver the table ahead of other media.

Evaluation: In order to understand the value of our interactive publication model, the shortcomings of its current form and practical directions for the future, a comprehensive evaluation needs to be conducted in collaboration with significant players in this enterprise. These would include publishers, authors and readers.

We anticipate providing our tools and procedures as they are developed to publishers, their contributing authors and designated peer-reviewers. In the course of authoring the publications, our collaborators will evaluate the tools and procedures on the following grounds: ease of creation, whether specialized skills are required and the skill level necessary. Reading the publications will be evaluated on the following: ease of use, speed of moving from one media type to another in the publication, speed in invoking the various media objects, and whether viewing and manipulating the objects require additional client tools. The results of this evaluation will inform the next stages of tool development. The validity of our hypothesis of the interactive publication in improving comprehension, learning, and degree of assessment of the research reported in the publication will need to be evaluated subsequently.

Preservation: The long-term preservation of all significant material in biomedicine is a mandated task for the National Library of Medicine (NLM), irrespective of the media or formats they come in. This will be the case for interactive publications as well. At present we are engaged in the design of systems for archiving historic documents [6]. This activity will be expanded to investigate factors relevant to the preservation of interactive publications, e.g., design of suitable archival systems, extraction of descriptive and technical metadata, and bulk migration of file formats.

5. Summary

⁸ <http://www.eclipse.org/community/osplugins.php>

This paper describes the Interactive Publication as envisioned by researchers at the National Library of Medicine. In contrast to a static document, an IP is a self-contained document with links to multimedia data: high resolution images, video, Web URLs and sufficiently detailed underlying tabular data relevant to the article. The development and characteristics of our Forge authoring tool and Panorama viewer are described. These platform-independent tools support common formats for most relevant media types and are designed to be extendable through third-party extensions. They will be open-sourced and made freely available. This article also discusses future tool development and plans for formal evaluation through usability studies and assessment of improved learning that an interactive publication would offer over a static document.

Acknowledgements

This research is supported by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM), and Lister Hill National Center for Biomedical Communications (LHNCBC).

References

1. G. R. Thoma, G. Ford, M. Chung, K. Vasudevan, S. Antani, Interactive Publications: Creation and Usage, Proc. SPIE Electronic Imaging, 6076(2006) 607603-(1-8).
2. D. Shotton, K. Portwin, G. Klyne, A. Miles, Adventures in Semantic Publishing: Exemplar Semantic Enhancements of a Research Article, PLoS Comput Biol., 5(4) (2009), doi: 10.1371/journal.pcbi.1000361, <http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.1000361>
3. M. Jern, M. Ranlöf, S. Palmberg, A. Nilsson, Coordinated Views in Dynamic Interactive Documents, Coordinated and Multiple Views In Exploratory Visualization, (2003) 95, doi: 10.1109/CMV.2003.1215007, <http://www.computer.org/portal/web/csdl/doi/10.1109/CMV.2003.1215007>
4. M. Jern, A. Nilsson, S. Palmberg, M. Ranlöf, "SmartDoc" - 3D Dynamic Interactive Documents, White Paper. ITN, Linköping University, Sweden. Available at (verified March 25, 2010): http://servus.itn.liu.se/smartdoc/project_results/documents/SmartDocPaper2003.pdf
5. Journal of Visualized Experiments (JoVE): <http://www.jove.com>
6. D. Misra, S. Chen, G. R. Thoma, A system for automated extraction of metadata from scanned documents using layout recognition and string pattern search models, Proc. IS&T Archiving 2009, (2009) 107-112.

```

<!-- "media" is the top level of the configuration. The "type" attribute
specifies what view (chart, table, video, etc.) to open, and the optional
"analysis" attribute specifies a particular analysis to run. -->
- <media type="chart" analysis="roc">
  <!-- Define list of parameters for the roc. These correspond to items
in the form view when ROC is selected. -->
  <parameter name="Pattern" value="Polyp Diameter (cm)" />
  <parameter name="Score" value="Colon Cancer" />
  <!-- Specify the dataset to open. The "value" attribute of the above
parameters are column names in this file. The filename is
assumed to be relative to the current working directory
but full url's may be used -->
  <dataset name="colon.csv" />
</media>

```

Figure 1. Example XML file created by Forge and read by Panorama for visualization.

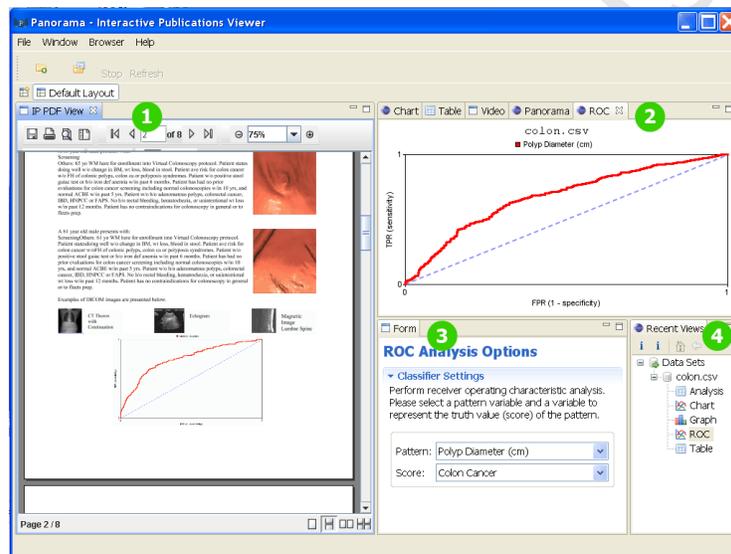


Figure 2. Panorama screenshot with labeled panels and depicting its Chart viewer.

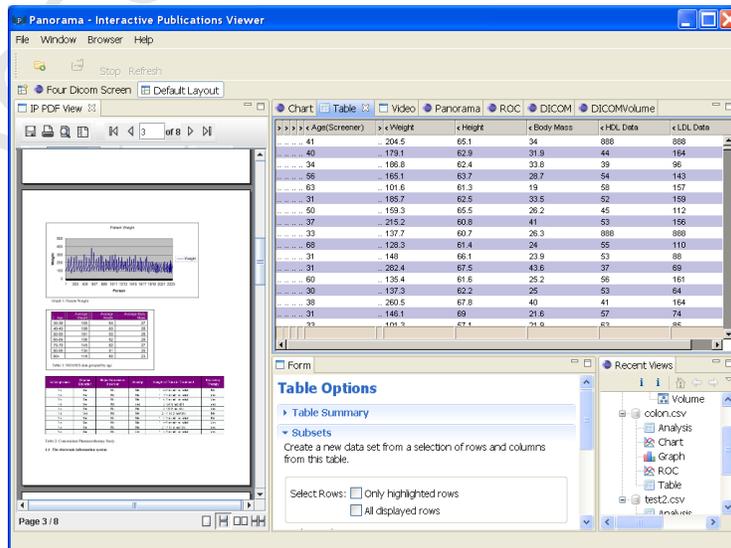


Figure 3. Panorama screenshot depicting its Table viewer.

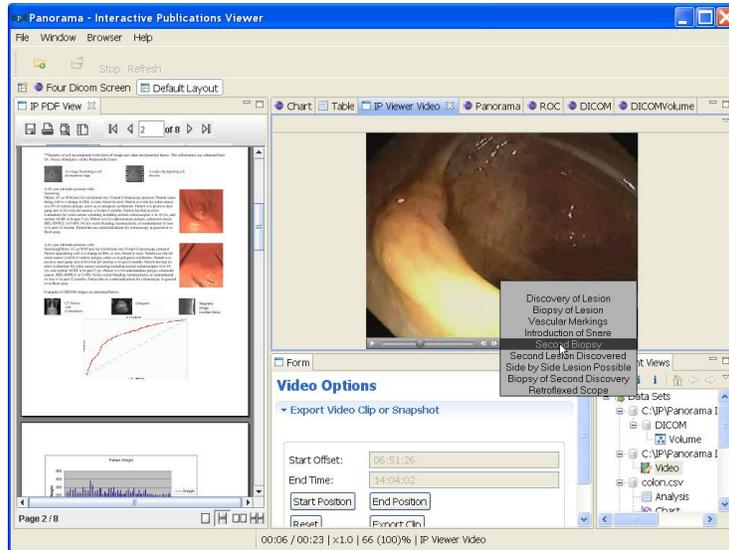


Figure 4. Panorama screenshot depicting its Video viewer.

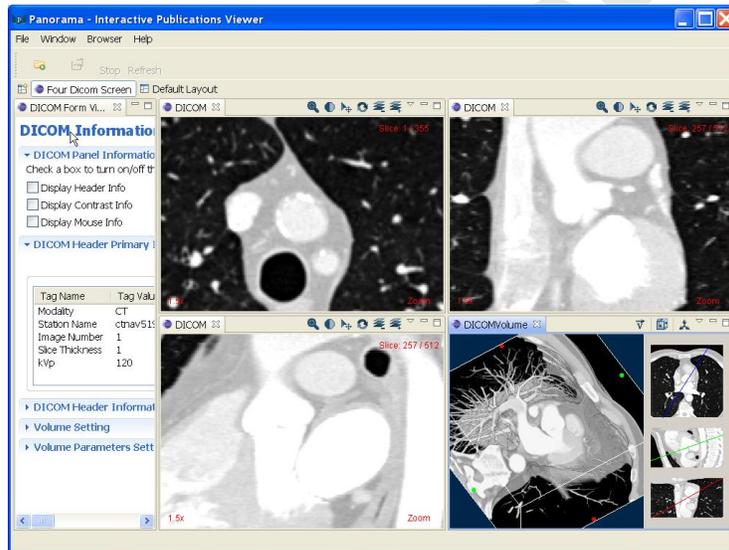


Figure 5. Panorama screenshot depicting the DICOM image viewer and 3D volume rendering tool (lower-right quadrant).