



**THE LISTER HILL NATIONAL CENTER
FOR BIOMEDICAL COMMUNICATIONS**

A research division of the U.S. National Library of Medicine

**TECHNICAL REPORT
LHNCBC-TR-2005-005**

Interactive Publications Research
A Report to the Board of Scientific Counselors

September 2005

George R. Thoma, Ph.D., Principal Investigator

Sameer Antani, Ph.D.

Glenn Ford

Michael Chung

Kirankumar Vasudevan, M.S.

U.S. National Library of Medicine, LHNCBC
8600 Rockville Pike, Building 38A
Bethesda, MD 20894



Table of Contents

1	BACKGROUND	1
2	PROJECT OBJECTIVES.....	2
3	PROJECT SIGNIFICANCE	4
4	METHODS AND PROCEDURES	4
4.1	OVERVIEW	4
4.2	STANDARDS AND MODELS	5
4.2.1	<i>Model description and drawbacks</i>	6
4.2.2	<i>Desirable attributes of an interactive publication</i>	8
4.3	ITAG: A TOOL FOR INTERACTIVE TABLES AND GRAPHS.....	10
4.4	PROTOTYPE PUBLICATIONS USING CURRENT DEVELOPMENTAL SOFTWARE.....	16
5	NEXT STEPS	20
5.1	ENHANCING ITAG	20
5.2	TOOL DEVELOPMENT.....	25
5.3	EVALUATION PLAN.....	26
5.4	PRESERVATION	26
6	SUMMARY	26
7	QUESTIONS FOR THE BOARD	27
	REFERENCES	28
	APPENDIX: AUTHORING PROCEDURES.....	29
	GLOSSARY	35
	CURRICULUM VITAE	37

Interactive Publications Research

1 Background

In almost every discipline, information in formats other than text keeps increasing in numbers and types. In biomedicine, such non-textual information (multimedia) comes in various forms: as digital x-rays, CT, MRI, ultrasound, ECG, video and audio. Far from being peripheral to text, these media types often provide the key intelligence for a reader to fully understand the essentials of the publication, though publishers today consider the textual part the “article,” and generally relegate the media objects to the status of “supplemental material.” Missing in this limiting characterization is the role these media might play to not only inform the reader of essential facts, but their potential to help the reader re-evaluate the conclusions in the publication itself. Re-evaluating an author’s conclusions may, for instance, be a necessary and welcome step in a comprehensive peer-review.

In a remarkably prescient narrative describing the multimedia digital libraries of tomorrow, Lindberg and Humphreys [1] envision a scenario featuring “rich interconnections among genetics research data, aggregated clinical and public health data, published literature, and high-quality health information in many languages.” Central to that vision is the existence of published literature, readily accessible online, that not only contains rich multimedia but also allows the reader to manipulate, use and analyze the information contained therein.

As envisioned here, an “interactive publication” is a composite digital document that has similarities to multimedia documents that have been in existence for a decade or more, but possesses specific differentiating characteristics. The latter in common usage refer to online entities that consist of text that links to images and video clips. These media, more often than not, reside in databases apart from the text, partly because their inclusion (as “supplemental material”) along with the text, all in a “container,” increases the document size to unmanageable levels for network distribution.

As commonly defined, such a “traditional” multimedia document has hyperlink relationships with other documents and media objects. It is, in effect, a *personalized multimedia presentation* system that enables a user to access media objects located in *external databases*; these objects are then organized and presented for viewing. In these traditional multimedia documents, the reader may: (a) link to other documents, and still and moving images; (b) use VCR-type navigation controls to start, stop or pause a video frame sequence. A recent example of such a document linked to two video sequences is an article by Clement and Marés in ELCVIA journal [2]. This paper describes a tracking algorithm for automatic instrument localization in robot-assisted laparoscopic surgery.

An interactive publication of the sort we envision implies a step beyond passively viewing multimedia objects and interacting with them in the traditional ways listed above. It should allow *reuse of the content*, so that the user can convert objects from one

form to another, e.g., from a graph to a table, for ease of analyzing data (invoking analytical tools such as SAS), or from a table to a graph to better visualize the results. The user should also be able to identify and extract data from the displayed object, e.g., an anterior osteophyte from the image of a cervical spine x-ray, and magnifying it for closer examination. In addition, the user should have the option of zoning parts of a document and have it summarized (by abstraction tools) or translated by machine translation tools.

While an interactive publication has many media objects as does the traditional multimedia document, it is a self-contained document, either as a single file with media files embedded within it (**embedded type**), or as a “folder” containing tightly linked media files (**folder type**). When using such a document, the reader should be able to: (a) view any of these objects on the screen; (b) hyperlink 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*.

It is the last objective that conclusively differentiates a traditional multimedia document from an interactive publication. Such a publication (none appears to exist today) would have the characteristics of a ‘document’ in the sense of a completed work of an author presenting hypotheses, findings and conclusions, but also giving a reader the facility to check the underlying data and possibly derive alternative conclusions leading, as stated earlier, to more in-depth peer reviews. In addition to its “research tool” aspect, from a library’s preservation point of view, a self-contained document is clearly preferable to avoid dependence on the content providers (publishers) to maintain their databases of supplemental multimedia items in perpetuity.

2 Project Objectives

Our long term objective is to develop open source and platform-independent tools and procedures for authors or publishers to create and disseminate interactive publications (IP). These tools and procedures are to be evaluated in terms of ease of use, adherence to established document standards (and if existing standards are inadequate, to contribute to establishing suitable ones.) Further, we intend to evaluate the role of IPs in enhancing comprehension and learning, and helping readers better assess the research reported in the publication. Our short term objective is to demonstrate the creation and use of IPs built on existing developmental platforms, viz., Adobe Acrobat, Microsoft Word, Flash and HTML.

Research objectives are in several areas:

Standards and platforms

- Evaluate ISO and W3C standards for formats and descriptive languages to determine their suitability for interactive publications. Examples are SMIL, HyTime and RTF. Identify gaps in the standards posing a barrier to the full realization of such publications.

- Assess the suitability of standards for publications in biomedicine, and identify special requirements for biomedical documents, e.g., the inclusion of clinical images following the DICOM standard.
- Investigate existing developmental software environments (“platforms”) for their suitability in creating and using interactive publications, viz., Adobe Acrobat, Microsoft Word, Flash and HTML.

Prototype development

- Create prototype interactive publications following existing standards in available developmental environments, to the extent possible, and including a range of media types. The range of media types should cover all possible ones in biomedicine. Prototype publications could have features and content such as:
 - Text with next/previous page and other navigation capability
 - 3D animations of anatomic structures
 - Graphs, charts and tabular data
 - Cell development images
 - Clinical images (in DICOM format)
- Expanding on the research tool aspect of the interactive publication, include the capability to enable the user to execute code to manipulate data in the document, such code either imbedded in the document or residing at the author’s Website.

Tool development

- Develop an authoring tool for scientific writers or publishers to create interactive publications without specialized programming skills. Such a tool should be platform-independent and provided freely.
- Develop a tool for the creation of interactive tables and graphs.
- Investigate availability of reading tools, and develop one if existing candidates are unsatisfactory.
- Conduct alpha testing on all tools to debug, verify accuracy and ease of operation.
- Conduct beta testing of tools with selected user/developer community.

Evaluation

- Establish performance criteria for interactive publications: comprehension, learning, evaluation of research reported in publication.
- Recruit biomedical publisher(s) to create and disseminate interactive publications, and tools for display and use of such publications.
- Evaluate the ability of authors and publishers in using authoring and reading tools.
- Evaluate reader’s use of interactive publication.

Preservation

- 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.

3 Project Significance

With an increasing awareness of the value of multimedia in the biomedical literature, online publications provide links from text to such media usually residing in publishers' databases. While a reader may go from text to viewing these media, there is little interaction with, and analysis of, the data. Besides this limited capability for the reader to fully use the publication, a major drawback for NLM and other major libraries is the difficulty in fulfilling our preservation mandate: it is difficult to archive the non-textual material scattered in the databases of publishers and authors. An integrated document containing all the material in one package ameliorates this situation.

4 Methods and Procedures

4.1 Overview

The subsequent sections of this report outline ongoing research in techniques to author and use interactive publications, and the creation of prototypes on several platforms (MS Word, Flash, Acrobat and HTML). In these prototypes, we include a range of media types encountered in the biomedical literature, such as: paginated text; 3D animations of anatomic structures; graphs, charts and tabular data; cell development images (video sequences); and clinical images such as CT, MRI and ultrasound in the DICOM format.

Our investigations are proceeding in several directions. First is an evaluation of ISO and W3C standards for formats and descriptive languages suitable for interactive publications, e.g., SMIL, HyTime and RTF. The objective here is to identify gaps in the standards posing a barrier to the full realization of such publications, in particular, for biomedical documents in which the inclusion of clinical images would be important.

Secondly, in the course of creating prototype interactive publications on existing platforms, and incorporating a range of media types, we are developing step by step procedures for an author or publisher to follow.

Thirdly, expanding on the research tool aspect of the interactive publication, we are developing the capability to enable the user to execute code to manipulate data in the document, such as analyzing the contents of large tables, or converting tables to graphs and vice versa.

This report presents the status of the project to date including: a tool to convert static tables or graphs into interactive entities, authoring procedures on the four platforms attempted, and advantages and drawbacks of each of these platforms. It also outlines

future work in evaluation, tool development and in meeting the challenge of network distribution for these large files.

4.2 Standards and models

Our first step, to seek existing standards and document models that may be suitable, is discussed here. For more than a decade, there have been several attempts to formally include multimedia data and increased interactivity into documents. Examples are two standards published by the International Standards Organization (ISO) and a currently active recommendation by the World Wide Web Consortium (W3C). These efforts do not, however, completely address the desired characteristics of an interactive publication intended for use by researchers and scientists authoring and reading scientific articles. Among the desirable attributes for interactive publications discussed in Section 4.2.2, a primary one is document integrity, i.e., all components of the document such as tables, images, video, audio, etc. should be suitably packaged with the article text and not be hosted on different servers on the Internet as in a typical Web publication today. To meet this requirement, these components could either be embedded into a single document file, or be contained in a folder with the text document tightly linked to these data. In order to achieve document integrity and satisfy other attributes it is important to design a framework and develop a set of supporting tools. Toward this development we have studied the proposed standards and recommendations for feasibility and adaptability to our work. Salient features in the document models and in language implementations were explored, and both useful features and shortcomings are described below. We also explore other present-day popular formats such as the Rich Text Format (RTF) designed by Microsoft, Adobe's Portable Document Format (PDF), Microsoft's Word, and HTML. Note that HTML is used here to include the set of Web-based technologies used to publish articles online today. Their properties and the extent to which they are useful are discussed in Section 4.4.

The document models we discuss here fall into three groups, viz., *open-open source*, *open-proprietary*, and *closed-proprietary*. An open-open source model is one which reveals complete details about the format in a freely accessible manner, and welcomes user input to improve the standard while also allowing users to create their own extensions to meet specific needs, e.g., W3C's SMIL Recommendation. An open-proprietary format is one which is freely released by its designers but is disallowed from modifications, e.g., Adobe PDF and Microsoft RTF. Closed-proprietary formats are, in contrast, neither openly described nor modifiable, e.g., Microsoft Word.

In the literature several terms are used with overlapping definitions to describe document design such as *standard*, *model*, *language*, and *format*. The distinction among these terms is not always clear and many are often used interchangeably; in general, a *standard* or a *model* stand for a specification of the features that may be available in a document authored in a particular *format* using (in some instances) the *language* specified by the model. In certain cases, a model can be the conceptual design, the format the physical structure, and the language a means to specify the structure. *Platform* is also used to refer to a software environment in which a document is developed. We try to use terms

consistently wherever possible, but, defer to standard terminology in certain situations as appropriate.

4.2.1 Model description and drawbacks

Open Document Architecture

The Open Document Architecture (or sometimes referred to as *Office* Document Architecture or just ODA) was designed by the ITU-T to be a standard document file format to replace all proprietary document file formats. ODA is detailed in the standards documents T.411-T.424, which is equivalent to ISO 8613, finally published in 1999. ODA defines a compound document format that can contain raw text, raster images and vector graphics. The document format possesses both logical and layout structures. Logically the text can be partitioned into chapters, footnotes and other sub-elements akin to HTML, and the layout fills a function similar to Cascading Style Sheets (CSS) in Web documents. Several companies including IBM, Siemens AG, ICL, and Olivetti attempted to develop tools for its use but the effort was a failure. However, concepts in its design have influenced present-day document formats such as HTML.

ODA addresses the encoding and inclusion of multimedia elements in a document, but does not address their use or how they would be packaged. Furthermore, no rules for interactivity were identified. While we cannot rule out inclusion of concepts from its design, it may be prudent to adopt applicable features from descendent technologies of the ODA for use in an interactive publication document model.

HyTime: Hypermedia/Time-based Structuring Language

HyTime is a markup language that is an "application" of SGML [3]. HyTime defines a set of hypertext-oriented element types that, in effect, supplement SGML and allow SGML document authors to build hypertext and multimedia presentations in a standardized way. It is an international standard published by the ISO and IEC (ISO/IEC 10744). The first edition was published in 1992, and the second edition in 1997. Some of the concepts formalized in HyTime were later incorporated into HTML and XML. Like ODA, the HyTime model created much research interest, but no references to it have been found later than 1999. It is unclear which features, if any, would be useful from HyTime directly. Again, it may be prudent to adopt these from technologies that are descendents of HyTime. Authoring and reading tools for HyTime were not found.

SMIL: Synchronized Multimedia Integration Language

SMIL™, (pronounced "smile"), enables simple authoring of interactive audiovisual presentations. SMIL is typically used for "rich media"/multimedia presentations which integrate streaming audio and video with images, text or any other media type. SMIL is an HTML-like language, and many SMIL presentations are written using a simple text-editor [4].

SMIL is a W3C Recommendation that enables authors to specify and control the precise time a sentence is spoken and make it coincide with the display of a given image. The Synchronized Multimedia (SYMM) Working Group completed SMIL 1.0 and SMIL 2.0 and is currently building SMIL 2.1 and a future version. Version 2.1 extends SMIL 2.0 functionalities into new or revised modules, examples of which are given below, and defines new profiles for the mobile communications industry. Additionally for Version 2.1, the Timed Text Working Group is designing an XML-based format used for the representation of streamed text synchronized with other timed media. Typical applications are real time subtitling of foreign-language movies, captioning for people lacking audio devices or having hearing impairments, karaoke, scrolling news items, and teleprompter applications. Several popular players including RealPlayer support SMIL.

A SMIL document is similar in structure to an HTML document in that it is typically divided between a <head> section and a <body> section. The <head> section contains layout and metadata information. The <body> section contains the timing information, and generally consists of combinations of two main tags: parallel ("`<par>`") and sequential ("`<seq>`"). It refers to media objects by URLs, allowing them to be shared between presentations and stored on different servers for load balancing. The language can also associate different media objects with different bandwidths. The language is defined as several modules which include animation controls, content control, layout, linking, media object, structure, timing, and transition effects among others.

Interactive publications should in essence emulate traditional documents, and therefore require a paginated look-and-feel, *enriched* by content interactivity. So while such a document could be created using SMIL, it would be quite cumbersome for the author to do so. Another problem with SMIL is that its structure consists of links and like a Web page it expects different servers to host the media elements, a characteristic that is unsuitable for long term archiving of publications in this format and not a desired feature in interactive publications. In addition, the readability and use of these documents are heavily dependent on network connectivity, server uptime, software versions, access rights, etc., making this a very fragile format. However, SMIL media could be elements in an interactive publications document. In addition, design concepts in SMIL offer several features that could be adopted for interactive publications.

There are authoring and reading tools for "traditional" multimedia documents as freeware listed in sourceforge.net. Examples of freeware for SMIL are: [ambulant](#) and [X-Smiles](#) (for reading SMIL documents); [SMILgen](#) for authoring.

RTF: Rich Text Format

RTF has been designed and continually developed by Microsoft since 1987 for cross-platform document interchange. Most word processors are able to read and write RTF documents. In addition, RTF is a markup language and is human-readable, i.e., it is written in plain text with control codes, e.g., `{\rtf\ansi\deff0{\fonttbl{\f0\froman Tms Rmn};}`, indicating particular display, layout or other characteristics. Multimedia

elements such as images and video are supported through Microsoft's object linking and embedding (OLE) technology also indicated by control codes. Using this any object can be included with RTF as long as its supporting application is available and also supports OLE. There are authoring and reading tools for "traditional" multimedia documents from Microsoft.

In several ways an RTF document has characteristics similar to those in a Word document, though unlike Word, it is an *open-proprietary* format. RTF is often pitched as a "future proof" format, but is closely tied to Microsoft technologies for multimedia data. To use its features, it also requires enriched editors, one of which is Microsoft Word. In the future, features from RTF could be considered for a document model for interactive publications.

4.2.2 Desirable attributes of an interactive publication

This section highlights some desirable attributes of interactive publications. Note that these attributes are a subset of a larger collection of desired attributes, 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. This implies 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.
 - Additional page forward and back links on document or in document reader would be beneficial.
- **Intra-page navigation:**
 - Traditional use of keyboard keys (cursor up/down/left/right) and mouse.
 - Additional use of control keys (as shortcuts) would be beneficial.
- **Image browsing**
 - Images should be natively supported, especially all relevant formats such as JPEG, TIFF, DICOM, GIF, JP2, BMP, PNG, etc.
 - Should be easy to encode some degree of interaction with these into the document model.
- **Navigating to an embedded / linked media object**
 - Since media objects include audio, video and other objects it is necessary to support mouse-click (or keyboard-based) activation.
 - Embedded or linked media objects should be able to invoke appropriate viewers or players.

- **Native support for interactivity**
 - The document model should include 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 then be made accessible to the reader to provide a reader-centric view to interactive documents.
 - Data in specialized and proprietary formats should be viewable using appropriate supporting application software.
- **Execute code**
 - The document model should be able to execute limited code in a secure fashion. For example, the code could be restricted to use only data within the document or document folder with actions limited to the document memory space.
- **Transmission**
 - The document model should support controlled order of transmission for “heavy weight” 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. In the former case the multimedia components are a part of the document software file. In a folder structure the multimedia components are separate files that are linked to from within the text document.
- **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.
 - In both embedded and folder formats, the document model should support document integrity by closely linking the text document to the multimedia components.

The question is whether existing open standards support the above requirements. A step in answering this was taken by researchers at the University of Ulm [5] who compared SMIL and HyTime, among other multimedia document models. They conclude that these have at least some of the specifications needed for content reuse, though not adaptability. In an attempt to rectify these shortcomings, they propose a new standard (ZyX) though this does not seem to have been adopted elsewhere [6]. They have used ZyX to create a training package for cardiology (Cardio-Op) that is not a self-contained document, but rather relies on a database repository of multimedia content. Papers on Cardio-Op are listed in PubMed [7,8].

We conclude that not all of the desirable characteristics listed are found in present-day file formats or published standards or recommendations, but many are. Current language structures also do not permit an instantiation of such a document. The lack of some of these desirable features in current document models provides impetus for our next steps

discussed in Section 5. In the next section we present the design of a tool necessary for incorporating dynamic tables in an interactive publication, following which we discuss the creation of prototype publications using current document models, and the extent to which these meet our objectives for interactive publications.

4.3 ITAG: A tool for interactive tables and graphs

Definition and capabilities

To fully realize the promise of an interactive publication, it must allow a reader to interact with, and analyze, data in the document. Most biomedical journal articles present data as tables or graphs which usually show only a part of the data analyzed in the study/experiment on which the paper is based. Often this is due to size limitations (e.g., number of allowable rows) in certain proprietary though commonly used spreadsheet programs such as MS Excel. Furthermore, the presentation is static, i.e., the reader does not, as a rule, have a ready way to manipulate or analyze the data. In an interactive document, the reader should be able to view the same data as a table or a graph (convert from one to the other), sort columns in ascending or descending order, create new tables from subsets of the data in an existing table, calculate statistical/mathematical quantities from the data, zoom into the graphical data for more detail, and save the results of analysis in formats compatible with Excel or other common analysis tools. The challenge is to enable a reader to do any of these functions with a mouse-click or two. To enable a reader to interact with this information, we need to make the tabular or graphical data dynamic and then to incorporate this 'active' object into the IP.

In searching for tools toward this end, we found dynamic tables offered by Knovel (<http://knovel.com>), a company that combines reference books and databases from more than 30 Sci-Tech publishers and professional societies including McGraw Hill, John Wiley, Elsevier, Springer, American Institute of Chemical Engineers and American Society of Mechanical Engineers. The content covers subject areas in Chemistry, Life Sciences, Materials and Engineering. The tables have some of the interactive features listed above. However, the software that makes the tables interactive is proprietary, and moreover is tailored specifically to handbook information and intended for Internet delivery from databases. It would not therefore meet our goal to have a non-proprietary, open source package that may be distributed to biomedical publishers or authors for use in research articles.

We also considered the use of Microsoft Excel as a natural choice for providing the interactive tables and graphs feature. The idea, however, was found problematic upon further analysis. Excel is not open-source software and its ubiquity cannot be guaranteed. Also, it is closely tied to the Microsoft Office suite of applications with limited platform independence, i.e., it can only operate in the PC or in the Macintosh environments. Excel does not support the ability to relate table data semantics to the charting and analytical software nor does it permit the use of an extension that could. Finally, Excel limits the number of cells in a worksheet to a grid of 65,536 rows by 256 columns which can be insufficient for biomedical applications (for example, CDC's NHANES II data consists

of over 2,000 columns). This number is further limited by available system memory. These factors underscored the need for open-source extensible software that provides interactivity with the tabular data. The development of a promising application that meets several of these requirements, its design, function, and proposed modifications as a tool for interactive tables and graphs in IP is discussed below.

Design approach

To address this problem, we developed a Java tool, *Interactive Tables and Graphs* or ITAG, which we based on an open source software package called Starlink TOPCAT (<http://www.star.bris.ac.uk/~mbt/topcat/>), funded by the Particle Physics and Astronomy Research Council, UK and developed for the British astronomy community. Though TOPCAT has much of the desired functionality, we customized it by eliminating functions specific to astronomy, and extended it to generate line graphs and to include command line arguments to incorporate tables (or graphs) into an IP. This last modification allows an author to make a dynamic table or graph part of the document being created, and a reader to invoke them for viewing and analysis.

The ITAG software has the capability of generating and displaying interactive tables and graphs from raw data in formats such as *CSV* (comma separated values). The software can generate an interactive table from raw data and then display its plot. It can also create an interactive graph from raw data and then generate a table from it. Figure 4.1 shows this functionality in a high level block diagram; here the author saves data from an external system such as MS Excel as a *.csv* file. Figure 4.2 shows a snapshot of the main control window of ITAG and Table 4.1 lists brief descriptions of icons on the ITAG main window.

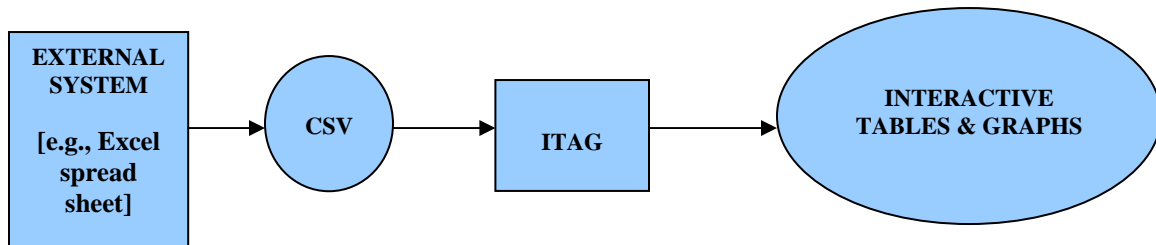


Figure 4.1 – The dataflow from an external system to visual display.

Incorporation of interactive tables and graphs in an IP

The steps to make a table or graph dynamic and to include it in an IP are specific to both document format and operating system. We outline the steps needed to incorporate a table or graph into a PDF document, and the creation of a Table Invoker module. Since this batch/script file is necessary for the reader to interact with the table, it must be included in the document by the author.

The author starts with raw data in the CSV format. The next step is to select a document format, say Microsoft Word, and start writing the document using its text editor. The raw data can be represented in the document using a static table, which can be created using a

custom menu for drawing tables. An author wishing to show a pictorial representation of the raw data may use ITAG to create a graph and then use the ITAG menu to save it as a .gif file, which then may be inserted at the appropriate place in the document. Assuming that the final document is to be in PDF format, the Microsoft Word document should then be converted to PDF format using Acrobat. The static tables and graphs in the document should then be linked to the ITAG software and the corresponding raw data (CSV) file in order to make them interactive. This is done in one shot by creating a Table Invoker for each interactive table and graph. The following are the steps to create a Table Invoker.

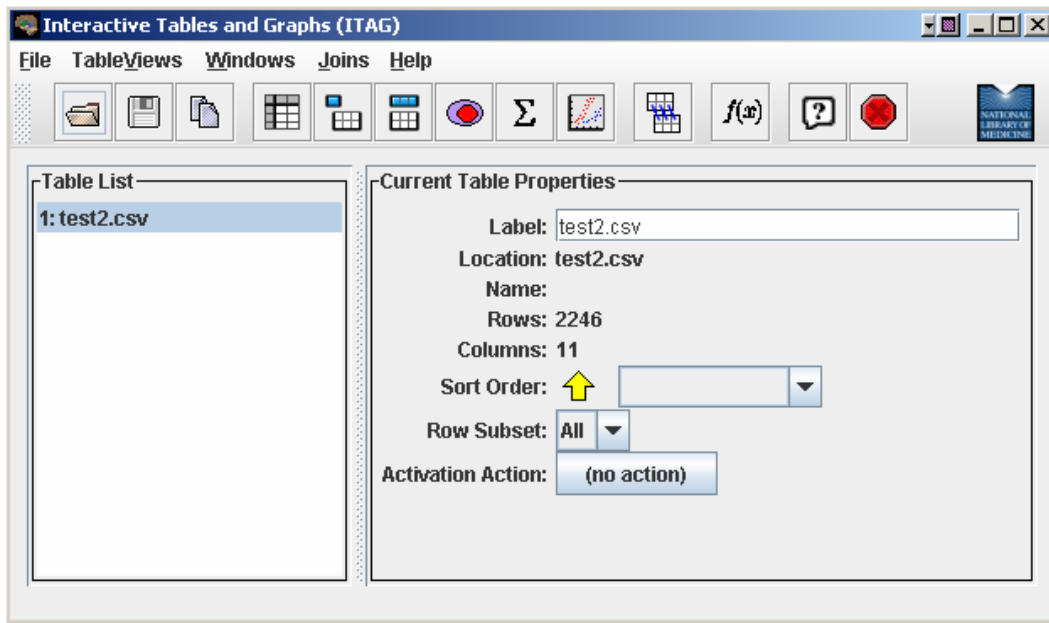


Figure 4.2 – Main control window in ITAG.














 Open File	 Show Column Metadata	 Copy Table
 Show Cell Data	 Show Table Metadata	 Save File
 Show Row Subsets	 Plot Column	 Join Tables
 Statistical Analysis	 Display Predefined Functions	 Help
 Close main window		

Table 4.1 – ITAG software icon definitions.

Steps:

1. Verify that the raw data is in proper CSV format as specified by ITAG.
2. Open a text editor, say Notepad, and type in the following text phrase:

```
start /min javaw -jar itag.jar -table -f csv <filename.csv>
```

`-table` is the option specified by ITAG for generating interactive tables. Similarly other options like `-graph`, `-x` and `-y` specified by ITAG could be used for generating interactive graphs. A combination of `-table` and `-graph` could be used for generating both. `<filename.csv>` should be replaced by the corresponding CSV file (raw data) supporting the table or graph.

3. Save the file in the format '`filename.bat`'. It is recommended that the filename reflects the corresponding table name.
4. Verify that the resulting saved file has a `.bat` extension and not a `.txt` extension. This file with `.bat` extension is called the Table Invoker.

Once the Table Invoker has been created, the author can use the menu provided by Adobe Acrobat to connect it to the corresponding static table or graph. The steps for this are briefly described and illustrated below.

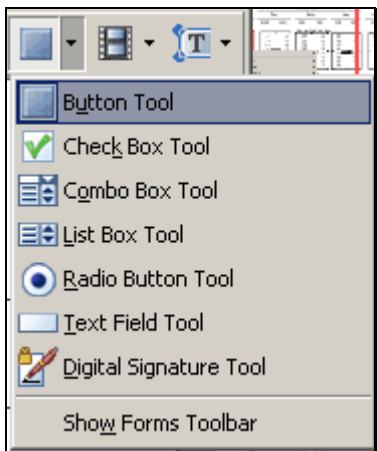


Figure 4.3 – Button for Tool Draw

Table A. Number of infant deaths and infant mother: United States, 2001–02 linked files

	Infant deaths	
	2002	2001
Total	7.0	6.8
White	5.8	5.7
Black.	13.8	13.3
American Indian	8.6	9.7
Asian or Pacific Islander	4.8	4.7
Hispanic.	5.6	5.4
Non-Hispanic white	5.8	5.7
Non-Hispanic black	13.9	13.5

A red rectangular box is drawn around the data cells for 'Total', 'White', 'Black', and 'American Indian' in the 2002 and 2001 columns. A small red box labeled 'Button1' is positioned over the value '8.6' in the 'American Indian' row, 2002 column.

Figure 4.4 – Rectangular Region Drawn

In Adobe Acrobat using the button tool (Figure 4.3) draw a rectangular button over the text or graphic (Figure 4.4) that is to serve as a link. Once this is drawn, a **Buttons Properties** dialog box will pop up. In the **Appearance tab** (Figure 4.5) the **Border Color** and **Fill Color** need to be changed to a white box w/ a red diagonal line (this is transparent). In the **Actions tab** (Figure 4.6), the author must ensure that the **Select Trigger** is set to **Mouse Up** and under **Select Action** choose **Open a file**, then click **Add**. When a **Select File to Open** dialog box pops up (Figure 4.7), the author must select the Table Invoker batch file as shown. A flowchart of the steps for creating dynamic tables and graphs within an interactive publication is given in Figure 4.8.

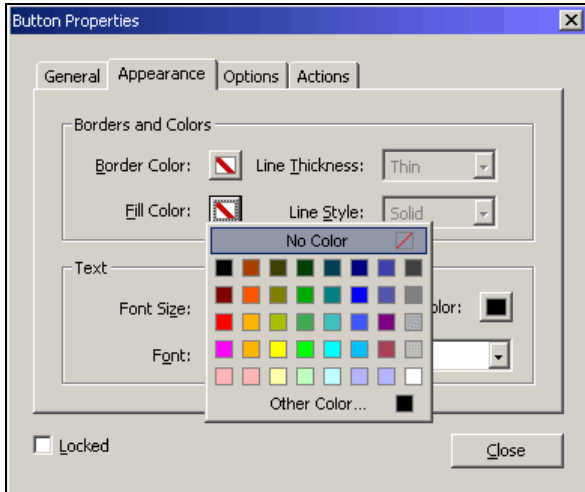


Figure 4.5 – Appearance Selection

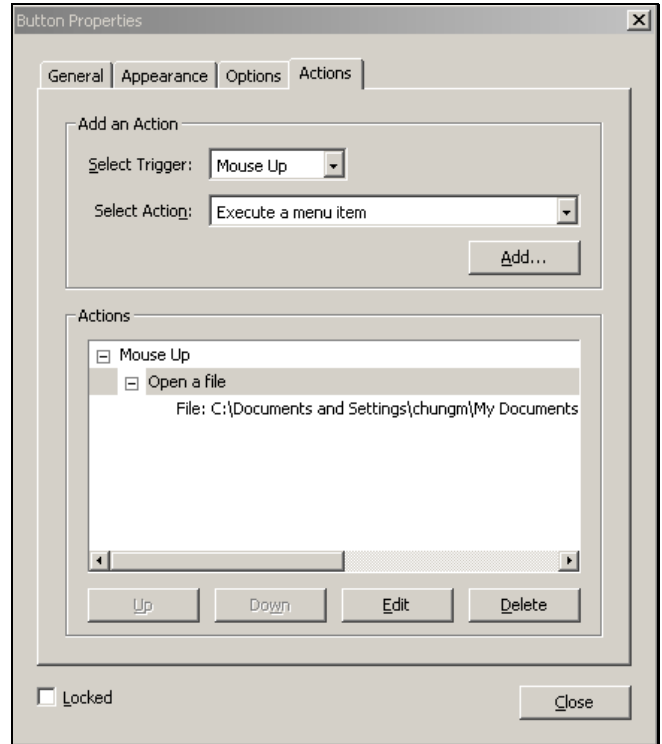


Figure 4.6 – Actions tab for newly drawn region.

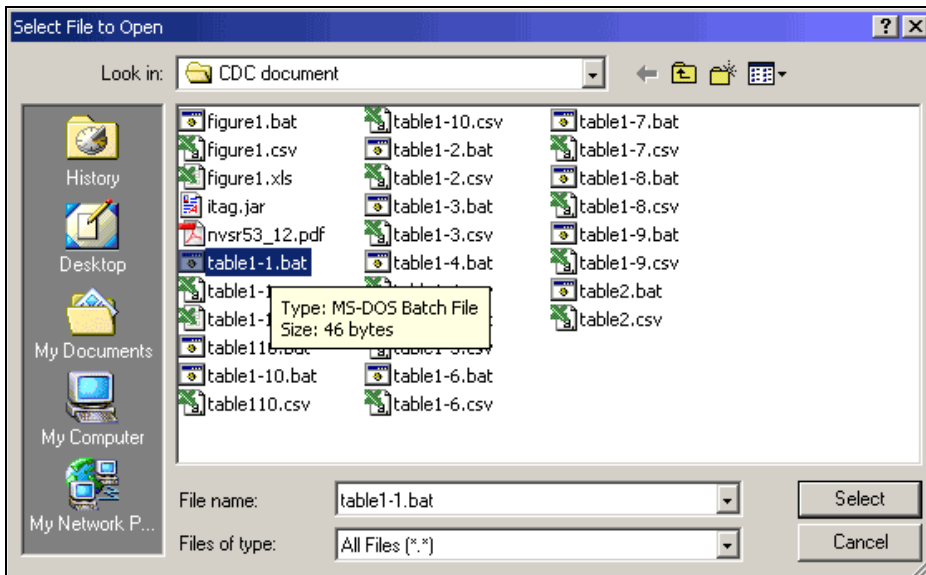


Figure 4.7 – File Select to select file to run for PDF region.

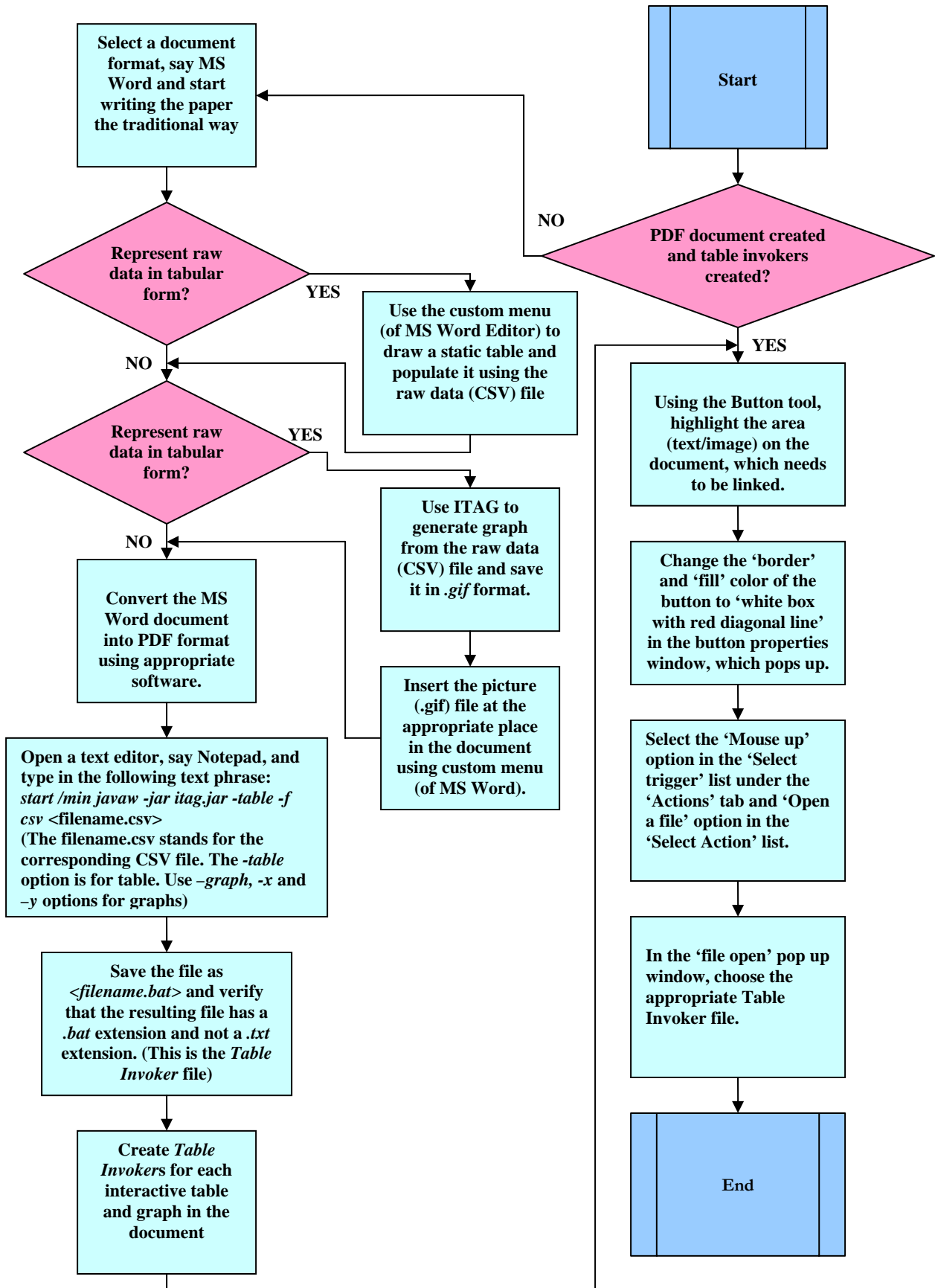


Figure 4.8 - A flowchart for creating dynamic tables and graphs in an interactive publication.

ITAG has been implemented in Java and is thus platform independent, i.e., it can work under Linux, Macintosh, Microsoft Windows, or any flavor of Unix. A batch file (.BAT) equivalent on non-Microsoft Windows systems is a shell script. While no changes are necessary in the script itself, for Unix, Linux, or Macintosh (OS X) machines, the batch file must be set to be an executable, i.e., the command `chmod ugo+x filename.bat` needs to be executed. Further development is necessary in ITAG to enable this in an automated way.

Limitations of ITAG

The current ITAG system is limited in certain ways. For instance, it is difficult to parse a CSV file for a complex table, i.e., one in which there is a hierarchy of main multiple-column headings and several subsidiary headings as shown in Figure 4.9. Such a complex table cannot be processed by the current version of ITAG. Other limitations include the need for authored startup script, inability to install ITAG as a client application (rather than requiring it to be packaged with each IP document as at present), lack of adequate charting tools, inability to create subsets using filters, need to save subsets as separate files, and lack of interoperability with other popular statistical packages. These shortcomings define our next steps in IP research and development.

Table A. Number of infant deaths and infant mortality rates, by age at death and by race and Hispanic origin of mother: United States, 2001–02 linked files

	Infant deaths		Neonatal deaths ¹						Postneonatal deaths ¹	
			Total		Early		Late			
	2002	2001	2002	2001	2002	2001	2002	2001	2002	2001
	Rate per 1,000 live births									
Total	7.0	6.8	4.7	4.5	3.7	3.6	0.9	0.9	2.3	2.3
White	5.8	5.7	3.9	3.8	3.1	3.0	0.8	0.8	1.9	1.9
Black	13.8	13.3	9.3	8.9	7.6	7.3	1.7	1.6	4.5	4.4
American Indian	8.6	9.7	4.6	4.2	3.2	3.1	1.4	1.1	4.0	5.4
Asian or Pacific Islander	4.8	4.7	3.4	3.1	2.7	2.5	0.7	0.6	1.4	1.6
Hispanic	5.6	5.4	3.8	3.6	3.0	2.9	0.8	0.8	1.8	1.8
Non-Hispanic white	5.8	5.7	3.9	3.8	3.0	3.0	0.8	0.8	1.9	1.9
Non-Hispanic black	13.9	13.5	9.3	9.0	7.6	7.4	1.8	1.6	4.6	4.5
	Number									
Total	27,970	27,523	18,791	18,275	15,020	14,622	3,771	3,653	9,179	9,248
White	18,395	18,087	12,352	12,078	9,804	9,571	2,548	2,506	6,044	6,009
Black	8,201	8,084	5,533	5,396	4,506	4,425	1,027	971	2,668	2,688
American Indian	366	404	195	176	137	129	58	47	171	228
Asian or Pacific Islander	1,006	947	710	624	573	496	138	128	296	323
Hispanic	4,927	4,630	3,360	3,105	2,673	2,439	687	666	1,567	1,526
Non-Hispanic white	13,327	13,300	8,853	8,817	7,002	6,979	1,851	1,839	4,474	4,483
Non-Hispanic black	8,031	7,938	5,399	5,293	4,386	4,337	1,014	956	2,632	2,645

¹Neonatal is less than 28 days of age, early neonatal is less than 7 days, late neonatal is 7–27 days, and postneonatal is 28 days through 11 months.

Figure 4.9. A complex table taken from Marian F. MacDorman *et al.*, “Explaining the 2001-02 Infant Mortality Increase: Data from the Linked Birth/Infant Death Data Set”, *National Vital Statistics Reports*, CDC, Volume 53, Number 12, January 24, 2005.

4.4 Prototype publications using current developmental software

To develop prototype interactive publications for demonstration purposes, we have considered four document models that exhibit many of the desired attributes listed in Section 4.2.2, viz., Adobe PDF, Microsoft Word, Macromedia Flash and HTML. PDF documents developed here are those supported in Adobe Acrobat 7.0.2. These document models were selected for the degree of interactivity they provide to multimedia elements, as well as the availability of editors or authoring tools.

The multimedia objects we assembled from contacts at the VA Hospital, Baltimore, MD and the Wadsworth Center (SUNY, Albany, NY) to include in our prototype publications

are: video sequences, including some of cell development, and clinical DICOM images. We took bitmapped images of spinal x-rays from our inhouse collection of NHANES II data. In addition, we created others: dynamic tables and graphs, and 3D animations of anatomic structures.

In the Appendix we list the steps an author would take to create interactive publications in the four formats attempted. As mentioned earlier, two types of publication were developed: the **embedded** (single file) as well as the **folder** type (linked files in a folder). Note that the 'embedded type' is actually a hybrid entity since not all multimedia objects (e.g., DICOM images) can be embedded in a single file, at the present state of development.

The prototypes developed were evaluated on the desired attributes listed in a previous section. Each was found to be inadequate on several counts. A summary of the attributes met and shortcomings are discussed below.

- **Appearance:**
 - **Desired:**
 - Paginated view of document similar to a traditional article.
 - **As Evaluated:**
 - Microsoft Word and Adobe PDF provide paginated views typical of traditional documents.
 - Macromedia Flash could provide a similar view, although with significant programming effort which could be cumbersome to most authors.
 - Web-based documents authored using markup languages such as HTML do not typically provide paginated views. Use of Cascaded Style-Sheets (CSS) could provide such views, though like Flash would require significant programming effort on part of the author.
- **Page transitions and intra-page navigation:**
 - **Desired:**
 - Traditional use of keyboard keys and mouse for character, line, and page control.
 - **As Evaluated:**
 - PDF, Word, and Web-based document models support these features.
 - A document authored in Flash would need to be programmed to respond appropriately to key strokes, which can be cumbersome for the author. A tool would need to be designed to enable this feature.
- **Embedding and linking feature**
 - **Desired:**
 - The document model should support both formats for including multimedia and tabular data.

- **As Evaluated:**
 - Word and PDF natively support object linking and embedding features.
 - Word requires the hosting application to be present, as well as the supporting OLE application.
 - Flash supports embedding and linking through programmatic means.
 - Web-based documents do not support embedding. Rather, every included object is linked to it. The object could reside in the same folder on the same server as the document, or potentially on any other Web-accessible folder or networked server.
- **Image browsing and navigating to embedded/linked media objects**
 - **Desired:**
 - Mouse-click activation with keyboard support.
 - Embedded objects should be able to evoke appropriate viewers or players.
 - Images should be natively supported, especially all relevant formats such as JPEG, TIFF, DICOM, GIF, JP2, BMP, PNG, etc.
 - Should be easy to encode some degree of interaction with these into the document model.
 - **As Evaluated:**
 - PDF, Word, Web-based document models support these features.
 - Flash supports such linking and embedding; however, programming is needed.
 - In general, Microsoft Word is compatible with media types associated with Microsoft software suites, but is incompatible with media types external to it.
 - All the four platforms supported linked objects well, but they failed in various degrees when tested with embedded objects. Microsoft Word was better than others, however, in supporting embedded object activation.
 - Adobe Acrobat, Macromedia Flash and HTML also have limited media type compatibility.
 - None of them provides native support for DICOM images.
- **Native support for interactivity**
 - **Desired:**
 - The document model should include native support for adding interactivity to tabular data, images and other multimedia data.
 - **As Evaluated:**
 - None of the selected formats provide native support of interactivity with tabular data.
 - The closest contender is Microsoft Word with the Office suite of applications including Excel for tabular data and Access for database.

- **Execute code**
 - **Desired:**
 - The document model should be able to execute limited code in a secure fashion.
 - **As Evaluated:**
 - Microsoft Word and Adobe PDF can link to executing programs that are platform specific and could be a security threat.
 - Flash could be programmed to perform similar actions.
 - Web-based technologies offer the best solution here through technologies such as Java, Javascript, and PHP.
- **Transmission**
 - **Desired:**
 - The document model should support controlled order of transmission for “heavy weight” multimedia items to enhance readability.
 - **As Evaluated:**
 - None of the technologies support elegant and reader-sensitive downloading.
- **Document Structure**
 - **Desired:**
 - The document model should allow for multimedia and tabular elements to be organized in a fashion that supports the transmission requirements as well as ease in archiving.
 - **As Evaluated:**
 - None meets this requirement.

As such, none of the platforms considered provides a comprehensive solution for all the desired attributes of an interactive publication. Flash and HTML are particularly deficient since authors would need specific programming skills for authoring documents in those formats, and moreover they support few of the desired attributes in interactive publications. However, MS Word and PDF show promise. Overall, MS Word was found to be the better authoring tool. Though it does not have an editor, PDF was determined to be the best supporting format since it adequately meets several essential characteristics of interactive publications, including *platform independence*, *look and feel*, *some degree of interactivity*, and *the ability to execute specific applications*. In addition, it is a widely accepted format.

The desired document integrity attribute can be expected of both Microsoft Word and PDF formats since they allow embedding of multimedia objects into the document. However, our evaluation has revealed several problems in both authoring and in reading. Authoring problems have workarounds, but reading tool behavior is unexpected and often unexplainable. Additionally, Word’s embedded and linked features are often limited to its Microsoft Office suite and other OLE applications. Table 4.2 gives a broad subjective assessment of authoring and reading IPs in the four formats.

In summary, present-day document tools only provide ad hoc support for interactive publications and are limited in scope. This finding underscores the need for developing a customized document model, supporting languages, appropriate software tools, and procedural framework. The shortcomings found suggest an excellent starting point for further research, and are addressed as topics for our next steps and listed in Section 5.

	PDF	MS Word	HTML	Flash
Authoring				
Ease of creation	Easy	Easy	Moderate	Difficult
Specialized skills required	No	No	Yes	Yes
Skill level needed	Low	Low	Moderate	High
Reading				
Ease of use	Easy	Moderate	Easy	Easy
Speed moving from one media type to another	Same across all formats			
Speed for invoking media types	Same across all formats			
Need special client tools for viewing	Yes	Yes	Yes	Yes
Need special client tools for manipulating	Yes	Yes	Yes	Yes
System/Software				
Cross Platform	Yes	No	Yes	Yes
Open source	No	No	Yes	No

Table 4.2 Platform comparison

5 Next Steps

Research in enabling interactivity in scientific publications requires a multi-prong approach which includes decisions in the choice of the appropriate document format, tool development for authoring, downloading, and viewing, and a thorough evaluation of the technology at various stages to ensure that the participating groups, viz., authors, publishers, researchers, and archivists benefit from the ubiquity of this type of publication. These topics are reflected in our next steps.

5.1 Enhancing ITAG

The current version of ITAG, though useful, needs to be enhanced to allow greater user interaction with the data. These enhancements include the ability to parse, display, and interact with complex tables, enriched charting techniques, greater interoperability with popular statistical analysis software, ability to install ITAG as a client application, automatic startup of ITAG software without the use of authored scripts, and not requiring

separate files for data subsets. Next steps on this topic are guided by these desirable features.

Table A. Number of infant deaths and infant mortality rates, by age at death and by race and Hispanic ethnicity of mother: United States, 2001–02 linked files

Column Group Row Group	Neonatal deaths ¹								
	Infant deaths		Total		Early		Late		
	2002	2001	2002	2001	2002	2001	2002	2001	
Total	7.0	6.8	4.7	4.5	3.7	3.6	0.9	0.9	
White	5.8	5.7	3.9	3.8	3.1	3.0	0.6	0.6	
Black	13.8	13.3	9.3	8.9	7.6	7.3	1.7	1.7	
American Indian	8.9	9.7	4.6	4.2	3.2	3.1	1.4	1.4	
Asian or Pacific Islander	4.8	4.7	3.4	3.1	2.7	2.5	0.7	0.7	
Hispanic	5.6	5.4	3.8	3.6	3.0	2.9	0.9	0.9	
Non-Hispanic white	5.8	5.7	3.9	3.8	3.0	3.0	0.8	0.8	
Non-Hispanic black	13.9	13.5	9.3	9.0	7.6	7.4	1.8	1.8	
Number									
Total	27,970	27,523	18,791	18,275	15,020	14,622	3,771	3,652	
White	18,395	18,087	12,352	12,078	9,804	9,571	2,548	2,500	
Black	8,201	8,084	5,538	5,396	4,506	4,425	1,027	1,027	
American Indian	367	404	195	176	137	129	59	59	
Asian or Pacific Islander	1,066	947	710	624	577	496	137	137	
Hispanic	4,927	4,630	3,360	3,105	2,673	2,439	687	687	
Non-Hispanic white	13,327	13,300	8,853	8,817	7,002	6,979	1,851	1,851	
Non-Hispanic black	8,031	7,938	5,399	5,293	4,396	4,337	1,014	1,014	

¹ Neonatal is less than 28 days of age, early neonatal is less than 7 days, late neonatal is 7–27 days, and postneonatal is 28 days through 11 months.

Figure 5.1 – A depiction of table, column, and row grouping in a complex table structure.

Management of complex tables

A complex table is a grouping of several sub-tables that share common rows or columns. Each data element in a complex table may be derived through a functional result on other tables. From the point of view of interactivity, it should be possible for the reader to explore these data sources as one would browse an information hierarchy on a Web site. We call this hierarchical relationship among tables a *TableGroup*. Further, relationships across data elements formed through common rows and/or columns is called *RowGroup* and *ColumnGroup*, respectively. Such complex tables may also have headers and footers.

For example, in Figure 5.1 which shows a part of a table on infant mortality from the National Vital Statistics Report published by the Centers for Disease Control, each column is a table in itself, while belonging to a *ColumnGroup* with a neighboring column. Hierarchical table grouping is seen in the header where “Neonatal deaths” is subdivided into “Total”, “Early” and “Late”. An example of column grouping is “Infant Deaths” which is composed of two columns: “2002” and “2001”. Row grouping (highlighted in red) is indicated by the header “Rate per 1,000 live births”, as shown. Table header and footer are “Neonatal Deaths” and “Neonatal is less than...”, respectively, as shown in the figure.

The current version of ITAG does not support complex tables. In order to support table groups, column groups and row groups, it is necessary to build relationships between and across elements of multiple data sets. We propose the use of a generalized XML structure which can support such groupings to indicate these relationships. The design in its

degenerate form would enable instantiation of a simple table, as possible now. In addition to enabling the use of XML structure, ITAG will also need to be redesigned to support management, presentation and interactivity with such complex data structures.

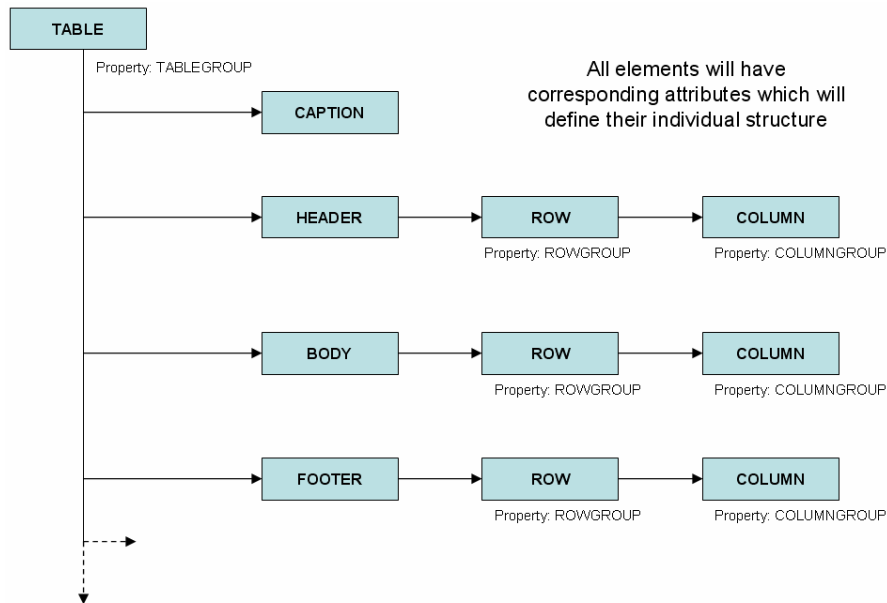


Figure 5.2 The initial XML DTD for complex tables

An abstraction for a DTD for our initial XML structure, which is currently under development, for an interactive complex table is illustrated in Figure 5.2. The element *Table* has a *TableGroup* property which can be set to indicate groups of tables. Similarly, the elements *Row* and *Column* have *RowGroup*, and *ColumnGroup* properties, respectively, which enable appropriate support for complex tables. This initial design of the DTD also supports the caption, headers and footers that commonly appear in such tables. Planned enhancements to this DTD include support for (multiple) subsets and special data relationships. An example of the correspondence of DTD elements with the complex table shown in Figure 5.1 is shown in Figure 5.3.

In order to enable interactivity in a complex table, raw CSV data needs to be converted into XML. In CSV no relationship between groups of data columns and/or rows is explicitly indicated and will need to be identified by the author (e.g., data types, functional relationships, sub-tables, row groups, column groups, etc.). We plan to develop a converter application written in Java to assist the author through a wizard-like interface in marking these table groups. A schematic indicating the role of the converter application in the generation of input files for ITAG is illustrated in Figure 5.4. Current functionality of using raw CSV data for simple tables is indicated as enclosed by the red box. Additional modules for complex tables and changes in the workflow are indicated within the green box.

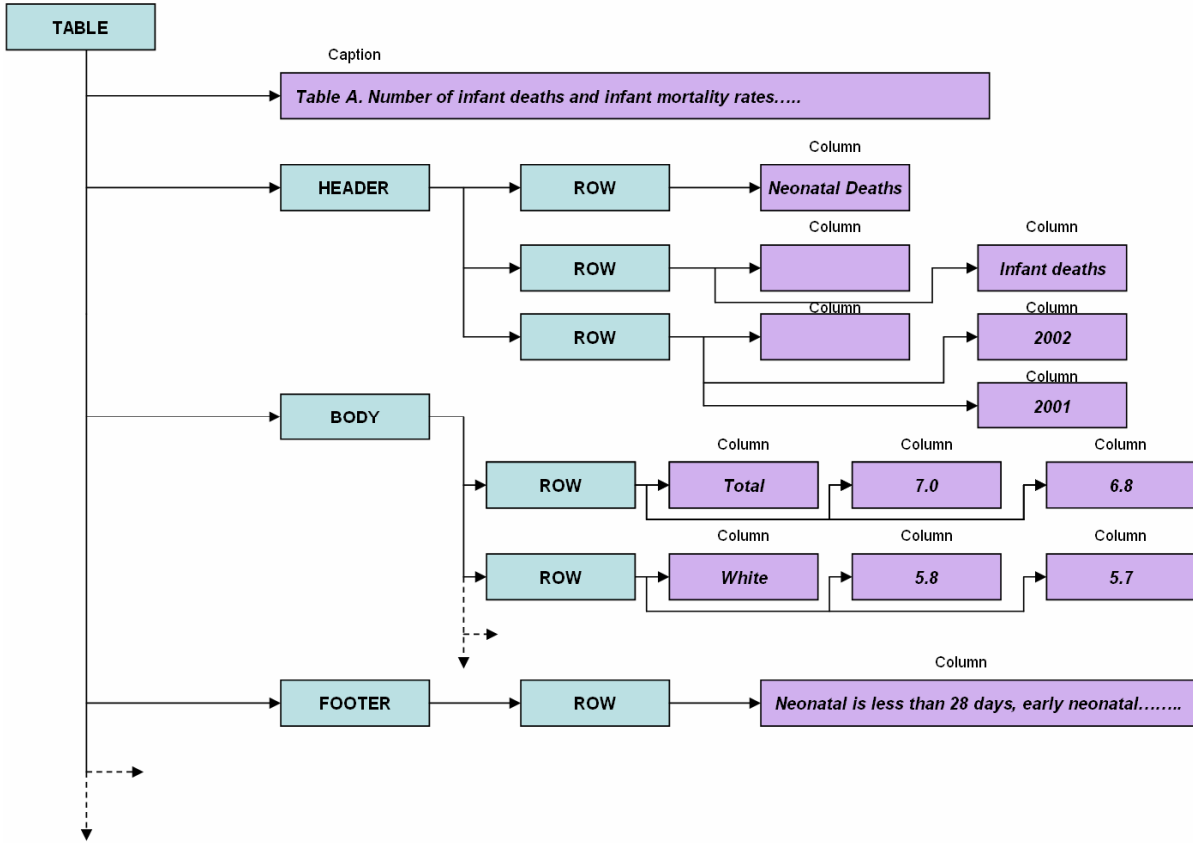


Figure 5.3 The table in Figure 5.1 presented in an XML abstract form.

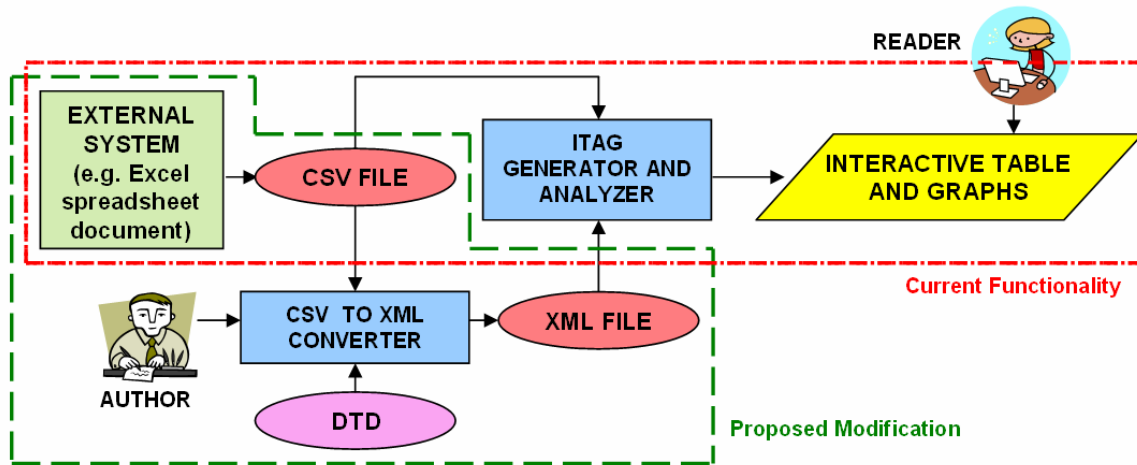


Figure 5.4 Proposed enhancements to ITAG input format for enabling interactive complex tables.

Other ITAG enhancements

In addition to changes necessary for supporting complex tables as outlined in the previous section, we have identified several other enhancements for ITAG, as listed below:

- **Charting/graphing features:** Current ITAG functionality allowing line graphs for only two columns needs to be expanded to allow multi-column plotting. In addition, other useful graphing methods such as bar graphs, stacked bar graphs, overlay of statistical information on bar graphs, and pie charts are to be developed.
- **Data filtering and subsets:**
 - ITAG currently has a limited form of data filtering capability. It is not possible, for instance, to interactively select a region of interest (ROI) on a dataset. Only rows can be selected and for each selection all *visible* columns may be selected. For arbitrary ROI marking (subsets), the author must hide columns that are not of interest before selecting rows, a cumbersome step. Proposed changes would enable arbitrary selection of ROI using data filtering techniques analogous to those in common use in database queries or by mouse-selection.
 - Proposed XML structure will support multiple tables and not require subsets to be saved as separate data files.
- **Command-line features:** ITAG has limited command-line functionality in which only a particular table, graph, or row-column selection can be invoked. Richer command-line features that allow additional ITAG capabilities to be selected at the command-line are proposed. In addition, changes to the DTD described above will enable individual XML data files to save computational functions by author and data ROI for ease of use.
- **Interoperability:**
 - ITAG can save data as CSV files. Additional data file formats specific to statistical analysis software will be explored.
 - Functions particular to various statistical software, such as Excel or SAS, should be executable directly from within ITAG through use of MIME-like application-function definitions.
- **Application installer:** Our prototype documents currently need to include the entire ITAG package to enable interactivity. In its current form, ITAG adds approximately 13MB to each document. This may not be much given current storage capacities, but can quickly add up if one were to store hundreds of articles on a personal computer¹. We propose the development of an application installer as used by other applications such as RealPlayer or QuickTime. The document would then use the system path or registry information to determine the presence and location of the installed application.

In addition to these enhancements, ITAG needs several support tools including those for creating linkages between text and tables presented in the article with the data sets and ITAG. Presently these are created by the author through executable batch or shell scripts. These scripts should be created automatically possibly through a GUI interface. Such a utility could be a component in a document authoring tool or a separate application. This

¹ Storing a copy of ITAG for every article would be prohibitive for the publishers, but simple download scripts on their respective websites could alleviate the problem by allowing them to maintain only a few copies and include one with each downloaded article.

GUI could also incorporate the ITAG tool, described above, for creating complex tables and defining relationships between various sub-tables, rows, columns, and respective groups.

5.2 Tool development

In order to enable the use of interactive publications, appropriate tools need to be developed for ease of authoring, review, and delivery. Some of these are:

Authoring tools

At the present state of development, the authoring tools proposed are primarily popular editors for document development accompanied by step by step procedures as instructions for the author. The author introduces interactivity for each media object (movie, DICOM image, table, etc.) by following a series of steps and selections, and has to keep almost every use situation in mind. This process can be tedious, repetitive, and possibly exasperating. Easy to use tools with wizard-like interfaces would be necessary to encourage widespread use.

Additionally, for most linked media objects, the interactivity is limited to the format in which the document was authored. A change of that format to another one for purposes of dissemination, for example, would not guarantee the interactivity originally designed. The author would need to recreate the interactivity in the target file format. A typical example is often seen in use today: most articles are created using MS Word or WordPerfect, but are placed on publishers' Web sites as PDF documents. The problem is that the interactivity introduced in the originals may not be retained automatically after format conversion to PDF. The solution is for tools that analyze source and target formats, correcting formatting problems as needed.

Download Manager

Whether the interactive publication is of the embedded or folder type, its size 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.

We propose an intelligent Download Manager utility to download the textual portion of the document first (allowing the reader to start perusing the text), while the data-intensive media objects arrive in the background. Moreover, the order in which these media are downloaded should be controllable by the reader based on his/her interest. For example, if a reader wishes to forego the introduction or methods sections typically found in research articles in favor of a dynamic table further down in the article, the Download Manager should be able to deliver the table first.

Progressive transmission somewhat analogous to this exists today in some image databases in which the stored images are decomposed by Digital Wavelet Transforms into layers at different resolutions. The user sees a low resolution version of the retrieved image immediately while details from the higher-resolution layers are filled in gradually.

We intend to explore these and similar concepts in our design of a Download Manager utility based on client-server architecture.

5.3 Evaluation plan

In order to understand the value of the interactive publication, its shortcomings in 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.

First, we intend to recruit one or more biomedical publishers with whose help we will establish a plan containing evaluation criteria for interactive publications: e.g., improvement of comprehension, learning, and degree of assessment of research reported in the publication. Next, we will provide our tools and procedures as they are developed to the publishers, their contributing authors and designated peer-reviewers, and train them in using our tools. In the course of authoring the publications, our collaborators will be expected to evaluate the tools and procedures provided 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.

5.4 Preservation

The long term preservation of all significant material in biomedicine is a mandated task for the 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 collaborative work with NLM's Preservation and Collection Management Section toward the design of systems for archiving scanned images and Web resources from various collections. 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.

6 Summary

This report outlines the research issues in creating and using interactive publications. Desirable attributes of such publications were defined, and existing document models were found deficient in some ways. Nevertheless, prototype publications were created in four common formats and evaluated against the established attributes. A tool for creating and using interactive tables and graphs was developed, and techniques were found to

incorporate such dynamic entities in a publication. Finally, next steps in research and development toward interactive publications were outlined.

7 Questions for the Board

1. The list of desirable attributes of an interactive publication is designed to promote its usefulness for better comprehension and learning as well as an aid to more in-depth peer review. Are there other useful attributes we should consider?
2. For our proposed collaboration with publishers, we have listed criteria to evaluate authoring and reading interactive publications. Are there other criteria we should consider?
3. Are there equal or better open-source tools for authoring or using the important media objects in interactive publications?
4. We have included several functions we consider useful in our interactive tables and graphs (ITAG) tool. Are there other functions we should consider?
5. Can you suggest collaborators in universities, industry or in research organizations to help us develop an authoring tool?

References

- [1] Lindberg, DAB and Humphreys BL. 2015 – The future of medical libraries. N. Engl. J. Med. (2005) 352; 11: 1067-70.
- [2] Climent J, Marés P. Automatic instrument localization in laparoscopic surgery. Electronic Letters on Computer Vision and Image Analysis 4(1): 21-31, 2004.
- [3] Buford JF, Rutledge L, Rutledge JL. Integrating object-oriented scripting languages with HyTime. Proc. International Conference on Multimedia Computing and Systems, May 1994, 425-434.
- [4] Sampaio PNM, Courtiat JP. Providing consistent SMIL 2.0 documents. Proc. IEEE International Conference on Multimedia and Expo, ICME 02, August 2002, Vol. 2, 337-40.
- [5] Boll S, Klas W, Westermann U. (1999) A comparison of multimedia document models concerning advanced requirements. Technical Report – Ulmer Informatik-Berichte. No. 99-01, Dept. of Computer Science, University of Ulm, Germany.
- [6] Boll S, Klas W. ZyX – A multimedia document model for reuse and adaptation of multimedia content. IEEE Trans. Knowledge and Data Engineering, Vol. 13, No. 3, May/June 2001.
- [7] Friedl R, Preisack M, Schefer M, Klas W, Tremper J, Rose T, Bay J, Albers J, Engels P, Guilliard P, Vahl CF, Hannekum A. CardioOp: an integrated approach to teleteaching in cardiac surgery. Stud Health Technol Inform. 2000; 70:76-82.
- [8] Friedl R, Klas W, Westermann U, Rose T, Tremper J, Stracke S, Godje O, Hannekum A, Preisack MB. The CardioOP-Data Clas (CDC). Development and application of a thesaurus for content management and multi-user teleteaching in cardiac surgery. Methods Inf Med. 2003; 42(1):68-78.

Appendix: Authoring procedures

Here we list the steps an author would take to create interactive publications in the four formats attempted: MS Word, PDF, HTML and Flash. Both types of publication are included: the **embedded** (single file) as well as the **folder** type (linked files in a folder). Note that the ‘embedded type’ is actually a hybrid entity since not all multimedia objects (e.g., DICOM images) can be embedded in a single file at the present state of development.

MS Word document: Folder type

1. Start with the text as an ordinary MS Word document. Assemble (save/copy/move) all the required multimedia material (bitmapped images, movies, clinical data, charts, other media formats) in the same folder as the document.
2. Insert hyperlinks to multimedia (images, animations, movies, DICOM, Excel spreadsheets, other file formats). In order to insert a hyperlink, select the text, right-click on the mouse to initiate a pop-up menu. Click on the Hyperlink menu item and then browse to find the corresponding multimedia file which needs to be linked. Alternately, the insert menu can also be used to provide hyperlinks. Click on the insert menu, select hyperlink submenu and then browse to the particular multimedia file. [CTRL+K is the keyboard shortcut for inserting hyperlinks. Hyperlinks can also be established by selecting an inserted image in the document using the above procedure.]
 - a. An animation file created in Flash of the type *.swf* format can be accommodated by first embedding it in an *HTML* file and then providing a link to the *HTML* page. Embedding a *.swf* file in a *HTML* page requires the author to use *<object>* tag to obtain the Flash player and then *<Param>* tag to embed the *.swf* file. [This can be done either manually or using a HTML editor.]
 - b. Clinical images in the DICOM format can be invoked using Java Applets or by linking to standalone applications capable of processing DICOM images.
3. Use internal hyperlinks in the MS Word document using the bookmark feature. This can be done by highlighting an image or text region and then selecting the Insert>Bookmark menu. A descriptive annotation can be given to the figure in the prescribed space. Then select the text which refers to that particular image and then right click to select the Hyperlink option. Once the Hyperlink window pops up, select the Place in this document option in the left menu of the window. This will display all the available bookmarks (annotations) in the document and the appropriate annotation can be selected to hyperlink that particular text to the image.

MS Word document: Embedded type

1. Start with the text as an ordinary MS Word document. Assemble (save/copy/move) all the required multimedia material (bitmapped images, movies, clinical data, charts, other media formats) in the same folder as the document.
2. Embed all images as objects into the Word document at the appropriate spatial positions. The supported types can be found by selecting the Insert>Object option. It will display a list of objects under the Create Object menu, such as Bitmap Image, Microsoft Paintbrush, etc.

Note that to view the object, the reader will need to have the corresponding tool on his/her computer properly configured to view that kind of object. For example, if we have a Bitmap Object and if the reader's computer is configured such that bitmap objects are opened, by default, using Microsoft Paint application, then every time the reader wants to view the object, Microsoft Paint will be invoked. On the other hand, if say, Adobe Photoshop is available, the reader's machine may be configured to open the bitmap objects, by default, by Adobe Photoshop.

There are two ways to embed images:

- Bitmap image objects – Create a new Bitmap Image object by selecting the Insert>Object menu in MS Word. Open images in .gif, .tif or other formats in another window using Microsoft Paint. Now copy the entire image and paste it onto the newly created Bitmap Object in the Word document, in the space provided by the object. The bitmap image is now embedded in the document.
 - Pictures – From the MS Word menu select Insert->Picture. There will be a number of options, the most common is From file. Use this to insert a picture into the document.
3. Internal hyperlinking in MS Word document requires the author to use the bookmark feature. This can be done by highlighting an image, and then selecting the Insert>Bookmark menu. A descriptive annotation can be given to the figure in the prescribed space. Then select the text which refers to that particular image and right click to select the Hyperlink option. Once the Hyperlink window pops up, select the Place in this document option on the left menu of the window. This will display all the available bookmarks (annotations) in the document and the appropriate annotation can be selected to hyperlink that particular text to the image.
 4. Embedding MS Excel spreadsheets is similar to that of embedding an image object. Create a Microsoft Excel Worksheet object after selecting the Insert>Object menu. Now select the Microsoft Excel Worksheet option from the resulting window. The corresponding spreadsheet to be inserted is opened in another window using the

MS Excel application. The entire content is then copied and pasted onto the newly created Excel Worksheet object.

5. Similarly, video files can be embedded as *video clip* objects. Create a video file object by selecting the Insert>Object>Video Clip from the main menu. Double click on the resulting icon, which opens a window having *insert clip* as one of the options. Select that option to browse and find the required video file. The file gets inserted in the document.
6. Similar steps can be performed for any of the other object types supported by MS Word, such as Calendar, MS Graph, etc.

PDF document: Folder type

1. Start with the text as an ordinary MS Word document. Assemble all the supplemental material (bitmapped images, movies, clinical data, charts).
2. Convert the text document to PDF format using Adobe Acrobat, following the steps File>Create PDF>From File. Navigate to the file to be converted, then click Select. The result is a PDF document identical to the original Word document, including all its images.
3. Create bookmarks for the document (to allow readers to jump to any point in the document, much like an interactive table of contents) by selecting the Bookmarks tab that appears to the left of the document. Move the document to the position for the bookmark, and select Options>New Bookmark. In the highlighted area, enter the name of the new bookmark.
4. Move all supplemental files (tables, graphs, videos, etc.) into the same folder as the document.
5. Use the Button tool in Adobe Acrobat to define interactive areas of the document. Typical interactive areas are tables that when clicked will open ITAG, or images that will open a video when clicked.
6. On defining an interactive area a dialogue box opens that permits changing the appearance of the button, as well as altering action settings. For appearance, set the Border and Fill to No Color, so that no additional graphical elements clutter the document when printing.
7. Next, select the Actions tab. In the Add an Action box, choose Mouse Up for the Trigger, and Open a File for the Action, then click Add. When a dialogue box opens, navigate to the File Location, select the file, then click Select. This will ensure that when the interactive area is clicked, the selected file will be launched.

PDF document: Embedded type

1. Start with the text as an ordinary MS Word document. Assemble all the supplemental material (bitmapped images, movies, clinical data, charts).
2. Convert the text document to PDF format using Adobe Acrobat, following the steps File>Create PDF>From File. Navigate to the file to be converted, then click Select. The result is a PDF document identical to the original Word document, including all its images.
3. Create bookmarks for the document (to allow readers to jump to any point in the document, much like an interactive table of contents) by selecting the Bookmarks tab that appears to the left of the document. Move the document to the position for the bookmark, and select Options>New Bookmark. In the highlighted area, enter the name of the new bookmark.
4. Import all supplemental files (tables, graphs, videos, etc.) into the document. For videos the command is Tools>Advanced Editing>Movie Tool. For large images it is best to go to the pages tab that appears to the left of the document. Then select Options>Insert Pages. A dialogue box will open allowing the author to select the image to be imported into the page.
5. Use the Button tool in Adobe Acrobat to define interactive areas of the document. Typical interactive areas are tables that when clicked will open ITAG, or images that will open a video when clicked.
6. On defining an interactive area a dialogue box opens that permits changing the appearance of the button, as well as altering action settings. For appearance, set the Border and Fill to No Color, so that no additional graphical elements clutter the document when printing.
7. Next, select the Actions tab. In the Add an Action box, choose Mouse Up for the Trigger, and Go to a page for the Action, then click Add. Next, a box will open that says *Use the mouse, scrollbars, and zoom tools to select the target view, then press Set Link to create the link destination*. Move the page that houses the video or image and press Set Link. Now, when a user clicks the button it will jump to a page at the end of the document that contains the specific media element. It is best to add a button to this page that will return the user to the document.

Macromedia Flash document

1. Start with the text as an ordinary MS Word document. Assemble all supplemental material (images, movies, clinical data, charts).
2. Copy the text from the document and paste them into several different frames in Macromedia Flash. [Each Flash frame corresponds to a page in the Word document.]
3. Add functionality, through actionscript, to move the playhead to the next or previous frame based on mouse interaction on a button. Add this actionscript to graphical buttons that can either be created in Flash or imported into Flash.
4. Import graphical elements from the Word document and place them on the appropriate frames. If these elements are links to external interactive objects, they are converted to graphical buttons and given actionscript that will load the objects. [These elements, such as video or large images, are actually separate Flash files that are loaded into empty movieclips that reside in the top, parent-level movieclip. This allows the various elements to be loaded as needed and independently of each other, so that a reader may begin viewing the document text as soon as it is downloaded, while other external files are loaded in the background.]
5. Create separate Flash files containing the external interactive elements. [This would also include software such as an image viewer that allows for zooming functionality.] Leave the first frame of these external files blank and put all content in the second frame. This will ensure that when the original parent file is loaded, these other files are loading in the background; however, because the first frame is blank, only the text document would be viewed. Once the reader clicks the corresponding button to view an interactive element, the playhead of the external Flash file will move into the contents frame and the content will display on top of the text document. Similarly, closing the file will simply park the playhead on the empty first frame.
6. For file types not supported by Flash (e.g., DICOM images), add a link for the file itself and let the user's operating system handle the file viewing. [For example, in the Windows OS a DICOM file with the .dcm extension might be set to open using DICOMworks. When a file string is entered in this fashion it is the equivalent of double clicking the file itself.]
7. Export all the separate Flash movies into .swf files and save them into the same folder.
8. For the **embedded type** of Flash document, the authoring procedure is identical to

the above, except that the various media elements are directly entered into Flash. The author, however, should create multiple movie clips and place the media elements into these movie clips. The only difference is that these movie clips do not need to be saved as external Flash files and can be kept within Flash itself. Some file types will still need to be external, however, such as DICOM.

HTML document

1. Popular HTML editors like Macromedia Dreamweaver, Microsoft Frontpage, among others, can be used to create interactive documents in HTML. In this case, authors should use customized functions provided by these HTML editors for creating an interactive HTML document.
 - a. In order to have a traditional paper look, start with an MS Word document and convert it to a folder type publication using the procedure described above. Click on the *save as* submenu on the *File* menu provided by MS Word to save the document as a Web page (.html file). The *view* menu of the Word editor provides an option to view the document as a Web page at the time of editing itself.

Glossary

BMP	A lossless bitmap image file format used internally by the Microsoft Windows® graphics subsystem, and used commonly as a simple graphics file format on that platform.
CSV	Comma separated values; a common format for tabular data.
CSS	Cascading Style Sheets is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents (http://www.w3.org/Style/CSS)
CSS2	W3C recommendation on Cascading Style Sheets, Level 2.
CT	Computed tomography (CT), also known as computed axial tomography or computer-assisted tomography (CAT), is a medical imaging method used to generate a 3D image from a series of 2D X-ray images taken around a single axis of rotation.
DICOM	Digital Imaging and Communications in Medicine; a format standard for interfaces to clinical equipment.
DTD	Document Type Definition is a set of declarations that conform to a particular markup syntax and that describe a class, or "type", of XML documents, in terms of constraints on the structure of those documents.
ECG	Electrocardiogram (also known as EKG from the German Elektrokardiogramm) is a graphic which records the electrical voltage in the heart in the form of a continuous strip graph.
ELCVIA	Electronic Letters in Computer Vision and Image Analysis; an online journal published in Barcelona, Spain.
GIF	Graphical Interchange Format is a compressed lossless bitmap image format for pictures and animations that use 256 (or fewer) distinct colors.
HTML	Hyper-Text Markup Language. Current release HTML4. (http://www.w3.org/TR/html401)
IEC	The International Electrotechnical Commission (IEC) is an international standards organization dealing with electrical, electronic and related technologies.
IP	Interactive Publications, as used in this document. Other common uses of the acronym include Internet Protocol and Intellectual Property.
ISO	International Organization for Standardization, a global federation of over a hundred national standards bodies with central secretariat in Geneva, Switzerland.
ITAG	Interactive Table and Graphics; viewer and editor software tool developed by CEB, and derived from Tool for OPerations on Catalogues And Tables (TOPCAT), a Starlink product released under the GNU General Public License (http://www.star.bris.ac.uk/~mbt/topcat).
ITU-T	The ITU Telecommunication Standardization Sector (ITU-T) coordinates standards for telecommunications on behalf of the International Telecommunication Union (ITU), based in Geneva, Switzerland. Prior to 1992, it was known as the International Telegraph and Telephone Consultative Committee (CCITT).
JPEG	Joint Photographic Experts Group that defined this lossy image compression standard relying on the Discrete Cosine Transform.
JP2	JPEG2000, a new open standard for compressed images relying on Digital Wavelet Transforms. It is intended to replace the current JPEG standard.
MRI	Magnetic resonance imaging, a clinical imaging modality using a magnetic field and electromagnetic radiation to create three dimensional images of internal anatomy.
OLE	Object Linking and Embedding is a distributed object system and protocol developed by Microsoft. OLE allows an editor to "farm out" part of a document, such as an image, to another editor and then re-import any changes. It is currently known as the distributed component object model (DCOM).

- PDF** Portable Document Format is developed by Adobe Systems, Inc. for representing documents in a manner that is independent of the original application software, hardware, and operating system used to create those documents.
- PNG** Portable Network Graphics is a lossless bitmap image format created to replace GIF.
- TIFF** Tagged Image File Format; common format for scanned documents and images.
- XML** The Extensible Markup Language (XML) is a W3C-recommended general-purpose markup language for creating special-purpose markup languages. It is a simplified subset of SGML, capable of describing many different kinds of data.
- XSL** The eXtensible Stylesheet Language (XSL) is a family of languages which allows one to describe how files encoded in the XML standard are to be formatted or transformed.
- W3C** World Wide Web Consortium (<http://www.w3.org>)

Curriculum Vitae

George R. Thoma

Chief, Communications Engineering Branch, LHNCBC, NLM

Education and Training:

Swarthmore College	B.S.	1965	Electrical Engineering
University of Pennsylvania	M.S.	1967	Electrical Engineering
University of Pennsylvania	Ph.D.	1971	Electrical Engineering

Research and Professional Experience:

1984-present Chief, Communications Engineering Branch, LHNCBC, NLM

Directs and conducts R&D in projects involving document image analysis, biomedical image processing, image animation and communications engineering. Project descriptions and results appear in the published literature (see Publications), and are synopsized in the branch's Web page <http://archive.nlm.nih.gov>. Recent work includes the following:

Digital Preservation Research: Developed an initial prototype of a *System for the Preservation of Electronic Resources* (SPER) that possesses the essential functions of a digital preservation system including ingest, automated metadata extraction, transfer of metadata from other databases, and bulk file migration.

Interactive Publications Research: Began a project to investigate the technology to create a comprehensive, self-contained and platform-independent multimedia documents.

Medical Article Records System (MARS): Led the development and implementation of systems to automate the extraction of bibliographic records to build MEDLINE citations from both scanned as well as online journal articles (MARS and WebMARS). Initiated the design of systems to meet the goals of NLM's *Indexing 2015* initiative to increase the efficiency of creating citations for MEDLINE so that the expected doubling of the citation rate in a few years can be accommodated through automation.

TurningThePages: Animation/touchscreen technology for photorealistic rendition of rare books.

Biomedical Imaging Research: Internet-accessible multimedia database of digitized x-rays, uterine cervix images, histological images and associated text; content-based image indexing; online digital x-ray atlas; algorithms for image indexing by shape, texture and color features.

AnatQuest: end-user access to a database of Visible Human cryosection and rendered images.

DocView: research in document imaging for end-user Internet access to electronic documents.

1974-84 Senior Electronics Engineer, Communications Engineering Branch, LHNCBC Conducted R&D in communications systems and signal processing techniques applied to electronic document storage, retrieval and display; biomedical image data compression, text recognition and image enhancement, satellite communications, echo suppression, videodisk premastering, motion-adaptive video compression.

1973-74 Systems Engineer, General Electric Co., Space Division, Valley Forge PA.
1971-73 Systems Engineer, AII Systems, Moorestown NJ.
1968-71 Ford Foundation Fellow, University Research Fellow and Post-doctoral
Research Associate, University of Pennsylvania, Philadelphia

Honors

- * General Chair, 14th IEEE Symposium on Computer-Based Medical Systems, 2001
- * Fellow of the SPIE, International Society for Optical Engineering
- * Awards: NIH Merit Award 1998, NLM Regents Award 1998, Federal Computer Week's *Federal 100 Award* 1995, NLM Staff Recognition Awards (1997-2002); On-the-spot Award (1998); Special Act/ Service Award (1997); Special Act Group Award (1992); Merit Awards (1984-97).
- * Member, Internet2 Applications Strategy Council.
- * Member, Maryland Governor's Task Force on High Speed Networks (1998- 2003).
- * Keynote speaker at SPIE Electronic Imaging, San Jose CA, February 1996; and at 8th IEEE Symposium on Computer-based Medical Systems, Lubbock TX, June 1995.
- * Member, Blue Ribbon Panel at NIST for document file format standards (1994-95).
- * Invited referee for: IEEE Journal on Special Areas in Communications; IEEE Computer Journal; IEEE Parallel and Distributed Technology; Information Processing and Management Journal; Journal of Clinical Engineering; SPIE and IEEE conferences.
- * Association of Image Information Management Certificate of Service (1995)
- * American College of Physicians Certificate of Accomplishment (1993)
- * NASA Certificate of Appreciation (1993)

Publications (2005 only):

Thoma GR. Public access to anatomic images. Chapter in: *Medical Informatics: Knowledge Management and Data Mining in Biomedicine*. Chen H, Fuller S, Friedman C, Hersh W, Eds., Norwell, MA: Springer Science + Business Media, 2005; 299-332.

Long LR, Antani SK, Thoma GR. Image informatics at a national research center. *Computerized Medical Imaging and Graphics* 29 (2005) 171-193.

Mao S, Misra D, Seamans J, Thoma GR. Design strategies for a prototype electronic preservation system for biomedical documents. *Proc. IS&T Archiving 2005 Conference*, Washington DC; April 2005; 48-53.

Walker FL, Thoma GR. Image preservation through PDF/A. *Proc. IS&T Archiving 2005 Conference*, Washington DC, April 2005; 259-63.

Antani SK, Natarajan M, Long JL, Long LR, Thoma GR. Developing a comprehensive system for content-based image retrieval of image and text from a national survey, *Proc. SPIE Medical Imaging*, February 2005; San Diego, CA; vol. 5748:152-161.

Antani SK, Long LR, Thoma GR. Applying vertebral boundary semantics to CBIR of digitized spine x-ray images. *Proc. IS&T/SPIE Electronic Imaging, Conference on Storage and Retrieval Methods and Applications for Multimedia 2005*. Lienhart RW, Babaguci N, Chang EY, Eds., January 2005, San Jose CA, SPIE Vol. 5682; 98-107.

Curriculum Vitae

Sameer Antani

Staff Scientist, Communications Engineering Branch (CEB), LHCNBC, NLM

Education and Training:

University of Pune, Pune India	B.E.	1994	Computer Engineering
The Pennsylvania State University	M.Eng.	1998	Computer Science and Engineering
The Pennsylvania State University	Ph.D.	2001	Computer Science and Engineering

Research and Professional Experience:

4/2002 – Present Staff Scientist, CEB, LHCNBC, NLM

12/2001 – 4/2002 Post-Doctoral Research Fellow, LHCNBC, NLM

Conducts and coordinates R&D in topics that include, but are not limited to:

Content-based image retrieval: Image segmentation, compression, feature extraction, feature organization, medical validation, image archives and indexing, feature similarity, combined text/image retrieval, multimedia medical information databases.

Interactive publications: Investigate and develop techniques for creation, use, and archival of comprehensive, self-contained, and platform-independent multimedia documents.

Video preservation and archiving: Investigate and develop techniques for content-based indexing, delivery, and long term preservation of video and media data.

8/1995 – 5/2001 Researcher, Instructor, and Systems Administrator (Computer Vision Laboratory), Department of Computer Science and Engineering, The Pennsylvania State University.

5/1998 – 8/1998 Research Intern, Philips Research, NY

1/1995 – 5/1995 Project Officer, Advanced Systems and Computer Training Ltd., India.

7/1994 – 12/1994 Instructor, Pune Institute of Computer Technology (PICT), India

1/1993 – 5/1995 Proprietor, SamssoftTech, Pune, India

Honors:

- General Co-Chair 17th IEEE Symposium on Computer-Based Medical Systems, 2004, Bethesda, Maryland.
- Staff Achievement Awards, National Library of Medicine, NIH, DHHS, 2003, 2004
- Special Achievement Award, National Library of Medicine, NIH, DHHS, 2004
- Session co-chair and co-organizer on special invited session for “Shape and Pattern Analysis in Content-Based Diagnostic Information Retrieval” at 8th World Multi-Conference on Systemics, Cybernetics, and Informatics (SCI) 2004, Orlando, FL.
- Session co-chair and co-organizer on special invited session on “Knowledge Discovery, and Image and Signal Processing in Medicine” at IEEE International Joint Conference on Neural Networks (IJCNN) 2003, Portland, OR.

- Program Committee / Reviewer for ACM Multimedia 2002, ACM Multimedia Systems Journal, International Journal for Document Analysis and Recognition, IEEE Transactions on Systems, Man, and Cybernetics, IEEE Transactions on Circuits and Systems for Video Technology, and other prestigious international conferences and journals on computer vision, pattern recognition, image processing, document analysis, and visual (and multimedia) information systems.
- President, Engineering Graduate Student Council, Pennsylvania State University, (1997-1998)
- Other university-wide student leadership positions. (1996 – 2000)
- Student Member, Strategic Planning Committee, Department of Computer Science and Engineering, Pennsylvania State University, 1998, 1996.
- Secretary, IEEE Student Chapter, Pune, India - (1993 – 1994).
- Earned B.E. degree with *First-Class with Distinction*, University of Pune, India, 1994
- Rotary International Award for Computer Excellence, 1994
- National Merit Scholarship, Government of India, 1987

Publications:

- Long LR, **Antani SK**, Thoma GR. Image informatics at a national research center Image informatics at a national research center. Computerized Medical Imaging and Graphics 29 (February 2005) 171-193
- **Antani S**, Lee DJ, Long LR, Thoma GR. Evaluation of shape similarity measurement methods for spine x-ray images Journal of Visual Communication and Image Representation Special issue: Multimedia Database Management Systems - Edited by Y. Li, J. R. Smith, T. Zhang and S.-F. Chang. 2004, 15(3):285-302.
- Chamarchy P, Stanley RJ, Cizek G, Long R, **Antani S**, Thoma G. Image analysis techniques for characterizing disc space narrowing in cervical vertebrae interfaces. Computerized Medical Imaging and Graphics, 2004, 28(1-2):39-50
- Crandall D, **Antani S**, Kasturi R. Extraction of Special Effects Caption Text Events from Digital Video. International Journal of Document Analysis and Recognition. Vol 5(2-3). 2003:138-57
- **Antani S**, Kasturi R, and Jain R. A Survey on the use of Pattern Recognition Methods for Abstraction, Indexing and Retrieval of Images and Video Pattern Recognition Journal, 35(4) :945-965, 2002.
- Xu X, Lee DJ, **Antani S**, Long LR. Relevance Feedback for Spine X-ray Retrieval. To appear in Proceedings of 18th IEEE Symposium on Computer-Based Medical Systems. Los Alamitos CA: IEEE Computer Society. June 2005.
- **Antani SK**, Long LR, Thoma GR. Applying vertebral boundary semantics to CBIR of digitized spine x-ray images Proceedings IS&T/SPIE Electronic Imaging Science and Technology, Conference on Storage and Retrieval Methods and Applications for Multimedia 2005, Lienhart RW, Babaguci N, Chang EY, Eds., January 18-19, 2005, San Jose, California, USA, SPIE Vol. 5682, pp.98-107
- **Antani S**, Long LR, Thoma G. Content-Based Image Retrieval for Large Biomedical Image Archives Proceedings of 11th World Congress on Medical Informatics (MEDINFO) 2004 CDROM, Print: Part 2, September 7-11 2004.

Curriculum Vitae

Glenn Ford
Senior Software Engineer

Education and Training:

Montgomery College, Germantown MD	A.A.	1989	Computer Science
University of Maryland, Baltimore MD	B.S.	1991	Computer Science

Research and Professional Experience:

1996 – Present Communications Engineering Branch, LHNCBC, NLM

- Lead software developer for the PubMed On Tap research project. PubMed on Tap is an application for PDAs that facilitates Internet access to NLM's MEDLINE database using PubMed utilities. The client is written in C++ using CodeWarrior 9.0 and the servlet is in Java with Tomcat.
- Lead developer for a web-mediated ground truth database for biomedical journals. Developed a set of ground truth data and tools for the analysis of information extraction from scanned articles. Algorithms that researchers can explore are for page segmentation, zone identification, OCR character features, and data manipulation. Website is at <http://marg.nlm.nih.gov>.
- Conducts research and development for an automated data entry system for MEDLINE medical journals. The system, MARS (Medical Article Records System), automatically detects and extracts citation information from biomedical journal articles.
- Designed and developed WILL (Workstation for Interlibrary Loan). WILL encapsulated all the technologies necessary to capture, recognize and deliver medical articles via the Internet, fax, or print. WILL was used in production at the National Institutes of Health Library.

1997 Consultant, Panda Publishing Inc. : West Group Publishing

- For West Group Publishing, Mr. Ford was lead design engineer in charge of developing the Estate Planning System (EPS) for Windows. The EPS software, completed in December 1998, handles Estate Tax Forms 706, 709 and 1041, including all schedules and worksheets. The system uses an image rendering tool developed internally to allow on-screen WYSIWYG form data entry.

1997 Consultant, Panda Publishing Inc. :
Naples Florida Chamber of Commerce

- For the Chamber of Commerce in Naples, Florida Mr. Ford developed a Recreational Travel CD-ROM system to highlight the attractions in Naples,

Florida. Designed and developed a system that uses over 2,000 images taken, developed and processed by Panda Publishing, using a full-text engine and hyperlink.

1994 – 1995 Consultant, Corporate Software : Hearst Business

- Designed and developed a car dealer inventory system (Black Book), for Hearst Business. As project manager for Hearst, Mr. Ford was responsible for two other developers and served as the project lead on all of Hearst's software systems.

Publications

- Alexander GL, Hauser SE, Steely K, Ford G, Demner-Fushman D. A Usability Study of the PubMed on Tap User Interface for PDAs Proceedings of 11th World Congress on Medical Informatics (MEDINFO) 2004 Imaging Informatics. September 7-11 2004; San Francisco, CA, USA. 1411-15
- Hauser SE, Demner-Fushman D, Ford G, Thoma G. PubMed on Tap: Discovering Design Principles for Online Information Delivery to Handheld Computers Proceedings of 11th World Congress on Medical Informatics (MEDINFO) 2004 Imaging Informatics. September 7-11 2004; San Francisco, CA, USA. 1430-33
- Hauser SE, Demner-Fushman D, Ford G, Thoma GR. A testbed system for mobile point-of-care information delivery Proc. 17th IEEE Symposium on Computer-Based Medical Systems. Los Alamitos CA: IEEE Computer Society. June 2004, 147-52.
- Ford G, Thoma GR. Ground truth data for document image analysis Proceedings of 2003 Symposium on Document Image Understanding and Technology, 9-11 April 2003, Greenbelt, MD: 199-205.
- Thoma GR, Ford G, Le DX, Li Z. Text verification in an automated system for the extraction of bibliographic data Proc. 5th International Workshop on Document Analysis Systems, Springer-Verlag: Berlin, August 2002, 423-32.
- Thoma GR, Ford G. Automated data entry system: performance issues Proc. SPIE: Document Recognition and Retrieval IX, Vol. 4670, January 2002, 181-90.
- Ford G, Hauser SE, Le DX, Thoma GR. Pattern matching techniques for correcting low confidence OCR words in a known context Proc. SPIE, Vol. 4307, Document Recognition and Retrieval VIII, January 2001, 241-9.
- Ford GM, Hauser SE, Thoma GR. Automated reformatting of OCR text from biomedical journal articles Proc. 1999 Symposium on Document Image Understanding Technology, April 1999; College Park, MD: Institute for Advanced Computer Studies; 321-5.
- Thoma GR, Hauser SE, Ford G. Automating document delivery: a case study Proc. the 12th Annual Computers in Libraries. Arlington, VA. Published by Information Today, Inc. March 10-12, 1997; 137-9.

Curriculum Vitae

Michael Chung
Multimedia Artist/Animator

Education and Training:

Montgomery College, Rockville MD	A.A.	1997	Computer Graphics
University of Maryland, Baltimore MD	B.A.	1999	Visual Art

Research and Professional Experience:

2003 – present Communications Engineering Branch, LHNCBC, NLM

- Creates all graphics and animation assets for Turning the Pages project at National Library of Medicine. This includes 2D manipulation of high resolution page scans, creations of 3D simulation of book, animation of 3D book, modification of code in Macromedia Director for kiosk version of book, and creation of web versions of books in Macromedia Flash.
- Conducts R&D in interactive documents using Adobe Acrobat (PDF), and Macromedia Flash.

2000 – 2003 Lead Animator, Infomatic Corporation

- Supervised a small staff of animators in the creation of online learning modules using Macromedia Flash.
- Conducted R&D in real-time, interactive Web 3D programming and content creation.

Fall 2000 Instructor, Montgomery College

- Taught introductory courses in Adobe Photoshop and Alias Maya
- Created course curriculum for Alias Maya (first Maya instructor at Montgomery College).

Spring 2000 Graphic Designer, National Association for Music Education

- Responsible for the design and printing of a variety of printed material including books, magazines, brochures, and forms.
- Responsible for the redesign and maintenance of the National Association for Music Education website.

Curriculum Vitae

Kirankumar Vasudevan

Graduate Student Fellow, Communications Engineering Branch, LHNCBC, NLM

Education and Training:

University of Calicut, Kerala, India	BTECH	1999	Electrical & Electronics Engineering
University of Bridgeport, Bridgeport, CT	MS	2003	Electrical Engineering
University of Maryland College Park, MD	PhD (student)		Electrical & Computer Engineering

Research and Professional Experience:

2005-Present Graduate Student Fellow, CEB, LHNCBC, NLM

- Conducting R&D toward developing a robust standard to create open source, platform independent tools for authoring, publishing and archiving Interactive Publications. Also involved in the active prototyping of Interactive Publications.

2003-2004 Research Assistant, Dept. of EE, State University of New York at Buffalo

- Performed research in the Materials, Devices and Circuits simulation laboratory, SUNY, Buffalo. Numerical device simulation based on Schrödinger's algorithm using FORTRAN, Java. Growth and characterization of nanostructures.
- Teaching Assistant for EAS 200 (EE Concepts): included working with students, helping during recitation hours, preparing, proctoring and grading exams.

2002-2003 Graduate Assistant, Dept. of EE, University of Bridgeport, CT

- Developed SPICE libraries, VHDL coding, performed design and simulation of circuit components using EDA tools. Fixed, ran and tested equipment in the Electronics lab.
- Taught EDA tools to graduate students for the Analog Digital IC Design course. Extensively used MATLAB to derive results for different research projects.

2000-2002 Software Engineer, India and US

- Developed applications in Java and C++ platforms at Hextelsoft Inc., San Jose, CA; and at BSTL and Unique Info Software, India.

1999-2000 Maintenance Electrical Engineer, India

- Fixed and maintained machines for the production of newspapers and tires at The Mathrubhumi Printing and Apollo Tires, India.

Honors and Professional Activities:

- Academic Achievement Award in Electrical Engineering (MS) at the School of Engineering, University of Bridgeport
- Selected to Who's Who Among Students in American Universities and Colleges; Academic Excellence Award 2003-2004, 2002-2003
- Member of IEEE Solid State Devices and Circuits Society, Electron Devices Society and Nanotechnology Council.

Research & Development skills:

- Used numerical approximation algorithms to solve non-linear engineering problems in Java and C++. Coded functions in C to enhance the signal processing capability of MATLAB.
- Designed and developed a capacitive imaging array for sensing the capacitance of biological cells. A novel Mixed-Signal capacitive sensing chip using dynamic biasing, switched capacitive techniques was designed and simulated at the *Integrated Biomorphic Systems Laboratory (IBIS), Dept. of ECE, University of Maryland College Park*. Used Cadence for simulation and layout.
- Designed a MEMS biosensor for sensing biochemical agents and for calibrating an Optical Profiler System in *MEMS Sensors and Actuators Lab (MSAL), Dept. of ECE, University of Maryland College Park*. Developed the circuit section of the sensor. Used Cadence for simulation and layout.
- Master's Thesis: Design of a novel architecture for a low voltage rail-to-rail CMOS opamp used for low-voltage applications.
- Designed logic circuits using differential signaling techniques (PECL, CMOS, LVDS) to improve speed, noise margin and power dissipation.

Computer Skills:

- Programming in Java, C++, C, VHDL, UNIX.
- Extensive usage of EDA tools (Mentor Graphics, Cadence, PSPICE)
- Comfortable with Math and laboratory packages (Matlab, Simulink)
- Familiar with Oracle, MS-SQL, Web-development tools (HTML, XML, JavaScript)
- Working knowledge of server-side programming in Java (Servlets, JSP)