Developing a Workflow for the Integration of Patient Education Materials into EPIC

Muhammad-Sharif Moustafa

Mentors: Kristine Petre and Linda Schwartz

Library Services

Abstract

The purpose of this project was to create a simple, maintainable, and flexible workflow for patient education and library staff to incorporate custom LVHN patient education documents into the EPIC medical records system so that end users of EPIC may be able to easily document that they gave patients education materials. The workflow consisted of a series of well-defined steps and a computer program that automated several steps of the workflow. Corresponding documentation was written for both the workflow and computer program to ensure the simple usage and maintenance of the workflow beyond the end of this project. The workflow was tested by converting several patient education documents to HTML and uploading the documents to EPIC. The patient education staff also tested the workflow and provided feedback, which was incorporated back into the workflow.

Introduction

Patient education is an important part of health care. Effective patient education can improve the health of patients by reducing anxiety and improving compliance, which in turn reduces the cost of care at hospitals. (“Patient Education in Rehabilitation”, 2010). Patient education can include providing patients with reading materials that explain their condition, possible treatments, and care instructions. At LVHN, when health professionals need to give patients education materials related to their diagnoses, it must be documented in the EPIC medical records system. To do this, they manually type in the titles of the documents that they gave to the patients and request the documents from the print shop. While this is a simple process, it is neither efficient nor free from errors. Since most of the process of documenting patient teaching is done by people, there are many opportunities for error. A health professional may make any number of these mistakes when documenting patient education: mistype the title; forget to record the title; record the title in the incorrect section of the record; etc… This issue is well known, and can be avoided by integrating the patient education process into EPIC.

Integrating the patient education process into EPIC would mean that health professionals would be able to search and document the dissemination of patient education materials using interfaces in EPIC specifically built for this purpose. This would reduce the opportunities for error because EPIC will do the error-prone tasks for users (e.g. recording the title of a document) and allow users to focus on the more important aspects of the process (e.g. which patient education materials are to be given to the patient).

LVHN has a repository of more than 1500 patient education documents that need to be integrated into EPIC. Thus a clear and efficient workflow that could be followed by the patient education staff needed to be formulated and tested in this project. Although not a requirement, automating the steps of the workflow greatly increased the robustness of the workflow.

Methods

To gather the requirements of the workflow, interviews were held with the Director of Knowledge Management, Senior Medical Librarian, and patient education staff. There was also correspondence between the director and other administrators at the University of Michigan and University of Virginia about their current workflows. Those workflows served as the starting point for our workflow. A preliminary workflow was drafted and the staff reviewed it and provided feedback. When the first version of the workflow was created after being approved by the staff, the implementation of it was detailed. The workflow consisted of a program written in-house that interfaced with Microsoft Word and other programs and led users through the process of converting documents to HTML and inserting metadata.

When the workflow was in its final stages, it was tested by having the patient education staff use the program to create several HTML documents with metadata and upload them to EPIC. For the first trial run, each employee used the workflow to convert a few documents to HTML and add metadata to them under supervision so that they could ask questions and give feedback as they used it. Their documents were uploaded to EPIC and reviewed by the director. After the first trial run, they used the workflow without supervision, relying only on the documentation that was provided for the workflow. They asked questions and provided feedback after they used the workflow. Their feedback was incorporated into the documentation, workflow, and program as they tested the workflow.

Results and Discussion

The patient education staff wanted LVHN patient education materials to be integrated into EPIC and to utilize certain features of EPIC. Specifically, they wanted to have the documents be displayed at the top of the "Relevant Documents" section when a health professional is selecting documents to give to the patient. It was previously known from the University of Michigan and University of Virginia that documents needed to be in HTML format to be uploaded to EPIC, required the correct metadata to be displayed in the "Relevant Documents" section, and needed an asterisk in the document title in the metadata in order for the documents to appear at the top of the "Relevant Documents" section. These requirements and their corresponding actions are listed in Table 1.

Table 1: Requirements for the workflow

Requirement	Action That Will Satisfy Requirement
Integrate LVHN patient education materials into EPIC	Convert the documents to HTML
Display LVHN patient education documents in the "Relevant Documents" section in EPIC	Insert a specific set of metadata into the HTML documents
Display LVHN patient education documents at the top of the list of the "Relevant Documents" section	Add an asterisk to the beginning of the title of the document in the metadata

The workflow needed to accomplish each of these actions. However, it also needed to be usable by non-technical users. Thus, an unofficial fourth requirement was that the workflow needed to be designed with non-technical users in mind. Ultimately this requirement was satisfied by creating a well-documented computer program with a graphical user interface that would perform most of the technical actions in the workflow.

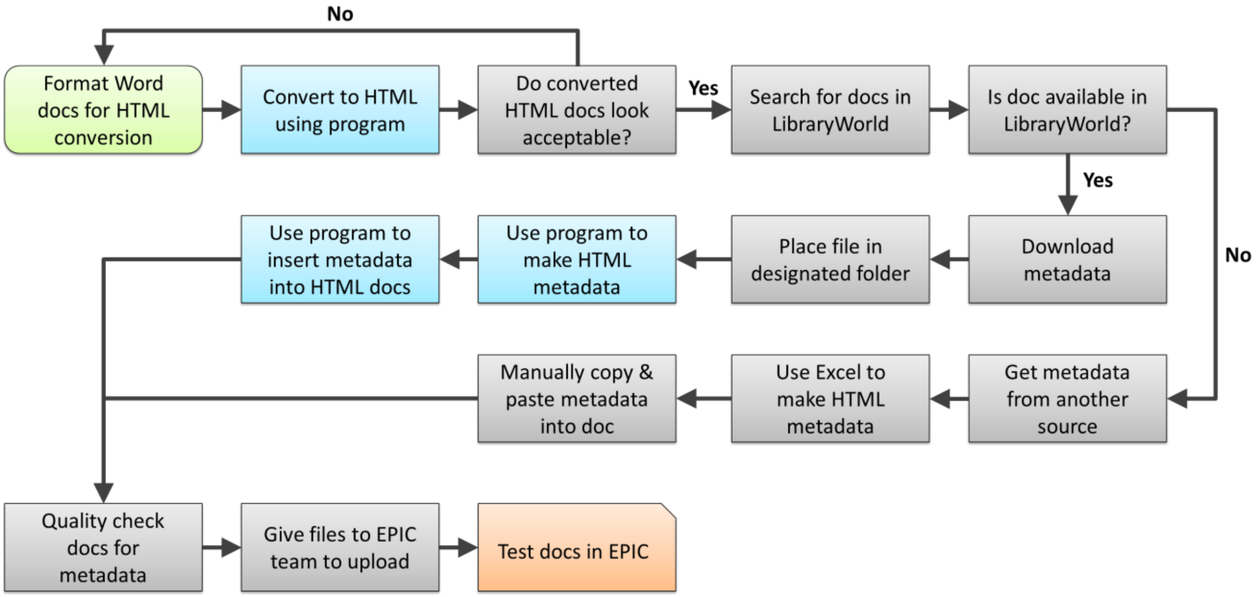

No
Format Word docs for HTML conversion
Convert to HTML using program
Do converted HTML docs look acceptable?
Yes
Search for docs in LibraryWorld
Is doc available in LibraryWorld?
Yes
Download metadata
Place file in designated folder
Use program to make HTML metadata
Use program to insert metadata into HTML docs
No
Get metadata from another source
Use Excel to make HTML metadata
Manually copy & paste metadata into doc
Quality check docs for metadata
Give files to EPIC team to upload
Test docs in EPIC

Figure 1: Flow chart representation of the workflow. The green rounded rectangle is the start of the workflow, the blue rectangles signify usage of the program, the gray rectangles are manual steps, and the orange single snip corner rectangle is the end of the workflow.

The program had three modules: a conversion module, a metadata generation module, and a metadata insertion module. In the conversion module, users placed patient education documents as Microsoft Word documents in a folder and used the program to convert them to HTML. The program achieved this by first feeding each document to a macro in Microsoft Word to convert them to HTML and then converting the documents to XHTML (a stricter form of HTML) using another program called html2xhtml. In the metadata generation module, users downloaded the metadata for the documents from LibraryWorld in MARC format and used the program to format the metadata as HTML metadata. In the metadata insertion module, users registered the HTML documents and HTML metadata with the program and proceeded to match the documents with their corresponding metadata. In the case that a document did not have its metadata in LibraryWorld, the user skipped that document and added the metadata to the document manually. This involved obtaining the metadata from another source (librarian, nurse, etc), inputting the metadata into an Excel worksheet with formulas to generate HTML metadata, and copying and pasting the metadata into the document.

The graphical interface of the computer program allowed users to complete each of the workflow steps without requiring familiarity with the technical aspects involved in each step. The user interface was made to be as simple and effective as possible so that users would be able to use the computer program with minimal technical skill and training. To build the user interface, a list of qualities of good user interfaces was compiled and adhered to as much as possible. A simple and effective user interface has these qualities ("Designing the User Interface", 1998):

1. Very low probability of crashing
2. Reduces the memory load on the user
3. Clearly shows the consequences of actions
4. Reduces user error when using the program
5. Clearly highlights destructive actions and prevents the user from performing them accidentally
6. Uses color to convey nonessential information only to accommodate color-blind users
7. Provides instruction on using the program either within the program or in an easily accessible area
8. Clearly shows the intended workflow of the program
9. Easy to use for novice users and efficient to use for experienced users

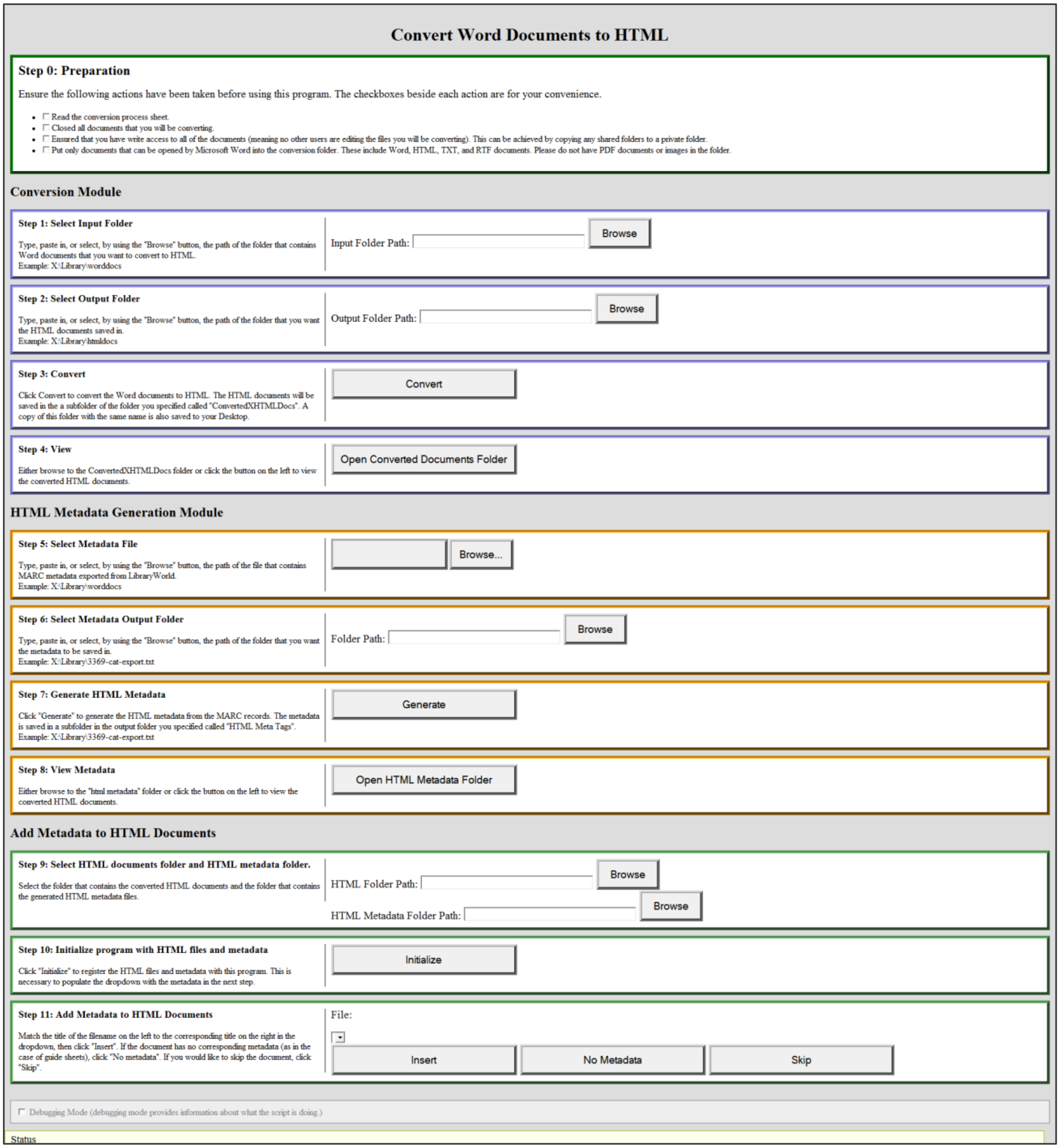
The program was built to include as many of these qualities as possible. Table 2 lists some of the program's features and the corresponding user interface quality each exhibits. Other features outside of the user-interface were built into the program to make it easy to use. One of these is that most of the EPIC integration workflow can be done within the program, reducing the need for users to switch to other programs. This promotes consistency, which reduces the opportunity for errors and simplifies the user experience. ("Designing the User Interface", 1998) Using only one program to perform a task is easier and more efficient than using many programs. The program was also made to be maintainable by splitting the application logic into smaller logical parts and documenting the source code's syntax and semantics. This will allow maintainers to work on small segments of code and be able to understand them with little effort.

As a last resort in the case the program cannot be used, the lower branch of the workflow shown in Figure 1 involving Microsoft Excel can be used as a manual backup. This backup enhances the robustness of the workflow because it allows the patient education staff to use the workflow even if there is no one able to maintain the program.

Table 2: A list of user interface qualities and the features of the program that exhibit that quality.

User Interface Quality	Corresponding Feature in Program
Very low probability of crashing	Extensive error checking to prevent crashes in most situations
Reduces the memory load on the user	Helpful hints throughout the program to remind the user of the required actions to be performed in the program
Clearly shows the consequences of actions	Detailed explanations of the consequences of each action are provided next to each step in the program

Reduces user error when using the program	Form checking to prevent errors from user input
Clearly highlights destructive actions and prevents the user from performing them accidentally	Avoidance of overwriting files when the program runs to protect the user from erroneous destructive actions
Uses color to convey nonessential information only to accommodate color-blind users	Steps are outlined in different colors depending on which module they are part of to assist users in quickly associating steps together. Modules are also labeled so that the color coding is not the only way to differentiate modules.
Provides instruction on using the program either within the program or in an easily accessible area	Each step has corresponding usage instructions beside it in the interface as well as a thorough user manual that is distributed with the program
Workflow of the program should be obvious to the user	The user interface is divided into modules, which are further divided into numbered steps that strongly suggest to the user to complete each step in order
Easy to use for novice users and efficient to use for experienced users	Every step has instructions beside it so that a novice will know where to look for more information about a step and an experienced user will be able to easily ignore the instructions.



Convert Word Documents to HTML

Step 0: Preparation

Ensure the following actions have been taken before using this program. The checkboxes beside each action are for your convenience.

- □ Read the conversion process sheet.
- □ Closed all documents that you will be converting.
- □ Ensured that you have write access to all of the documents (meaning no other users are editing the files you will be converting). This can be achieved by copying any shared folders to a private folder.
- □ Put only documents that can be opened by Microsoft Word into the conversion folder. These include Word, HTML, TXT, and RTF documents. Please do not have PDF documents or images in the folder.

Conversion Module

Step 1: Select Input Folder

Type, paste in, or select, by using the "Browse" button, the path of the folder that contains Word documents that you want to convert to HTML.
Example: X:\Library\worddocs

Step 2: Select Output Folder

Type, paste in, or select, by using the "Browse" button, the path of the folder that you want the HTML documents saved in.
Example: X:\Library\htmldocs

Step 3: Convert

Click Convert to convert the Word documents to HTML. The HTML documents will be saved in the a subfolder of the folder you specified called "ConvertedXHTMLDocs". A copy of this folder with the same name is also saved to your Desktop.

Step 4: View

Either browse to the ConvertedXHTMLDocs folder or click the button on the left to view the converted HTML documents.

HTML Metadata Generation Module

Step 5: Select Metadata File

Type, paste in, or select, by using the "Browse" button, the path of the file that contains MARC metadata exported from LibraryWorld.
Example: X:\Library\worddocs

Step 6: Select Metadata Output Folder

Type, paste in, or select, by using the "Browse" button, the path of the folder that you want the metadata to be saved in.
Example: X:\Library\3369-cat-export.txt

Step 7: Generate HTML Metadata

Click "Generate" to generate the HTML metadata from the MARC records. The metadata is saved in a subfolder in the output folder you specified called "HTML Meta Tags".
Example: X:\Library\3369-cat-export.txt

Step 8: View Metadata

Either browse to the "html metadata" folder or click the button on the left to view the converted HTML documents.

Add Metadata to HTML Documents

Step 9: Select HTML documents folder and HTML metadata folder.

Select the folder that contains the converted HTML documents and the folder that contains the generated HTML metadata files.

Step 10: Initialize program with HTML files and metadata

Click "Initialize" to register the HTML files and metadata with this program. This is necessary to populate the dropdown with the metadata in the next step.

Step 11: Add Metadata to HTML Documents

Match the title of the filename on the left to the corresponding title on the right in the dropdown, then click "Insert". If the document has no corresponding metadata (as in the case of guide sheets), click "No metadata". If you would like to skip the document, click "Skip".

□ Debugging Mode (debugging mode provides information about what the script is doing.)

Status

Figure 2: The user interface of the program features large buttons, extensive instructions, numbered steps, and color-coded groups.

Testing the workflow with the patient education staff provided valuable feedback that was used to improve the usability of the workflow. User testing should be as representative as possible of the real usage of the workflow. (“Usability Engineering”, 1994) During the tests, users followed the steps detailed in the documentation with limited supervision. The tests realistically mimicked the way they would be using the workflow. The supervisor intervened when there were shortcomings in the documentation that were revealed during testing. Otherwise, users progressed through the workflow independently. These tests were then used to clarify the workflow. After the supervised test, users were able to use the workflow without supervision.

When testing the HTML documents in EPIC, it was found that the language tag in the metadata was causing issues when displaying documents written in languages other than English. When different language versions of the same document were uploaded to EPIC, only the English document would display. The other versions were hidden completely from both the “Relevant Documents” section and search. This was an issue because patient education documents must be available in the languages of the patients who will read them. The issue was resolved by omitting the language tag from all foreign language documents and encoding them in the title of the document.

Conclusion

A workflow that performed all of the actions specified in the requirements for integrating patient education documents in EPIC was created and tested with the patient education staff. Detailed documentation was written for the workflow to guide users and standardize the procedure. The computer program was an effective component of the workflow because it allowed users to complete the majority of the workflow while requiring little technical knowledge. It also increased the efficiency of employees tasked with integrating patient education materials into EPIC. As of the end of this project, several documents have been converted and tested in EPIC using this workflow. However, they have not been tested in a clinical setting yet.

Future Work

The next steps for this project are to convert all of the patient education documents pertaining to a specific disease and test them in EPIC in a clinical setting. This would mean having health professionals access patient education materials in EPIC and giving materials to real patients. Currently, patient education documents are created as both Microsoft Word documents and Adobe InDesign documents. Research is needed to include the conversion of Adobe InDesign documents into the workflow so that all patient education documents can be integrated into EPIC.

Acknowledgments

I would like to thank my mentor Kristine Petre for guiding me in this project, familiarizing me with the library infrastructure, teaching me about MARC records, and assisting me in writing this article and the poster. I am grateful for the cooperation and leadership of Linda Schwartz on the creation of the workflow, the metadata, correspondence with other librarians, and EPIC testing. Special thanks to Laura Harstine and Marie Bartos for their time spent in testing the workflow and providing invaluable feedback. Finally, thank you to Matthew Vonfrisch for uploading our test documents to EPIC and troubleshooting EPIC issues.

References

Bartczak, Jeremy (May 2015). *University of Virginia metadata scheme for patient education materials in EPIC*. University of Virginia.

Chowdhury, G. G., & Chowdhury, S. (2007). *Organizing information: from the shelf to the Web.* London: Facet.

Dreeben, O. (2010). *Patient Education in Rehabilitation.* Sudbury, Massachusetts: Jones and Bartlett Publishers.

Nielsen, J. (1994). *Usability Engineering.* San Francisco, California: Morgan Kaufmann Publishers.

Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (3rd Edition ed.). Reading, Massachussetts: Addison Wesley Longman.

Volk, Ruti (May 2015). *University of Michigan metadata scheme for patient education materials in EPIC*. University of Michigan.