

Electronic medical textbook generator

Assadaporn Nuchprayoon*

Issarang Nuchprayoon** Chanisa Thananuwat*

Nuchprayoon A, Nuchprayoon I, Thananuwat C. Electronic medical textbook generator. Chula Med J 2000 Apr; 44(4): 253 - 66

Background : *Medical textbooks are important educational resources for medical students and physicians. Medical texts are often cross-referenced, and not easy to search. Computer technology using "hypertext" links allows easy access to the cross-referenced topics. Electronic textbooks are increasingly available as a substitute for heavy multi-volume textbooks. Despite its usefulness, electronic textbook editing is a laborious task.*

Objective : *To develop an electronic medical textbook based on Hypertext links*

Material and Methods : *We created a new software program that automatically converts an electronic document into HTML (hypertext mark-up language) documents which can be viewed by a web browser, thereby generating an electronic medical textbook from a simple textbook. Two chapters on pediatric hematology were used as a model. The efficiency of the generator program was tested against the manual editing process in creating an electronic textbook.*

Results : *This software allows a textbook author to produce an electronic hypertext-linked textbook without the tedious task of coding words to be linked.*

* Department of Statistics, Faculty of Commerce and Accountancy, Chulalongkorn University

**Department of Pediatrics, Faculty of Medicine, Chulalongkorn University

Conclusion : *This electronic medical textbook generator would be helpful for textbook editors to generate an electronic "hypertextbook" in a short period of time.*

Key words : *Electronic textbooks, HTML, Hypertextbook.*

Reprint request : Nuchprayoon A, Department of Statistics, Faculty of Commerce and Accountancy, Chulalongkorn University, Bangkok 10330 Thailand.

Received for publication. August 15,1999.

อัญญาพร นุชประยูร, อิศรางค์ นุชประยูร, ชนิศา ฐานานุวัฒน์. โปรแกรมสำเร็จรูปเพื่อสร้างตำราแพทย์อิเล็กทรอนิกส์. จุฬาลงกรณ์เวชสาร 2543 เม.ย; 44(4): 253 - 66

ความเป็นมา : ตำราเป็นองค์ประกอบสำคัญของทั้งนักศึกษาแพทย์และแพทย์ประจำบ้านในการศึกษาวิชาการแพทย์ ตำราทางการแพทย์นั้น มักจะมีการอ้างอิงระหว่างคำหรือหัวข้อในบทต่าง ๆ กัน ทำให้สืบค้นได้ลำบาก เทคโนโลยีทางคอมพิวเตอร์ที่เรียกว่า ไฮเปอร์เทกซ์ ช่วยให้การค้นหาความรู้ที่มีการอ้างอิงถึงกันเป็นไปได้โดยง่าย ในปัจจุบันตำราอิเล็กทรอนิกส์นี้ได้เริ่มเข้ามาทดแทนตำราแพทย์ที่มีเนื้อหาหนักได้เป็นอย่างดี แต่การผลิตตำราอิเล็กทรอนิกส์นั้น สิ้นเปลืองแรงงานและเวลาในการทำดังนี้

วัตถุประสงค์ : เพื่อผลิตสื่อการสอนที่เป็นตำราแพทย์อิเล็กทรอนิกส์ในลักษณะของ HTML

วิธีวิจัย : ผู้วิจัยได้พัฒนาโปรแกรมที่สามารถเปลี่ยนตำราต้นฉบับ ซึ่งจัดทำโดยใช้โปรแกรมประมวลผลคำ ให้เป็นตำราอิเล็กทรอนิกส์ ที่มีการเชื่อมโยงกันในเนื้อหาของตำราในรูปของเอกสาร HTML ซึ่งสามารถเปิดตำรานี้จาก Web browser อันได้แก่ Netscape Navigator เป็นต้น ผู้วิจัยได้นำบทเรียนในเรื่องโลหิตวิทยาในเด็กมาเป็นตัวแบบในการสร้างตำราอิเล็กทรอนิกส์ และได้มีการทดลองใช้เพื่อเปรียบเทียบกันระหว่างการสร้างตำราอิเล็กทรอนิกส์ด้วยมือและการใช้โปรแกรมใหม่ที่พัฒนาขึ้น

ผลการศึกษา : โปรแกรมสำเร็จรูปที่พัฒนาขึ้นช่วยให้เจ้าของตำราสามารถสร้างตำราอิเล็กทรอนิกส์โดยไม่ต้องอ่านและทำการแก้ไขเอกสารด้วยตนเองซึ่งจะทำให้สะดวกและประหยัดเวลาได้อย่างมาก

สรุป : โปรแกรมสำเร็จรูปเพื่อสร้างตำราแพทย์อิเล็กทรอนิกส์นั้นจะเป็นประโยชน์มากสำหรับผู้เขียนตำราและเหมาะสมในการสร้างตำราแพทย์อิเล็กทรอนิกส์ที่มีการเชื่อมโยงความสัมพันธ์ระหว่างเนื้อหาต่าง ๆ ในตำราได้อย่างง่ายดาย

Textbooks are indispensable tools for medical education. Textbooks contain established medical facts, written for physicians and students, to comprehensively review a medical subject. In addition to text, they also contain illustrations such as figures, diagrams, tables of contents, and indices.

With continuous progress in medicine, medical textbooks today have become considerably larger in content. Many textbooks have attained large size with small print and multi-volumes. The search for a specific medical term or answer has become less convenient due to the mass of material.

The advance of computer technology has allowed compaction of large amounts of information into small-sized computer disks or compact disks (CD-ROM). These so-called 'electronic textbooks' allow colorful presentation of the content and are considerably easier to search for a specific term than by using the index of the classic textbook. When the readers come across specific terms, they may be able to jump to the details of the term or topic immediately through a process called a 'hypertext' link. This 'hypertext' cross-referencing feature was never possible in the classic paper textbook. In addition, printing a specific text of interest from an electronic version is possible at any time.

The conversion of an electronic textbook from an existing paper textbook is a laborious and time consuming process. The editor must have certain computer expertise, and usually must painstakingly read through each paragraph for key words used to establish links to other chapters using 'Hypertext Mark-up Language', or HTML. Most authors of medical textbooks do not have the ability or time to spend on this task. We report here an automation of the process

that generated an HTML-linked electronic textbook from a plain medical textbook. This automation tool is called an electronic medical textbook generator.

Materials and Methods

An Intel[®] Pentium 166 MMX personal computer PC with 32 MB of RAM and a 2.1 GB hard disk drive was used to develop the electronic medical textbook generator. There were two phases in the development of this program. The first phase was to develop a prototype of the desired electronic textbook, and the second was to develop a functional electronic medical textbook generator.

The prototype of an electronic medical textbook was developed using general-purpose application software such as Microsoft[®] Word (MS-Word), Notepad, and Microsoft[®] Internet Explorer (IE). A set of pre-typed textbook chapters was edited and saved into HTML format using MS-Word 97. Additional HTML tags were added to provide links at the keywords in the documents. The prototype was tested on IE.

After the prototype was completed, the concept was transferred into the development of a working generator. This second phase involved a series of software tools, which included:

1. Microsoft[®] Word 97 (MS-Word 97)

This word processor was used to create a document template for the content material. This document template contained a set of macros to provide functions such as to save the document into an HTML format, to search for a particular index key word in file, etc.

2. Microsoft[®] Excel 97 (MS-Excel 97)

This spreadsheet program was used during

the automatic process of creating indices for the chapters. It kept records of the chapters and file names of all of the chapters. MS-Excel 97 was used to create a spreadsheet file named `table1.xls` containing indices and filenames of each document. This file contained 8 sheets of data, one for main index, five sheets for four different types of document, and two sheets for subtopics of 'symptom' documents and 'disease' documents.

3. Microsoft® Visual Basic for Application (VBA)

This software development tool helped in generating an application on top of Microsoft® Office Suite (MS-Office). Macros created in MS-Office generated VBA codes. For example, we used VBA to automatically save the material contents into HTML format in MS-Word 97, to add an index of the material content in MS-Excel 97 and create Hypertext links to other material content in MS-Word 97, and to scan for

a key word appearing in the material contents in MS - Word, etc.

4. Microsoft® Visual Basic 5.0 (VB5)

This software development tool for Windows applications was used to create the entry point to the generator and link all VBA codes together. VB5 can manage documents in MS-Word 97 and MS-Excel 97 through the Object Linking and Embedding (OLE) technology.

5. JavaScript™

JavaScript™ is a set of instructions (script) added to a document in HTML format. A script was attached to each figure to instruct web browsers to open a new window. This allowed viewing a figure in the resulting electronic medical textbook.

6. InstallShield® 5.0 Professional

InstallShield® was used as to make an installer diskette set after the development was completed.

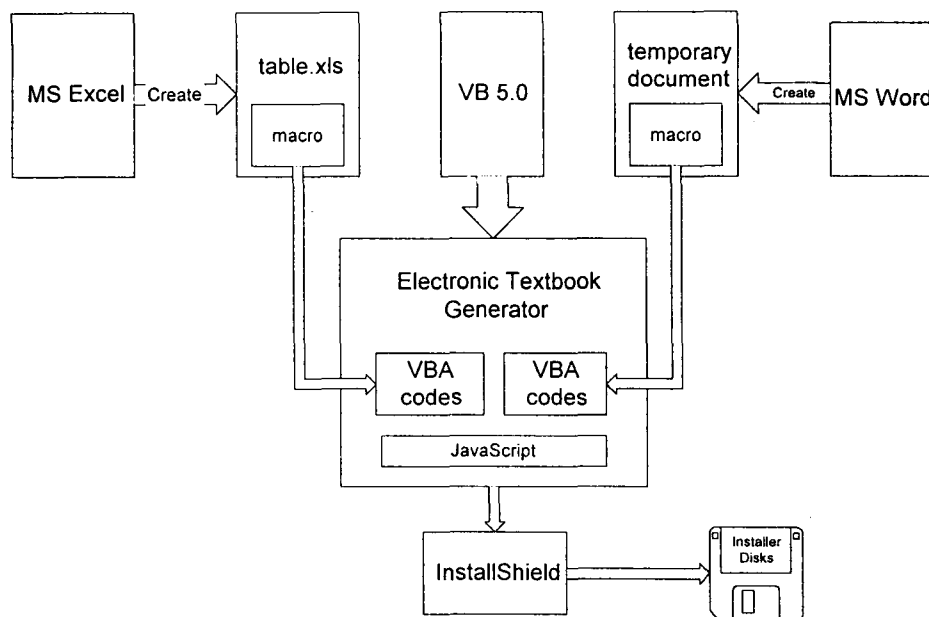


Figure 1. Diagram of the development of electronic medical textbook generator.

The relationship of each tool is summarized in figure 1. First, macros were recorded in a temporary MS-Word 97 document and in an MS-Excel file named *table1.xls*. These macros automatically generated VBA codes. Then some VBA codes were edited and added. VB5 was used to prepare a control module for all the codes, including JavaScript™.

The program was tested with a hypothesis that the generator program reduced the author's time in creating an electronic textbook. Two topics in pediatrics, anemia and bleeding disorders,⁽¹⁻³⁾ were used in this test. Subjects were six third-year students in Statistics who were familiar with MS-Word 97 and HTML. They were given several chapters forming the 2 topics in pediatrics. Each chapter was a file with the name of the disease or the symptom. Each of the subjects selected any six chapters consisting of at least one symptom to form an electronic textbook for this test. Time to complete a task and the number of interactions were recorded as parameters of the generator's efficiency. Time was measured in minutes for each chapter which the subjects worked on. The number of keystrokes and mouse clicks was counted to represent the number of interactions.

Subjects were divided into two groups. The first three subjects were asked to create an electronic medical textbook from the files given manually. They first created a new notepad document as a table of contents with HTML code and then the name of the chapter (either a symptom or a disease) must be linked to the file name of the chapter. For each chapter, they had to find the keywords, which were the disease names and the symptom names, and wrote a set of HTML code to create links from

those keywords to the matching file names. The other three subjects were asked to use the generator program we developed to create an electronic textbook. For each chapter of the electronic textbooks that both groups of subjects created, an observer marked the time and counted the number of interactions each subject made.

Results

The outcome of our research project is a CD-ROM containing an electronic medical textbook generator ready to be installed. It requires MS-Windows 95, and at least 10 MB of hard disk space for installation. It also requires MS-Office to make use of the program.

The development process of the program was the design of input, process, and output of the generator. The program testing results were expected to be very helpful to the authors of the textbooks.

Input

The electronic medical textbook generator required textual input material prepared electronically. There were four main types of input materials, three of which were textual materials and one was pictorial.

1. Symptom documents

This textual document contained the material about each symptom. It must be prepared by a word processor and named after the symptom with the file extension *.doc*. The title of the chapter must be in both English and Thai. The subtopics for each symptom document must be in a specific format, for example:

บทนำ
แนวทางการวินิจฉัยเด็ก
ประวัติที่สำคัญ
การตรวจร่างกายที่สำคัญ
การตรวจทางห้องปฏิบัติการ
แนวทางการดูแลผู้ป่วย
การรักษา
โรคที่อาจมีลักษณะอาการดังกล่าว
เอกสารอ่านประกอบ

The subtopics are used as a linking point from the table of content of the electronic textbook. Some subtopics could be omitted but subtopics had to appear in order. The authors could modify this subtopic format.

2. Disease documents

These documents contained the material about each disease. They had to be filed with the disease name and the extension *.doc*. The subtopics of the disease documents had to be specifically formatted, for example:

สิ่งสำคัญในการวินิจฉัยโรค
ข้อคิดเห็นทั่วไป
สิ่งที่ตรวจพบทางคลินิก
การตรวจทางห้องปฏิบัติการ
การวินิจฉัยแยกโรค
ภาวะแทรกซ้อน
การรักษา
การป้องกัน
การพยากรณ์โรค

The subtopics for the disease documents are used and maintained in the same way as those for the symptom documents.

3. Pictorial documents

This document contained a figure illustration. The pictures could be prepared by any graphic

software and saved in either GIF, BMP, TIF, or JPEG format. Figure file name must start with a number. The reference to the figure file, in both the symptom and disease documents, and must start with the key word "รูปที่".

4. Exercise documents

There were two types of exercises: practice cases and multiple-choice questions. For practice case exercises, each file contained information about a patient case to be diagnosed. A practice case document also had to be prepared by a word processor but in the format of a two column table. The left column stated the case information categories while the right column stated information value. The answers to the practice cases had to be provided in the last row of the table with the keyword *Diagnosis* in the left column. The author could give any name with the extension *.doc* to the file.

The multiple-choice document contained a set of multiple-choice questions pertaining to one symptom. For each question, an answer could be provided at the end of the question by use of the keyword *เฉลย*.

Functions

The available functions of the Electronic Medical Textbook Generator are to add a chapter, delete a chapter, update a chapter, and to edit the subtopic listings of the chapter format.

1. Adding a chapter

In order to add a chapter, the user must define the type of the chapter, along with the file name. The addition process could not be accomplished if there was already a chapter of the same name. Besides adding the file name to the index sheet, chapters must

be scanned for the diseases in order to add HTML codes for cross-referencing.

2. Deleting a chapter

The file name of the chapter to be deleted must be specified. If the file did not exist, no chapter was deleted. The deleting process included removing the file name from the corresponding index sheet and scanning chapters for any links to the deleted chapter.

3. Updating a chapter

Updating a chapter is a process to replace an old version of a chapter with a new one under the same name. If the file did not exist, there would be no replacement. This process is similar to the addition process.

4. Maintaining subtopics

Subtopics of symptom chapters and subtopics for disease chapters are maintained in MS-Excel sheets. The author is allowed to change these at will. The author could change the sequence of the subtopics as they appeared in the chapter, as well as the wording.

The first three services could be done by specifying the names of chapters to be added, deleted or updated. These three services could be activated on the same user interface screen as shown in the appendix. However, the fourth service could be accessed through a dropdown menu list. While using the generator, the author may need to recall which documents are already in the database. This service is also provided through a menu item.

Output

The output of the generator are documents and pictorial files in HTML format so that they could be viewed on a web browser such as Netscape

Navigator or IE. These files are linked together through the symptom names and the disease names, in both Thai and English. A snapshot of a running screen of the Electronic Textbook is illustrated in the appendix.

Each screen is divided into three areas: a title area, a table of contents area, and a content area. The title area is displayed at the top of the screen, displaying the name of the program. The table of contents area is the left frame of the screen and shows topics that a user can view. Each topic is a symptom followed by a set of related diseases. The content area shows the contents according to the user selection. The content area is the largest area and is in the lower right corner. In this area, the content of each chapter can be viewed. When a chapter is selected, the program lists all the subtopics of the chapter and then contents of the chapter.

Using the Electronic Textbook is the same as using a web browser to view web pages. Each page represents a chapter, either a symptom or a disease. For each chapter selected, a list of subtopics is displayed in the contents area. The users could use several options for study:

1. Studying from the beginning

The users could study the material in the textbook as if they were reading from the first page. Users could start by selecting the first chapter from the table of contents, which then revealed subtopics of the selected chapter in the contents area. By clicking the first selection, *เริ่มอ่านเนื้อหาตั้งแต่ต้น*, the user is able to study the chapter from the beginning. The user can switch to the next chapter by selecting that next chapter in the table of contents area.

2. Studying the symptoms

The users can select a symptom to study the diseases associated with it. This can be done by clicking on the symptom of interest from the table of contents. While studying a symptom, the user can jump to study a disease by clicking at a particular disease name appearing in the symptom contents. With the *back button* (of any web browser), the user can jump back to where he left off in the symptom document.

3. Studying a particular disease

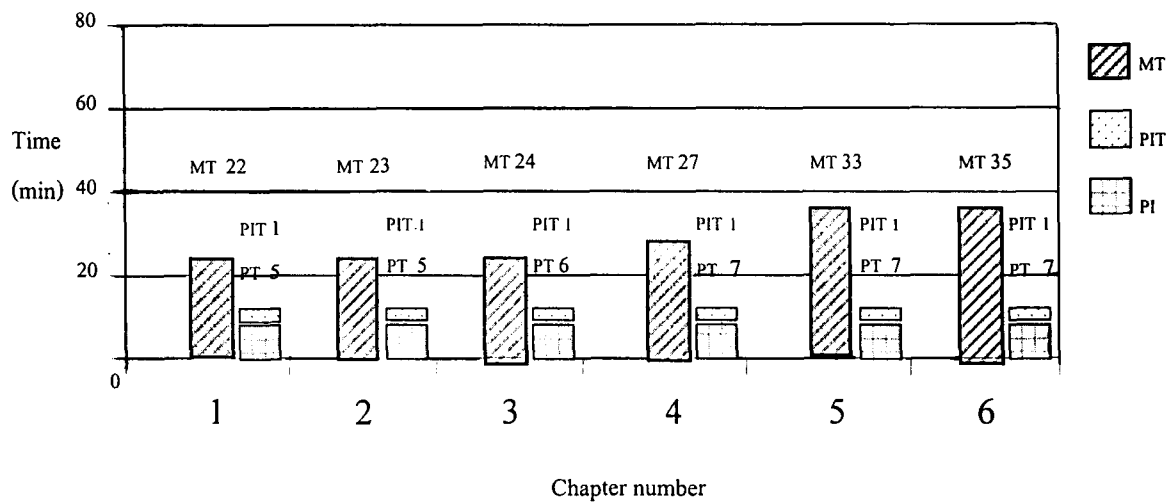
The users can select a particular disease from the table of contents. If there were related diseases or pictures in the contents, users could jump to those documents.

4. Exercises

Two types of exercises are provided: practice case diagnosis and multiple choice questions. These exercises can be accessed from the end of each symptom document. The questions are displayed without answers until the user clicks at the answer key word.

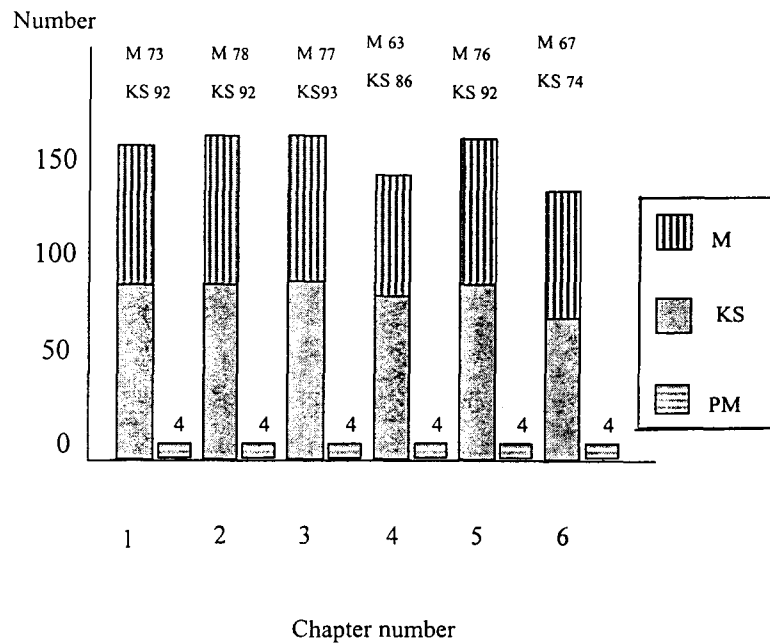
Testing the generator

The results of having six subjects create an electronic pediatric textbook consisting of six chapters is shown in *Figures 2 and 3*. Figure 2 presents the average time the subjects used to create each chapter by comparing between the manual method and using the generator we developed.



Note: MT (Manual Time) refers to the average of the overall time (which is also the interactive time) in minutes the users used in creating the chapter by hand.
 PT (Program Time) refers to the average of the overall time in minutes the users used in creating the chapter using the program.
 PIT (Program Interactive Time) refers to the average time in minutes the users used to interact with the system in creating the chapter using the program.

Figure 2. Average time the subjects used to create each chapter.



Note: KS (Keystroke) refers to the number of keystrokes the users made in creating the chapter manually.
M (Mouse click) refers to the number of mouse clicks the users made in creating the chapter manually.
PM (Program mouse) refers to the number of mouse clicks the users made in creating the chapter using the program.

Figure 3. Average number of interactions the subject used to create each chapter.

Figure 3 compares the number of interactions, in terms of the number of keystrokes and the number of mouse clicks, in creating each chapter in the electronic pediatric textbook manually against the use of the generator.

It is clear that the electronic medical textbook generator helps authors save a tremendous amount of time by using fewer keystrokes. Using the generator, it took less than ten minutes to create each chapter in the electronic pediatric textbook. During this time, the users had to actively interact with the computer for only one minute at an average then wait for the rest. In contrast, manually creating a chapter in an electronic textbook took more than 20 minutes in which the users had to interact with the computer continuously.

The number of interactions a user made can be concluded similarly. A user had to strike the keyboard more than 90 times and click the mouse more than 60 times in order to construct a chapter manually. It consistently took the user only 4 mouse clicks to add a chapter into the electronic textbook.

Fewer keystrokes and mouse clicks equates to less time consumption and also fewer typing errors. We conclude that using the generator will help authors save a lot of time and make fewer errors.

Discussion

Hardware Selection

The goal of the generator was to assist authors in creating an electronic textbook. The generator must

run on moderate specification hardware. We believe that a Pentium processor of 166 MHz with 16 MB RAM and at least 10 MB in hard disk will be generally suitable and these specifications are found in the average PC. We did not use specialized Computer Aided Instruction (CAI) software because it may require higher hardware specifications.

Software Selection

We first thought of an electronic medical textbook as being a CAI process. There are several types of CAI. Drill and Practice CAI provides questions to be practiced for the learners. Simulation CAI influences the learners to solve a problem through cases. CAI in testing style concentrates the learners on completing a test and learning about how they did. ⁽⁴⁾ However, an electronic textbook is not exactly the same since the textbook is more like a reference book where the topics are highly cross-referenced.

Therefore, we believe that Hypertext links are the most suitable way for readers to the contents of an electronic textbook. The hypertext concept is well established in the World Wide Web (WWW) and HTML is a standard language for most web Browsers,⁽⁵⁾ Electronic textbooks in HTML format allow users who are familiar with that current technology to efficiently use it.

The input material forms the contents of the textbook. Nowadays, the electronic format of a textbook created with word processors is commonly used by authors to prepare the manuscript of the textbook. In this particular case, our objective was to help authors and editors who already had their textbook content in electronic form. Thus, our program requires that the content be already formatted by word processor.

We chose a combination of software from MS-Office because of their ability to work together. We chose MS-Word 97 for the main working space due to its special function to save a normal word document into an HTML document, ready to be browsed through a web browser. MS-Excel 97 was the most convenient way to create a table matching the disease and symptoms to the actual file name, and could be coded to automatically insert a new chapter to the table of contents as well as to delete unwanted chapters. VB5 was chosen because it was the super set of the VBA codes generated from macros in MS-Office. The VB5 codes allows the author to enter the generator program by double - clicking on the program icon the same way as running other normal Windows programs.

In order to add a figure document in a pop-up window, a scripting code was needed. JavaScript™ was chosen. This scripting code was hidden inside the VB Code and written to the HTML document where the keyword to the figure was located. Other scripting languages such as VBScript can also be used.

The whole generator program could alternatively using a software development tool such as MS Visual Basic. We chose to use a combination of software to develop this generator because of the advantage of the object-oriented and reuse-oriented concept⁽⁶⁾, less development time, and fewer hidden errors (bugs) in the program. Moreover, MS-Office is widely available and can easily be customized .

Categories of documents

Using a pediatric textbook as a prototype, we found that chapters could be categorized into either symptoms or diseases. These two have a close relationship and when a student learns about a

symptom, he needs to know a differential diagnosis of diseases that might cause that symptom, or when one learns about a disease, he needs to know the details of the symptom and other similar diseases. Therefore, symptoms and diseases are the main types of chapters provide by the author. The table of contents is organized by symptoms. Each symptom is listed with related diseases. If there is a disease related to more than one symptom, it appears (in the table of contents) under all of the symptoms that it is related to.

Figures are important learning tools in medicine. When figures are attached to the text, graphics make document files considerably large and could slow the editing process. Therefore, we designed the program to accept figures as separate files. These files are then linked to the text through the 'figure number' in hypertext format.

Review questions are also important learning tools. One suitable type of questions for a medical textbook is practice case diagnosis. We designed each case to be presented to the user with a patient profile until a click at the keyword *Diagnosis* when an answer box pops up. Another type of review is multiple choice questions. Similarly, the correct answer is presented to the user for each question when the keyword **เฉลย** is clicked. While the author has to prepare only one practice case in a file, all multiple choice questions pertaining to one symptom must be gathered in the same file. When adding these review-question documents in the process, the author must specify which symptom topic they belong to.

Limitations of the program

When an author decides to create an electronic textbook, he has to provide the manuscript in chapters

of symptoms and diseases. No other category of chapters is allowed. Moreover, the chapters must be added into the electronic textbook one by one. The author has to wait until a chapter is completely added before adding another chapter.

The major advantage of this program is that the author does not have to provide a list of keywords for each chapter added into the electronic textbook being created. The program will search for and create an index from the symptoms and disease names it finds. However, the author cannot provide other keywords that might also be relevant for searching.

As shown in figure 2, the time used in creating chapters increases as more chapters are added to the electronic textbook. Due to the automatic process in scanning for disease names as keywords in existing chapters, the program requires more time as the number of chapters in the electronic textbook increases. This may result in a lengthy waiting time and the user may not know whether the program is still operating properly or if it is hung although the progress bar at the status bar is presented.

Conclusions

We proposed a prototype for an electronic medical textbook generator. This prototype helps an author to create an electronic textbook from a manuscript created on a word processor. A major advantage of using this program in creating an electronic textbook is to eliminate the laborious and tedious task of looking up keywords and thus save a lot of time. The electronic textbook generated by this program assists the reader in a self-directed learning process in tutorial style by providing Hypertext links among the chapters.

Appendix

1.

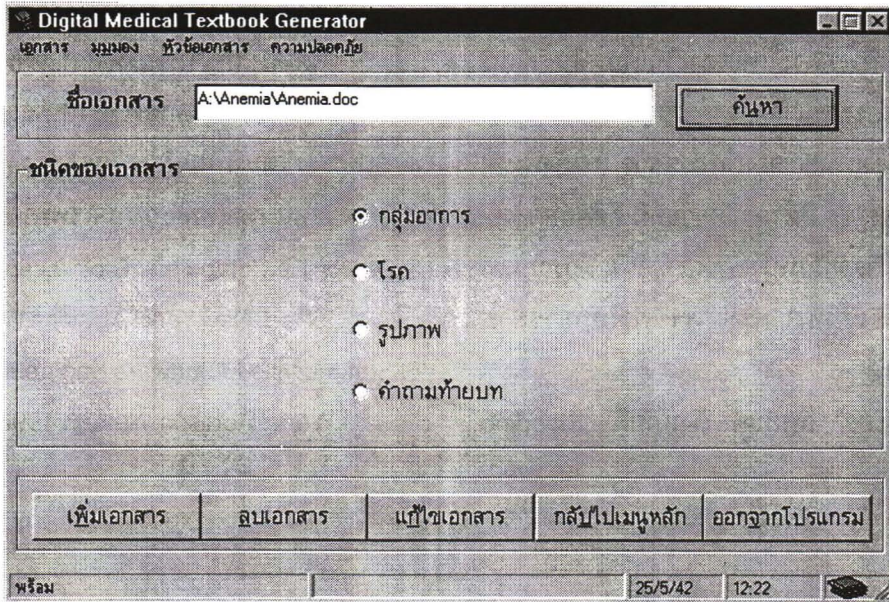


Figure 4. The main screen of the electronic medical textbook generator.

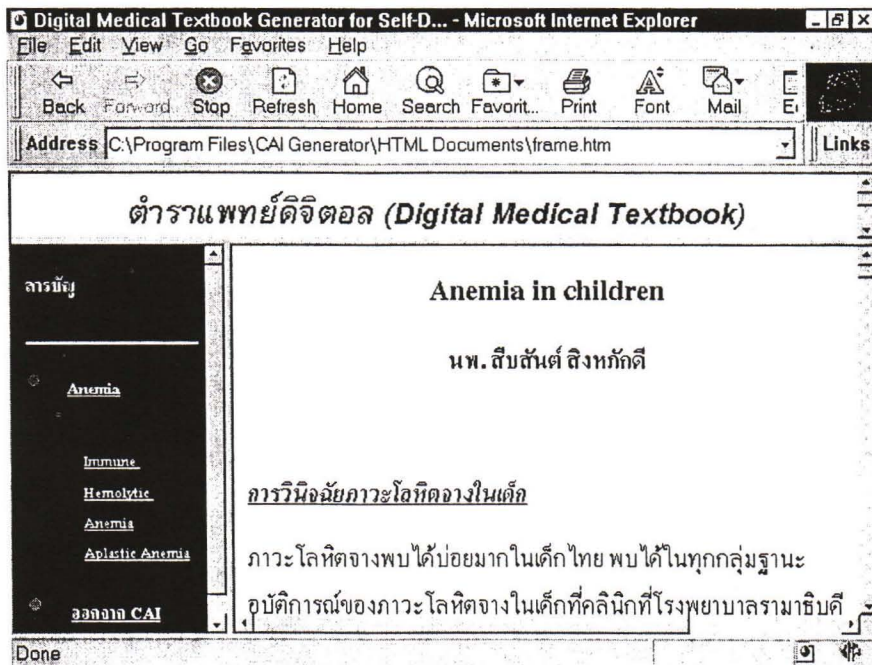


Figure 5. A snapshot of an electronic textbook.

Acknowledgements

We would like to thank Montra Katedara, Panita Phumithavorn, Suwimon Sangtubtimsang, and Wanwisa Metongmool for their assistance in setting up the test and generating all of the graphics in this

report. This research was funded by ทุนอุดหนุนโครงการการเรียนการสอนเพื่อเสริมประสบการณ์ของกองบริการการศึกษา, จุฬาลงกรณ์มหาวิทยาลัย และทุนพัฒนาอาจารย์ใหม่/นักวิจัยใหม่ของกองส่งเสริมและประสานงานวิจัย, จุฬาลงกรณ์มหาวิทยาลัย

References

1. สืบสันต์ สิงห์ภักดี. ภาวะโลหิตจางในเด็ก. ใน: สำหรับจิตตินันท์, บรรณาธิการ. ตำรากุมารเวชศาสตร์. เล่มที่ 3. กรุงเทพฯ : กรุงเทพเวชสาร, (กำลังพิมพ์)
2. อิศรางค์ นุชประยูร, ชุติ มิตรกุล. ภาวะที่มีเลือดออกง่าย. ใน: สำหรับ จิตตินันท์, บรรณาธิการ. ตำรากุมารเวชศาสตร์. เล่มที่ 3. กรุงเทพฯ : กรุงเทพเวชสาร, (กำลังพิมพ์)
3. อิศรางค์ นุชประยูร, สืบสันต์ สิงห์ภักดี. Bleeding disorder from coagulation defects. ใน: สำหรับจิตตินันท์, บรรณาธิการ. ตำรากุมารเวชศาสตร์. เล่มที่ 3. กรุงเทพฯ : กรุงเทพเวชสาร, (กำลังพิมพ์)
4. ถนอมพร (ตันพิพัฒน์) เลานจรัสแสง. คอมพิวเตอร์ช่วยสอน. กรุงเทพฯ: วงกลม โปรดักชั่น, 2540: 71-126
5. Capron HL. Computers: Tools for an Information Age, 5th ed. New York : Addison Wesley. 1998: 208
6. Sommerville I. Software Engineering. 5th ed. New York : Addison Wesley, 1996: 396 - 7