

ĐÓNG GÓI BÀI HỌC E-LEARNING

HÀ VIỆT HẢI*

TÓM TẮT

Sử dụng E-learning là một trong những xu hướng chủ đạo của giáo dục trong thời kỳ mới. Các bài giảng E-learning có thể được biên soạn ban đầu ở nhiều dạng khác nhau như văn bản tĩnh, trình diễn, trang web. Tất cả chúng đều cần được chuyển đổi về những dạng được hỗ trợ bởi các hệ quản lý học tập (Learning Management System – LMS) để có thể triển khai rộng rãi đến người học. Bài viết này trình bày cách chuyển đổi cho các bài học ở những định dạng ban đầu khác nhau và giải thuật đóng gói bài giảng ở dạng website thành gói theo chuẩn SCORM là dạng chuẩn được sử dụng nhiều nhất hiện nay trên các LMS.

Từ khóa: bài học E-learning, chuẩn SCORM, đóng gói SCORM, đóng gói E-learning.

ABSTRACT

Packaging E-learning resources

Using E-learning is one of the major trends in education nowadays. E-learning resources can be composed initially in different forms such as static text, multimedia presentations, or websites, all of which need to be converted to supported formats in Learning Management System (LMS) so as to be widely used by learners. The article presents the procedures in converting different original lessons and the algorithm for packaging lessons in the form of websites into SCORM, which is the most commonly used standard in LMS.

Keywords: E-learning resources, Scorm standard, packaging Scorm, packaging E-learning resources.

1. Mở đầu

Kinh tế tri thức đang phát triển như vũ bão. Lượng tri thức khổng lồ của nhân loại tăng và được cập nhật nhanh chưa từng có, đòi hỏi sự thay đổi tương ứng về giáo dục và đào tạo. Các phương thức dạy học truyền thống mặt đối mặt (face to face) và sử dụng tài liệu tĩnh (sách, tài liệu giấy) không còn đáp ứng được nhu cầu cập nhật nhanh và triển khai rộng được nữa. Những phương thức này cũng không đáp ứng được cho nhu cầu học tập ở mọi nơi và học tập suốt đời. E-learning, với khả năng cho phép học tập ở mọi nơi, mọi lúc, tiết kiệm chi phí, tiết kiệm thời gian, uyển chuyển và linh động, dễ cập nhật nội dung, dễ mở rộng số người học đã nhanh chóng được phát triển và ứng dụng rộng rãi ở nước ngoài. Ở Việt Nam, E-learning cũng đã bắt đầu được sử dụng bởi nhiều cơ sở đào tạo, phục vụ cho việc học chính khóa cũng như ngoại khóa.

* TS, Trường Đại học Sư phạm, Đại học Huế; Email: haviethaivn@yahoo.com

Các bài học E-learning có thể được soạn thảo theo nhiều dạng, từ đơn giản là các văn bản tĩnh đến các đối tượng động phức tạp như: video clip, trình diễn và trang web có sử dụng các đối tượng multimedia, hoặc phức tạp hơn nữa là các dạng tài liệu được thiết kế riêng với khả năng tương tác mạnh. Cách thức phân phối cũng rất khác nhau, từ các chuyên giao trực tiếp qua file và người học sử dụng trực tiếp trên máy tính cục bộ đến việc triển khai rộng hơn thông qua hình thức chia sẻ tài nguyên trên mạng cục bộ, và đến cách triển khai rộng khắp nhất là sử dụng Internet. Cùng với sự mở rộng của phạm vi triển khai, nhu cầu quản lý các khóa học, người học cũng tăng lên. Cả hai mặt này đòi hỏi phải có các nền tảng hạ tầng mạnh để phục vụ cho người học cũng như nơi tổ chức dạy học. Rất nhiều tổ chức đã phát triển những hệ thống để đáp ứng nhu cầu này, và nhìn chung, đại đa số các hệ thống đều dựa trên nền tảng web. Xây dựng, duy trì và phát triển một hệ thống như thế rất tốn kém cả về thời gian, tiền bạc và nhân lực. Vì vậy, sử dụng một hệ thống mã nguồn mở, miễn phí đã trở thành giải pháp được nhiều tổ chức chọn lựa, trong đó nổi bật lên là sử dụng Moodle. Với đa số các tổ chức, Moodle đã đáp ứng được tất cả các nhu cầu để triển khai, quản lý, khai thác các khóa học, người học và người dạy. Đối với những yêu cầu đặc biệt mà Moodle chưa cung cấp sẵn, tổ chức có thể mua các gói mở rộng hoặc tự mình hay thuê những công ti phần mềm khác lập trình để đáp ứng.

2. Các dạng chuẩn đóng gói bài học E-learning

Khi sử dụng Moodle nói riêng hay đa số các LMS nói chung, có hai cách chính để tạo các bài học E-learning. Cách thứ nhất là sử dụng các công cụ soạn thảo được tích hợp sẵn trên các LMS để soạn thảo trực tiếp bài học ở trên đó. Cách này đơn giản nhưng không đủ mạnh để đáp ứng đầy đủ các yêu cầu về định dạng văn bản cũng như yêu cầu về các đối tượng multimedia trong bài học. Cách thứ hai là sử dụng các công cụ soạn thảo độc lập bên ngoài để tạo bài giảng, sau đó chuyển đổi, đóng gói thành dạng được hỗ trợ bởi các LMS và tải lên trên các hệ này. Cách thứ hai khá thuận tiện cho người sử dụng bởi vì họ có thể dùng các công cụ soạn thảo mạnh mẽ mà mình quen thuộc. Những công cụ này cũng thường hỗ trợ nhiều định dạng và đối tượng multimedia khác nhau, đáp ứng cho nhu cầu đa dạng của bài học. Tuy nhiên, trong đa số trường hợp, các bài giảng được biên soạn theo cách thứ hai đều có định dạng nguyên thủy không được hỗ trợ bởi các LMS, do đó không thể được hiển thị và hoạt động trực tiếp trên đó. Do vậy, chúng cần được chuyển đổi định dạng để có thể được đưa lên các LMS. Từ đây về sau của bài báo chỉ đề cập đến cách soạn thảo bài học theo cách thứ hai.

Về vấn đề định dạng của tài liệu trên các LMS, trước hết cần nhắc lại một đặc điểm chung của đại đa số các hệ này là vận hành trên nền tảng web. Vì vậy, các dạng tài liệu được hỗ trợ trên web đều có thể được đưa lên và sử dụng trên chúng. Tuy nhiên, ngoài việc cung cấp khả năng vận hành các bài học như các đối tượng web bình thường, các LMS còn sử dụng và tạo nhiều thông tin hệ thống khác về bài học, khóa học, tiến trình học... Mặt khác, với những bài học có cấu trúc phức tạp như ở dạng các trang web hoặc trình diễn chứa trên nhiều file và có các đối tượng multimedia thì việc

tổ chức lưu trữ, kết nối và triển khai bài học cần phải tuân thủ các quy tắc chặt chẽ để tránh các lỗi về sau. Tất cả những điều này dẫn đến sự ra đời của các chuẩn đóng gói bài học E-learning. Các chuẩn này cũng tạo điều kiện cho việc tương thích của bài học trên các LMS cũng như tạo điều kiện thuận lợi cho các nghiên cứu, khai thác dữ liệu E-learning.

Các dạng chuẩn bài học E-learning được khởi tạo từ năm 1997, trong dự án ADL Initiative của Bộ Quốc phòng Mỹ phối hợp với các tổ chức khác gồm IEEE (Institute of Electrical and Electronics Engineers), AICC (Aviation Industry Computer-based Training Committee), IMS Global Learning Consortium... nhằm tạo các tài liệu học tập trên máy tính sao cho có tính năng dễ truy cập, tương thích, bền vững và tái sử dụng. Trong số các dạng chuẩn đưa ra, chuẩn AICC và SCORM được sử dụng rộng rãi nhất. [3]

Chuẩn AICC (Aviation Industry CBT [Computer-Based Training] Committee) [2] được áp dụng cho việc phát triển, phân phối và đánh giá các khóa đào tạo và thường được triển khai thông qua các LMS. AICC là một tổ chức quốc tế chuyên về các vấn đề đào tạo thông qua công nghệ. Một tài nguyên học tập được xem là tương thích với chuẩn AICC (AICC Compliant) khi nó tương thích với ít nhất là 1 trong số 9 hướng dẫn và kiến nghị của AICC (AICC Guidelines & Recommendations (AGR's)). Trường hợp thông dụng nhất là tương thích với AGR-006 (File-based CMI Systems) và AGR-010 (Web-based CMI Systems). Các AGR này xác định các giao tiếp giữa LMS và tài nguyên học tập.

Chuẩn SCORM (Sharable Content Object Reference Model), được định nghĩa [5]: “SCORM là một tập các chuẩn kỹ thuật dành cho các sản phẩm phần mềm E-learning. SCORM chỉ cho người lập trình cách để viết phần mã của họ sao cho nó có “chạy tốt” với các phần mềm E-learning khác”. SCORM được phát triển qua một số giai đoạn và hiện tại các version được sử dụng nhiều nhất là SCORM 1.2, SCORM 2004 (3rd Edition và 4th Edition). Phần sau của bài viết này sẽ tập trung nói về SCORM 2004.

3. Đóng gói bài học E-learning

Đóng gói bài học E-learning theo nghĩa đơn giản là chuyển nó về dạng được hỗ trợ bởi các LMS để có thể triển khai, truy cập và quản lý trong các khóa học E-learning. Về căn bản, các LMS đều dựa trên Web nên đều hỗ trợ các dạng tài nguyên web, bao gồm các trang web (có thể chứa các đối tượng tĩnh và động như audio, video, Flash movie...), tài liệu PDF, XML... Vì vậy, công việc căn bản nhất của việc đóng gói các tài nguyên học tập E-learning là chuyển các tài nguyên này về một trong các dạng nói trên. Ngoài ra, việc đóng gói về các dạng chuẩn còn đòi hỏi thêm về việc tuân thủ cấu trúc tổ chức các file chứa tài nguyên, bổ sung các thông tin hệ thống... Các vấn đề này sẽ được đề cập ở mục III.4.

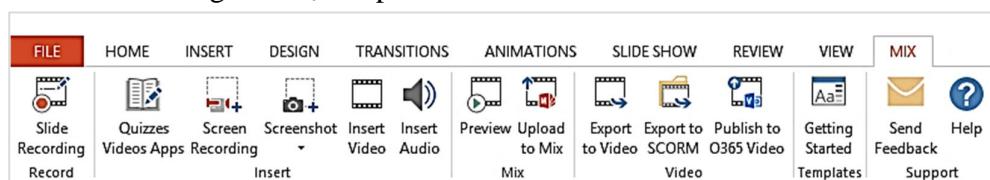
3.1. Các bài học ở định dạng đơn giản

Ở dạng đơn giản, các bài học E-learning có thể được biên soạn theo dạng văn bản thuần chữ (text), văn bản có các đối tượng phức tạp nhưng không có nội dung động. Với dạng này, ngoài cách tạo trực tiếp bằng các công cụ được tích hợp sẵn trên các LMS, các bài học cũng thường được tạo bằng các hệ soạn thảo văn bản thông dụng, chẳng hạn như Microsoft Word, Libre Office Write, trong đó chứa chữ, hình ảnh tĩnh, biểu đồ, công thức... Các trình diễn (được biên soạn bằng Microsoft PowerPoint, Libre Office Impress, Violet,...) chỉ chứa các đối tượng tĩnh cũng có thể được xếp vào dạng đơn giản. Những bài học ở dạng này thì chỉ cần chuyển đổi sang một trong những thể loại được hỗ trợ bởi các LMS như đã nêu ở trên. Điều này có thể được thực hiện bằng các công cụ tích hợp chuyển đổi định dạng trong các hệ soạn thảo văn bản và trình diễn. Riêng với các trình diễn có tính năng động thì có thể dùng các công cụ bên ngoài để chuyển đổi thành các dạng phim, Flash. Vì vậy, không quá khó khăn để việc chuyển đổi các bài học đơn giản về định dạng được hỗ trợ bởi các LMS.

3.2. Các dạng bài học có định dạng phức tạp

Hai định dạng thông dụng của bài học loại này là trình diễn đa phương tiện và website.

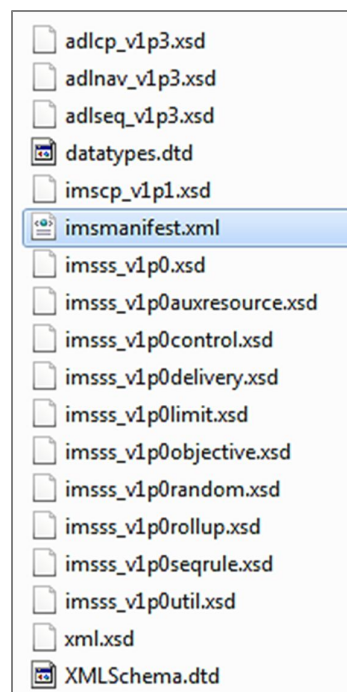
Các trình diễn đều không thể sử dụng một cách trực tiếp trên nền web nói chung và trên các LMS nói riêng, vì thế chúng cần được chuyển đổi về dạng thích hợp. Mặt khác, các trình diễn có sử dụng các đối tượng động như clip âm thanh, phim, Flash thì vấn đề càng phức tạp hơn vì những đối tượng này đều không được lưu trữ tích hợp trong file chứa trình diễn mà ở những file riêng. Vì thế, nếu chỉ chuyển đổi và đưa file trình diễn lên các LMS thì có nguy cơ không sử dụng được các đối tượng động. Tuy nhiên, vấn đề này hiện nay đã được giải quyết được dễ dàng bằng các công cụ tích hợp sẵn hoặc công cụ cài thêm (add in) trong phần mềm soạn thảo trình diễn. Chẳng hạn như Violet đã có sẵn công cụ để đóng gói trình diễn về chuẩn SCORM. PowerPoint thì có thể dùng các công cụ độc lập bên ngoài hoặc các công cụ của các add-in như Adobe Presenter [1], Microsoft Office Mix [4] để thực hiện. Bản chất công việc được thực hiện bởi các công cụ này cũng không quá phức tạp, bao gồm việc chính là chuyển đổi định dạng sang hình thức tương thích với LMS (Flash), thêm một số thông tin hệ thống và đóng gói thành một file nén. Do khuôn khổ có giới hạn của bài viết, chúng tôi không trình bày ở đây chi tiết phần đóng gói cho các trình diễn mà chỉ trình bày phần cho các website trong các mục tiếp theo.



Hình 1. Thanh công cụ của Microsoft Office Mix khi được cài lên PowerPoint

3.3. Kiến trúc cơ bản của một bài học dạng web

Các trang web là tài nguyên đầu tiên và căn bản nhất trên Internet. Ban đầu, các trang web chỉ mang tính chất tĩnh, với nội dung chỉ chứa phần chữ (text) và các hình ảnh tĩnh. Cùng với thời gian và nhu cầu sử dụng, các kỹ thuật ở mức độ khác nhau được phát triển mạnh mẽ trên để cung cấp các tính năng động, tương tác, quản trị cơ sở dữ liệu... Những kỹ thuật ở mức cao cũng đòi hỏi các trang web phải được hỗ trợ bởi các web server có tính năng tương ứng. Đa số các bài học E-learning thì chỉ cần sử dụng các tính năng cơ bản, bao gồm việc trình bày bài học có dữ liệu multimedia (chữ, ảnh, công thức, biểu đồ, âm thanh, phim...) và khả năng trình bày, vận hành, đánh giá bài kiểm tra. Với những yêu cầu này, chỉ cần sử dụng các tính năng cơ bản của công nghệ web như HTML, JavaScript, Flash, HTML5, hay nói cách khác là chỉ cần sử dụng các trang web tĩnh bổ sung thêm khả năng lập trình ở client, là đủ để đáp ứng. Khi chỉ sử dụng các trang web tĩnh, kiến trúc cơ bản của một website nói chung tĩnh nói chung và bài học E-learning nói riêng sẽ bao gồm các trang web được chứa trong các file HTML và các tài nguyên multimedia được chứa trong các file có định dạng tương ứng. Để dễ dàng quản lý và tránh các lỗi đứt liên kết tiềm ẩn, tất cả chúng phải được chứa trong một cây thư mục. Tất nhiên là còn có thể có các tài nguyên bên ngoài được liên kết qua các URL (Uniform Resource Locator), nhưng các tài nguyên này có thể xem là luôn tồn tại độc lập với website đang xét và không cần phải quan tâm khi đóng gói cũng như khi triển khai website.



Hình 2. Các file của chuẩn SCORM 2004 4th

3.4. Kiến trúc của gói SCORM cho một website

Một gói SCORM cho một website nói chung, ở trong ngữ cảnh của bài này là một bài học nói riêng, bao gồm 2 phần chính là các file hệ thống của SCORM và các file chứa bài học, mỗi phần có thể được tổ chức trong một số thư mục khác nhau. Toàn bộ chúng được đóng gói thành một file zip.

Các file hệ thống của SCORM chứa những thông tin hệ thống. Những file này là khác nhau, phụ thuộc vào version của chuẩn SCORM. Người dùng có thể download các file này về tại địa chỉ: <http://scorm.com/scorm-explained/technical-scorm/content-packaging/xml-schema-definition-files/>. Trong Hình 2 là danh sách các file hệ thống của SCORM 2004 4th Edition. Trong danh sách này, tất cả các file đều cố định, ngoại trừ file imsmainfest.xml có nội dung thay đổi để mô tả các tài nguyên của bài học.

Các file chứa các tài nguyên của bài học được đặt vào trong cùng thư mục với các file hệ thống SCORM. Trường hợp có nhiều file thì có thể tổ chức trong một cây thư

mục. Thông thường thì file chứa trang chủ (trang đầu tiên) của bài học được đặt trong cùng thư mục với các file hệ thống SCORM, còn các file khác thì để trong các thư mục con.

Như vậy việc tạo gói SCORM bao gồm các việc:

1. Tạo 1 thư mục gốc để chứa các file thành phần;
2. Tạo (copy) các file hệ thống của SCORM vào thư mục gốc;
3. Tạo (copy) các file của bài học vào thư mục gốc;
4. Biên tập file imsmanifest.xml cho phù hợp với các file của bài học;
5. Đóng gói tất cả các file vào một file zip.

```
<?xml version="1.0"?>
<!-- Manifest template to demonstrate the proper XML namespace declarations for SCORM 2004 4th Edition
manifests that do not use metadata. Provided by Rustici Software - www.scorm.com. -->
<manifest xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
http://www.adlnet.org/xsd/adlcp_v1p3 adlcp_v1p3.xsd
http://www.adlnet.org/xsd/adlseq_v1p3 adlseq_v1p3.xsd
http://www.adlnet.org/xsd/adlnav_v1p3 adlnav_v1p3.xsd
http://www.imsglobal.org/xsd/imsss imsss_v1p0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:imsss="http://www.imsglobal.org/xsd/imsss"
xmlns:adlnav="http://www.adlnet.org/xsd/adlnav_v1p3"
xmlns:adlseq="http://www.adlnet.org/xsd/adlseq_v1p3"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
xmlns="http://www.imsglobal.org/xsd/imscp_v1p1" version="1"
identifier="com.scorm.manifesttemplates.scorm2004.4thEd.nometadata">
- <metadata>
  <schema>ADL SCORM</schema>
  <schemaversion>2004 4th Edition</schemaversion>
</metadata>
- <organizations default="B0">
  - <organization identifier="B0" adlseq:objectivesGlobalToSystem="false">
    <title>Title</title>
    - <item identifier="i1" isVisible="true" identifierref="r1">
      <title>Title</title>
    </item>
  </organization>
</organizations>
- <resources>
  - <resource identifier="r1" href="index.html" adlcp:scormType="sco" type="webcontent">
    <file href="index.html"/>
  </resource>
</resources>
</manifest>
```

Hình 3. Nội dung của file imsmanifest.xml của một bài học dạng website

Các việc 1-3 là khá đơn giản và có thể dễ dàng thực hiện một cách thủ công cũng như bằng cách lập chương trình. Việc 5 cũng có thể dễ dàng được thực hiện thủ công bằng cách sử dụng một chương trình nén file. Lưu ý khi đóng gói vào file zip là chọn tất cả các file và thư mục con của bài học để đóng gói chứ không phải là chọn thư mục chứa các tài nguyên này. Lập trình cho việc 5 cũng khá dễ nếu sử dụng các ngôn ngữ lập trình hiện đại như C#, Java, nhờ vào các thư viện về việc nén file có sẵn trong các ngôn ngữ này. Việc khó khăn duy nhất chính là ở việc 4 và được xem xét chi tiết dưới đây.

Trong Hình 3 là nội dung của template `imsmanifest.xml` theo chuẩn SCORM 2004 4th Edition được download về từ địa chỉ nói ở đoạn trên.

Nhìn vào nội dung của file này, có thể thấy thẻ loại XML của nó, như đã được chỉ ra ở phần mở rộng của tên file.

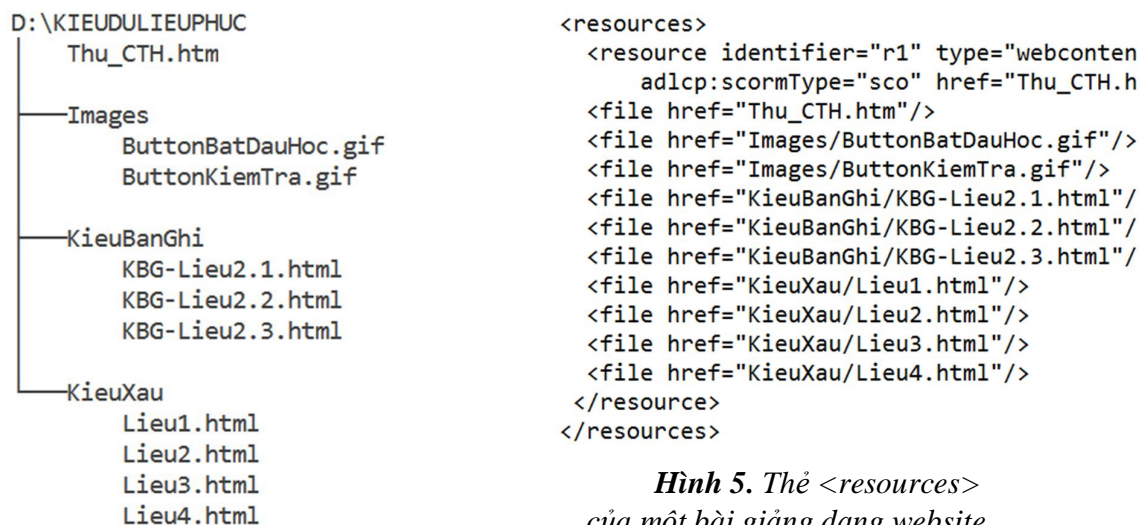
Theo thứ tự từ trên xuống, các phần chính của file có ý nghĩa và sau đây:

Bảng 1. Vai trò của các thẻ trong file `manifest.xml`

| TT | Thẻ | Ý nghĩa | Chỉnh sửa |
|----|--|--|--|
| 1 | <code><?xml version="1.0"?></code> | Version XML | Không cần |
| 2 | <code><!--Manifest template... --></code> | Chú thích về mục đích và xuất xứ của file | Không cần |
| 3 | <code><manifest identifier= ... ></code> | Các địa chỉ hệ thống của file | Không cần |
| 4 | <code><metadata> ... </metadata></code> | Tên của cấu trúc file | Không cần |
| 5 | <code><organizations...>...</organizations></code> | Tên của tổ chức | Tên của tổ chức Thay Title bằng tên của tổ chức thực tế |
| | <code><item> ... </item></code> | Xác định các tài nguyên trong tổ chức | Có thể thay đổi các thuộc tính cho tương ứng với thực tế. Mỗi thẻ <code><item></code> tương ứng với một thẻ <code><resource></code> về sau |
| 6 | <code><resources> ... </resources></code> | Các tài nguyên trong website | Thay nội dung chứa trong các thẻ con <code><resource></code> để mô tả hết tất cả các tài nguyên trong website |
| | <code><resource ...> ... </resource></code> | Tài nguyên đầu tiên – trang chủ (Home) của website | Thay đổi để chỉ đến trang chủ của website Bổ sung các thẻ <code><file...></code> để chỉ đến các tài nguyên khác được sử dụng trong website |

Chi tiết về thẻ <resources>

Thẻ resources là một thẻ phức (complex tag) chứa trong đó thẻ <resource> chỉ định thẻ loại của gói cũng như trang chủ (trong trường website) và các thẻ <file> để định vị các tài nguyên. Đây chính là nơi quan trọng nhất khi xét về mặt người dùng phải thiết lập các giá trị để đóng gói bài học. Lưu ý là mỗi thẻ thành phần <resource> phải tương ứng với một thẻ <item> có cùng giá trị identifierref ở phía trên.



Hình 5. Thẻ <resources> của một bài giảng dạng website

Hình 4. Cây thư mục của một bài giảng dạng website

Hình 5 minh họa việc thiết lập thẻ này cho một bài học cụ thể với các tài nguyên được mô tả trong cây thư mục ở Hình 4, trong đó trang chủ được chứa trong file ThuCTH.html.

3.5. Giải thuật để đóng gói cho một gói dạng website

Từ những phân tích trên, giải thuật được xác định dưới đây.

Input:

- Website của bài học được chứa trong [folderBaiHoc]
- Các file hệ thống SCORM được download về và chứa trong [folderSCORM] (xem mục III.2.b)

Output:

- File zip là đóng gói của bài học, chứa trong [folderKetQua]

Giải thuật:

```

Function writeResources(Folder sourceFolder, file fileManifest)
    Foreach (file in sourceFolder)
        Ghi thẻ <file href="[đường dẫn tới file]"/> vào fileManifest
    Foreach (subFolder in sourceFolder)
        writeResources(subFolder, fileManifest)
End Function

Function Main
    Tạo [folderKetQua]
    Copy tất cả các file trong [folderSCORM] đặt vào
    [folderKetQua]
    Copy tất cả các file trong [folderBaiHoc] đặt vào
    [folderKetQua]
    Tạo file imsmanifest.xml trong [folderKetQua]
    Mở file imsmanifest.xml và trong file này:
    Ghi thẻ từ 1 đến 5 //tham khảo Bảng 1
    Ghi thẻ <resources>
    Ghi thẻ <resource identifier="r1" tipe="webcontent"
    adlcp:scormTipe="sco" href="[Tên_file_chứa_trang_chủ]">
        writeResources(folderBaiHoc, fileManifest)
    Ghi các thẻ </resource>, </resources>, </manifest>
    Đóng file imsmanifest.xml
    Tạo file zip với tên file là gói chứa kết quả và chứa tất cả
    các file trong [folderKetQua]
End Function

```

Sử dụng giải thuật trên, độc giả có thể dễ dàng phát triển một phần mềm đóng gói cho bài giảng dạng website của mình. Độc giả cũng có thể vào địa chỉ dưới đây để tham khảo phần mềm thử nghiệm của chúng tôi được viết bằng ngôn ngữ C# trên môi trường phát triển Microsoft Visual Studio.

<https://drive.google.com/folderview?id=0B2hBkiGsF7qWdjFydnBzMmQ0UVU&usp=sharing>

Lưu ý khi sử dụng phần mềm trên là cần phải tùy biến trước file manifest theo mẫu nằm trong file SCORMFiles.zip. Độc giả có thể sử dụng trực tiếp các file trong file SCORMFiles.zip này để làm nguồn đóng gói.

Để thử nghiệm, độc giả có thể tạo một tài khoản miễn phí và upload các gói đã được đóng lên đám mây của SCORM tại địa chỉ: <http://scorm.com/scorm-solved/scorm-cloud-features/>.

4. Kết luận

Bài viết này trình bày cách thức để chuyển đổi những tài liệu học tập được tạo ở các định dạng khác nhau về các dạng thích hợp để có thể đưa lên các hệ thống quản lí học tập. Các nguyên tắc cũng như giải thuật cho phần mềm đóng gói các bài giảng dưới hình thức website về dạng chuẩn SCORM cũng đã được trình bày. Độc giả có thể dựa vào giải thuật này để tiến hành đóng gói một các thủ công hoặc viết phần mềm để thực hiện việc đóng gói một các tự động. Giải thuật này cũng có thể được phát triển thêm để đóng gói những bài học dạng website nhưng có cấu trúc phức tạp hơn.

TÀI LIỆU THAM KHẢO

1. Adobe Presenter, <http://www.adobe.com/products/presenter.html>, truy cập ngày 02/02/2016.
2. Advanced Distributed Learning (ADL) Initiative, <http://adlnet.gov>, truy cập ngày 04/01/2016.
3. Edward R. Jones. *Implications of SCORM and Emerging E-learning Standards On Engineering Education*. Proceedings of the 2002 ASEE Gulf-Southwest Annual Conference, The University of Louisiana at Lafayette, March 20 – 22, 2002.
4. Microsoft Office Mix, <https://mix.office.com/en-us/Home>, truy cập ngày 02/02/2016.
5. What is SCORM?, <http://scorm.com/scorm-explained>, truy cập ngày 24/3/2016.

(Ngày Tòa soạn nhận được bài: 15-02-2016; ngày phản biện đánh giá: 14-3-2016;
ngày chấp nhận đăng: 17-3-2016)

CÁC SỐ TẬP CHÍ KHOA HỌC SẮP TỚI:

- Số 4(82)/2016: Khoa học giáo dục
- Số 5(83)/2016: Khoa học xã hội và nhân văn
- Số 6(84)/2016: Khoa học tự nhiên và công nghệ.

Ban biên tập Tạp chí Khoa học rất mong nhận được sự trao đổi thông tin của các đơn vị bạn và được bạn đọc thường xuyên cộng tác bài vở, góp ý xây dựng.