

Một mô hình kho câu hỏi trắc nghiệm theo chuẩn QTI trong đào tạo điện tử với hệ cơ sở dữ liệu XML nguyên sinh

Nguyễn Đình Hóa*

Viện Công nghệ Thông tin, Đại học Quốc gia Hà Nội, 144 Xuân Thủy, Hà Nội, Việt Nam

Nhận ngày 30 tháng 6 năm 2010

Tóm tắt. Chuẩn QTI đối với các hệ thống sát hạch trắc nghiệm bằng máy tính có vai trò tương tự như chuẩn SCORM với các hệ thống đào tạo điện tử (e-learning). Nghiên cứu này trình bày một mô hình hệ thống kho câu hỏi trắc nghiệm theo chuẩn QTI, sử dụng các công cụ phần mềm mã nguồn mở có quy mô nhỏ, gọn nhẹ, dễ dàng triển khai để bước đầu khuyến khích phát triển chuẩn QTI. Mô hình đề xuất là một ứng dụng web với cơ sở dữ liệu nguyên sinh XML eXist và ngôn ngữ XQuery cho phép phát huy lợi thế quản trị nội dung các tài liệu XML. Kiến trúc MVC mang lại cho hệ thống tính linh hoạt cao, mã lệnh ngắn gọn và dễ bảo trì.

1. Đặt vấn đề

Với mục đích tương tự như chuẩn SCORM nhằm chia sẻ dùng chung các tài nguyên đào tạo điện tử giữa các hệ thống khác nhau, tổ chức IMS Global Learning Consortium cũng quan tâm nghiên cứu các hệ thống sát hạch trắc nghiệm trực tuyến và đề xuất chuẩn đặc tả câu hỏi trắc nghiệm QTI (*Question and Test Interoperability*). Như tên gọi đã hàm ý, chuẩn này có mục đích nâng cao tính khả chuyển để sử dụng chung ngân hàng câu hỏi trắc nghiệm giữa các phần mềm hệ thống trắc nghiệm khác nhau. Các câu hỏi đánh giá kiến thức và kỹ năng rất phong phú và đa dạng. Tuy nhiên có thể mô hình hóa và phân loại theo câu hỏi theo phương thức hỏi cũng như quy tắc xử lý đáp án một cách thống nhất để khuyến khích việc sử dụng lại ngân hàng câu hỏi. QTI v.1.2 là phiên

bản được hiểu thấu đáo và áp dụng rộng rãi nhất hiện nay, mặc dù bản v.2.0 đã được phát hành từ đầu năm 2005 và phiên bản mới nhất v.2.1 còn đang được thảo luận, chưa đi đến thống nhất.

Một câu hỏi trắc nghiệm theo chuẩn QTI là một tệp XML đặc tả rõ nội dung và cách trình bày câu hỏi, đáp án đúng, cách xử lý đáp án của thí sinh. Chuẩn QTI cũng đưa ra đặc tả bài trắc nghiệm, cũng là một tệp XML xác định rõ các thành phần của bài trắc nghiệm, tập hợp các câu hỏi chứa trong bài trắc nghiệm đó và nhiều thuộc tính khác nữa. Mô hình dữ liệu XML cho phép dễ dàng quản trị nội dung kho câu hỏi và các bài trắc nghiệm, hỗ trợ giáo viên tìm kiếm các câu hỏi và bài thi theo nội dung, lựa chọn các câu hỏi theo những tiêu chí cụ thể để soạn ra các đề trắc nghiệm đáp ứng yêu cầu từng đợt sát hạch khác nhau.

Nhằm khuyến khích áp dụng chuẩn QTI, đã có các dự án phát triển công cụ phần mềm hỗ

* ĐT: 84-4-37547879.

E-mail: hoand@vnu.edu.vn

trợ cho chuẩn này do một số trường đại học ở Anh tiến hành. Những dự án này làm ra sản phẩm là các mô đun công cụ có thể dùng như các thành phần hay các dịch vụ web để xây dựng thành một hệ thống sát hạch bằng máy tính tuân thủ chuẩn QTI dựa trên web. Tuy nhiên, một nghiên cứu về hiệu năng [1] cũng cho thấy rằng hệ thống lắp ghép các dịch vụ web giảm hiệu năng đáng kể khi số người dùng đồng thời tăng cao. Nghiên cứu cũng khuyến cáo nên trực tiếp xây dựng hệ thống từ các thư viện hàm công cụ.

Chúng tôi đặt vấn đề nghiên cứu một mô hình tổ chức lưu trữ và khai thác kho câu hỏi trắc nghiệm theo chuẩn QTI phục vụ cho các hệ thống sát hạch trắc nghiệm bằng máy tính dựa trên web, quy mô nhỏ, gọn nhẹ, dễ dàng triển khai để bước đầu khuyến khích phát triển chuẩn QTI. Hệ thống dựa trên các công cụ mã nguồn mở, tận dụng các lợi thế của mô hình dữ liệu XML và các công nghệ tiêu chuẩn cho ứng dụng web do W3C phát triển.

2. Các dự án phần mềm công cụ hỗ trợ chuẩn QTI

Nhằm khuyến khích áp dụng chuẩn QTI, một số trường đại học Vương quốc Anh tham gia chương trình nghiên cứu phát triển e-learning đã hợp tác thực hiện một nhóm dự án phát triển công cụ phần mềm hỗ trợ cho chuẩn này.

AsDel (<http://asdel.ecs.soton.ac.uk/>) rút gọn của “*Assessment Delivery*”, là một dự án do Khoa Điện tử và Khoa học máy tính, Trường Đại học Southampton tiến hành. Dự án nhằm phát triển hệ thống tạo ra và phân phối bài trắc nghiệm tuân thủ chuẩn QTI v 2.1 có thể chạy như một ứng dụng độc lập hoặc như một dịch vụ thành phần trong hệ thống theo kiến trúc

hướng dịch vụ. Dự án đã kết thúc vào năm 2008 và sản phẩm là các dịch vụ Web để lắp ghép thành ứng dụng.

AquRate (<http://aquarate.kingston.ac.uk>) là một dự án mã nguồn mở nhằm xây dựng công cụ soạn thảo câu hỏi trắc nghiệm hợp chuẩn QTI (*QTI Authoring Tool*), độc lập với các nền hệ điều hành. Dự án này do nhóm Learning Technology Research Group, thuộc Trường Đại học Kingston, cùng thực hiện kết hợp với đối tác là nhóm CARET, Trường Đại học Cambridge. Sản phẩm của dự án là phần mềm để soạn thảo các câu hỏi trắc nghiệm tuân thủ chuẩn QTI. Yêu cầu đặt ra là kết quả phải có khả năng liên tác với các sản phẩm của 2 dự án thuộc nhóm dự án phần mềm công cụ hỗ trợ chuẩn QTI là Asdel (đã nói trên) và Minibix (sẽ nói dưới đây).

Hiện nay, Aquarate đã kết thúc và cho sản phẩm là phần mềm soạn thảo một số dạng câu hỏi cốt lõi thường gặp nhất. Kết quả của dự án này được tiếp tục mở rộng bởi dự án mới là Mathqurate, cho phép xử lý các câu hỏi có chứa nhiều ký hiệu toán học.

Minibix (<http://qtitools.caret.cam.ac.uk/>) là một dự án mã nguồn mở nhằm phát triển một hệ thống quản trị ngân hàng câu hỏi tuân thủ chuẩn QTI do nhóm CARET, Trường Đại học Cambridge thực hiện. Dự án có tham vọng phát triển hệ thống kho câu hỏi đáp ứng nhiều yêu cầu ứng dụng khác nhau từ mức thấp đến mức cao trong nghiên cứu và ứng dụng QTI, với nhiều chức năng, kể cả việc quản lý nhiều phiên bản tài liệu XML. Minibix dựa trên một hệ thống cơ sở dữ liệu đã được phát triển từ trước tại Cambridge.

Tóm lại, có thể nói các dự án về QTI thuộc chương trình nghiên cứu phát triển e-learning của một số trường đại học Vương quốc Anh nhằm đến việc xây dựng hệ thống ở quy mô

lớn và chuyên nghiệp. Việc lắp ghép xây dựng một hệ thống như vậy không đơn giản và đang trong quá trình hoàn thiện để giải quyết những vấn đề phát sinh, ví dụ vấn đề hiệu năng.

Dưới đây trình bày một cách tiếp cận khác, xây dựng một hệ thống quy mô nhỏ, gọn nhẹ, dễ dàng triển khai, hỗ trợ giáo viên trong việc tổ chức lưu trữ và khai thác kho câu hỏi trắc nghiệm theo chuẩn QTI để bước đầu khuyến khích phát triển chuẩn QTI.

3. Phân tích lựa chọn công nghệ

3.1. Tại sao nên dùng cơ sở dữ liệu XML nguyên sinh

Kho câu hỏi trắc nghiệm và bài thi theo chuẩn QTI là một tập hợp các tệp XML nên việc nghiên cứu dùng cơ sở dữ liệu XML để quản trị kho câu hỏi trắc nghiệm là lựa chọn tự nhiên. Các hệ cơ sở dữ liệu XML về bản chất chia làm hai họ, các hệ cơ sở dữ liệu *khả XML (XML-enable)* và các hệ cơ sở dữ liệu *nguyên sinh XML (native XML)*.

Các hệ cơ sở dữ liệu quan hệ nổi tiếng như Oracle, DB2, MySQL...đều có phần mở rộng khả XML, cho phép tổ chức lưu trữ và quản trị các tài liệu XML trên nền mô hình dữ liệu quan hệ. Cơ sở dữ liệu nguyên sinh XML là hệ cơ sở dữ liệu được phát triển dành riêng cho các tài liệu XML, coi toàn bộ tài liệu XML là đơn vị lưu trữ lô gic cơ sở.

Từ góc độ ứng dụng khai thác các tài liệu XML, có thể xếp cơ sở dữ liệu XML vào hai nhóm khác nhau: *hướng dữ liệu (data centric)* và *hướng tài liệu (document-centric)*. Cơ sở dữ liệu XML hướng tài liệu nhằm đến các ứng dụng sử dụng toàn bộ tài liệu XML như một đơn vị trọn vẹn. Cơ sở dữ liệu XML hướng dữ

liệu phục vụ cho các ứng dụng có thể cần trích ra những phần dữ liệu dưới dạng các phần tử hay thuộc tính chứa trong nhiều tài liệu XML.

Một đặc tính quan trọng cần xem xét là khả năng *hoàn nguyên (round-trip)* một tài liệu XML. Như đã biết, mô hình dữ liệu DOM của một tài liệu XML là mô hình phân cấp các nút, không kiểm soát trình tự anh em trong các con của cùng một nút cha. Các cơ sở dữ liệu XML có thể “bấm nhỏ” tài liệu XML làm nhiều phần dựa trên mô hình phân cấp DOM và tổ chức lưu trữ vật lý ở những vùng nhớ khác nhau. Khi lấy ra một tài liệu XML, các phần tử XML được lắp ghép trả lại, trình tự các nút con của cùng một nút cha có thể không giống như tài liệu gốc. Một số ứng dụng, ví dụ các văn bản pháp luật, hợp đồng kinh tế,... đòi hỏi phải giữ nguyên dạng văn bản gốc. Tính chất này gọi là khả năng hoàn nguyên văn bản.

Bài toán tổ chức kho câu hỏi trắc nghiệm theo chuẩn QTI là xây dựng cơ sở dữ liệu XML hướng tài liệu, có một số yêu cầu đặc thù như sau đây:

- Quản lý một lượng lớn các tài liệu XML ngắn, không có tài liệu dài, kích thước lớn;
- Lưu trữ, lấy ra, chuyển tải toàn bộ tài liệu XML trọn vẹn, không có (hay ít khi) yêu cầu trích xuất từng phần nhỏ của nhiều tài liệu khác nhau.
- Yêu cầu khả năng “hoàn nguyên” tài liệu XML. Trình tự các phần trong tài liệu phải được kiểm soát.
- Có yêu cầu quản trị nội dung, tìm kiếm theo nội dung để khai thác sử dụng hay cập nhật sửa chữa các tài liệu XML chứa trong cơ sở dữ liệu.

Hệ cơ sở dữ liệu nguyên sinh XML đáp ứng tốt những yêu cầu trên.

3.2. Tại sao chọn eXist

Để phục vụ các cơ sở đào tạo, chúng tôi hạn chế chỉ tập trung nghiên cứu các hệ cơ sở dữ liệu nguyên sinh XML mã nguồn mở. Nhìn chung các hệ cơ sở dữ liệu nguyên sinh XML nguồn mở đều hỗ trợ mô hình phát triển ứng dụng web theo các công nghệ tiêu chuẩn của W3C. Đó là khuôn dạng dữ liệu (HTML, XHTML, XML), giao thức truyền thông (HTTP, REST, Web Service, ...) và các ngôn ngữ truy vấn XML và phát triển ứng dụng (XPath, XLT, XQuery). Dưới đây trình bày thêm những nét nổi bật của một số hệ cơ sở dữ liệu đáng chú ý. Nguồn tư liệu tham khảo từ các trang web chính thức của từng dự án và các bài nghiên cứu so sánh [2].

BaseX (<http://inf.uni-konstanz.de/dbis/basex/>) là hệ cơ sở dữ liệu nguyên sinh XML mã nguồn mở viết bằng Java. BaseX có thể chạy như một ứng dụng độc lập hay theo chế độ khách chủ. Hỗ trợ truy vấn XPath/XQuery và tìm kiếm toàn văn bản (*full text*). Hỗ trợ lưu trữ tệp XML kích thước rất lớn. Hỗ trợ nhiều giao diện lập trình ứng dụng (API) như REST/JAX-RX, XQJ, XML:DB.

Berkeley DB XML (<http://oracle.com/database/berkeley-db/xml/>) là hệ cơ sở dữ liệu nguyên sinh XML xây dựng trên nền Berkeley DB. Nó đúng hơn là một cơ sở dữ liệu nhúng trong ứng dụng chứ không chạy theo chế độ khách - chủ.

dbXML (<http://dbxml.com/>) là hệ cơ sở dữ liệu nguyên sinh XML hỗ trợ 4 phương thức lưu trữ tài liệu XML khác nhau. Mô hình tổ chức dữ liệu thành hệ phân cấp các sưu tập giống như thư mục trong hệ thống tệp. Sưu tập có thể chứa tệp nhị phân (ví dụ như tệp ảnh) nhưng không chứa lẫn lộn tệp XML với tệp nhị phân. *dbXML* cũng hỗ trợ các giao diện lập trình cơ sở dữ liệu nhúng trong ứng dụng.

Xindice (<http://xml.apache.org/xindice/>) là hệ cơ sở dữ liệu nguyên sinh XML. Hỗ trợ truy vấn XPath và ngôn ngữ cập nhật dữ liệu là XML:DB XUpdate. Xindice hỗ trợ XML:DB API để phát triển ứng dụng với Java và cũng hỗ trợ XML-RPC API để phát triển với các ngôn ngữ khác. Xindice đặc biệt có ích khi làm việc với các tài liệu XML có cấu trúc rất phức tạp.

MyXMLDB

(<http://myxml.db.sourceforge.net/>) là hệ cơ sở dữ liệu nguyên sinh XML triển khai bên trên MySQL. Lưu trữ tài liệu XML như các BLOB, có thể lớn đến 256 MB. Cho phép lưu trữ các tài liệu XML rất lớn là một điểm mạnh của MySQL, vì hầu hết các CSDL nguyên sinh XML đều hạn chế tệp XML kích thước lớn do chúng làm việc với cây DOM của tài liệu mà mô hình DOM thì rất tốn bộ nhớ.

eXist (<http://exist-db.org/>) là hệ cơ sở dữ liệu nguyên sinh XML sử dụng mô hình lưu trữ cây B+ và tệp có phân trang. eXist có thể chạy trong chế độ khách chủ như máy chủ cơ sở dữ liệu, như một cơ sở dữ liệu nhúng bên trong ứng dụng và đặc biệt là eXist có thể chạy trong một máy servlet của ứng dụng Web ví dụ như Tomcat. Rất thuận lợi khi phát triển ứng dụng Web.

eXist được phân phối kèm với nhiều mô đun chứa các hàm tiện ích do người sử dụng tự phát triển, hỗ trợ nén dữ liệu, xử lý ngày giờ, các phép toán số học, làm việc với hình ảnh, tệp nhị phân và cả dữ liệu địa lý. Một ưu điểm là eXist có tài liệu hướng dẫn rất đầy đủ dành cho người phát triển ứng dụng.

Chúng tôi lựa chọn sử dụng eXist để xây dựng hệ thống kho câu hỏi.

3.3. Dùng ngôn ngữ XQuery để phát triển ứng dụng Web

Bất cứ khi nào cần truy cập và thao tác dữ liệu XML thì một lựa chọn cần xem xét là dùng

XQuery. XQuery là ngôn ngữ lập trình *khai báo* (*declarative*), không chỉ sử dụng làm ngôn ngữ truy vấn dữ liệu XML mà còn sử dụng làm ngôn ngữ chuyển đổi dữ liệu giữa các khuôn dạng khác nhau và làm ngôn ngữ tích hợp dữ liệu giữa các hệ thống. Khả năng diễn đạt của XQuery không kém bất cứ ngôn ngữ lập trình mục tiêu chung nào. Hoàn toàn có thể và rất hiệu quả khi phát triển toàn bộ trọn vẹn một ứng dụng web hoàn chỉnh quy mô lớn (*enterprise web application*) bằng XQuery [3-5]. Hơn nữa, phát triển ứng dụng web bằng XQuery còn có một số ưu điểm như sau:

- Mã lệnh ngắn gọn và đẹp đẽ. Trong [4] cũng đã so sánh số dòng mã lệnh khi triển khai cùng một ứng dụng Web cơ sở dữ liệu bằng Java và bằng XQuery. Hỗ trợ XLTS của XQuery cho phép thực hiện các trang XHTML rất ngắn gọn. Là một ngôn ngữ khai báo (*declarative*), XQuery cũng cho phép triển khai các dịch vụ kho câu hỏi đơn giản hơn nhiều so với Java.

- Đơn giản hóa kiến trúc tổng thể của ứng dụng và cải thiện tính linh hoạt. Một ứng dụng XQuery gồm nhiều mô đun, là những tệp XQuery. Một mô đun XQuery gồm nhiều hàm, có thể được gọi trực tiếp từ một mô đun khác hoặc đưa ra qua REST hay như một dịch vụ Web. XQuery rất thuận tiện để xây dựng ứng dụng web theo kiến trúc MVC.

Một ứng dụng XQuery gồm các thành phần sau:

- Các mô đun là tệp các lệnh XQuery thực hiện các truy vấn dữ liệu XML cũng như chuyển đổi dữ liệu và hiển thị kết quả thành trang web.

- Bộ xử lý XQuery thi hành các vấn tin XQuery. Đây có thể là một thành phần chạy độc lập hay phần mở rộng của hệ cơ sở dữ liệu tích hợp sẵn, có khả năng xử lý XQuery.

- Một hệ cơ sở dữ liệu XML chứa dữ liệu cần truy vấn và có hỗ trợ XQuery.

- Một giao diện lập trình ứng dụng XQuery để triệu gọi và xử lý các truy vấn từ trong ứng dụng.

4. Mô hình ứng dụng web theo kiến trúc MVC với cơ sở dữ liệu nguyên sinh XML eXist và ngôn ngữ XQuery

4.1. Phân tích yêu cầu

Mục tiêu bước đầu là xây dựng một hệ thống quy mô nhỏ, gọn nhẹ, dễ dàng triển khai, hỗ trợ giáo viên trong việc tổ chức lưu trữ và khai thác kho câu hỏi trắc nghiệm theo chuẩn QTI. Hệ thống này không gồm công cụ soạn thảo câu hỏi trắc nghiệm và công cụ phân phối bài thi tới thí sinh.

Giáo viên là tác giả các câu hỏi trắc nghiệm đưa vào kho câu hỏi. Trước mỗi kỳ sát hạch, giáo viên có thể rà soát kho câu hỏi thực hiện các cập nhật cần thiết trước khi lựa chọn các câu hỏi để làm thành bài trắc nghiệm. Giáo viên có thể truy vấn kho câu hỏi theo một số tiêu chí chính xác như theo dạng câu hỏi mà chuẩn QTI đã định nghĩa (câu hỏi dạng lựa chọn một trong nhiều đáp án, câu hỏi điền vào khoảng trống, câu hỏi có tương tác với hình ảnh, ...); theo điểm số gán cho đáp án đúng; theo tên tác giả, thời gian tạo ra, thời gian cập nhật ...

Điều đặc biệt quan trọng là mô hình dữ liệu XML cho phép quản lý nội dung. Có thể thực hiện tìm kiếm và xử lý theo nội dung. Có thể tìm kiếm và xử lý đến từng phần tử XML hay thuộc tính cụ thể trong các tài liệu XML. Có thể tìm theo một (các) từ khóa có xuất hiện trong lời văn câu hỏi hay thậm chí một thành phần cụ thể của câu hỏi, trong phần lời dẫn, trong phần phương án trả lời. Công nghệ XML mang lại

tiềm năng lớn về quản trị nội dung kho câu hỏi. Hệ thống cần tận dụng lợi thế này.

Kho câu hỏi phải được tích hợp dễ dàng với hệ thống trắc nghiệm bằng máy tính, phân phối bài thi đến các thí sinh khi đăng nhập thực hiện kỳ thi. Nói rộng hơn người dùng kho câu hỏi còn là các thí sinh thông qua trung gian là mô đun phân phối bài thi. Nhiều hệ thống trắc nghiệm bằng máy tính sinh bài thi theo cách đơn giản là chọn ngẫu nhiên từ các câu hỏi trong kho. Giải pháp đơn giản này có các hạn chế, ví dụ như không cho phép xây dựng các bài sát hạch đáp ứng những điều kiện nhất định về nội dung cần sát hạch mà giáo viên dự kiến. Giáo viên có thể mong muốn ví dụ bài thi sẽ gồm 4 câu hỏi về vấn đề X, 5 câu hỏi về kỹ năng Y, ... Với hệ thống có khả năng quản trị nội dung, yêu cầu trên được hỗ trợ giải quyết dễ dàng.

4.2. Kiến trúc MVC

Kho câu hỏi trắc nghiệm theo chuẩn QTI là một ứng dụng cơ sở dữ liệu dựa trên nền Web. Một ứng dụng điển hình dạng này thường có kiến trúc 3 tầng, gồm tầng giao diện người sử dụng, tầng truy cập dữ liệu và tầng xử lý. Khung kiến trúc MVC được sáng tạo ra để thiết kế các ứng dụng dạng này.

Mô hình-khung nhìn-bộ điều khiển hay MVC (*Model-View-Controller*) là một kiến trúc phần mềm theo khuôn mẫu, tách biệt phần thực hiện chức năng xử lý với dữ liệu đầu vào và trình bày trang giao diện dành cho người sử dụng. Mục đích của MVC là thông qua việc tách biệt xử lý dữ liệu với hiển thị trang web sẽ làm thiết kế kiến trúc đơn giản hơn cho phép

tăng tính linh hoạt, có thể phát triển, kiểm thử và bảo trì từng phần độc lập nhau [6,7].

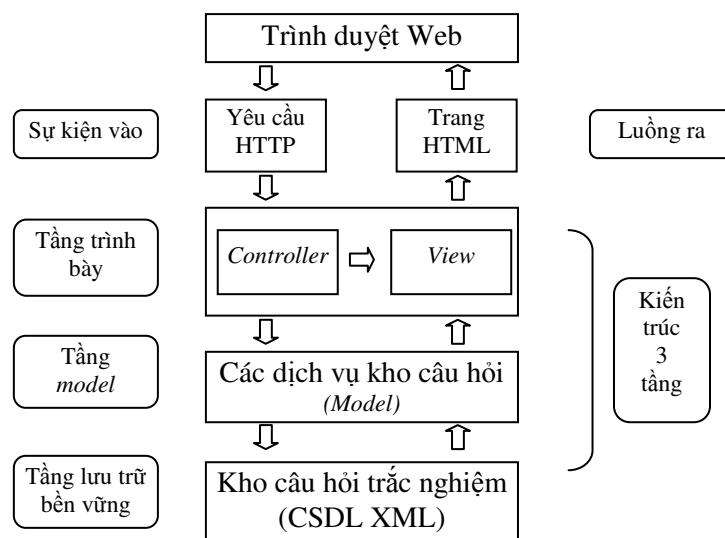
Mô hình (*model*) là nói về mô hình biểu diễn dữ liệu và tính toán xử lý mà ứng dụng thực hiện. Khi có thay đổi dữ liệu trong mô hình, nó sẽ báo cho khung nhìn để làm tươi lại, phản ánh kịp thời những thay đổi đó. Nhìn chung các ứng dụng sử dụng một cơ sở dữ liệu để lưu trữ dữ liệu bền vững. Trong khung kiến trúc MVC, tầng truy cập dữ liệu được hiểu là nằm trong phần mô hình.

Khung nhìn (*view*) là phần biểu hiện mô hình xử lý sao cho dễ tương tác, thường là dưới dạng các phần tử giao diện người sử dụng. Có thể có nhiều khung nhìn đối với một mô hình dành cho từng mục đích khác nhau.

Bộ điều khiển (*controller*) nhận yêu cầu đầu vào từ người sử dụng và khởi tạo các đáp ứng (*response*) bằng cách gọi các đối tượng xử lý của mô hình.

Một ứng dụng MVC có thể là một tập hợp các bộ ba mô hình/ khung nhìn/ bộ điều khiển, mỗi bộ ba chịu trách nhiệm về một phần tử giao diện người sử dụng khác nhau. MVC thường áp dụng cho các ứng dụng web, ở đây khung nhìn là tài liệu HTML/XHTML mà ứng dụng sinh ra. Bộ điều khiển tiếp nhận dữ liệu từ các yêu cầu GET hay POST và quyết định chuyển giao xử lý tới các đối tượng (tức là mô hình) chứa quy tắc xử lý.

Dựa trên khung kiến trúc MVC cho một ứng dụng web với phía khách gậy (*thin-client*) điển hình nêu trong [7], hình dưới đây minh họa kiến trúc MVC của ứng dụng web triển khai hệ thống kho câu hỏi trắc nghiệm theo chuẩn QTI dựa trên hệ cơ sở dữ liệu XML của chúng ta.



Hình 1. Kiến trúc MVC của ứng dụng kho câu hỏi trắc nghiệm theo chuẩn QTI với cơ sở dữ liệu XML.

4.3. Tổ chức các sưu tập tài liệu XML

Hệ cơ sở dữ liệu eXist, cũng như một số hệ cơ sở dữ liệu nguyên sinh XML khác, tổ chức dữ liệu ở mức lô gic thành hệ cây phân cấp các sưu tập (*collection*) tương tự như các thư mục trong hệ thống tệp nói chung [8]. Trong mỗi sưu tập chứa nhiều tệp XML, có thể ứng với các lược đồ (*schema*) XML khác nhau. Nếu một sưu tập chỉ chứa các tệp XML ứng với một lược đồ XML duy nhất thì việc làm chỉ mục sẽ đơn giản hơn và truy vấn hiệu quả hơn. Sưu tập cũng có thể chứa các khối dữ liệu nhị phân, ví dụ tệp hình ảnh, âm thanh .. chứa trong tài liệu đa phương tiện. Nói chung, một cách tổ chức hợp lý là nên tách các khối dữ liệu nhị phân đưa vào sưu tập riêng.

Khi thiết kế lưu trữ lô gic kho câu hỏi cần xem xét theo hai chiều đan chéo nhau: theo dạng câu hỏi định nghĩa bởi chuẩn QTI và theo môn học. Tổ chức các sưu tập theo dạng câu hỏi tức là một sưu tập chỉ chứa các câu hỏi một dạng duy nhất, ứng với một lược đồ XML duy

nhất. Lựa chọn này cho phép làm chỉ mục sẽ đơn giản và tăng hiệu năng truy vấn. Tuy nhiên, các câu hỏi của cùng một môn học nằm trong nhiều sưu tập khác nhau, đặt ra những khó khăn về kiểm soát truy cập. Hơn nữa, khi giáo viên tìm kiếm theo các tiêu chí khác với dạng câu hỏi thì phải truy vấn nhiều sưu tập, hiệu năng cuối cùng sẽ giảm.

Chúng tôi lựa chọn tổ chức các sưu tập theo môn học.

- Một môn học ứng với một sưu tập cơ sở nhỏ nhất, không phân chia tiếp thành sưu tập con nhỏ hơn nữa. Một sưu tập cơ sở chứa tất cả các câu hỏi trắc nghiệm và bài thi trắc nghiệm của môn học đó.

- Trên cấp sưu tập cơ sở có các sưu tập mức cha ứng với từng ngành hay nhóm ngành đào tạo, phản ánh việc tổ chức các chương trình đào tạo.

- Toàn bộ kho câu hỏi là sưu tập gốc.

Lựa chọn này được định hướng bởi yêu cầu sử dụng kho câu hỏi. Hơn nữa, dễ dàng mở

rộng quy mô kho câu hỏi khi cần thêm các môn học mới hay ngành đào tạo mới. Lô gic xử lý hầu như không phải thay đổi gì.

4.4. Kiểm soát truy cập

Để đảm bảo an toàn dữ liệu, các ứng dụng cơ sở dữ liệu đều phải có kiểm soát truy cập dữ liệu đúng đắn. Mô hình kiểm soát truy cập dữ liệu thường dựa trên vai trò chức năng của người sử dụng [9].

Các giáo viên sử dụng cơ sở dữ liệu để lưu trữ, tìm kiếm các câu hỏi, sau đó có thể sửa đổi cập nhật và biên tập các bài thi. Mỗi giáo viên thường phụ trách một (hay vài) môn học. Giáo viên cần có mọi quyền đối với kho câu hỏi trắc nghiệm của môn học mà mình phụ trách: tạo ra, đọc, viết, xóa các tệp XML trong cơ sở dữ liệu. Cần định nghĩa nhóm người dùng *giáo viên-môn học* ứng với từng môn học.

Các thí sinh là người sử dụng kho câu hỏi thông qua hệ thống phân phối bài thi. Mỗi lần đăng nhập hệ thống, thí sinh làm bài thi của một môn học. Thí sinh chỉ cần có quyền đọc. Ta định nghĩa nhóm người dùng *thí sinh-môn học* với quyền truy cập chỉ đọc sưu tập môn học tương ứng.

Như vậy, tổ chức lưu trữ logic theo hệ thống sưu tập ứng với từng môn học cũng phù hợp với yêu cầu phân quyền và kiểm soát truy cập kho câu hỏi.

4.5. Kiểm soát truy cập trong eXist

eXist thực hiện phân quyền truy cập từng tài liệu XML, từng sưu tập trong cơ sở dữ liệu tương tự như quyền truy cập hệ thống tệp/thư mục của linux/unix [10]. Quyền truy cập mỗi tệp XML hay mỗi sưu tập có ba mức khác nhau:

- R là *Read/Execute*: cho phép đọc nội dung đối với tệp hoặc liệt kê danh sách các tệp, các sưu tập con có trong sưu tập.

- W là *Create/Delete*: cho phép thêm mới hoặc xóa bỏ tệp, sưu tập con.

- U là *Modify/Change*: cho phép cập nhật, sửa đổi nội dung. Lưu ý rằng nếu người sử dụng chỉ có quyền U thì không thể thêm tệp mới vào sưu tập.

Tập hợp người sử dụng hệ thống được chia thành 3 lớp: chủ sở hữu (*owner*), nhóm người dùng (*group*) và khách (*world*). Một trường gồm ba bit ứng với RWU, mỗi bit sẽ được bật hay tắt để cấp quyền tương ứng cho từng lớp người sử dụng.

Mô hình dữ liệu XML còn cho phép phân quyền và kiểm soát truy cập mịn hơn nữa, đến mức các phần tử trong mô hình DOM của tài liệu XML. Tuy nhiên, kho câu hỏi của chúng ta không có yêu cầu kiểm soát truy cập mịn đến mức ấy.

Phác thảo mô hình kiểm soát truy cập các sưu tập và tài liệu XML trong kho câu hỏi như sau:

- Mỗi giáo viên là người dùng chủ sở hữu tài liệu XML do mình tạo ra.

- Nhóm giáo viên-môn học là nhóm người dùng (*group*) của sưu tập-môn học tương ứng.

- Nhóm thí sinh-môn học chỉ cần quyền truy cập khách (*world*).

Người quản trị hệ thống cơ sở dữ liệu chỉ cần có quyền với các sưu tập mức trên cơ sở, không cần phải có quyền đọc viết cập nhật với các sưu tập mức cơ sở.

5. Phần mềm thực nghiệm

5.1. Thiết kế các mô đun

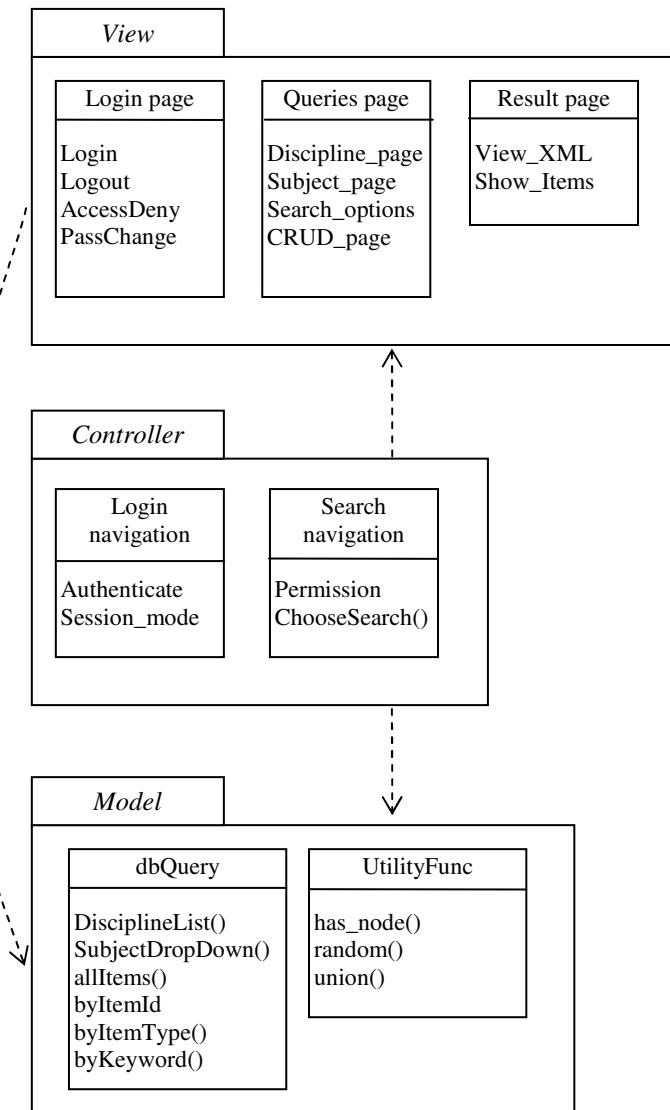
Giáo viên đăng nhập và khai thác kho câu hỏi như một người sử dụng cơ sở dữ liệu. Các thao tác cơ sở dữ liệu cơ bản gồm thêm mới, lấy

ra, cập nhật, và xóa (CRUD). Phần xử lý ở đây chỉ hạn chế là các truy vấn thao tác dữ liệu được chuyển đến tầng truy cập dữ liệu và nhận kết quả trả về. Kết quả đầu ra có thể dưới hai định dạng: trang web hiển thị trong cửa sổ trình duyệt và tệp mã nguồn XML. Các tệp mã nguồn XML chính là đầu vào chuyển đến công

cụ soạn thảo câu hỏi hay hệ thống phân phối bài thi tối thí sinh.

Thuật ngữ QTI gọi câu hỏi là *item* và bài trắc nghiệm là *test*.

Sơ đồ lớp UML với các gói trong hệ thống như dưới đây. Các tên gọi đã được chọn nhằm phản ánh nội dung của từng yếu tố.



Hình 2. Sơ đồ lớp UML trong các gói của phân hệ “Giáo viên”.

5.2. Kết quả triển khai

Chúng tôi đã phát triển kho câu hỏi trắc nghiệm theo chuẩn QTI, sử dụng các công cụ phần mềm nguồn mở và áp dụng mô hình thiết kế như đã trình bày trên. Phân hệ “Giáo viên” được phát triển trước. Các chức năng hỗ trợ giáo viên truy vấn kho câu hỏi theo những tiêu chí chính xác hoặc tìm kiếm theo nội dung dựa trên các từ khóa đã trình bày trong phần phân tích yêu cầu có thể triển khai bằng các truy vấn XQuery. Kết quả trả về có thể dưới hai khuôn dạng: hiển thị như trang Web trong trình duyệt, như thí sinh sẽ nhìn thấy hoặc là mã nguồn XML. Giáo viên có thể chọn các câu hỏi, lập ra các bài thi đáp ứng những điều kiện nhất định về nội dung cần sát hạch trong kỳ thi.

Các công cụ cập nhật XML còn chưa phát triển chín muồi. Đặc tả XUpdate (2000) gần như bị lãng quên và phần mở rộng XQuery Update Facility mới bắt đầu được phát triển. Hiện này, việc sửa đổi cập nhật câu hỏi vẫn phải thông qua công cụ soạn thảo câu hỏi (*authoring tool*), sửa đổi từng câu hỏi, chưa được tích hợp vào hệ thống.

Nếu đăng nhập như thí sinh, sau khi chọn môn học, hệ thống sẽ hiển thị một bài thi. Các chức năng xử lý quá trình thi, xử lý bài làm của thí sinh, hiển thị kết quả mới dừng ở mức các thao tác cơ bản nhất. Điều này chỉ nhằm chứng tỏ tính khả thi nếu tiếp tục phát triển phân hệ phân phối bài thi và xử lý quá trình thi.

6. Kết luận

Hoạt động đào tạo kỹ năng công nghệ thông tin tại Viện Công nghệ thông tin Đại học quốc gia Hà Nội đặt ra yêu cầu nghiên cứu triển khai đào tạo điện tử nói chung và phát triển hệ thống sát hạch trắc nghiệm bằng máy tính nói riêng.

Xây dựng hệ thống phù hợp chuẩn QTI sẽ đảm bảo cho kho câu hỏi có thể dễ dàng chia sẻ sử dụng chung với các hệ thống sát hạch trắc nghiệm khác trong tương lai. Nghiên cứu này đưa ra một mô hình hệ thống kho câu hỏi trắc nghiệm theo chuẩn QTI, sử dụng các công cụ phần mềm mã nguồn mở. Mô hình ứng dụng web gồm cơ sở dữ liệu nguyên sinh XML eXist và ngôn ngữ XQuery với kiến trúc MVC cho phép tăng tính linh hoạt, dễ dàng phát triển, mã lệnh ngắn gọn đồng thời phát huy lợi thế quản trị nội dung dựa trên XML. Một phần mềm thử nghiệm đã được phát triển, với các chức năng cơ bản nhất có thể hỗ trợ giáo viên trong lưu trữ và khai thác sử dụng kho câu hỏi trắc nghiệm, bước đầu khuyến khích phát triển chuẩn QTI.

Lời cảm ơn

Nội dung nghiên cứu trên được thực hiện với sự hỗ trợ của đề tài nghiên cứu khoa học cấp ĐHQG HN QG-07-28.

Tài liệu tham khảo

- [1] G. B. Wills, *et. al.*, “A Delivery Engine for QTI Assessments”, School of Electronics and Computer Science, University of Southampton, Southampton, UK, 2008, <http://eprints.ecs.soton.ac.uk/15980>.
- [2] R. Bourret, “XML Database Products”, 2010, www.rpbouret.com/xml/XMLDatabaseProds.htm.
- [3] I. Ivanov, “Interactive Web Applications with XQuery”, 2003, <http://www.xml.com/pub/a/2003/05/14/xquery.html>.
- [4] M. Kaufmann, D. Kossmann, “Developing an Enterprise Web Application in XQuery”, *Web Engineering, Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Volume 5648/2009, p. 465-468.
- [5] W. Meier, “Developer's Guide: Writing Web Applications using XQuery”, <http://exist.sourceforge.net/devguide.html>.

- [6] E. Palmitesta, "MVC-driven Web Application Framework: Why and How", *Mark Logic User Conference*, May 12-14, 2009, San Francisco.
- [7] D. J. Anderson, "Using MVC Pattern in Web Interactions", white paper, 2000, <http://www.uidesign.net/.../UsingMVCPatterninWebInter.html>.
- [8] W. Meier, "eXist: An Open Source Native XML Database", <http://www.springerlink.com/index/dv46dpqmcnd4e1f3.pdf>.
- [9] A. Gabillon, "An Authorization Model for XML DataBases", *Conference: ACM Workshop on Secure Web Services*, October 29, 2004, Fairfax VA, USA, p16-28.
- [10] W. Meier, "User Authentication and Access Control", <http://exist.sourceforge.net/devguide.html>.

A model for the QTI assessment items bank in e-Learning using a native XML database

Nguyen Dinh Hoa

Information Technology Institute, VNU, 144 Xuan Thuy, Hanoi, Vietnam

In computer-based assessment systems, the QTI Standard is the same as the SCORM Standard for the e-learning systems. This paper presents a model for the QTI assessment items bank, using open source software tools with a scalable light-weight design and an easy implementation. The purpose is to promote the QTI standard in the educational organizations. It is a database web application implemented by using the native XML database eXist and the programming language XQuery. The system has advantages of the XML data model in content management and the MVC architecture gives it a higher flexibility, a compactness of coding and a better maintainability.