

NGHIÊN CỨU XÂY DỰNG HỆ THỐNG SINH TỰ ĐỘNG MÃ NGUỒN CÁC WEBSITE ASP.NET MVC

Nguyễn Trần Quốc Vinh¹, Nguyễn Văn Vương², Nguyễn Ngọc Cương³

^{1,2}Trường Đại học Sư phạm - Đại học Đà Nẵng, ³Học viện An ninh Nhân dân

¹ntquocvinh@gmail.com, ²vanvuong2610@gmail.com, ³cuongnn.hvan@gmail.com

TÓM TẮT - Tác giả tiến hành nghiên cứu lý thuyết sinh mã nguồn tự động và ứng dụng để tạo ra chương trình sinh tự động mã nguồn các website ASP.NET theo mô hình MVC của Microsoft. Đầu vào là siêu dữ liệu từ cơ sở dữ liệu (CSDL), các khuôn mẫu các trang web có cấu trúc theo tiêu chuẩn của thư viện StringTemplate và cấu hình của người dùng. Đầu ra là một ứng dụng web hoạt động theo mô hình MVC trên nền tảng ASP.NET. Website được sinh ra có các chức năng thao tác dữ liệu như hiển thị kết quả thực thi truy vấn, thêm mới, cập nhật, xóa trên các bảng, khung nhìn. Đối với khung nhìn đã có sẵn trong CSDL hoặc truy vấn do người dùng định nghĩa, dựa trên siêu dữ liệu của các bảng tham gia thu được từ CSDL, bài báo đề xuất thuật toán xác định bảng để khuyến cáo cập nhật, thêm mới và xóa dữ liệu cũng như giới hạn các cột có thể chọn cho các trang web đó. Người dùng cũng có thể chọn khuôn mẫu riêng biệt cho mỗi trang web. Website được chương trình sinh ra có thể được triển khai ngay trên máy chủ web IIS mà không cần thêm bất cứ thành phần nào khác.

Từ khóa - sinh tự động mã nguồn, khuôn mẫu, trang web aspx, mô hình MVC, thao tác dữ liệu, truy vấn và khung nhìn

I. ĐẶT VẤN ĐỀ

Sinh mã nguồn là kỹ thuật viết và sử dụng các chương trình để xây dựng các ứng dụng hay hệ thống mã cụ thể nào đó một cách tự động [1]. Sinh mã nguồn tự động sẽ giúp tiết kiệm thời gian viết mã. Mã nguồn được sinh ra dễ đọc và đảm bảo ít lỗi do được sinh ra trên cơ sở các khuôn mẫu chuẩn. Các mô tả cho mã nguồn đầu ra là độc lập với nền tảng công nghệ, tức là không phụ thuộc vào công nghệ sử dụng là .NET, Java hay là PHP.

Ngày nay, việc phát triển ứng dụng công nghệ thông tin, đặc biệt là ứng dụng web đang trở nên phổ biến. Các nhà phát triển ứng dụng web thường xuyên phải lặp lại các công việc tương tự nhau trong suốt quá trình phát triển các website, các trang web một cách vô nghĩa. Việc đó chiếm một khoảng thời gian lớn, đồng nghĩa với gia tăng một khoản chi phí đáng kể đối với doanh nghiệp. Để giảm bớt khối lượng công việc cho các nhà phát triển, sinh mã nguồn một cách tự động là giải pháp được đánh giá là tốt nhất hiện nay.

Trên thế giới, có nhiều phần mềm sinh mã nguồn tự động với rất nhiều phiên bản miễn phí hoặc thương mại. Nổi bật nhất là phần mềm ASP.NET Maker [2] và CodeSmith Generator [3]. Cả hai phần mềm đều tính phí sử dụng. Công trình [4] dùng StringTemplate thực hiện sinh mã nguồn website đơn giản trên nền tảng ASP.NET. Nhìn chung, các hệ thống sinh mã nguồn hiện tại gặp phải các hạn chế sau:

- Chưa sinh được mã nguồn website trên nền tảng ASP.NET theo mô hình MVC của Microsoft sử dụng Razor View Engine.

- Không tạo được chương trình hoàn chỉnh, chỉ sinh mã cho các loại đối tượng cụ thể, ví dụ chỉ sinh mã nguồn C#, không sinh được giao diện;

- Mã nguồn được sinh ra cần phải biên dịch mới có thể triển khai ứng dụng trên máy chủ web được;

- Người sử dụng không thể thêm, bớt hoặc kế thừa mã nguồn được sinh ra;

- Không thể tùy chỉnh khuôn mẫu, chỉ cho phép chọn một khuôn mẫu cho mỗi dự án, không cho phép cấu hình bố trí cho từng trang web;

- Không sinh được mã cho chức năng thêm, sửa, xóa đối với bảng tham gia vào truy vấn và khung nhìn. Hạn chế này là nghiêm trọng, bởi lẽ hầu hết người dùng trong hầu hết trường hợp đều có nhu cầu thao tác dữ liệu dựa trên ngữ cảnh – thao tác dữ liệu ở bảng này nhưng cần thông tin hỗ trợ từ nhiều bảng khác. Nếu sinh mã thao tác dữ liệu chỉ dựa trên thông tin từ một bảng thì sẽ rất khó khăn. Chẳng hạn cho trường hợp bảng điểm của sinh viên có thể chỉ bao gồm các cột mã-môn-học, mã-sinh-viên và điểm.

Mô hình MVC (model-view-controller) tiêu chuẩn có nhiều ưu điểm so với mô hình web truyền thống. Ngoài các đặc điểm chung của mô hình MVC tiêu chuẩn, mô hình MVC của Microsoft còn được bổ sung các tính năng mở rộng khác. Tác giả nghiên cứu việc sinh mã nguồn tự động các trang web ASP.NET theo mô hình MVC của Microsoft. Các website được sinh ra cho phép tìm kiếm và thao tác dữ liệu trên các bảng cũng như khung nhìn và truy vấn. Tác giả tiến hành nghiên cứu kỹ thuật sinh mã nguồn tự động bằng cách lập trình sử dụng thư viện StringTemplate, ứng dụng các kỹ thuật đó để xây dựng chương trình sinh mã nguồn trên CSDL SQL Server. Kết quả mong đợi là hệ thống chương trình có thể sinh ra một website hoàn chỉnh có thể chạy ngay trên máy chủ web IIS cho mỗi CSDL, với đầy đủ các chức năng thêm, sửa, xóa và tìm kiếm trên từng bảng, từng khung nhìn; cũng như khả năng chọn khuôn mẫu cho mỗi trang web và tùy chỉnh bố trí giao diện cho mỗi trang web sinh ra. Hệ thống được sinh ra phải được tích hợp quản

lý đăng nhập và phân quyền. Việc cho phép chọn khuôn mẫu cho từng trang rất ý nghĩa, vì thông thường, các trang dành cho quản trị có giao diện khác so với các trang dành cho người dùng. Tính năng cho phép thay đổi thứ tự các cột là chưa đủ, vì các cột có độ rộng khác nhau, đòi hỏi không gian khác nhau để hiển thị giá trị. Theo đó, tính năng nâng cao cho phép điều chỉnh bố trí trang web là cần thiết cho các trang thêm, sửa, xem một bản ghi. Ngoài ra, chương trình sinh tự động mã nguồn còn tạo ra mã nguồn gồm các tính năng khác như: quản lý tệp trên máy chủ (tải lên, đổi tên, xóa), tính năng bảo mật, quản lý tài khoản người dùng và phân quyền; sử dụng AJAX...

II. XÂY DỰNG CHƯƠNG TRÌNH SINH TỰ ĐỘNG MÃ NGUỒN

A. *StringTemplate*

StringTemplate là một thư viện máy sinh theo mẫu (Template Engine Library) dùng để tạo ra các văn bản dựa trên các mẫu dữ liệu có cấu trúc [5,6]. Thư viện này được tạo ra do nhu cầu phát triển các trang web “động” với nhiều khuôn mẫu khác nhau với mong muốn làm cho phát triển ứng dụng web dễ dàng hơn, cải thiện tính linh hoạt, giảm chi phí bảo trì và cho phép viết song song mã nguồn xử lý (C#, Java, PHP) và mã nguồn giao diện (HTML).

B. *Sinh mã mô-đun làm việc với CSDL*

Kiến trúc MVC (Model-View-Controller) đã trở thành khuôn mẫu tốt nhất trong việc phát triển phần mềm hiện nay. Kiến trúc MVC thể hiện sự chia tách mạnh mẽ các thành phần trong một phần mềm, ví dụ tách giữa thao tác dữ liệu, xử lý lô-gic và hiển thị nội dung. Việc tách rõ các mối quan tâm giúp chia nhỏ sự phức tạp khi thiết kế một ứng dụng, tăng khả năng kiểm thử độc lập các thành phần trong quá trình phát triển, dễ dàng hơn trong việc nâng cấp.

Kiến trúc MVC chia ứng dụng thành 3 thành phần chính, đó là: i) Model gồm một tập các lớp mô tả các dữ liệu mà người dùng đang làm việc và các quy tắc ràng buộc sự thay đổi dữ liệu; ii) View định nghĩa cách thức giao diện người dùng sẽ được hiển thị; iii) Controller là một tập các lớp điều khiển các luồng dữ liệu của người dùng, xử lý các sự kiện trong hệ thống.

Trong ASP.NET MVC, model thực hiện đóng gói dữ liệu được lưu trữ trong CSDL, điều khiển và cập nhật các thay đổi của model vào các đối tượng cụ thể trong CSDL. Model sử dụng công cụ Entity Framework để truy xuất dữ liệu. Đối với view, ASP.NET MVC sử dụng Razor View Engine hoặc ASPX View Engine để xử lý hiển thị. Giống như ASPX View Engine, Razor cho phép viết mã HTML và C# hoặc VB.NET trên cùng 1 tệp. Tuy nhiên, với Razor người dùng sử dụng ít ký tự để viết mã hơn. Ví dụ khai báo khối lệnh C#, ASPX dùng cặp đánh dấu `<% ... %>`, còn Razor chỉ cần khai `@`. Ngoài ra, Razor cho phép viết đang mã HTML bên trong mã C# (VB.NET). Sự phối hợp này mang lại tính đơn giản và dễ sử dụng khi viết mã nguồn, đặc biệt là trong kiểm thử tự động. Cuối cùng là controller, thành phần điều khiển các yêu cầu của người dùng, làm việc với các model và lựa chọn view để hiển thị lên giao diện người dùng.

Như vậy, để sinh mã nguồn cho một ứng dụng ASP.NET MVC ngoài sinh mã nguồn cho 3 thành phần chính, chúng ta cần sinh mã mô-đun làm việc với CSDL theo mô hình Entity Framework. Bên cạnh đó, cú pháp Razor sử dụng trên view phải được tuân thủ chặt chẽ nếu không muốn ứng dụng bị lỗi khi chạy.

C. *Xác định cấu trúc dữ liệu*

Chương trình sinh tự động mã nguồn sử dụng các siêu dữ liệu (meta-data) lấy từ CSDL như thông tin về bảng, về khung nhìn và các cột, cũng như thông tin về khoá chính và khoá độc nhất (unique, sau đây gọi chung là khoá) và khoá ngoại. Cụ thể, siêu dữ liệu có thể bao gồm tên bảng hoặc khung nhìn; tên các trường (field name), còn được gọi là tên cột (column), cấu tạo nên bảng hoặc khung nhìn cùng với kiểu dữ liệu, kích thước và cho phép trống (NULL) hay không; thông tin khoá; thông tin khoá ngoại; truy vấn định nghĩa các khung nhìn. Mối quan hệ giữa các bảng được xác định thông qua điều kiện của phép nối giữa các bảng trong truy vấn cùng với thông tin về khoá của các bảng.

Mỗi bảng, khung nhìn sẽ được biểu diễn bằng một lớp với tên gọi là thực thể (Entity). Tập hợp các thực thể biểu diễn mô hình khái niệm của CSDL. Việc ánh xạ giữa cấu trúc lưu trữ CSDL và mô hình khái niệm được quản lý bởi những quy ước trong Entity Framework [7].

Ngoài ra, toàn bộ siêu dữ liệu của CSDL là những thông tin có dạng phân cấp, các bảng, khung nhìn và cột có mối quan hệ với nhau. Với ngôn ngữ XML, dữ liệu được lưu trữ dưới dạng cây [8], nên sử dụng XML để lưu lại các siêu dữ liệu. Việc đọc và ghi thông tin bằng XML rất thuận lợi và nhanh chóng thông qua các thư viện được hỗ trợ trong C#.

D. *Sinh mã giao diện sử dụng cú pháp Razor View Engine*

Razor View Engine được sử dụng lần đầu tiên trong ASP.NET MVC 3. Razor sử dụng Layout Page làm trang chính và View Page, View Page with Layout làm trang con. Razor có cú pháp đơn giản, sử dụng ký tự `@` để bắt đầu viết mã C# (VB.NET). Điều này giúp cho việc lập trình giao diện trở nên nhanh hơn, ví dụ: để hiển thị một đoạn dữ liệu ta chỉ cần viết `@obj.Attribute` hoặc viết trong khối lệnh `@if (foo) {<p>Plain Text</p>}`. Ngoài ra, Razor hỗ trợ kiểm thử đơn vị cho các view mà không cần controller hoặc chạy ứng dụng trên máy chủ web.

Với cú pháp Razor, chương trình sinh mã cần nhận biết được vị trí mã nguồn C# và vị trí mã nguồn HTML để tiến hành sinh mã chính xác.

E. Quá trình sinh mã

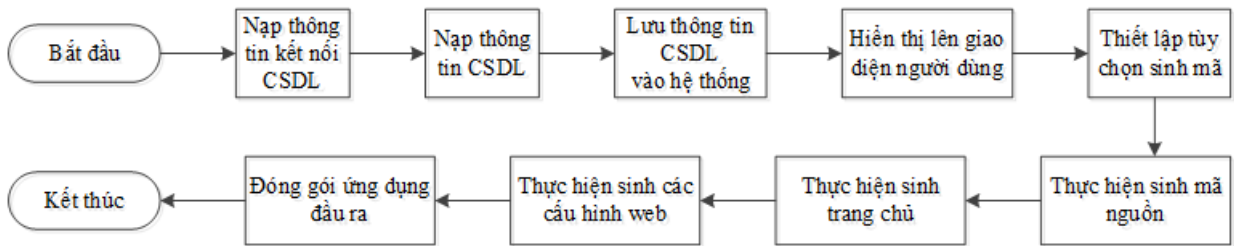
Quá trình sinh mã nguồn được tiến hành theo quy trình chuẩn, bao gồm các bước thu thập, trích lọc, chuyển đổi thông tin CSDL về dạng phù hợp, áp dụng các quy tắc sinh mã và nạp khuôn mẫu.

Các thông tin CSDL đầu vào phục vụ cho việc sinh mã được lấy từ bất kỳ CSDL quan hệ nào trên nền tảng SQL Server.

Các khuôn mẫu có sẵn là một đầu vào khác của hệ thống. Các khuôn mẫu bao gồm các nhóm: nhóm khuôn mẫu view (.cshtml), nhóm khuôn mẫu controller (.cs), nhóm khuôn mẫu model (.cs), nhóm khuôn mẫu entities (.cs). Các khuôn mẫu được tổ chức lưu trữ thành các tệp trên đĩa cứng. Nội dung của các khuôn mẫu gồm hai phần, đó là phần không thay đổi (phần tĩnh) và phần thay đổi (phần động) trong suốt quá trình sinh mã. Khuôn mẫu được nạp ngay khi hệ thống có được các thông tin CSDL cần thiết và tùy vào việc sinh mã cho nhóm nào thì nhóm khuôn mẫu tương ứng sẽ được nạp.

Việc sinh mã là công việc rất phức tạp, đòi hỏi phải trải qua nhiều bước và tuân thủ nhiều quy tắc. Tác giả đề xuất xây dựng một công cụ riêng để thực hiện việc này bằng ngôn ngữ lập trình C#.

Sau khi nghiên cứu các kỹ thuật sinh mã nguồn tự động, tác giả đề xuất cấu trúc của hệ thống bao gồm các thành phần: giao diện cấu hình CSDL, giao diện tùy chọn sinh mã và ghi vết chương trình, giao diện cấu hình đầu ra, giao diện tạo khung nhìn và thực thi truy vấn, giao diện người dùng, bộ sinh mã, các khuôn mẫu, các gói tiện ích, biên dịch và đóng gói. Các thành phần này hoạt động độc lập với nhau và được điều khiển bởi bộ sinh mã để thực hiện chức năng sinh mã của hệ thống.



Hình 1. Sơ đồ hoạt động của hệ thống

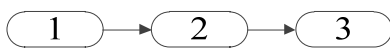
Tác giả sử dụng công cụ Visual Studio 2013 và .Net Framework 4.5 để xây dựng các thành phần của hệ thống có cách thức hoạt động được mô tả trên hình 1.

Trong đó, bước thứ 6 – thực hiện sinh mã nguồn gồm nhiều chức năng. Các chức năng đó là gọi tiện ích đọc tệp và nạp khuôn mẫu phù hợp; truyền thông tin CSDL từng bảng và từng khung nhìn vào khuôn mẫu; xuất ra kết quả dạng chuỗi từ khuôn mẫu đã nạp dữ liệu; ghi tệp kết quả ra đĩa cứng; ghi nhật ký cho quá trình sinh mã cho từng bảng và từng khung nhìn.

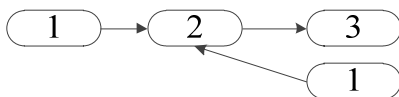
F. Sinh mã nguồn cho khung nhìn và truy vấn

Hệ thống sinh mã nguồn quan tâm đến một số cấu trúc đặc biệt của các truy vấn tạo khung nhìn từ CSDL, truy vấn tạo khung nhìn do người dùng định nghĩa trực tiếp trên hệ thống sinh mã nguồn hoặc các truy vấn. Khi các bảng tham gia tạo khung nhìn có các mối quan hệ theo một quy tắc nào đó, nếu hệ thống phát hiện được các cấu trúc này thì hoàn toàn có thể tạo các chức năng thêm, cập nhật, xóa dữ liệu. Sau khi nghiên cứu nhiều truy vấn, tác giả nhận thấy các truy vấn có cấu trúc chung rơi vào các trường hợp sau đây:

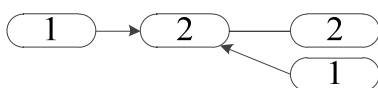
Trường hợp 1:



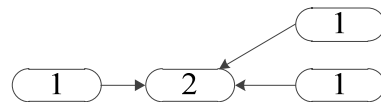
Trường hợp 3:



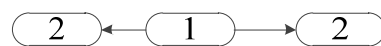
Trường hợp 5:



Trường hợp 2:



Trường hợp 4:



Các hình vẽ trên đây thể hiện xu hướng mối quan hệ của các bảng tham gia vào truy vấn. Tất nhiên, mô hình có thể được mở rộng. Trong đó, mỗi hình ô-van cùng với hệ số được gắn kèm đại diện cho một bảng dữ liệu, mũi tên có hướng biểu thị mối quan hệ một – nhiều, đường nối không có hướng biểu thị mối quan hệ một – một giữa hai bảng. Việc xác định mối quan hệ được thực hiện dựa trên thông tin khoá của các bảng tham gia vào truy vấn của các cột tham gia vào điều kiện phép nối. Thông tin về khoá ngoại chỉ được dùng để khuyến cáo tạo điều kiện phép nối trong quá trình thiết kế truy vấn. Nếu một đầu là khoá và đầu kia không phải là khoá, thì mối quan hệ là một – nhiều. Nếu cả hai đầu đều là khoá, thì mối quan hệ là một – một. Việc lựa chọn bảng để sinh mã thực hiện thao tác thêm, sửa, xoá được khuyến cáo theo mức độ ưu tiên tỉ lệ thuận với hệ số của mỗi bảng. Để có thể thực hiện gắn hệ số cho mỗi bảng, bài viết đề xuất xây dựng ma trận vuông $n \times n$, với n – số lượng bảng tham gia vào truy vấn, để biểu diễn đồ thị thể hiện mối quan hệ giữa các bảng. Mỗi phần tử trong ma trận biểu thị mối quan hệ giữa hai bảng đứng ở hàng và cột giao nhau. Mỗi bảng trong mối quan hệ một – nhiều được đánh số sao cho giá trị của bảng này bằng tổng hoặc bằng hiệu của bảng liền kề với 1 và phải luôn lớn hơn 0 theo nguyên tắc: cộng cho 1 nếu đi cùng hướng mũi tên, trừ cho 1 nếu ngược lại. Đối với trường hợp mối quan hệ một – một, hai bảng có giá trị bằng nhau.

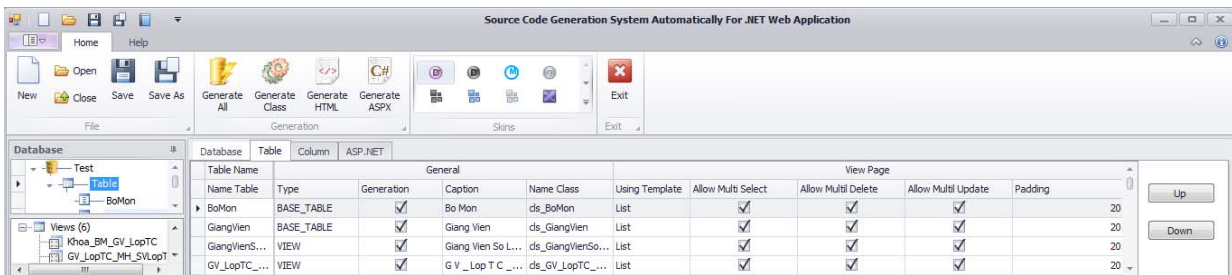
Đồ thị thu được sau khi đã gắn hệ số cho các bảng được dùng vào hai mục đích, đó là để khuyến cáo người dùng chọn bảng để sinh mã nguồn thực hiện các thao tác dữ liệu và xác định các cột có thể tham gia vào trang web thao tác dữ liệu cũng như vai trò của nó. Thứ nhất, những bảng có hệ số cao hơn sẽ được khuyến cáo với mức độ cao hơn. Thứ hai, một khi một bảng có hệ số k được chọn để sinh mã nguồn thực hiện thao tác dữ liệu, i) chỉ các cột thuộc bảng đó mới có thể tham gia vào trang web với vai trò tiếp nhận giá trị mới (editable) hoặc hiển thị giá trị đã có; ii) các cột thuộc bảng có hệ số nhỏ hơn hoặc bằng k có thể xuất hiện trong trang web thao tác dữ liệu, nhưng chỉ với vai trò hiển thị giá trị; iii) các cột thuộc bảng có hệ số lớn hơn k không được phép chọn cho vào trang web thao tác dữ liệu.

Với trường hợp truy vấn không bao gồm các hàm thống kê, chỉ bao gồm phép nối, phép chọn và phép chiếu, hệ thống luôn đề xuất lấy bộ giá trị khoá của bảng để gắn liền với mỗi bản ghi. Khi khoá không có mặt trong mệnh đề SELECT trong truy vấn, người dùng không thể chọn để cho phép hiển thị, nhưng luôn tồn tại các đối tượng ẩn được sinh ra chứa các bộ giá trị khoá đó gắn liền với mỗi bản ghi được hiển thị. Việc này nhằm mục đích hỗ trợ các thao tác sửa và xoá dữ liệu. Với trường hợp truy vấn đầu vào bao gồm các hàm thống kê, bài viết chỉ xét các bảng tham gia vào truy vấn với các cột đều nằm trong mệnh đề GROUP BY và trong số các cột đó phải bao gồm các cột tạo nên khoá chính hoặc một khoá độc nhất nào đó.

III. KẾT QUẢ VÀ ĐÁNH GIÁ

A. Kết quả

Hệ thống sinh tự động mã nguồn được thiết kế để sinh mã nguồn cho các CSDL quan hệ khác nhau. Trước tiên, phải thiết lập cấu hình kết nối CSDL thông qua giao diện hệ thống. Người dùng nhập thông tin máy chủ CSDL, thể hiện của hệ quản trị CSDL, tên CSDL, công kết nối, tên đăng nhập và mật khẩu. Tất nhiên, yêu cầu đối với tài khoản đăng nhập phải có đủ quyền để có thể đọc được siêu dữ liệu. Sau khi hệ thống kết nối thành công đến CSDL thì giao diện sẽ tương tự như hình 2.

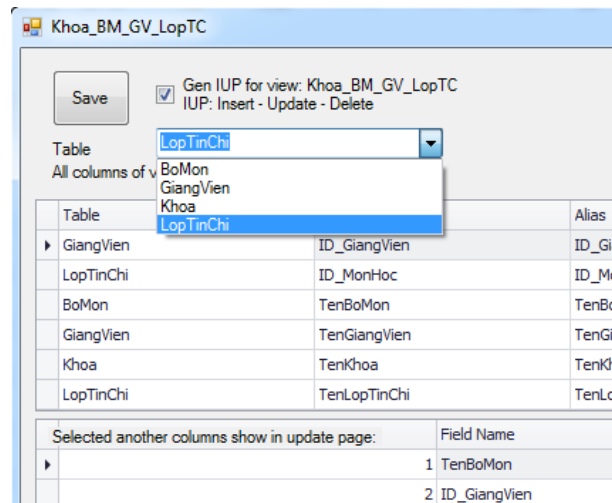


Hình 2. Giao diện chính của chương trình

Trên giao diện này, người dùng có thể thấy phía bên trái là danh sách các bảng và khung nhìn có trong CSDL. Bên phải là tùy chọn cấu hình cho từng bảng. Người dùng có thể lựa chọn khuôn mẫu bố trí cho từng trang web ở thông số “Using Template”. Người dùng có thể định nghĩa các khuôn mẫu bố trí trang web khác nhau ngoài hai khuôn mẫu phổ biến đã được hệ thống cung cấp sẵn, đó là bố trí theo dạng bảng chi tiết (list) và bố trí theo kiểu ma trận (matrix). Danh sách các khuôn mẫu bố trí tự động được nạp khi có khuôn mẫu mới. Khi người dùng nhấn chuột vào từng bảng cụ thể sẽ xuất hiện giao diện cấu hình thông tin cho các cột của bảng như trên hình 3.

Field Name	Data Type	Size	Primary Key	Identity	Caption	Viewable	Nullable	Validate	Control Edit	Allow Search	Search Option
GioiThieu	nvarchar	200	<input type="checkbox"/>	<input type="checkbox"/>	Gioi Thieu	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IsString	<input checked="" type="checkbox"/> TextBoxSi...	<input checked="" type="checkbox"/>	Like
ID_BoMon	int	4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	I D _Bo Mon	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IsInt	<input checked="" type="checkbox"/> TextBoxSi...	<input checked="" type="checkbox"/>	=
ID_Khoa	int	4	<input type="checkbox"/>	<input type="checkbox"/>	I D _Khoa	<input checked="" type="checkbox"/>	<input type="checkbox"/>	IsInt	<input checked="" type="checkbox"/> Dropdown...	<input checked="" type="checkbox"/>	=
TenBoMon	nvarchar	50	<input type="checkbox"/>	<input type="checkbox"/>	Ten Bo Mon	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	IsString	<input checked="" type="checkbox"/> TextBoxSi...	<input checked="" type="checkbox"/>	Like

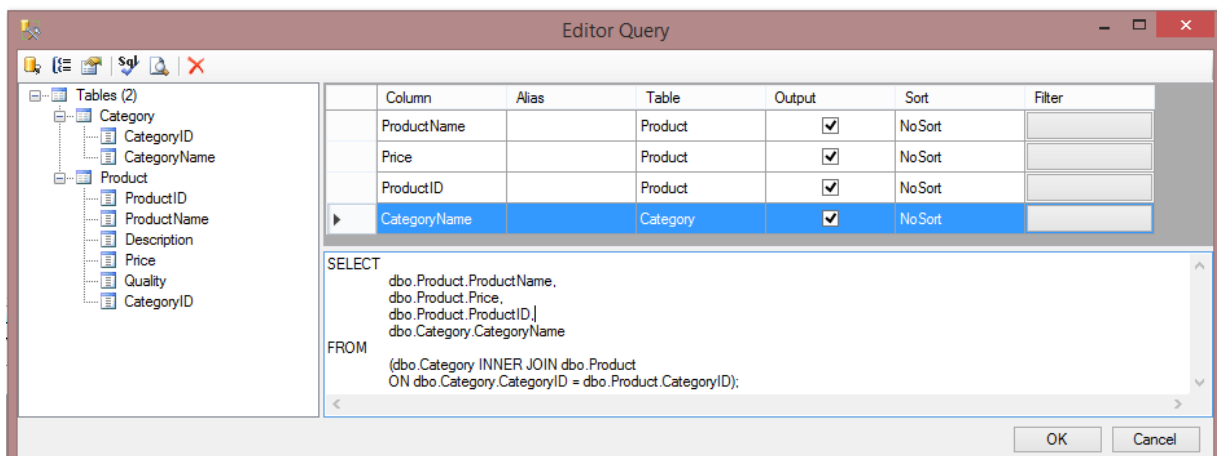
Hình 3. Giao diện tùy chỉnh cột



Hình 4. Giao diện tùy chọn bảng để thao tác dữ liệu

Hình 4 cho thấy hệ thống khuyến cáo lựa chọn bảng LopTinChi để tạo trang web thao tác dữ liệu đối với khung nhìn Khoa_BM_GV_LopTC sử dụng 4 bảng theo thứ tự tăng dần của hệ số (xem II.F): Khoa, BoMon, GiangVien và LopTinChi. Trên trang web này, có thể thao tác dữ liệu trên các cột thuộc bảng LopTinChi. Các cột từ các bảng còn lại chỉ có thể được chọn để hiển thị trong chế độ chỉ đọc. Nếu người dùng chọn bảng BoMon, thì tương ứng sẽ là bảng Khoa; người dùng không thể chọn các cột từ các bảng GiangVien và LopTinChi để hiển thị trên trang web thao tác dữ liệu bảng BoMon. Đối với các trang web hiển thị dữ liệu cho khung nhìn hoặc truy vấn, hệ thống cung cấp tùy chọn để cho phép thao tác dữ liệu trên bảng bất kỳ tham gia vào khung nhìn hoặc truy vấn. Khi đó, người dùng sẽ được điều hướng sang trang web khác cho phép thao tác dữ liệu với bảng có trường được chọn, trên đó có thể hiển thị các cột từ các bảng có hệ số thấp hơn.

Ngoài ra, hệ thống cho phép người dùng tạo khung nhìn bằng công cụ thiết kế truy vấn cơ bản. Người dùng chỉ cần chọn và kéo thả bảng hoặc cột vào lưới thì sẽ tự động sinh ra truy vấn như trên hình 5.



Hình 5. Giao diện thiết kế truy vấn

Đối với các truy vấn được tạo ra trực tiếp trên giao diện hỗ trợ viết truy vấn và các truy vấn tạo nên khung nhìn trong CSDL, chương trình đề xuất giao diện (hình 4) chọn bảng để sinh các trang web thực hiện thêm, cập nhật, xóa. Theo nguyên tắc đã đưa ra ở trên, tùy thuộc vào bảng được chọn, tại giao diện này có thể chọn các cột khác nhau từ các bảng khác nhau tham gia vào trang web thao tác dữ liệu và có một số cột từ một số bảng có thể không nằm trong danh sách lựa chọn này.

B. Đánh giá

Kết quả đạt được khi ứng dụng hệ thống sinh tự động mã nguồn cho thấy hệ thống đã sinh ra được mã nguồn _CSHTML, CS và các cấu hình cần thiết cho ứng dụng. Ứng dụng web được sinh ra chạy theo mô hình MVC. Với mô hình MVC, người dùng có thể dễ dàng kiểm thử, nâng cấp và chỉnh sửa chức năng giúp hệ thống hoạt động ổn định hơn. Hệ thống cũng đã đưa ra nhiều tùy chọn cấu hình, tạo nên một sự mềm dẻo, linh hoạt trong quá trình sinh mã. Đồng thời, hệ thống sinh mã đã khắc phục được các hạn chế của các hệ thống trên thị trường hiện nay.

Về mặt hiệu năng, tuy chưa tiến hành kiểm tra các website được sinh ra bằng các công cụ chuyên dụng, nhưng có thể khẳng định rằng, chúng không thể có hiệu năng tốt như trường hợp website được xây dựng thủ công hoàn toàn với sự chú tâm tối ưu trong xử lý đến từng lệnh SQL đến CSDL.

IV. KẾT LUẬN

Trong phạm vi nghiên cứu, tác giả đã sử dụng thư viện StringTemplate để xây dựng được hệ thống sinh tự động mã nguồn với đầu vào là một CSDL quan hệ bất kỳ. Đầu ra của chương trình là một ứng dụng web theo mô hình MVC với các chức năng tương tác dữ liệu trên các bảng và các khung nhìn như thêm mới, cập nhật, xóa, tìm kiếm và trình bày dữ liệu.

Hệ thống sinh tự động mã nguồn về căn bản đã giải quyết được vấn đề sinh mã nguồn toàn diện cho ứng dụng. Ngoài khả năng sinh mã như các hệ thống khác trên thị trường, hệ thống đã khắc phục một số nhược điểm của các hệ thống hiện có trên thị trường bằng các tính năng như khả năng chọn khuôn mẫu riêng biệt cho từng trang web, tùy chỉnh bố trí cho từng trang web, sinh mã thao tác dữ liệu cho truy vấn và khung nhìn. Nghiên cứu đã đề xuất các thuật toán để xác định bảng đề khuyến cáo người dùng chọn bảng để sinh mã nguồn thực hiện các thao tác dữ liệu và xác định các cột có thể tham gia vào trang web thao tác dữ liệu cũng như vai trò của nó trên giao diện thao tác dữ liệu. Trên cơ sở đó, nghiên cứu xây dựng tính năng sinh mã nguồn thao tác dữ liệu cho trường hợp truy vấn hoặc khung nhìn từ CSDL. Chương trình cũng cung cấp công cụ thiết kế truy vấn thân thiện.

Tuy nhiên, mã nguồn được sinh ra chỉ chạy ổn định nếu sử dụng công nghệ .NET 4.5. Việc sinh mã nguồn thao tác dữ liệu cho trường hợp các truy vấn chỉ dừng lại ở các truy vấn nối trong (INNER JOIN). Đối với truy vấn nối ngoài (OUTER JOIN), truy vấn lồng và truy vấn đệ quy, hệ thống chỉ có thể sinh mã nguồn các trang web hiển thị, chưa sinh được trang web thao tác dữ liệu. Hệ thống cũng chưa hỗ trợ các tính năng cho phép người dùng lập trình xử lý sự kiện trên các đối tượng khác nhau. Đây chắc chắn là hướng phát triển của nghiên cứu.

V. TÀI LIỆU THAM KHẢO

- [1] Jack Herrington, “Code Generation”, <http://onjava.com/onjava/2003/09/03/generation.html>. (truy cập: 12/01/2013)
- [2] “ASPMaker v.12”, <http://www.hkvstore.com/ASPMaker/>. (truy cập: 24/05/2015)
- [3] “CodeSmith Generator”, <http://www.codesmithtools.com/product/generator>. (truy cập: 24/05/2014)
- [4] Nguyễn Trần Quốc Vinh, Nguyễn Văn Vương, “Nghiên cứu xây dựng hệ thống sinh tự động mã nguồn các website .ASPX trên cơ sở dữ liệu”, Tạp chí Khoa học và Công nghệ - Đại học Đà Nẵng, số 8(69), tr. 182-188. 2013.
- [5] Terence Parr, “StringTemplate”, <http://wwwantlr.org/wiki/display/ST4/StringTemplate+4+Documentation>. (truy cập: 14/05/2015)
- [6] Collin Fagan, “Generating a Builder Class using StringTemplate”, <https://www.java.net/blogs/aberrant>. (truy cập: 05/01/2015)
- [7] Morgridge Dane, “Entity Framework 4.1, Code First”, <http://www.codemag.com/Article/1108051>. (truy cập: 11/5/2015)
- [8] Doug Tidwell, “Giới thiệu về XML”, <http://www.ibm.com/developerworks/vn/edu/xmlintro>. (truy cập: 20/04/2015)

BUILDING THE APPLICATION THAT GENERATES CODE AUTOMATICALLY FOR ASP.NET MVC WEBSITES

Nguyen Tran Quoc Vinh, Nguyen Van Vuong, Nguyen Ngoc Cuong

ABSTRACT - In this paper, the authors have done research on the theory of automatic code generating and apply this theory to build the software that can generate automatically code for ASP.NET MVC website. The input of the program is the database metadata, user's configurations and templates of web pages. The output is the .NET MVC web application. The structure of the templates is based on the standard of stringtemplate library. The generated websites can manipulate data such as query, insert, update, delete records on tables directly as well as through the database views and user's defined views. For the database views or user's defined queries, the paper puts forward algorithm to identify table for updating, inserting and deleting data. The user can choose template for each web page. The created by the program website can be directly implemented on IIS webserver without any modification.

Keywords - automatic source code generating, templates, ASPX web page, MVC model, data manipulation, query and view.