

Xây dựng và thử nghiệm công cụ CODEGER-UWE phát triển ứng dụng Web hướng mô hình

CODEGER-UWE - Model Driven Web Application Development Tool

*Trần Đình Diễn, Huỳnh Quyết Thắng, Bùi Thị Mai Anh,
Hoàng Thành Đạt, Nguyễn Phi Lê**

*Trường Đại học Bách khoa Hà Nội, Hà Nội, Việt Nam
Email: lenp@soict.hust.edu.vn*

Tóm tắt

CODEGER-UWE là công cụ hỗ trợ phát triển ứng dụng web hướng mô hình. Trong bài báo này, chúng tôi trình bày kiến trúc công cụ CODEGER-UWE. Công cụ đã được tích hợp các quy tắc chuyển đổi mô hình và sinh mã để tạo ứng dụng web cho nền tảng JSF một cách tự động. CODEGER-UWE được triển khai như một plugin cho công cụ MagicDraw cho phép sinh mã nguồn từ các mô hình UWE tương ứng với các thành phần trong mô hình Web MVC. Đây là một công cụ với quy trình phát triển rõ ràng, tiện dụng, áp dụng công cụ để phát triển ứng dụng web sẽ làm giảm rất nhiều thời gian cho việc phát triển ứng dụng Web

Từ khóa: CODEGER-UWE, phát triển phần mềm hướng mô hình, công cụ phát triển ứng dụng web, sinh mã tự động

Abstract

In this paper, we present a CODEGER-UWE tool to support model-driven web application development. The CODEGER-UWE tool has integrated rules for transformation model and code generation rules to create web applications for the JSF platform automatically. CODEGER-UWE implemented as a plugin for the MagicDraw tool which allows the generation of source code from UWE models that correspond to the components of the MVC model. This is a tool with a clear, handy development process, applying tools to develop web applications that will greatly reduce the time for developing Web applications.

Keywords: CODEGER-UWE, model driven software development, web application development tools, automated code generation

1. Giới thiệu

Công cụ hỗ trợ đầy đủ là yếu tố quan trọng quyết định thành công của phương pháp hoặc kỹ thuật phát triển phần mềm nói chung và của ứng dụng web nói riêng [1]. Việc sử dụng công cụ sinh mã sẽ cải thiện quy trình phát triển phần mềm và làm tăng chất lượng của mã được tạo ra [2]. Đối với phương pháp UWE (UML-based Web Engineering) cũng đã có một số công cụ như ArgoUWE, UWE4JSF [3], đây là công cụ tự động sinh mã ứng dụng Web với nền tảng JSF (JavaServer Faces) từ mô hình UWE, bao gồm một chuỗi các plugin được tích hợp trong môi trường mô hình hóa Eclipse EMF (Eclipse Modeling Framework), được xây dựng và đóng gói thành các thư viện Java (.jar), đây là minh chứng cho khả năng sinh mã nguồn cho ứng dụng Web hướng mô hình theo kỹ thuật UWE, nhưng hiện tại đã không còn được tiếp tục phát triển, và không thể vận hành với các phiên bản mới của Eclipse cũng như UWE profile. Công cụ MagicUWE hỗ trợ thiết kế hệ thống ứng dụng web với phương pháp UWE [4], giúp phát triển ứng dụng web các hoạt động lập mô hình bằng cách sử dụng cấu hình UWE và chuyển đổi giữa các

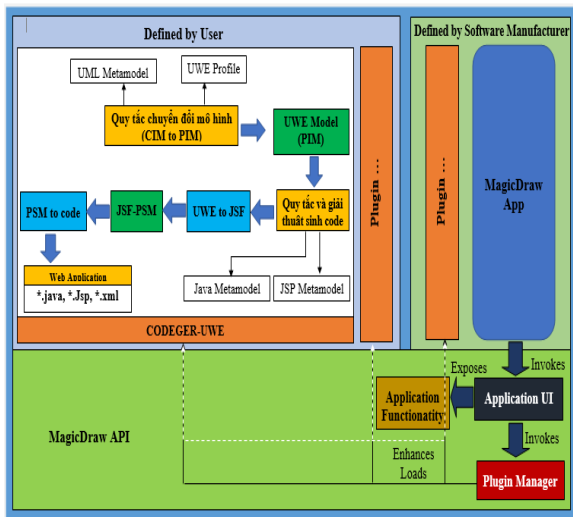
mô hình UWE. MagicUWE được triển khai như một plugin cho công cụ MagicDraw. Tuy nhiên, công cụ này chủ yếu tập trung vào chuyển đổi mô hình, chưa tập trung cho việc sinh mã. Trong bài báo này, chúng tôi trình bày công cụ CODEGER-UWE, một plugin được chúng tôi xây dựng và tích hợp trong phần mềm MagicDraw cho phép sinh mã nguồn từ bốn mô hình UWE tương ứng với các thành phần trong mô hình Web MVC (Model-view-controller) [5].

Quá trình phát triển của UWE là phát triển từ quy trình thủ công thông qua quy trình định hướng mô hình bán tự động (dựa trên các loại biến đổi mô hình khác nhau [6]). Áp dụng những nguyên lý của MDA, mục tiêu tiếp cận của UWE là xây dựng các tập của CIM (Computation Independent Model), PIM (Platform-Independent Model) và PSM (Platform-Specific Model) là kết quả của việc phân tích, thiết kế và cài đặt quá trình định hướng mô hình. Mục đích giai đoạn phân tích là tập hợp được một tập các yêu cầu ổn định, các yêu cầu chức năng sẽ được mô tả lại bởi ý nghĩa của các mô hình yêu cầu. Mô hình yêu cầu bao gồm các 'use cases' chuyên biệt và các 'class model' cho ứng dụng Web. Giai đoạn thiết kế bao gồm việc xây dựng một loạt các mô hình ở mức độ độc lập nền tảng. Sau đó các mô hình thiết kế được chuyển đổi để thực hiện trên các nền tảng cụ thể [7].

Quy trình bắt đầu bằng mức mô hình nghiệp vụ (CIM) định nghĩa một mô hình yêu cầu. Các mô hình thiết kế độc lập nền hệ thống (các PIM) được dẫn xuất từ mô hình yêu cầu này. Tập các mô hình thiết kế này biểu diễn các mối quan tâm khác nhau của các ứng dụng Web. Nó bao gồm nội dung (content), điều hướng (navigation), xử lý (process), trình bày (presentation). Trong các nghiên cứu [6, 8-10] các tác giả đã đề xuất các quy tắc chuyển đổi mô hình từ CIM sang PIM sang PSM và [11] bổ sung bộ quy tắc chuyển đổi từ mô hình yêu cầu sang mô hình xử lý và mô hình trình bày, nhằm cải thiện quá trình chuyển đổi tự động cho các ứng dụng web sử dụng UWE. Trong [6] đã đề xuất chuyển đổi mô hình hỗ trợ cho nhà thiết kế các hoạt động tạo lập mô hình bằng cách sử dụng cấu hình UWE và tạo mô hình bán tự động. Tuy nhiên, còn nhiều thành phần của mô hình vẫn phải xây dựng lại thủ công, việc sinh code mới chỉ dừng lại ở việc tạo ra các khung để lập trình viên viết code. Chính vì vậy, chúng tôi dựa vào các quy tắc chuyển đổi mô hình trong [11, 15] để đề xuất xây dựng một số quy tắc, giải thuật sinh code tự động nhằm cải tiến, nâng cao hiệu quả chuyển đổi từ mô hình sang code với công cụ CODEGER-UWE.

Nội dung còn lại của bài báo được cấu trúc như sau: Phần 2 trình bày cấu trúc của công cụ CODEGER-UWE, phần 3 thử nghiệm và đánh giá CODEGER –UWE, phần 4 kết luận và đề xuất.

2. Kiến trúc của công cụ CODEGER-UWE

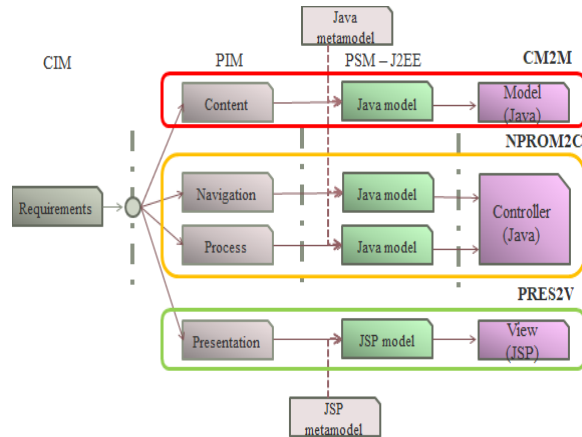


Hình 1. Kiến trúc CODEGER –UWE

Trong [11,15] chúng tôi bổ sung 11 quy tắc chuyển đổi mô hình CIM sang PIM. Trong [15], chúng tôi cũng đề xuất 10 quy tắc cùng với 03 giải thuật chuyển đổi từ PIM sang PSM và từ PSM sang code:

- CM2M (Content Model to Model Code Generation)
- CM2V (Presentation Model to View Code Generation)

- NPROM2C (Navigation and Processing Model to Controller Code Generation)
- Kiến trúc của công cụ CODEGER –UWE được thể hiện tại Hình 1.

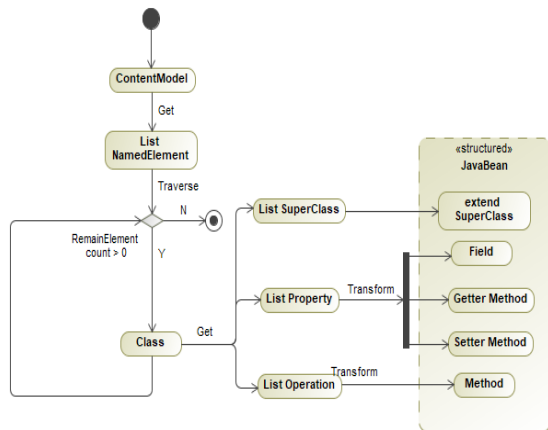


Hình 2. Chuyển đổi mô hình PIM sang PSM, mã nguồn trong CODEGER -UWE

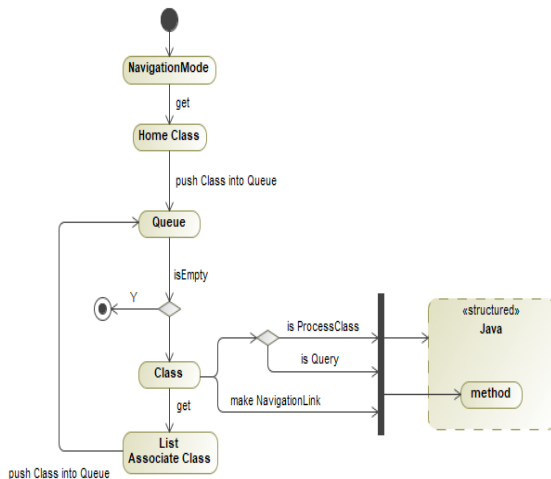
Chuyển đổi mô hình được phân chia thành một vài mô-đun. Công việc cần làm là thiết lập từng mô-đun cho mỗi công đoạn chuyển đổi mô hình UWE. Mô-đun các quy tắc chuyển đổi mô hình (CIM to PIM). Để chuyển đổi từ CIM sang PIM, đầu vào là mô hình yêu cầu (CIM), dựa trên UML Metamodel, UWE Metamodel và UWE Profile được ánh xạ với nhau với mô hình Trong đó, mô hình nội dung (content) chúng tôi đã tích hợp các luật chuyển đổi C2O, S2O; Mô hình điều hướng (navigation) các luật C2N, P2N, A2N; Mô hình xử lý (process) các luật U2C, S2C, U2U, S2S; Mô hình biểu diễn (presentation) gồm các luật D2G, P2E. UWE Profile và UWE Meta-model được ánh xạ với nhau, chúng gồm các phần có quan hệ tuyến tính với nhau. Ngoài ra, UWE Meta-model còn là sự mở rộng của UML Meta-model nên việc chuyển đổi gồm các luật (quy tắc) chuyển đổi khuôn mẫu UML (stereotype) sang các thành phần tương ứng của UWE Meta-model và các thành phần UML cơ bản được lồng ghép vào. Trong Hình 1, mô-đun các quy tắc và giải thuật sinh code, được chúng tôi tích hợp quy tắc CM2M sinh mã thành phần Model từ mô hình nội dung; Quy tắc PRES2V sinh mã thành phần View từ mô hình trình bày và Quy tắc NPROM2C sinh mã thành phần Controller từ mô hình điều hướng và xử lý. Chuyển đổi từ CIM sang PIM đơn thuần chỉ là ánh xạ giữa mô hình UML và mô hình UWE. Đối với chuyển đổi PIM sang PSM, trong CODEGER-UWE áp dụng phương pháp khuôn mẫu và metamodel, kết hợp với sự tương đồng về mặt định nghĩa của bốn mô hình trong UWE là mô hình nội dung, điều hướng, xử lý, trình bày với ba thành phần trong mô hình MVC, nhóm nghiên cứu đề xuất quy trình chuyển đổi MDA (Model-Driven Architecture) cho kỹ thuật UWE như Hình 2.

Mô hình nội dung (mức PIM) được chuyển đổi sang JavaModel (mức PSM), là một thể hiện của Java metamodel, sau đó chuyển đổi sang mã nguồn Java, tương ứng thành phần Model với giải thuật chuyển đổi có tên là CM2M.

Mô hình điều hướng và xử lý được chuyển đổi sang JavaModel, sau đó được chuyển đổi sang mã nguồn Java, tương ứng thành phần Controller (Hình 3). Giải thuật chuyển đổi có tên NPROM2C; Mô hình trình bày được chuyển đổi sang JSPModel, là một thể hiện của JSP metamodel, sau đó giải thuật PRES2V chuyển đổi sang mã JSP, tương ứng thành phần View (Hình 4, 5).



Hình 3. Quy trình sinh mã nguồn từ mô hình nội dung



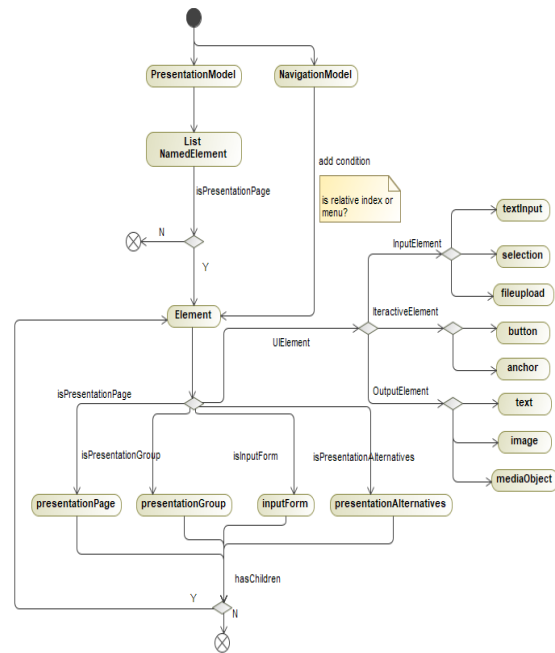
Hình 4. Quy trình sinh mã nguồn từ mô hình điều hướng

3. Thử nghiệm và đánh giá CODEGER –UWE

Công cụ CODEGER –UWE đã được hoàn thiện. Toàn bộ mã nguồn được cài đặt, lưu trữ trên Github [16], tuân thủ các quy định phần mềm nguồn mở. Trong công cụ này, quá trình chuyển đổi mô hình nền tảng độc lập PIM sang mô hình nền tảng cụ thể PSM là quá trình chuyển đổi được xây dựng trên một nền tảng Web cụ thể và các luật chuyển đổi được xây

dựng tương ứng với mô hình MVC, trong đó những mối quan tâm của một ứng dụng Web tương ứng với vai trò của Model (nội dung), View (trình bày) và Controller (điều hướng và xử lý). Mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác, giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp, bảo trì.

Để thử nghiệm và đánh giá công cụ CODEGER –UWE chúng tôi áp dụng ví dụ Music Portal [5]. Đây là một ứng dụng về công nghệ thông tin âm nhạc cho phép người dùng mua album và sau đó có thể được tải xuống dưới dạng tệp MP3, được sử dụng nhiều để thử nghiệm các thuật toán chuyển đổi mô hình



Hình 5. Quy trình sinh mã JSP từ mô hình trình bày
Quy trình thực hiện như sau:

Bước 1: Xác định đầu vào của công cụ

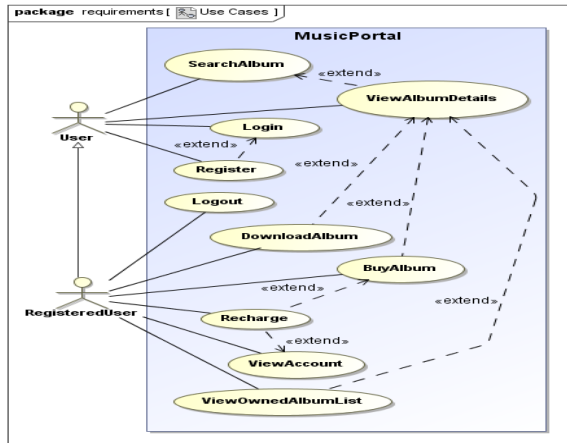
Các yêu cầu trong UWE được mô hình hóa bằng biểu đồ use case, CODEGER –UWE cho phép chuyển đổi từ mô hình yêu cầu (Use case) sang mô hình nội dung, điều hướng, xử lý và trình bày với các quy tắc chuyển đổi đã được chúng tôi bổ sung, phát triển, giúp cho quá trình chuyển đổi mô hình hiệu quả hơn.

Bước 2: Thực hiện chuyển đổi mô hình

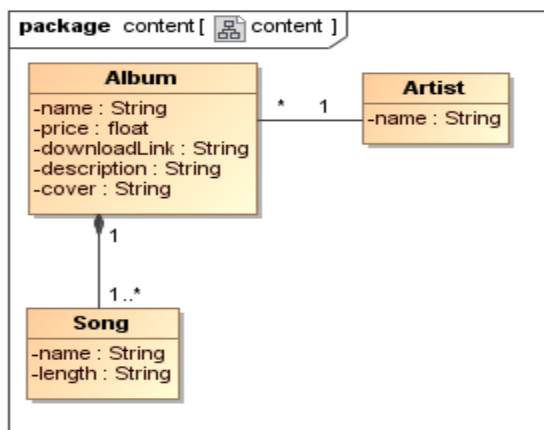
Lần lượt thực hiện chuyển đổi mô hình và thu được các kết quả như trong Hình 7, 8, 9. Trong khuôn khổ bài báo không liệt kê mô hình trình bày.

Bước 3: Thực hiện sinh mã sử dụng công cụ CODEGER –UWE.

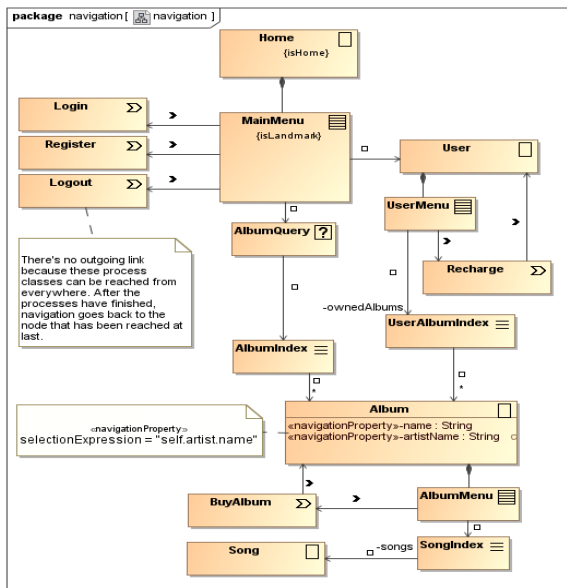
Công cụ đã chuyển đổi từ mô hình thực thi sang mã nguồn Java tự động tương ứng với nguyên tắc xây dựng các thành phần Model, View, Controller trong mô hình PSM.



Hình 6. Biểu đồ use case, mô hình đầu vào của ứng dụng

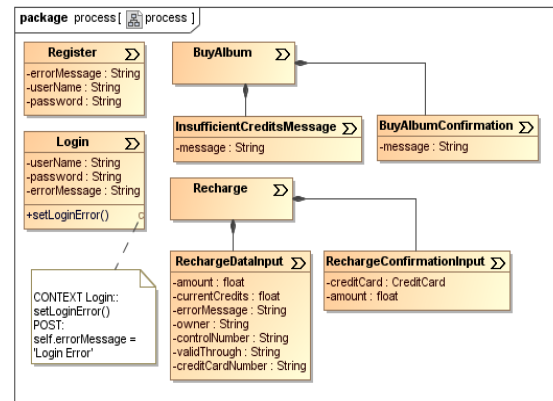


Hình 7. Mô hình nội dung

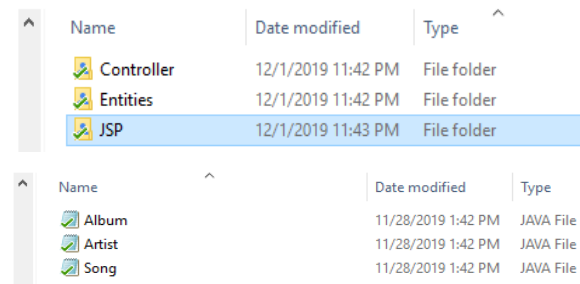


Hình 8. Mô hình Điều hướng

Kết quả thử nghiệm đã thu được một khung mẫu đơn giản của mô hình Web viết bằng mã nguồn Java (Hình 10). Trên cơ sở đó nhà phát triển ứng dụng có thể bổ sung thêm vào các file mã nguồn để xây dựng ứng dụng.

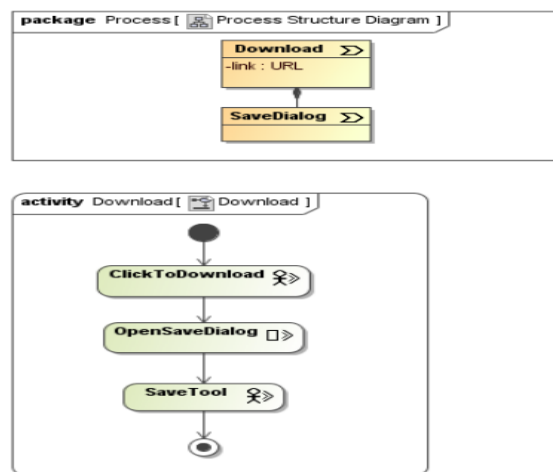


Hình 9. Mô hình Xử lý



Hình 10. File mã nguồn sinh ra từ mô hình UWE

Một trường hợp điển hình khác là ứng dụng UWE Website [14] thể hiện mô hình, các thành phần của trang web UWE, <http://uwe.pst.ifi.lmu.de/>. Mô hình này không có biểu đồ use case, song việc chuyển đổi mô hình sang mã nguồn Java vẫn có thể thực hiện. Sử dụng công cụ CODEGER-UWE, mã nguồn Java được sinh ra bằng cách chuyển đổi các mô hình nội dung, xử lý, điều hướng và trình bày. Kết quả thu được như ở Hình 11, 12, 13.



Hình 11. Mô hình xử lý của UWE Website

Trong đó, thư mục Controller chứa mã nguồn Process.java, là kết quả thu được từ việc chuyển đổi mô hình xử lý sang Java, Hình 12. Thư mục Entities chứa các thực thể, các đối tượng, lớp có trong mô hình nội dung, cùng với các thuộc tính và phương thức, như trong Hình 13. Thư mục JSP chứa mã

nguồn giao diện, file jsp là kết quả của việc chuyển đổi mô hình trình bày sang mã nguồn Java. File này đã chèn sẵn một số mã nguồn Java kết hợp với HTML.

```
package uwe.transformation.controller;
import java.util.*;

class SaveDialog{
}

class Download{
    private uwe.transformation.beans.URL _link;
    public uwe.transformation.beans.URL getLink(){
        return _link;
    }
    public void setLink(uwe.transformation.beans.URL _link){
        this._link=_link;
    }
}
```

Hình 12. Mã nguồn Process.java, sinh ra bởi chuyển đổi mô hình xử lý sang Java.



Hình 13. Các định nghĩa lớp được sinh ra bởi chuyển đổi mô hình nội dung sang Java.

5. Kết luận và đề xuất

Trong bài báo, chúng tôi đã trình bày công cụ CODEGER –UWE chuyển đổi mô hình và sinh mã tự động phát triển ứng dụng web từ mô hình UWE tương ứng với các thành phần trong mô hình Web MVC. Đây là một plugin được tích hợp trong phần mềm MagicDraw và là một phần mềm có tính mô-đun cao và dễ mở rộng. Trong thời gian tới chúng tôi sẽ hoàn thiện các quy tắc chuyển đổi cũng như sinh mã nguồn áp dụng vào các bài toán khác nhau và bổ sung thêm giải thuật cho mô hình tích hợp điều kiện OCL hướng tới việc cải thiện kết quả sinh mã cho phát triển ứng dụng web.

Lời cảm ơn

Nghiên cứu này được tài trợ bởi Trường Đại học Bách Khoa Hà Nội trong đề tài mã số T2018-PC-015.

Tài liệu tham khảo

[1] S. Jácome, J. M. Ferreira, and A. Corral, Software Development Tools in Model-Driven Engineering presented at the 5th International Conference in Software Engineering Research and Innovation, Mexico, 2017.

[2] E.V. Sunitha and P. Samuel, Object Constraint Language for Code Generation from Activity Models, Information and Software Technology, vol. Volume 103, pp. Pages 92-111, 2018.

[3] C. Kroiss, N. Koch, and I. Knapp, UWE4JSF: A Model Driven Generation Approach for Web Applications, presented at the In Proc. 9th Int. Conf. Web Engineering (ICWE'09), 2009.

[4] M. Busch and N. Koch, MagicUWE – A CASE Tool Plugin for Modeling Web Applications, in 9th International Conference, ICWE 2009, Spain, 2009, pp. 505-508.

[5] UWE – UML-based Web Engineering <http://uwe.pst.ifi.lmu.de/examplePhiloponella.html>

[6] K. Nora, Transformation Techniques in the Model-Driven Development Process of UWE, in sixth international conference on Web engineering California, USA, 2007, p. 10.

[7] K. Andreas, K. Alexander, and K. Nora, Model-Driven Generation of Web Applications in UWE, in 3rd International Workshop on Model-Driven Web Engineering, Italy, 2007, pp. 23-38.

[8] M. Busch and M. Á. G. d. Dios, ActionUWE: Transformation of UWE to ActionGUI Models, Institute for Informatics, Germany, 2012.

[9] L. Abdellatif, M. Chhiba, Abdelmoumen Tabyaoui, and O. Mjihil, Model driven architecture approach for application security integration, Journal of Theoretical and Applied Information Technology, vol. 8, p. 13, 30th April 2017.

[10] A. F. Subahi, A Business User Model-Driven Engineering Method for Developing Information Systems, Department of Computer Science, University of Sheffie, Englan <https://github.com/zbloodz/CODEGER-UWE> d, UK, 2015.

[11] T. D. Dien, H. Q. Thang, and T. Q. Khanh, Development of The Rules for Model Transformation with OCL Integration in UWE, in 7th International Conference on Frontiers of Intelligent Computing: Theory and Application (FICTA 2018), Da Nang, Viet Nam, 2018.

[12] K. Andreas, Model Driven Software Engineering for Web Applications, PhD, Faculty of Mathematics, Computer Science and Statistics, University of Malaga, Spanien, 2007.

[13] <http://uwe.pst.ifi.lmu.de/toolMagicUWE.html>.

[14] <http://uwe.pst.ifi.lmu.de/exampleUWEWebsite.html>.

[15] Quyet-Thang Huynh, Dinh-Dien Tran, Thi-Mai-Anh Bui, Phi-Le Nguyen (2019). Development of Rules and Algorithms for Model-Driven Code Generator with UWE Approach. New Trends in Intelligent Software Methodologies, Tools and Techniques, Volume 318, 2019, pp. 531 – 544, ISBN 978-1-64368-012-5, <https://doi.org/10.3233/FAIA190078>, WoS, SCOPUS Indexed.

[16] <https://github.com/zbloodz/CODEGER-UWE>