

## เครื่องมือควบคุมการปรับแต่งแม่แบบเว็บแอปพลิเคชันแบบไดนามิก\* Dynamic Template for Customize of Web Applications (DTCoW)

นวลศรี เด่นวัฒนา<sup>1\*\*</sup> , มาโนชญ์ ใจกว้าง<sup>2</sup> , ณัฐพร ภัคดี<sup>3</sup>  
Nuansri Denwattana<sup>1\*\*</sup> , Manot Jaikwang<sup>2</sup> , Nuttapon Phakdee<sup>3</sup>

<sup>1</sup>สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา  
เลขที่ 169 ถนนลงหาดบางแสน ตำบลแสนสุข อำเภอเมือง จังหวัดชลบุรี 20130

<sup>1</sup>Software Engineering, Faculty of Informatics, Burapha University  
169 Long-Hard Bangsaen Road, Saen Sook Sub-District, Mueang District, Chonburi 20131

<sup>2</sup>ห้องปฏิบัติการวิจัยวิศวกรรมระบบสารสนเทศ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา  
เลขที่ 169 ถนนลงหาดบางแสน ตำบลแสนสุข อำเภอเมือง จังหวัดชลบุรี 20130

<sup>2</sup>Information System Engineering Research Laboratory, Faculty of Informatics, Burapha University  
169 Long-Hard Bangsaen Road, Saen Sook Sub-District, Mueang District, Chonburi 20131

<sup>3</sup>เขตอุตสาหกรรมซอฟต์แวร์ภาคตะวันออก คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา  
เลขที่ 169 ถนนลงหาดบางแสน ตำบลแสนสุข อำเภอเมือง จังหวัดชลบุรี 20130

<sup>3</sup>Eastern Software Park, Faculty of Informatics, Burapha University  
169 Long-Hard Bangsaen Road, Saen Sook Sub-District, Mueang District, Chonburi 20131

### บทคัดย่อ

การนำแอปพลิเคชันที่มีการใช้งานอยู่ไปให้บริการแก่ผู้ใช้รายใหม่ จะต้องทำให้แอปพลิเคชันมีเอกลักษณ์ที่ต่างกัน ส่วนที่มีการเปลี่ยนแปลงสำคัญของแอปพลิเคชันซึ่งมีผลต่อผู้ใช้รายใหม่ คือ ส่วนติดต่อกับผู้ใช้งาน (User Interface) การเปลี่ยนแปลงในส่วนนี้อาจจะดูเหมือนเป็นการเปลี่ยนแปลงเพียงเล็กน้อย แต่ก็มีผลกระทบต่อการทำงานโดยรวมและอาจจะส่งผลต่อการทำงานที่ผิดพลาดของแอปพลิเคชัน งานวิจัยนี้จึงนำเสนอเครื่องมือสำหรับควบคุมการปรับแต่งแม่แบบบนเว็บแอปพลิเคชันแบบไดนามิก (DTCoW) ซึ่งเป็นแอปพลิเคชันโอเพนซอร์ส (Open-Source Software) สำหรับการปรับเปลี่ยนแม่แบบ (Template)

\* ปรับปรุงเพิ่มเติมเนื้อหาจากบทความ เรื่อง “เครื่องมือควบคุมการปรับแต่งแม่แบบเว็บแอปพลิเคชัน” ที่นำเสนอในการประชุมวิชาการ “2016 International Computer Science and Engineering Conference (ICSEC 2016)” วันที่ 14-17 ธันวาคม 2559 ณ โรงแรมเชียงใหม่ ออร์คิด จังหวัดเชียงใหม่

\*\* ผู้เขียนหลัก

อีเมลล์: nuansri@buu.ac.th



จากแม่แบบเดิมสู่แม่แบบใหม่ เพื่ออำนวยความสะดวกให้แก่นักพัฒนาซอฟต์แวร์ ผลการเปรียบเทียบหลังนำเครื่องมือ DTCoW มาใช้งานจริง จำนวน 6 แอปพลิเคชัน พบว่า สามารถช่วยให้นักพัฒนาซอฟต์แวร์แก้ไขไฟล์เพียง 4 ไฟล์ ได้แก่ Dynamic Template, Layout, Config และ Class Abstract และการทดสอบประสิทธิภาพของเครื่องมือ DTCoW กับกลุ่มตัวอย่าง 3 กลุ่ม ได้แก่ (1) นักพัฒนาซอฟต์แวร์ ประสบการณ์ 0-1 ปี จำนวน 6 คน (2) นักพัฒนาซอฟต์แวร์ ประสบการณ์ 1-3 ปี จำนวน 5 คน และ (3) นักพัฒนาซอฟต์แวร์ ประสบการณ์มากกว่า 3 ปี จำนวน 4 คน พบว่า เครื่องมือ DTCoW มีประสิทธิภาพในการปรับเปลี่ยนแม่แบบ ทั้งสามกลุ่มไม่แตกต่างกัน ที่ระดับนัยสำคัญ .05 แสดงว่า การใช้เครื่องมือ DTCoW ไม่จำเป็นต้องมีประสบการณ์ในการทำงานมากก็สามารถใช้งานเครื่องมือนี้ได้ ปัจจุบันเครื่องมือนี้ได้มีการนำไปใช้งานจริงกับระบบแผนและติดตามการใช้งบประมาณ ระบบบริการวิชาการ ระบบทำนุบำรุงศิลปวัฒนธรรม ระบบบริหารจัดการครุภัณฑ์ ระบบบุคลากรและส่งเสริมการพัฒนาบุคลากร และระบบจัดการผลงานวิจัยและวิชาการให้กับวิทยาลัยพยาบาลในสังกัดกระทรวงสาธารณสุข

## คำสำคัญ

แม่แบบ เว็บแอปพลิเคชัน ส่วนติดต่อกับผู้ใช้ การปรับแต่งค่า

## Abstract

To propose an existing application to new users, it is importance to make it's unique. In addition, an obvious changing that affect the new user is the developer has to create the user interface. Even if there is a slightly effect, the affect will be a cause of developing process and application failures. Therefore, this tool is an open source software to facilitate software developer for switching old template to the new template faster and more convenience.

The comparative result from the experiment with 6 applications found that this tool assist the software developer to edit only 4 files. These files consists of Dynamic template, Layout, Config and Class abstract. The sample consist of 3 group of users were as follow: 1) 6 of 0-1 year experienced programmer, 2) 5 of 1-3 year experienced programmer and 3) 4 at least 3 year programmer. The finding, according to user satisfaction, was the difference of using is statistically significant at .05 level. This software can adopted in real application without the experience of the user. The software has been applied to Plan and Budget Management system, Academic Service Management System, Culture Conservation Management System, Asset Management System, Human Resource and Development Support System and Researcher Management System for software suite for the Thailand Nursing institutes, Ministry of Public Health.

## Keywords

Dynamic Template, Web Application, User Interface, Customization.

## บทนำ

เมื่อสังคมก้าวเข้าสู่ยุคดิจิทัล ภาครัฐและภาคธุรกิจมีการปรับตัวและเร่งพัฒนาองค์กรให้สอดคล้องกับการเปลี่ยนแปลง เห็นได้จากการนำซอฟต์แวร์มาใช้ในการดำเนินงาน เพื่อรองรับการแข่งขันและการขยายตัวทางเศรษฐกิจ ส่งผลให้ทีมพัฒนาซอฟต์แวร์ต้องเร่งการพัฒนาซอฟต์แวร์ให้แล้วเสร็จและส่งมอบให้ลูกค้า ในระยะเวลาที่จำกัด ซอฟต์แวร์ที่ถูกพัฒนาขึ้นจึงมีความหลากหลาย และต้องยืดหยุ่นตามความต้องการของผู้ใช้งาน ซึ่งองค์ประกอบสำคัญที่ทำให้ผู้ใช้เห็นถึงความทันสมัย และความแตกต่างกันของซอฟต์แวร์ก็คือ ส่วนที่ติดต่อกับผู้ใช้งาน (User Interface) ซึ่งจำเป็นต้องทำให้ส่วนนี้มีความแตกต่างและมีเอกลักษณ์สอดคล้องกับการดำเนินธุรกิจหรือการดำเนินงานขององค์กร

อย่างไรก็ตาม สำหรับการพัฒนาซอฟต์แวร์ โดยเฉพาะอย่างยิ่งซอฟต์แวร์ที่มีขนาดใหญ่และมีความซับซ้อน การปรับเปลี่ยน ปรับปรุง และแก้ไขส่วนติดต่อกับผู้ใช้ จำเป็นต้องใช้เวลาในการปรับเปลี่ยนค่อนข้างมาก ทั้งในส่วนของการพัฒนา (Development) การดูแลบำรุงรักษา (Maintenance) และการปรับหรือระบบ (Reengineering) การดำเนินงานในส่วนนี้จึงมีต้นทุนที่สูง และใช้เวลานาน รวมถึงในการปรับแก้ระบบอาจส่งผลกระทบต่อการทำงานของซอฟต์แวร์ (Business Process) และก่อให้เกิดข้อบกพร่อง (Defects) อีกด้วย ถึงแม้ว่า มีเครื่องมืออัตโนมัติในการจัดการแม่แบบ อาทิเช่น Dynamic Joomla Templates (2010) แต่เครื่องมือดังกล่าวเหมาะสำหรับแอปพลิเคชันที่มีลักษณะเป็นพาณิชย์อิเล็กทรอนิกส์ (E-commerce) และแม่แบบ Meteor (Meteor Template) แต่ไม่สามารถใช้ได้กับการให้บริการแอปพลิเคชัน (On Demand Applications) และไม่สนับสนุนการพัฒนาแบบที่อ้างอิงสถาปัตยกรรมแบบ MVC

บทความนี้เป็นการนำเสนอเครื่องมือสำหรับอำนวยความสะดวกให้กับนักพัฒนาซอฟต์แวร์ (Developers) ซึ่งชื่อว่า เครื่องมือควบคุมการปรับแต่งแม่แบบบนเว็บแอปพลิเคชัน (DTCoW) เพื่อนำไปใช้ในการควบคุมการปรับแต่งแม่แบบบนเว็บแอปพลิเคชัน ที่มีสถาปัตยกรรมแบบ MVC (Model-View-Controller) หรืออื่นๆ ที่เป็นเว็บแอปพลิเคชัน (Web-Based Application) ที่รองรับภาษา PHP Framework โดยเครื่องมือนี้จะสอดคล้องกับมาตรฐานการพัฒนาซอฟต์แวร์ขององค์กร โดยลำดับในการนำเสนอเนื้อหาของบทความนี้ประกอบไปด้วย ลำดับที่ 2 จะเป็นการอธิบายกรอบแนวคิดและเทคโนโลยีที่เกี่ยวข้อง ในส่วนของลำดับที่ 3 เป็นการนำเสนอเครื่องมือควบคุมการปรับแต่งแม่แบบเว็บแอปพลิเคชัน ในส่วนของผลการดำเนินงาน จะอภิปรายในลำดับที่ 4 และลำดับสุดท้ายจะกล่าวถึงการสรุปผลและแผนงานในอนาคต

## บททวนวรรณกรรม

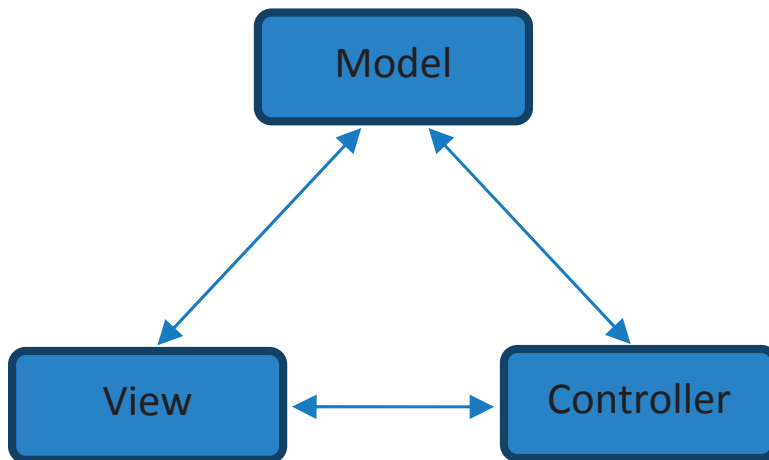
ในส่วนนี้จะกล่าวถึงแนวคิดที่เกี่ยวข้องข้องในการพัฒนาเครื่องมือสำหรับควบคุมการปรับแต่งแม่แบบบนเว็บแอปพลิเคชันซึ่งได้แก่ แม่แบบหรือ Template ของหน้าจอบเว็บไซต์ สถาปัตยกรรม MVC รวมถึงงานวิจัยที่เกี่ยวข้อง ประกอบด้วย การปรับเปลี่ยนเนื้อหาข้อมูลเว็บเพจอัตโนมัติ Dynamic Joomla Templates และกรอบการทำงาน Meteor ซึ่งมีการนำมาใช้ในการดำเนินงานครั้งนี้

### 1. แม่แบบ (Template)

แม่แบบหรือแบบฟอร์มเว็บไซต์ที่ยังไม่มีการใส่เนื้อหาเข้าไป ผู้ผลิตสร้างขึ้นมาเป็นเว็บไซต์สำเร็จรูป ถือเป็นต้นแบบสำหรับการสร้างเว็บไซต์ นักพัฒนามาใช้เพื่อช่วยในการประหยัดเวลาในการสร้างเนื้อหาของแม่แบบจะประกอบด้วยสิ่งที่ทุกเว็บไซต์มีเหมือนกัน เช่น การจัด Layout และรูปแบบของเนื้อหาต่างๆ ได้แก่ โลโก้ เมนู รูปแบบตัวอักษร ขนาด สีของข้อความ สีไฮเปอร์ลิงก์ เป็นต้น อีกทั้งยังมีการกำหนดพื้นที่ไว้สำหรับให้ใส่เนื้อหาแต่ละเว็บเพจแตกต่างกันได้ นอกจากนี้เมื่อแม่แบบถูกแก้ไข ก็จะส่งผลไปยังทุกเว็บเพจที่ใช้แม่แบบดังกล่าวโดยอัตโนมัติ ในปัจจุบันมีแม่แบบ จำนวนมากเพื่อให้ผู้ใช้ได้เลือกใช้ตามความต้องการ (Babasaheb & Santosh, 2015)

### 2. สถาปัตยกรรม Model View Controller (MVC)

MVC เป็นสถาปัตยกรรมซอฟต์แวร์ชนิดหนึ่งที่ได้รับคานิยมในงานด้านวิศวกรรมซอฟต์แวร์ ซึ่งแบ่งออกเป็น 3 ส่วน ประกอบด้วย Model Component, View Component และ Controller Component รูปแบบ MVC ใช้เพื่อแยกส่วนซอฟต์แวร์ในส่วน ตรรกะเนื้อหา (Domain Logic) ได้แก่ความเข้าใจในระบบของผู้ใช้และส่วนการป้อนข้อมูลและแสดงผล (GUI) ซึ่งช่วยให้การพัฒนาการทดสอบ และการดูแลรักษาซอฟต์แวร์ แยกออกจากกันซึ่งแต่ละ Component สามารถพัฒนา Application โดยไม่ขึ้นกับ Component ใด Component หนึ่ง นั่นหมายถึง เมื่อมีการเปลี่ยนแปลงจะไม่มีผลกระทบต่อ Component อื่นๆ (Little Bear, 2013)



ภาพที่ 1: สถาปัตยกรรมแบบ MVC

จากภาพที่ 1 อธิบายสถาปัตยกรรมแบบ MVC โดยมีองค์ประกอบ 3 ส่วนหลัก มีรายละเอียดดังต่อไปนี้

1. Model หมายถึง ส่วนของซอฟต์แวร์ที่ใช้ในการแปลงการทำงานของระบบ ทำหน้าที่ในการจัดการนำข้อมูลขึ้นมาจากฐานข้อมูล และเก็บข้อมูลไว้ นอกจากนี้ยังมีส่วนของตรรกะหรือกระบวนการทางธุรกิจ เช่น การเข้าถึงข้อมูล การยืนยันความถูกต้องของข้อมูลและตรรกะการเก็บข้อมูล เป็นต้น

2. View แสดงผลลัพธ์ใน Model ในรูปแบบที่เหมาะสมต่อการปฏิสัมพันธ์กับผู้ใช้ในแต่ละ Model สามารถมี View ได้หลายแบบ เพื่อใช้ในจุดประสงค์ที่ต่างกัน

3. Controller ทำหน้าที่เป็นตัวสื่อกลางที่ทำงานประสานกันระหว่าง Model และ View โดย Controller ทำหน้าที่ส่งคำร้องขอที่ได้จาก Client แล้วดูว่าคำร้องขอที่ได้นั้นเรียก Model ตัวใดให้ทำงาน และเมื่อ Model ทำงานเสร็จจะส่งการตอบรับกลับมาที่ Controller และ Controller จะเป็นตัวควบคุมอีกทีว่าการตอบรับนี้จะให้ View ตัวไหนแสดงผลพร้อมออกมาพร้อมกับข้อมูลที่ได้มาจากชั้นของ Model (Supaartagon, 2011)

### 3. การปรับเปลี่ยนเนื้อหาข้อมูลเว็บเพจอัตโนมัติ

Sanjay et al. (2008) ได้นำเสนอระบบและวิธีการในการพัฒนาเว็บเพจ (System and Method for Developing a Dynamic Web Page) ที่สามารถสร้าง การแก้ไข และ/หรือการประมวลผลเนื้อหาข้อมูลได้อย่างอัตโนมัติ โดยความสามารถของเครื่องมือนี้ สามารถสร้างโค้ดแบบ Static ด้วยภาษา HTML หรือสามารถสร้างโค้ด (Coding) แบบ Dynamic ด้วยภาษาจาวาสคริปต์ เนื้อหาข้อมูลที่สร้างได้ยังรวมถึงการสร้างข้อความทั้งย่อหน้า หรือหัวข้อของเนื้อหาข้อมูล โดยลักษณะของเนื้อหาที่สามารถสร้างได้นั้นจะขึ้นอยู่กับตัวแปรที่หลากหลาย เช่น เว็บเบราว์เซอร์ หรือตำแหน่งที่ผู้ใช้ ใช้งานแอปพลิเคชันนั้นอยู่ สถาปัตยกรรมการทำงานของระบบจะมี JSP container ทำงานร่วมกับฐานข้อมูลในฝั่งเครื่องแม่ข่าย โดย JSP container จะทำหน้าที่รับและส่งคำร้องในรูปแบบโค้ดไปยังผู้พัฒนาซอฟต์แวร์ ซึ่งหากได้รับคำร้องจะประมวลผลโค้ดที่ได้รับมา และจะส่งโค้ดกลับในรูปแบบ Static หรือ Dynamic

### 4. กรอบการทำงาน Meteor

Greif (2015) ได้นำเสนอ Meteor ซึ่งเป็นกรอบการทำงานเพื่อการสร้างเว็บแอปพลิเคชัน บนมือถือ หรือ Web API แบบเรียลไทม์ที่พัฒนาต่อยอดมาจาก Node.js กรอบการทำงาน Meteor สนับสนุนการทำงานทั้งในสภาพแวดล้อมที่เป็น Mac, Windows หรือ Linux ทำงานร่วมกับฐานข้อมูล MongoDB แต่ไม่สนับสนุนสถาปัตยกรรมการทำงานแบบ MVC โดย Meteor จะทำงานอยู่ระหว่างฐานข้อมูล (Database) และส่วนติดต่อผู้ใช้ (User Interface) เพื่อทำให้ข้อมูลทั้งสองฝั่งนั้นทำงานสอดคล้องกันอยู่เสมอ ซึ่งการพัฒนาต่อยอดจาก Node.js ทำให้ Meteor ใช้ภาษาจาวาสคริปต์ได้ทั้งบนฝั่งไคลเอนต์และเซิร์ฟเวอร์ นอกจากนี้ Meteor จะผูกไฟล์ CSS จาวาสคริปต์ รวมทั้ง HTML เข้าด้วยกันทำให้เว็บไซต์ตอบสนองต่อข้อมูลและเหตุการณ์ต่างๆ ได้ทันที (Reactive and Real-time)



## 5. Dynamic Joomla Templates

Dynamic Joomla Templates (2010) ถูกพัฒนามาให้สามารถปรับเปลี่ยนแม่แบบของเว็บไซต์ที่พัฒนาโดยใช้สถาปัตยกรรม MVC เหมาะสำหรับการสร้างและบริหารเว็บไซต์แบบสำเร็จรูป ซึ่งตัวอย่างเนื้อหาที่อยู่ในเว็บไซต์ ที่สามารถจัดการได้ เช่น ข้อความ ภาพ วิดีโอ เอกสาร เพลง ฯลฯ Dynamic Joomla Templates ถูกพัฒนาเพื่อทำให้การปรับเปลี่ยนแม่แบบของเว็บไซต์ทำได้ง่ายผ่านเครื่องมือที่เป็น Graphic User Interface สามารถเลือกเปลี่ยนแม่แบบจากแม่แบบสำเร็จรูป ที่ Joomla ได้เตรียมไว้ให้ หรือสามารถดาวน์โหลดแม่แบบเพิ่มเติม สามารถเลือกรับเปลี่ยนเนื้อหาเพียงบางส่วน หรือจะเลือกรับเปลี่ยนเนื้อหาให้เป็นไปตามแม่แบบของแม่แบบใหม่ทั้งหมดได้ พัฒนาโดยใช้ภาษา PHP และทำงานร่วมกับฐานข้อมูล MySQL แม่แบบ Joomla มีผู้ใช้งานมากมาย เช่น Harvard University, eBay, Orange, IKEA, Pizza Hut รวมถึง McDonalds

## 6. ส่วนติดต่อกับผู้ใช้ (User Interface: UI)

6.1 ส่วนที่เป็นสื่อกลางในการติดต่อระหว่างผู้ใช้กับระบบ เพื่อรองรับการนำเข้าสู่ข้อมูลหรือคำสั่งเข้าไปสู่ระบบ ตลอดจนนำเสนอสารสนเทศกลับมายังผู้ใช้ ไม่ว่าจะเป็นการติดต่อทางด้านฮาร์ดแวร์หรือซอฟต์แวร์ โดยสามารถแบ่งได้เป็น 2 ประเภท คือ

6.1.1 คอมมานด์ไลน์ (Command Line) เป็นส่วนประสานงานกับผู้ใช้ที่อนุญาตให้ป้อนคำสั่งที่เป็นตัวหนังสือ (Text) สั่งการลงไปด้วยตนเองเพื่อให้คอมพิวเตอร์ทำงานตามความต้องการที่ละบรรทัดคำสั่ง

6.1.2 แบบกราฟฟิก (Graphic User Interface: GUI) เป็นส่วนติดต่อกับผู้ใช้ที่นักพัฒนานำมาเป็นเครื่องมือในการสร้างสื่อประสมมีรูปแบบ สีสันสวยงามและใช้งานได้ง่ายขึ้น

6.2 องค์ประกอบของส่วนติดต่อกับผู้ใช้ (User Interface Elements) เมื่อนักพัฒนาต้องออกแบบส่วนติดต่อกับผู้ใช้ นักพัฒนาต้องคำนึงถึงองค์ประกอบของส่วนที่ติดต่อกับผู้ใช้ เพื่อเลือกหรือพิจารณาว่าองค์ประกอบแบบใดที่เหมาะสมกับการใช้งาน และเพื่อให้การทำงานของระบบมีประสิทธิภาพอันได้แก่

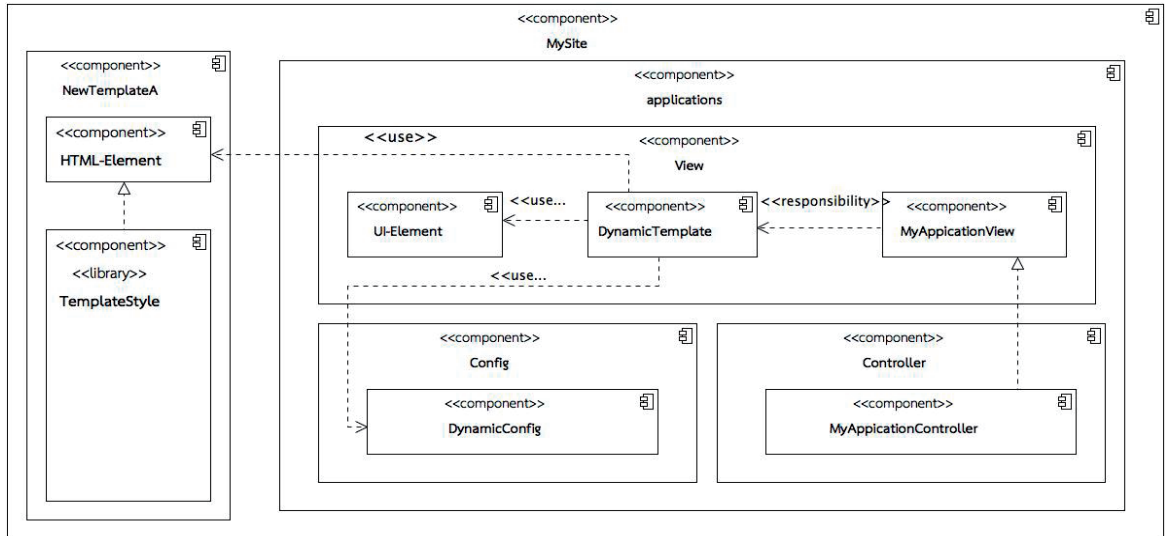
6.2.1 ส่วนของการควบคุมการรับข้อมูลเข้า (Input Controls) เช่น Checkboxes, Radio Buttons, Dropdown Lists, List Boxes, Buttons, Toggles, Text fields, Date Field

6.2.2 ส่วนของการควบคุมทิศทาง (Navigational Components) เช่น Breadcrumb, Slider, Search Field, Pagination, Slider, Tags, Icons

6.2.3 ส่วนของ Containers เช่น Accordion เป็นต้น

## เครื่องมือควบคุมการปรับแต่งแม่แบบเว็บแอปพลิเคชัน

เนื้อหาในส่วนนี้จะนำเสนอองค์ประกอบของ DTCoW ในลักษณะองค์ประกอบที่มีความสัมพันธ์กันภายใต้สถาปัตยกรรมแบบ MVC ซึ่งประกอบไปด้วย 4 องค์ประกอบ ดังปรากฏในภาพที่ 2



ภาพที่ 2: DTCoW Component

สถาปัตยกรรมองค์ประกอบ (Component Design) จากภาพที่ 2 ประกอบด้วย 4 องค์ประกอบหลัก ได้แก่

### 1. UI-Element

องค์ประกอบนี้จัดเป็นองค์ประกอบเริ่มแรกของ DTCoW โดยทำหน้าที่ในการกำหนด Element ต่างๆ ของภาษา HTML ตัวอย่างเช่น <table>, <img>, <div> เป็นต้น ในลักษณะ Abstract class เพื่อรอการเรียกใช้ โดย UI-element จะถูกเรียกใช้ (Extending) ด้วย Dynamic Template ในลำดับถัดไป

**อัลกอริทึม :** สร้าง Abstract class ของ Element ต่างๆ ในภาษา HTML

**ข้อมูลนำเข้า :** ชื่อ Element ของ HTML

**ผลลัพธ์ :** Abstract class ของ Element ต่างๆ ในภาษา HTML

- 1: ประกาศตัวแปรชนิด String
- 2: ทำซ้ำจนกว่าจะออกจากโปรแกรม
- 3: รับข้อความชื่อ Element ของภาษา HTML
- 4: Generate abstract ตามชื่อ Element
- 5: บันทึก Abstract ใหม่ลงไฟล์

ภาพที่ 3: UI-Element



## 2. Dynamic Template

สำหรับองค์ประกอบนี้จะทำหน้าที่ในการดำเนินการ (Implement) Abstract Class ตามที่ได้กล่าวไว้ใน UI-element เหล่านั้น ในลักษณะ Override Method นอกจากนี้ Dynamic Template ยังทำการจับคู่ (Matching) Element ที่อยู่ในแม่แบบ (Template) ที่ต้องการจะใช้กับ Method ของ Abstract ที่ได้ประกาศไว้

**อัลกอริทึม :** จับคู่ Element ของภาษา HTML ใน Abstract กับ Element ที่ปรากฏในแม่แบบ (Template) ที่ต้องการ

**ข้อมูลนำเข้า :** 1) ชื่อ Abstract  
2) ชื่อ Element ของแม่แบบ (Template)  
3) ชื่อ Template

**ผลลัพธ์ :** Base class ของ Template

- 1: ประกาศตัวแปรชนิด String
- 2: ทำซ้ำจนกว่าจะออกจากโปรแกรม
- 3: อ่านไฟล์ Abstract class
- 4: เลือก Abstract ที่ต้องการจับคู่
- 5: รับข้อความชื่อ Element ของแม่แบบ (Template)
- 6: รับชื่อ Template
- 7: ถ้าชื่อ Template ซ้ำ
  - 8: Generate method ตามชื่อ Abstract และให้
  - 9: Element (Abstract) => Element (Template)
  - 10: บันทึก method ใหม่ลงท้ายไฟล์ตามชื่อ Template
- 11: ไม่ซ้ำ
  - 12: Generate method ตามชื่อ Abstract และให้
  - 13: Element (Abstract) => Element (Template)
  - 14: สร้างไฟล์และ Class ตามชื่อ Template
  - 15: บันทึก method ใหม่ลงไฟล์ใหม่ที่ตั้งชื่อตาม Template

ภาพที่ 4: Dynamic Template



### 3. Dynamic Config

ทำหน้าที่ในการตั้งค่าการเรียกใช้งานแม่แบบที่ต้องการใช้งานทั้งหมดและเป็นองค์ประกอบที่กำหนดการเปลี่ยนแปลงของแม่แบบ (Template) ภายในแต่ละซอฟต์แวร์

- อัลกอริทึม :** ตั้งค่าการเรียกใช้งานแม่แบบ  
**ข้อมูลนำเข้า :** ชื่อ Template class  
**ผลลัพธ์ :** Template ที่สามารถเปลี่ยนไปตามการกำหนดค่า
- 
- 1: ประกาศตัวแปรชนิด String
  - 2: วนซ้ำจนกว่าจะออกจากโปรแกรม
  - 3: แสดง Template class ทั้งหมด
  - 4: เลือก Template class ที่ต้องการตั้งค่า
  - 5: Generate เงื่อนไข และการเรียกใช้ Template
  - 6: บันทึกเงื่อนไข และการเรียกใช้ Template ใหม่ลงไฟล์

### ภาพที่ 5: Dynamic Config

### 4. MyApplication View

เป็นส่วนติดต่อกับผู้ใช้ (User Interface) ในส่วนนี้นักพัฒนาต้องพัฒนาภายใต้ Element ของ Abstract Class ที่ถูกกำหนดและจับคู่ใน Dynamic Template เท่านั้น ในที่นี้หมายถึงในส่วนของ View (V) คณะผู้วิจัยจึงตั้งชื่อ View ในส่วนนี้ว่า “MyApplicationView” โดยการใช้ View ดังกล่าว ผู้พัฒนาซอฟต์แวร์ต้องเรียกใช้ (Include) Dynamic Template Class และทำการสร้าง Object เพื่อทำการเรียกใช้ Element ต่างๆ ใน Object นั้นๆ

### ผลการดำเนินงาน

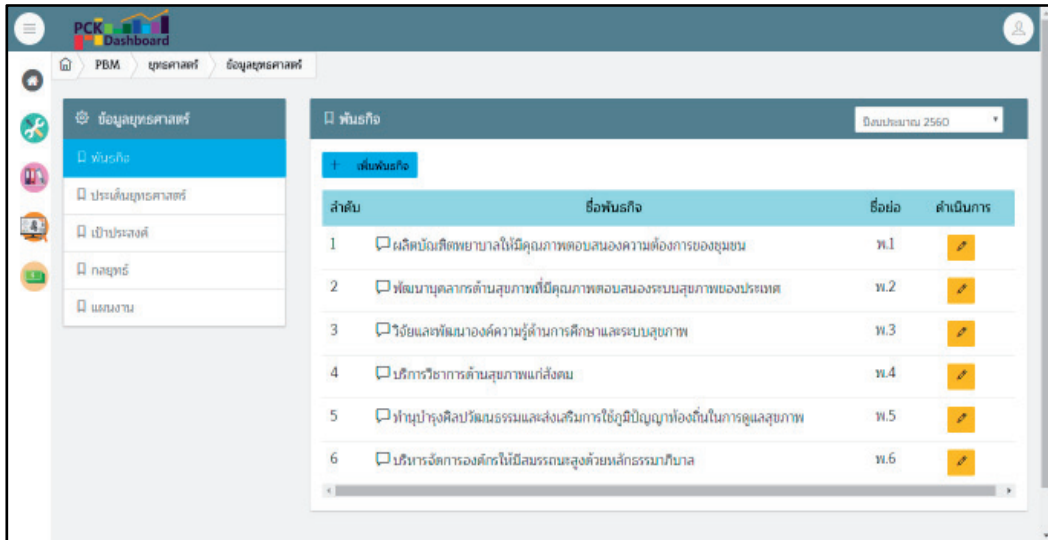
เครื่องมือ DTCoW ได้ถูกนำมาพัฒนาเป็นซอฟต์แวร์ต้นแบบ โดยพัฒนาภายใต้เครื่องมือโอเพนซอร์สซอฟต์แวร์ ได้แก่ (1) ภาษา PHP เวอร์ชัน 7.0 โดยใช้ Framework ชื่อ Codeigniter PHP Framework (CI) (2) ติดตั้งบนระบบปฏิบัติการลินุกซ์ Debian เวอร์ชัน 8.6 (3) เว็บเซิร์ฟเวอร์ Apache เวอร์ชัน 2.4.10 และ (4) โปรแกรม NetBeans เวอร์ชัน 7.3.1 โดยในบทนี้ได้นำเสนอ ผลการนำเครื่องมือไปใช้จริงและผลลัพธ์เชิงเปรียบเทียบ รายละเอียด ดังนี้

#### 1. ผลการนำเครื่องมือ DTCoW ไปใช้งานจริง

คณะผู้วิจัยได้นำเครื่องมือนี้มาใช้ในการปรับแต่งแม่แบบเว็บแอปพลิเคชัน ได้แก่ ระบบแผนและติดตามการใช้งบประมาณ (Plan and Budget Management System: PBMS) ระบบบริการวิชาการ (Academic Service Management System: ASMS) ระบบงานทำนุบำรุงศิลปวัฒนธรรม (Culture Conservation Management System: CMS) ระบบบริหารจัดการครุภัณฑ์ (Asset Management System:

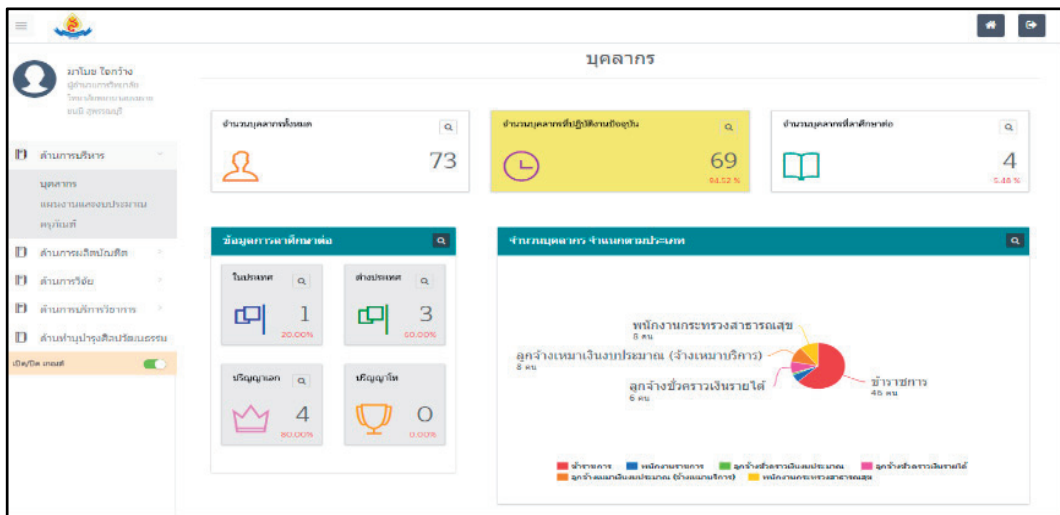


AMS) ระบบบุคลากรและส่งเสริมการพัฒนาบุคลากร (Human Resource Development Support System: HRDSS) และระบบจัดการผลงานวิจัยและวิชาการ (Researcher Management System: RMS) ตัวอย่างหน้าเว็บแอปพลิเคชันของระบบ แสดงดังภาพที่ 6 และภาพที่ 7



ภาพที่ 6: ระบบแผนและติดตามการใช้งบประมาณ

จากภาพที่ 6 นำเสนอส่วนติดต่อกับผู้ใช้ที่นำเครื่องมือ DTCoW มาใช้งานในการปรับแต่งแม่แบบเว็บแอปพลิเคชัน ซึ่งประกอบไปด้วย Navigator Bar แถบเมนู และส่วนแสดงเนื้อหา ได้แก่ ตาราง และ ฟอรัมของการนำเข้าสู่ข้อมูล



ภาพที่ 7: ระบบบุคลากรและส่งเสริมการพัฒนาบุคลากร

## 2. ผลลัพธ์เชิงเปรียบเทียบ

เครื่องมือควบคุมการปรับแต่งแม่แบบเว็บแอปพลิเคชัน (DTCoW) มีการนำไปประยุกต์ใช้จริงให้กับชุดซอฟต์แวร์ระบบสารสนเทศของวิทยาลัยการพยาบาล ในสังกัดกระทรวงสาธารณสุข จำนวน 7 วิทยาลัย ซึ่งมีผลการเปรียบเทียบของการนำเครื่องมือ DTCoW มาใช้ กับการทำงานแบบดั้งเดิม

### ตารางที่ 1

แสดงการเปรียบเทียบหลังนำเครื่องมือ DTCoW มาใช้งานจริง

ลำดับ	แอปพลิเคชัน	จำนวนไฟล์ Views	จำนวนไฟล์ที่มีการปรับแก้	
			การทำงานแบบเดิม	การทำงานโดยใช้ DTCoW มาช่วย
1.	PBMS	55	55	4*
2.	OMS	15	15	4*
3.	CMS	15	15	4*
4.	AMS	38	38	4*
5.	HRSS	227	227	4*
6.	RMS	161	161	4*

จากตารางที่ 1 เป็นการเสนอข้อมูลการเปรียบเทียบ Web Application ที่ได้นำเครื่องมือ DTCoW ไปประยุกต์ใช้งานจริง พบว่า เมื่อมีการเพิ่มแม่แบบใหม่ วิธีการทำงานแบบเดิมผู้พัฒนาซอฟต์แวร์ต้องแก้ไขไฟล์ที่อยู่ในส่วนของ Views ตาม Framework ที่มีการทำงานเป็นแบบ MVC เท่ากับจำนวนไฟล์ Views ของระบบเดิม แต่เมื่อใช้เครื่องมือ DTCoW ผู้พัฒนาซอฟต์แวร์แก้ไขไฟล์เฉลี่ยจำนวน 4 ไฟล์ ได้แก่ Dynamic template, Layout, Config และ Class Abstract หรืออาจจะมีไฟล์อื่นเพิ่มเติมที่ต้องมีการแก้ไข ขึ้นอยู่กับความซับซ้อนของแม่แบบ

นอกจากนี้ ชุดซอฟต์แวร์ที่พัฒนาโดยใช้เครื่องมือควบคุมการปรับแต่งแม่แบบบนเว็บแอปพลิเคชัน (DTCoW) ที่นำไปใช้กับวิทยาลัยพยาบาลทั้ง 7 แห่ง เป็นซอฟต์แวร์ชุดเดียวกันแต่ติดตั้งในเครื่องแม่ข่ายที่แตกต่างกัน

## 3. ผลลัพธ์เชิงเปรียบเทียบกับผู้ใช้เครื่องมือ

การศึกษาคั้งนี้ คณะพัฒนาได้นำเครื่องมือการปรับแต่งแม่แบบบนเว็บแอปพลิเคชันไปทดสอบประสิทธิภาพการใช้งานกับกลุ่มตัวอย่าง จำนวน 3 กลุ่ม ได้แก่ (1) นักพัฒนาซอฟต์แวร์ ประสบการณ์ 0-1 ปี จำนวน 6 คน (2) นักพัฒนาซอฟต์แวร์ ประสบการณ์ 1-3 ปี จำนวน 5 คน และ (3) นักพัฒนาซอฟต์แวร์ ประสบการณ์มากกว่า 3 ปี จำนวน 4 คน เก็บข้อมูลจากระยะเวลาที่ใช้เครื่องมือ DTCoW ภายใต้สิ่งแวดล้อมเดียวกันต่อแอปพลิเคชันของกลุ่มตัวอย่างแต่ละกลุ่มเปรียบเทียบกับ โดยใช้ MVC เดิม และเมื่อนำผลที่ได้มา



วิเคราะห์เปรียบเทียบด้วยค่าเฉลี่ยของเวลา คณะผู้วิจัยจึงใช้สถิติอนพาราเมตริก ได้แก่ สถิติ Kruskal-Wallis Test เนื่องจากกลุ่มตัวอย่างมีขนาดเล็ก พบว่า ค่าเฉลี่ยของระยะเวลาที่ใช้เครื่องมือ DTCoW ของนักพัฒนาซอฟต์แวร์ ทั้งสามกลุ่มไม่แตกต่างกัน ที่ระดับนัยสำคัญ .05 (ไคสแควร์เท่ากับ 6.299,  $df = 2$ ) แสดงว่า การใช้งานของนักพัฒนาซอฟต์แวร์ ด้วยเครื่องมือ DTCoW ภายใต้ MVC เดิม นั้น มีประสิทธิภาพในการทำงานไม่แตกต่างกัน อย่างมีนัยสำคัญที่ .05 นั่นคือ การที่นักพัฒนาซอฟต์แวร์ จะนำเครื่องมือ DTCoW ไปใช้ ไม่จำเป็นต้องประสบการณ์ในการทำงานมากก็สามารถใช้งานเครื่องมือนี้ได้

## สรุป

บทความนี้เป็นกรนำเสนอ เครื่องมือสำหรับควบคุมการปรับแต่งแม่แบบเว็บแอปพลิเคชัน มาใช้ในการปรับเปลี่ยนส่วนแม่แบบเดิมสู่แม่แบบใหม่ โดยการนำไปทดสอบการใช้งานกับนักพัฒนาซอฟต์แวร์ 3 กลุ่ม ที่มีประสบการณ์แตกต่างกัน ผลการทดสอบแสดงให้เห็นว่า เครื่องมือนี้มีประสิทธิภาพการใช้งานไม่แตกต่างกัน นั่นคือ เครื่องมือนี้สามารถนำไปใช้งานได้จริง โดยนักพัฒนาซอฟต์แวร์ไม่จำเป็นต้องมีความเชี่ยวชาญทางการเขียนโปรแกรมก็สามารถใช้งานได้ไม่ต่างกับนักพัฒนาซอฟต์แวร์ที่มีความเชี่ยวชาญสูง และคณะผู้วิจัยได้นำซอฟต์แวร์ดังกล่าวไปใช้งานจริงกับการพัฒนาระบบแผนและติดตามการใช้งบประมาณ ระบบบริการวิชาการ และทำนุบำรุงศิลปวัฒนธรรม ระบบบริหารจัดการครุภัณฑ์ ระบบบุคลากรและส่งเสริมการพัฒนาบุคลากร และระบบจัดการผลงานวิจัยให้กับวิทยาลัยพยาบาลในสังกัดกระทรวงสาธารณสุข

ขณะนี้คณะผู้วิจัย มีการพัฒนาต่อยอดให้ผู้พัฒนาซอฟต์แวร์ใช้งานได้สะดวกขึ้น เช่น ทำการเพิ่มเครื่องมืออัตโนมัติเพื่อแนะนำคำสั่ง ทำ Template Extension เพื่อรองรับการปรับเปลี่ยนของแม่แบบในอนาคตให้สะดวกมากขึ้น และมีแผนในการนำเครื่องมือดังกล่าวไปใช้ในการพัฒนาซอฟต์แวร์ของห้องปฏิบัติการวิจัยวิศวกรรมระบบสารสนเทศ มหาวิทยาลัยบูรพา พร้อมทั้งให้นิสิตในสาขาวิศวกรรมซอฟต์แวร์นำเครื่องมือดังกล่าวมาใช้ในรายวิชาที่เกี่ยวข้องกับการพัฒนาซอฟต์แวร์โดยจะมีการเก็บรวบรวมข้อมูลเพื่อนำมาวิเคราะห์ประสิทธิภาพการใช้งาน และยกระดับของซอฟต์แวร์ให้ตอบสนองกับนักพัฒนาซอฟต์แวร์ที่มีทักษะที่หลากหลาย

## เอกสารอ้างอิง

- Babasaheb, S. K. & Santosh, S. L. (2015). Study of content management systems. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2015(5), 670 – 672.
- Supaartagron, C. (2011). PHP Web authoring for database management based on MVC Pattern. In *Proceeding of the World on Engineering and Computer Science 2011*, 19-21 October 2011. San Francisco, USA. (pp. 406-411). California: Newswood.



- Dynamic Joomla Templates. (2010). **Dynamic Joomla Templates**. Retrieved June 1, 2017, from <http://www.dynamic-template.com/type/joomla-templates>.
- Ghorecha, V. & Bhatt, C. (2013). A guide for selecting content management system for web application development. **International Journal of Advance Research in Computer Science and Management Studies**. 2013(1), 13 – 17.
- Greif, S. (2015). **Dynamic template includes with blaze**. Retrieved June 1, 2017, from <https://www.discovermeteor.com/blog/blaze-dynamic-template-includes/>.
- Little Bear. (2013). **MVC Architecture Pattern**. Retrieved September 10, 2016, from <http://www.softganz.com/upload/pics/mvc.png>.
- Sanjay, D. ; Stephen, T. ; Srinivasan, T. R. & Predrag, P. (2008). **System and method for developing a dynamic web page**. Retrieved June 1, 2017, from <https://www.google.com/patents/US7316003>.
- Welling, L. & Thomson, L. (2005). **PHP and MYSQL Web Development**. Indianapolis: Sams Publishing.