การรวมรูปแบบยูเอ็มแอล สำหรับระบบการศึกษาแบบฝัง

สุขแสง คูกนก*

บทคัดย่อ

ระบบแบบฝัง (Embedded Systems) ที่ปัจจุบันนับวันมีการใช้งานเพิ่มมากขึ้นทุกวัน ได้แก่ อุปกรณ์อิเล็กทรอนิกส์แทบทุกชนิด ได้แก่ โทรศัพท์มือถือ กล้องถ่ายรูปดิจิตอล เป็นต้น การพัฒนาการเขียนโปรแกรมภายใน Embedded นั้นต้องการความรวดเร็ว และไม่ผิดพลาด ทุกวันนี้ได้มีการนำ Real Time Operating Systems (RTOS) มาใช้งานกับระบบแบบฝัง ทั้งนี้เพื่อความรวดเร็วในการผลิต Firmware แม้จะต้องใช้ทรัพยากรด้านหน่วยความจำเพิ่มขึ้นก็ตาม UML (Unified Modeling Language) เป็นภาษาในลักษณะของการใช้แผนภาพ และมีลักษณะพื้นฐานการทำงานในเชิงวัตถุ การเขียน Firmware สำหรับระบบแบบฝังในแบบเดิมยังเป็นการเขียนโปรแกรมในแนวความคิดเชิงการทำงานที่ยังต้องอาศัย Flow Chart และเชิงวัตถุในบางส่วนที่จะต้องใช้เครื่องมือที่เป็น Object Oriented เช่น C++ แต่การเขียนในลักษณะนี้ยังคงเกิดปัญหาในเรื่องของการสูญเสียทรัพยากรไปบางส่วนในด้านของการนำ Firmware กลับมาใช้ใหม่ รวมทั้งปัญหาในด้านการปรับเปลี่ยนคุณลักษณะบางประการของ Firmware เพื่อสร้างผลิตภัณฑ์ใหม่ จาก Firmware เดิม

การเขียนวิเคราะห์ และออกแบบ เพื่อแก้ปัญหาของระบบแบบฝังโดยการนำ UML สามารถที่จะแก้ปัญหาที่เคยเกิดขึ้นจากการเขียน Firmware แบบเก่าได้ โดยการแทรกขั้นตอนของ UML ให้อยู่ในส่วนหน้าสุดของขั้นตอนเดิมทั้งหมด แล้วปรับเปลี่ยนกระบวนการเขียน Code ของ Firmware ให้อยู่ในขั้นตอนใหม่ของ UML ในบทความนี้จะเป็นการถ่ายทอดกระบวนการทั้งหมดเหล่านี้ไปยังผู้เรียน ซึ่งทั้งนี้จำเป็นที่ผู้เรียนต้องมีความเข้าใจ และมีองค์ความรู้โดยผ่านการศึกษามาจากสาขาเป็นพื้นฐานก่อน การถ่ายทอดองค์ความรู้ในเรื่องนี้ เป็นการที่จะต้องสรุปองค์ความรู้จากหลายสาขาแล้วปรับแต่งให้เป็นรูปแบบใหม่ที่ทำให้ผู้เรียนเข้าใจง่าย จากรูปแบบพื้นฐาน ซึ่งจะทำให้ผู้เรียนสามารถพัฒนาและนำไปประยุกต์กับรูปแบบของงานจริงได้

1. บทนำ

เนื่องจากความต้องการที่ซับซ้อนของผู้ใช้งาน และหน้าที่ของระบบอิเล็กทรอนิกส์ในชีวิตประจำวันมีความหลากหลายมากขึ้น ทำให้ระบบ Embedded ซึ่งตอบสนองความต้องการดังกล่าว ในปัจจุบันมีความสลับซับซ้อนมากขึ้นตามไปด้วย [20] เช่น ความต้องการใช้ประโยชน์ที่หลากหลาย มีการใช้มัลติมีเดียมากขึ้น มีการทำงานกับทั้งข้อมูลภาพ ข้อมูลเสียง และฐานข้อมูลอื่น ๆ เช่น แผนที่ เป็นต้น ทำให้มีความจำเป็นในการติดต่อสื่อสารกับอุปกรณ์ต่าง ๆ เพิ่มมากขึ้น [10] เช่นกัน อีกทั้งเมื่อมีการทำงานที่หลากหลาย ทำให้ระบบรักษาความปลอดภัยในตัวมันเอง ก็มีความจำเป็นที่เพิ่มมากขึ้นเช่นกัน

ด้วยลักษณะของความต้องการที่เพิ่มมากขึ้น ทำให้การพัฒนาด้านหน่วยประมวลผล ซึ่งเป็นหัวใจของการทำงานในระบบ Embedded มีการพัฒนาไปได้รวดเร็วมาก ซึ่งทำให้การพัฒนาด้านซอฟต์แวร์ ต้องมีการปรับตัวให้ทันกับสภาพแวดล้อมที่เปลี่ยนแปลงไปอย่างรวดเร็วด้วยเช่นกัน เนื่องจากความกดดันด้านความต้องการของผู้ใช้มีมาก ทำให้ผู้ที่ปรับตัวทันกับความเปลี่ยนแปลงดังกล่าวนี้อยู่ในสถานะที่ได้เปรียบด้านธุรกิจ [19] ทำให้เกิดการผลักดันให้มีการค้นคิดและพัฒนาด้านการเขียนโปรแกรม เพื่อให้ระบบทำงานได้ตามความต้องการได้อย่างรวดเร็ว เมื่อมองในมุมของการพัฒนาโปรแกรมแล้ว จะพบว่า มีการใช้อุปกรณ์ Embedded ในลักษณะที่ซ้ำ ๆ กันอยู่เสมอ [10] ซึ่งทำให้เกิดความคิดว่าจะทำอย่างไรในลักษณะของการนำซอฟต์แวร์มาใช้ใหม่ [19] ซึ่งเป็นการทำให้นวัตกรรมไม่สูญเปล่า ไม่เสียเวลาในการสร้างใหม่ ทั้งฮาร์ดแวร์ และซอฟต์แวร์ อีกทั้งไม่สูญเสียทรัพยากรทางความคิดอีกด้วย ซึ่งในที่สุดหมายถึง ค่าใช้จ่ายที่น้อยกว่า และธุรกิจจะอยู่ในตำแหน่งที่ได้เปรียบคู่แข่ง

มีการค้นคว้าในวิศวกรรมซอฟต์แวร์ ว่าจะนำโปรแกรมมาใช้ใหม่ได้อย่างไร สำหรับระบบ Embedded แล้วมีการค้นคิดและเขียนโปรแกรมในลักษณะที่เรียกว่า Real-time ซึ่งอาจจะมีการใช้ Operating System ที่มีลักษณะของ Real-time อยู่ด้วยที่เรียกว่า RTOS (Real-Time Operating System) RTOS บางตัว มีคุณลักษณะของการทำงานในเชิงวัตถุ ซึ่งจะสนับสนุนการนำโปรแกรมที่เขียนมาใช้ใหม่ได้ เนื่องจากเป็นการมองอย่างเป็นระบบ โดยให้มีผู้กระทำและผู้ถูกกระทำเป็นวัตถุและกิจกรรมที่เกิดขึ้น เป็นกิจกรรมที่เกิดขึ้นเป็นปฏิสัมพันธ์ระหว่างกัน เช่น การทำงานของเครื่องจักรอุปกรณ์ต่าง ๆ เป็นกิจกรรรมที่เกิดขึ้นระหว่างวัตถุกับวัตถุ

* นักศึกษาปริญญาเอก ภาควิชาคอมพิวเตอร์ศึกษา คณะครุศาสตร์อุตสาหกรรม สจพ.

การมองแบบวัตถุนั้นเปรียบเสมือนการนำจิ๊กซอว์มาต่อกัน โดยจิ๊กซอว์แต่ละตัวเปรียบเสมือนวัตถุแต่ละตัว การต่อกันเปรียบเสมือนกิจกรรมที่เกิดขึ้นระหว่างกัน ซึ่งหมายถึง การแบ่งแยกการทำงานได้ จิ๊กซอว์แต่ละตัวมีการทำงานของมันเอง สิ้นสุดในตัวมันเอง ดังนั้นหากเกิดปัญหาในส่วนไหน หรือต้องการการเปลี่ยนแปลงในส่วนใด ก็สามารถนำวัตถุนั้นออกไป แล้วนำตัวใหม่มาใส่ ซึ่งทำให้ไม่มีปัญหาหรือผลกระทบกับส่วนอื่น ๆ

การเขียนโปรแกรมเชิงวัตถุนั้น จะมีการแบ่งแยกวัตถุแยกย่อยออกเป็น Class ต่าง ๆ ซึ่งแต่ละ Class จะมีกิจกรรมที่กระทำระหว่างกัน โปรแกรมการทำงานเชิงวัตถุบางตัวทำงานกับสิ่งแวดล้อมที่ไม่เหมือนกัน ทำให้มี Class ต่าง ๆ แยกย่อยออกไปที่ไม่เหมือนกัน ซึ่งเป็นการออกแบบ และมองทั้งระบบ ซึ่งการมองลักษณะนี้ทำให้มีการค้นคิดวิธีการเขียนโปรแกรมแนวใหม่ ซึ่งมีผู้เสนอแนวคิดที่เรียกว่า Unified Modeling Language (UML) ปัจจุบันนี้ UML มีการพัฒนาไปจนถึง UML 2.0 ซึ่งมี Diagram ทั้งหมด 13 Diagram โดยแบ่งเป็น 3 หมวดคือ Structural Diagrams, Behavior Diagrams, Model Mangement Diagrams [8] โดย diagram ทั้งหมดจะครอบคลุมการพัฒนาได้ในทุกส่วน แต่อาจมีบางส่วนที่ใช้มากน้อยแตกต่างกัน ตามแต่ application ที่ทำ

UML เริ่มต้นด้วยการออกแบบโดยการตั้งปัญหาและแก้ปัญหา เริ่มจากการตั้งปัญหาในเชิงธุรกิจ การแก้ปัญหาโดยเริ่มที่การตลาด การตอบสนองความต้องการของตลาด การออก-แบบให้ทันกับความต้องการของตลาด การวางแผนงาน การแบ่งแยกหน้าที่ การเร่งการพัฒนา การวิเคราะห์คอขวด การพัฒนาผลิตภัณฑ์ให้เร็วทั้งด้านซอฟต์แวร์ และฮาร์ดแวร์

จากกระแสความต้องการที่เปลี่ยนไปอย่างรวดเร็วดังกล่าว ทำให้การเรียนการสอนในเรื่องของระบบ Embedded นี้จะต้องมีความทันสมัยอยู่เสมอไม่แพ้กัน ในบทความนี้จะนำหลักการเรียนการสอนแบบมีส่วนร่วม [9] โดยมุ่งเน้นให้ผู้เรียนคิดเป็น ทำเป็น และแก้ปัญหาเป็น โดยพัฒนาความรู้ ความสามารถเต็มศักยภาพด้วยตนเองเป็นสำคัญ โดยผู้สอนเป็นเพียงผู้ชี้แนะและคอยช่วยเหลือให้การจัดการเรียนการสอนประสบผลดังจุดประสงค์ที่ตั้งไว้ โดยการนำหลักการของ UML แนะนำสู่นักศึกษา อภิปราย และตั้งปัญหาด้าน Embedded ให้นักศึกษาแก้ด้วยวิธีที่ได้สรุปไว้

ในตอนที่ 2 จะเป็นเรื่องของ Embedded UML และงานวิจัยที่มีผู้ที่ได้นำเสนอไว้ในตอนที่ 3 เป็นเรื่องของงานวิจัยทางด้านการศึกษา ที่เกี่ยวกับ Embedded และ UML และในตอนที่ 4 เป็นสิ่งที่ได้นำเสนอในบทความนี้ และตอนที่ 5 เป็นบทสรุป

2. Embedded UML Pattern

UML ดังที่ได้กล่าวแล้วว่าเป็น UML (Unified Modeling Language) เป็นภาษาในลักษณะของการใช้แผนภาพ และมีลักษณะพื้นฐานการทำงานในเชิงวัตถุ มีกระบวนการขั้นตอนการเขียน และเรียงลำดับอย่างเป็นระบบ เช่น ถือว่า Product จะยังใช้งานไม่ได้จนกว่า จะทำเอกสารประกอบเสร็จ [14] และมีกระบวนการขั้นตอนที่เริ่มต้นจากสถาปัตยกรรม การวิเคราะห์ การออกแบบ การนำไปใช้งาน การทดสอบ ตลอดจนถึงการส่งงาน กระบวนการนี้เน้นการคิดก่อนเริ่มทำ ทำให้งานที่ได้ค่อนข้างชัดเจน มีแบบแผนที่แน่นอน เมื่อแรกเริ่มนั้น UML สร้างขึ้นโดย Rational Software และปัจจุบันนี้ บำรุงรักษาโดยองค์กรมาตรฐาน Object Management Group (OMG) [5] และในที่สุดเป็น industry-standard language ซึ่งทำให้เราได้มีวิธีการที่จะสร้างสถาปัตยกรรม และการออกแบบได้ตรงกับความต้องการอย่างชัดเจน [17] ซึ่ง UML ปัจจุบันพัฒนาไปถึง Version 2.0 โดยมีส่วนเพิ่มจากเดิมหลายประการ โดยเฉพาะอย่างยิ่งในส่วนที่เกี่ยวกับ Service-Oriented Architect (SOA) และ Model Driven Architecture (MDA) [10] ใน Version 2.0 ได้พัฒนาจาก Version 1.0 จากเดิม 9 Diagram เป็น 13 diagram [8] ดังนี้ Activity Diagram, Class Diagram, Communication Diagram, Component Diagram, Composite Structure Diagram, Deployment Diagram, Interaction Overview Diagram, Object Diagram, Package Diagram, Sequence Diagram, State Machine Diagram, Timing Diagram และ Use Case Diagram

ต่อมาได้มีนักพัฒนาระบบนำ UML ไปใช้งานและปรับปรุงเพิ่มเติมไปบ้าง เช่น Rational Unified Process (RUP) เป็นตัวอย่างหนึ่งของการนำ UML มาใช้ได้อย่างมีประสิทธิภาพ [17] โดยมีหลักการปฏิบัติที่มีประสิทธิภาพ 6 ประการคือ Develop software iteratively, Manage requirements, Use component based architectures, Visually model software, Verify software quality และ Control changes to software

ในส่วนของระบบ Embedded นั้น ก็ได้มีนักพัฒนานำ UML ไปประยุกต์ใช้เช่นกัน ซึ่งได้แก่ B. P. Douglass. [2] ได้นำเสนอในเรื่องของความแตกต่างของส่วนประกอบระหว่าง ฮาร์ดแวร์

และซอฟต์แวร์ในเรื่องของ requirements patterns และหน้าที่ของมันที่จะต้องทำ Sascha Konrad และ Betty H.C. Cheng [7] ได้นำเสนอ Requirements Patterns for Embedded Systems ซึ่งเป็นโครงร่างสำหรับการออกแบบปัญหาทั่วไป ซึ่งสามารถนำไปใช้กับโครงงานอื่นได้ Lutz Bichler, Ansgar Radermacher และ Andreas Schurr [13] ได้ขยายส่วนของ UML ไปใช้กับระบบ Real-Time โดยการนำเสนอร่วมกับการพัฒนาอย่างเร็วของเครื่องควบคุมเครื่องปรับอากาศ Object Managment Group (OMG) ได้เสนอให้มีส่วนเพิ่มของ UML โดยสร้าง UML “profiles” [15] ซึ่งบรรจุความต้องการที่เพิ่มขึ้น

นอกจากนี้แล้วยังมีบุคคลหลายท่านที่นำเสนอ ในรูปแบบต่าง ๆ กันเช่น Kjetil Svarstad, Gabriela Nicolesucu และ Ahmed A. Jerraya [12] ได้นำเสนอตัวแบบที่แสดงถึงการติดต่อสื่อสารระหว่าง Aggregate Objects ในระดับ Specification และ design ของระบบแบบฝัง Grant Martin, Luciano Lavagno และ Jean Louis-Guerin [6] ได้นำเสนอโดยนำส่วนดีของ UML นำมาประยุกต์กับงาน Real-time โดยเสริมด้วยแนวคิดด้าน Mapping ของ Embedded ท่านทั้งหลายเหล่านี้มีส่วนให้เกิดการพัฒนา และประยุกต์งานต่าง ๆ เข้าด้วยกัน ซึ่งเป็นประโยชน์อย่างมากในการทำงานด้าน Embedded ดังนั้น จะเห็นได้ว่า UML สามารถที่จะนำมาใช้ร่วมกับ ระบบ Embedded ได้ และที่สำคัญก็คือ UML เป็น New Object-Oriented Modeling Language ที่ให้ความสำคัญกับคุณลักษณะ และเทคนิคการออกแบบ

3. Embedded Systems Education

ดังที่ได้ทราบกันดีแล้วว่า ความสลับซับซ้อนของปัญหาด้าน Embedded มีเพิ่มมากขึ้นเรื่อย ๆ ดังนั้นจึงต้องมีการพิจารณาถึงรูปแบบการเรียนการสอนที่เหมาะสม โดยมองไปในอนาคต ซึ่งมีผู้คาดการณ์ไว้ว่า การเรียนการสอนโดยการพิจารณาโต้ตอบในเรื่องของส่วนประกอบจะน้อยลง และเน้นไปเรื่องการพิจารณาโต้ตอบในส่วนของการวิเคราะห์และออกแบบ [20] ส่วนประกอบต่าง ๆ จะเพิ่มความสลับซับซ้อนมากขึ้น ความสำคัญของวิธีการจะเพิ่มมากขึ้น (เช่น UML) ตามความซับซ้อนของชิปที่ใช้ และการออกแบบซอฟต์แวร์ และยิ่งไปกว่านั้น Wayne Wolf and Jan Madsen [20] เชื่อว่าระบบการเรียนการสอนที่ดีที่สุดในการบรรจุความรู้อันเป็นพื้นฐาน ได้แก่ การเพิ่ม การสอน การให้การบ้าน และลดการทดลองไปบ้าง มากกว่าหลักสูตรการเรียนการสอนไมโครโปรเซสเซอร์ที่ขึ้นอยู่กับตัวโปรเซสเซอร์ และเขายังเชื่อว่าการทดสอบซอฟต์แวร์ และคุณลักษณะของระบบมีความสำคัญมากในการให้ทักษะแก่นักศึกษา ในการออกแบบ Embedded Systems

ในด้านการเรียนการสอน Embedded Systems Education นั้น The National Institute of Applied Sciences of Toulouse (NIAST) โดย Jean-Claude Geffroy, Claude Baron และ Gilles Motet [11] ได้นำเสนอหลักสูตรการเรียนการสอนสำหรับ ระบบ Real-Time Systems ที่เชื่อถือได้ โดยมี 5 ขั้นตอนในการพัฒนาระบบดังนี้ Specification, Design, Implementation, Production, และ Utilization ซึ่งทั้งหมดเป็นส่วนหนึ่งของ UML

ในด้านเทคนิคการเรียนการสอน ก็เป็นส่วนหนึ่งที่สำคัญในการถ่ายทอดให้นักศึกษาได้เกิดการเรียนรู้ เข้าใจ และสามารถนำไปใช้งานได้อย่างยั่งยืน ไม่แพ้เนื้อหาของรายวิชาเช่นกัน ซึ่งระบบการเรียนการสอนในปัจจุบันเน้นการเรียนการสอน ในลักษณะผสมผสานระหว่าง Problem-base Learning และการเรียนรู้โดยให้ผู้เรียนเป็นศูนย์กลาง ในส่วนของ Problem-base Learning นั้น ที่ Iowa State University โดย Aaron Striegel และ Diane T. Rover [1] ได้สร้างโมเดลการเรียนการสอนในวิชา Embedded Systems ที่เรียกว่า 3C5I ขึ้นโดย 3C หมายถึง Concepts within Course within a Curriculum และ 5I หมายถึง Introduction, Illustration, Instruction, Investigation และ Implementation โดยใช้ Problem-based learning ขยายความรู้ในส่วนของ Investigation และ Implementation ในส่วนของเนื้อหายังคงใช้เนื้อหาในแบบเดิม

ส่วนการเรียนรู้ที่เน้นผู้เรียนเป็นศูนย์กลางนั้น เป็นการเรียนการสอนที่เน้นผู้เรียนเป็นศูนย์กลาง [9] โดยอาจารย์ผู้สอนเป็นผู้ชี้แนะ โดยมีหลักการดังนี้ 1) ผู้เรียนมีส่วนในกระบวนการเรียนรู้ ให้ผู้เรียนเป็นผู้สร้าง (Construct) ความรู้ด้วยตนเอง ทำความเข้าใจ สร้างความหมาย ของสาระความรู้ให้แก่ตนเอง ค้นพบข้อความด้วยตนเอง 2) ผู้เรียนมีปฏิสัมพันธ์ (Interaction) ต่อกันและกัน และได้เรียนรู้จากกันและกัน ได้แลกเปลี่ยนข้อมูลกัน ความคิดและประสบการณ์แก่กันและกัน ให้มากที่สุดเท่าที่จะทำได้ 3) ผู้เรียนมีบทบาทมีส่วนร่วม (Participation) ในกระบวนการเรียนรู้ให้มากที่สุด 4) ผู้เรียนได้เรียนรู้ “กระบวนการ” (Process) ควบคู่ไปกับ “ผลงาน” (Product) และ 5) ผู้เรียนนำความรู้ที่ได้ไปใช้ใน

ลักษณะใดลักษณะหนึ่ง (Application)

การจัดประสบการณ์การเรียนรู้ที่เน้นผู้เรียนเป็นศูนย์กลาง จึงเป็น “กระบวนการจัดการเรียนการสอน ที่มุ่งส่งเสริมให้ผู้เรียนเกิดการเรียนรู้ โดยเน้นให้ผู้เรียนได้คิดค้น สร้าง และสรุปข้อความรู้ด้วยตนเอง สามารถทำงานร่วมกับผู้อื่น และนำความรู้ไปใช้ประโยชน์ได้” [9]

4. The Integration

ในการสร้าง Pattern สำเร็จรูป ในเชิงเนื้อหาของ Embedded UML ประยุกต์เข้าสู่ระบบการเรียนการสอน ที่เน้น Problem-base และเน้นผู้เรียนเป็นศูนย์กลางนั้น จะมีการบรรจุเนื้อหาของ UML ทั้งหมด 13 Diagram [11] ดังนี้

1) Activity Diagram เป็นเรื่องของการแสดงให้เห็นถึงกระบวนการทางธุรกิจขั้นสูง ประกอบด้วย Data Flow หรือกระบวนการทางตรรกะในระบบ ซึ่งใน Activity Diagram ในส่วนของ Business Model นี้ ในเนื้อหาของ Embedded จะไม่เน้นในส่วนนี้

2) Class Diagram เป็นการรวบรวมกลุ่มของส่วนประกอบต่าง ๆ เช่น Class, Type เนื้อหา และความสัมพันธ์ของมัน

3) Communication Diagram เป็นเรื่องของการติดต่อสื่อสารระหว่างวัตถุที่ได้นิยามไว้

4) Component Diagram เป็นส่วนประกอบ และความสัมพันธ์ระหว่างกัน รวมถึงส่วนของงานประยุกต์ และระบบ

5) Composite Structure Diagram เป็นการประกอบกันเข้าของโครงสร้าง เช่น Class, Component หรือ Use Case

6) Deployment Diagram เป็นการแสดงถึงการทำงานของสถาปัตยกรรมของระบบ

7) Interaction Overview Diagram เป็นการแสดงถึงกิจกรรมระหว่าง Node

8) Object Diagram เป็นการแสดงถึงวัตถุและความสัมพันธ์ระหว่างกัน ในช่วงของเวลา

9) Package Diagram แสดงให้เห็นองค์ประกอบของโมเดลที่จัดกลุ่มเป็นแพ็คเกจ

10) Sequence Diagram เป็นโมเดลที่มีความสัมพันธ์กับลำดับการทำงานตามเวลาที่เปลี่ยนไป

11) State Machine Diagram อธิบายถึงวัตถุหรือการทำงานภายในระหว่างกัน

12) Timing Diagram อธิบายถึงการเปลี่ยน State หรือเงื่อนไขตามเวลาที่เปลี่ยนไป

13) Use Case Diagram แสดงถึง Use Cases, Actors และความสัมพันธ์ภายในระหว่างกัน

นอกเหนือไปจาก Diagram ข้างต้น มีการเพิ่มเติมในส่วนของข้อจำกัด พฤติกรรม และ รูปแบบของการออกแบบ [18] โดยมี Pattern ดังนี้ Pattern Name and Classification: เป็นชื่อเรียกของ Pattern ที่มีรายละเอียดของ Pattern และการจัดกลุ่มที่ขึ้นกับวัตถุประสงค์ของ Pattern นั้น Intent เป็นเจตนาหรือวัตถุประสงค์ที่ต้องการแก้ปัญหานั้น Motivation เป็นเป้าหมาย หรือวัตถุประสงค์ของระบบที่จะใช้ Pattern Constraints เป็นข้อจำกัดที่มีอยู่ในระบบ Applicability ลักษณะงานของ Pattern ที่นำไปใช้ Structure แสดงรายละเอียดของ Class และ ความสัมพันธ์ของมัน ในเทอมของ UML Class Diagram Behavior แสดงพฤติกรรมของกิจกรรมระหว่างกันของ Class และ Object โดยการใช้ UML state และ Sequence Diagram Participants แยกรายการของ Class และ Object ซึ่งอยู่ในความต้องการของ Pattern และการตอบสนองของมัน Collaborations อธิบายถึง Object และ Class มีปฏิสัมพันธ์กันอย่างไร Consequences Pattern ผลที่จะเกิดขึ้นเป็นอย่างไร และบรรลุจุดประสงค์ได้อย่างไร Design Patterns [4] ถูกนำมาใช้เฉพาะใน Pattern ที่จะทำให้ Requirement Pattern ชัดเจนยิ่งขึ้น Also Known As แสดงชื่ออื่นเพื่อใช้ใน Requirement Pattern Related Requirements Patterns มี Requirement pattern อื่นที่มีความสัมพันธ์กับมันอีกไหม เช่น รูปแบบสำหรับการวิเคราะห์ระบบ Embedded ดังนี้ Controller Decompose แยกส่วนของระบบ Embedded ออกเป็นส่วนตามภาระที่ต้องตอบสนอง Actuator-Sensor Pattern จะแบ่งแยกความแตกต่างระหว่าง Sensor และ Actuators ในแต่ละชนิดได้อย่างไร Watchdog Pattern การเฝ้าระวังอุปกรณ์ และเริ่มต้นทำงานที่ถูกต้องใหม่ เมื่อเกิดการทำงานที่ผิดพลาดได้อย่างไร Examiner Pattern มีการเฝ้าระวังอุปกรณ์ และเก็บลักษณะความผิดพลาดที่เกิดขึ้นได้อย่างไร Fault Handler Pattern จะมีการรับมือกับการทำงานที่ผิดพลาดในระบบ Embedded ได้อย่างไร Mask Pattern จะลดภาระของ Computing component เมื่อมี Sensor และ Actuator หลายตัวในระบบ Moderator Pattern จะมีการเตรียมการ และรับมือกับ Subsystem ที่ซับซ้อนได้อย่างไร User Interface Pattern มีการระบุรายละเอียดของ User interface ที่มีส่วนขยาย และนำมาใช้ใหม่ได้อย่างไร Channel Pattern มีการจัดการการติดต่อสื่อสารระหว่าง

อุปกรณ์ 2 ชนิดได้อย่างไร Monitor-Actuator Pattern จะเพิ่มความปลอดภัย และความเชื่อถือได้ โดยการเฝ้าติดตามความผิดพลาดของ actuator behavior ได้อย่างไร

การเรียนการสอนโดยการใช้ Pattern นี้ร่วมกับแนวคิดในการเรียนการสอน แบบ Problem-base Learning [3] ผสมผสานกับการเรียนการสอนแบบเน้นผู้เรียนเป็นศูนย์กลางนั้น คาดว่าจะทำให้เกิดผลดีต่อผู้ที่ศึกษาในด้านผลสัมฤทธิ์และการนำไปใช้ดังนี้ ให้นักศึกษามีส่วนร่วม และได้เรียนรู้ในสิ่งที่ต้องการ เกิดการเรียนรู้ใหม่ ๆ ที่ท้าทายต่อเนื่อง และเป็นฝ่ายกระทำ (Active Learning) นักศึกษาสามารถค้นคว้าจากสื่อต่าง ๆ กัน วิธีการประยุกต์ ทำได้โดยการยกตัวอย่างให้นักศึกษานำมาอภิปราย และได้รู้จักนำมาคิดพิจารณาในเรื่องของการประยุกต์ใช้ เช่น ลักษณะของ Keyboard จะมีการนำมาใช้ใหม่ได้อย่างไร เนื่องจาก Keyboard มีการนำมาใช้เป็นปุ่มกดเพื่อพิมพ์ข้อความเข้าคอมพิวเตอร์ มีการนำ Keyboard มาใช้เป็นปุ่มกดในเครื่องรับโทรศัพท์ บ้านและมือถือ หรือมีการนำมาใช้ในเครื่อง palm เป็นต้น หรือลักษณะของการติดต่อสื่อสารควบคุมหน้าจอ ที่เป็น LED เป็นต้น ซึ่งมีใช้ทั้งในโทรศัพท์บ้าน มือถือ เครื่องซักผ้า เตาไมโครเวฟ ฯลฯ

5. บทสรุป

จากความต้องการของมนุษย์ที่มีเพิ่มมากขึ้น ในด้านความสะดวกสบาย และการติดต่อสื่อสารข้อมูล ทำให้การออกแบบระบบ Embedded มีความสลับซับซ้อนมากขึ้น การเรียนการสอนในระบบ Embedded จึงต้องมีการปรับเปลี่ยนให้ทันกับความเปลี่ยนแปลงด้วย การที่จะทำให้ผู้เรียนมีความสามารถสูงเพิ่มขึ้นในด้านการเรียนรู้ และการนำไปใช้ มีความจำเป็นอย่างยิ่งที่จะต้องทำให้ผู้เรียนเพิ่มขีดความสามารถในการเรียนรู้มากขึ้นตามเนื้อหาที่เพิ่มขึ้น เนื้อหาจำเป็นที่จะต้องมีการปรับเปลี่ยนให้เป็นระบบ เพื่อให้มีพัฒนาความคิดอย่างเป็นระบบตามไปด้วย UML เป็นคำตอบหนึ่งของระบบ Embedded ในปัจจุบันการเรียนการสอนจะต้องมีรูปแบบสำเร็จที่เป็นแนวทางให้ผู้เรียนได้เรียนรู้และพัฒนาไปตามแนวทางที่ได้วางไว้ นอกเหนือจากนั้นจำเป็นที่จะต้องมีการใช้ระบบการเรียนการสอนที่มีประสิทธิภาพควบคู่ไปด้วย การผสมผสานการเรียนรู้ด้วยวิธีการ Problem-base Learning และการเรียนรู้ที่เน้นผู้เรียนเป็นศูนย์กลางนั้น ได้พิสูจน์แล้วว่าทำให้ผู้เรียนมีผลสัมฤทธิ์ทางการเรียนได้มาก อย่างไรก็ตามการเรียนรู้ ระบบ Embedded โดยใช้ Pattern สำเร็จรูป กับแนวทางการเรียนการสอนแบบผสมผสานนี้ ยังไม่ได้มีการนำไปวิจัยว่า จะได้ผลสัมฤทธิ์ดีเพียงใด ซึ่งเป็นสิ่งที่จะต้องทำวิจัยในด้านการศึกษาต่อไป

บรรณานุกรม

[1] Aaron Striegel, Diane T. Rover, “Problem-Based Learning in an Introductory Computer Engineering Course,” *32nd ASEE/IEEE Frontiers in Education Conference,* Boston, MA, *November 6-9, 2002*.

[2] B. P. Douglass. *Doing Hard Time: Developing Real Time Systems with UML, Objects, Frameworks and Patterns.* Addison-Wesley, 1999.

[3] Doug L. Maskell and Peter J. Grabau, *A Multidisciplinary Cooperative Problem-Based Learning Approach to Embedded Systems Desing*, IEEE.

[4] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

[5] Grady Booch, Ivar Jacobson, and James Rumbaugh, *Unified Modeling Language 1.3*, White paper, Rational Software corp., 1998.

[6] Grant Martin, Luciano Lavagno, Jean Louis-Guerin, *Embedded UML: a merger of real-time UML and co-design,* White paper on the Profile mechanism, 1999. document ad/99-04-07.

[7] http://community.borland.com/article/0,1410,31881, 00.html.

[8] http://www.agilemodeling.com/essays/uml Diagrams.htm.

[9] http://www.termsak.com/teaching18.html

[10] Jean J.Labrosse. *Embedded Systems Building Blocks, Complete and Ready-to-Use Modules in C*. R&D Technical Books, 1995.

[11] Jean-Claude Geffroy, Claude Baron, Gilles Motet, *Dependability Issues for a Curriculum in Real-Time Systems*, White paper, LESIA, DGEI/INSA, Complexe Scientifique de Rangueil, 31077 Toulouse, France.

[12] Kjetil Svarstad, Gabriela Nicolesucu, Ahmed A. Jerraya, *A model for Describing Communication between Aggregate Objects in the Specification and Design of Embedded Systems*, 1530-1591/01, IEEE 2001, pp.77-84.

[13] Lutz Bichler, Ansgar Radermacher, Andreas Schurr, "Evaluating UML Extensions for Modeling Real-time Systems," *Proceeding of the Seventh International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS 2002).*

[14] Michael Priestley, "A Unified Process for Software and Documentation Development," *0-7803-6413-7, IEEE 2000,* pp. 211-237.

[15] OMG, *White paper on the Profile mechanism*, 1999. document ad/99-04-07.

[16] Patrick Tessier, Sebastien Gerard, Chokri Mraidha, Francosi Terrier, Jean-Marc Geib, "A Component-Based Methodology for Embedded System Prototyping," *Proceedings of the 14th IEEE International Workshop on Rapid Systems Prototyping (RSP'03),* 2003.

[17] Rational Unified Process "Base Practices for software Development Teams" A Rational Software Corporation White Paper, 1998.

[18] Sascha Konrad and Betty H.C. Cheng, "Requirements Patterns for Embedded Systems," *Proceedings of the IEEE Joint International Conference on Requirements Engineering,* 2002.

[19] Uwe Rastofer, Frank Bellosa. *An Approach to Component-Base Software Engineering from DistributedEmbedded Real-Time Systems.* White paper, Departmant of Computer Science IV, University of Erlangen-Nurnberg, MartensstraBe 1, 91058 Erlangen, Germany.

[20] Wayne Wolf and Jan Madsen, "Embedded Systems Education for the Future," *Proceedings of the IEEE, Vol. 88, No.1*, January 2000, pp. 23-30.