การขยายแบบจำลองการออกแบบที่อิสระจากแพลตฟอร์มให้มีคุณสมบัติด้านความปลอดภัย

Extending PIM with Security Features

ณัฐกานต์ สรรพจักร* และ ทรงศักดิ์ รองวิริยะพานิช*

บทคัดย่อ

การพัฒนาซอฟต์แวร์แบบ MDA เป็นแนวคิดในการใช้แบบจำลอง (Software Model) ในการขับเคลื่อนกระบวนการพัฒนาซอฟต์แวร์ซึ่งการพัฒนาซอฟต์แวร์ส่วนใหญ่นักวิเคราะห์และออกแบบระบบจะสร้างแบบจำลอง PIM โดยให้ความสำคัญกับความต้องการหลัก (Functional Requirement) ของระบบเป็นหลัก ในขณะที่ความต้องการด้านความปลอดภัย (Security) จะถูกนำไปเพิ่มเติมให้กับแอพพลิเคชันหลังกระบวนการพัฒนาเสร็จเรียบร้อยแล้ว ซึ่งทำให้เกิดปัญหาเรื่องค่าใช้จ่ายในการพัฒนาสูงขึ้น งานวิจัยนี้จึงเสนอแนวทางการในการเพิ่มคุณสมบัติด้านความปลอดภัยลงในแบบจำลอง PIM ครอบคลุมเรื่องการพิสูจน์ตัวจริง (Authentication) การรักษาความลับ (Confidentiality) และการรักษาความสมบูรณ์ (Integrity) โดยใช้วิธีการเพิ่มขยายเมตาโมเดลของ UML และเสนอแนวทางในการแปลงแบบจำลอง PIM ที่มีการกำหนดคุณสมบัติความปลอดภัยเป็นแบบจำลอง PSM ที่รองรับการรักษาความปลอดภัยได้ ภายใต้การพัฒนาแอพพลิเคชันเว็บเซอร์วิส และมีการกำหนดวิธีการรักษาความปลอดภัยโดยใช้ WS-Security

คำสำคัญ: เอ็มดีเอ แบบจำลองที่อิสระจากแพลตฟอร์ม ความปลอดภัย เมตาโมเดลของความปลอดภัยบนเว็บเซอร์วิส

Abstract

The Model-Driven Architecture concept is a recent evolution of software development that uses models to drive software process. Developing application in conventional way, analyst and designer will concentrate first on the functional requirement to create PIM. While the security requirement, will be incorporated after the complete application development. This leads to greatly increased development costs when defects are found. In this paper, we propose an approach to incorporating the PIM with security requirements early in analysis and design phases. The security requirements cover authentication, confidentiality and integrity. Then we propose an approach to transforming this PIM to PSM. We used web service application as an example to validate our approach and WS-Security for the implementation of security requirement.

Keyword: MDA, PIM, Security, WS-Security Metamodel

1. บทนำ

ในกระบวนการพัฒนาซอฟต์แวร์ส่วนใหญ่ นักพัฒนาให้ความสำคัญในการพัฒนาแอพพลิเคชันในส่วนของความต้องการที่เป็นฟังก์ชันของระบบ (Functional Requirements) เป็นลำดับแรก ส่วนความต้องการที่ไม่ใช่เชิงฟังก์ชันหลักของระบบ (Non-Functional Requirements) โดยเฉพาะอย่างยิ่งความปลอดภัย (Security) จะถูกนำไปพัฒนาเพิ่มเติมหลังจากที่มีการพัฒนาแอพพลิเคชันเสร็จเรียบร้อยแล้ว ซึ่งจากเหตุการณ์ดัง กล่าวทำให้เกิดปัญหาในด้านค่าใช้จ่ายของการพัฒนาที่เพิ่มขึ้น ถ้าแอพพลิเคชันนั้นมีข้อบกพร่องเกิดขึ้นในระหว่างการพัฒนา [1] จึงมีงานวิจัยที่เสนอแนวทางในการนำความต้องการที่ไม่ใช่เชิงฟังก์ชันหลักของระบบ [2], [3], [4] และ [5] อย่างเช่น ความน่าเชื่อถือ (Reliability),

* คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

ประสิทธิภาพ (Performance) รวมถึงความปลอดภัย (Security) มารวมไว้ในขั้นตอนการวิเคราะห์และออกแบบระบบ เพื่อที่จะช่วยลดข้อผิดพลาดและค่าใช้จ่ายในการพัฒนาแอพพลิเคชัน นอกจากนี้การเพิ่มขอบ เขตของขั้นตอนการวิเคราะห์และออกแบบให้ครอบคลุมความต้องการที่ไม่ใช่เชิงฟังก์ชันหลักของระบบจะเป็นการเพิ่มความสามารถในการผลิต (Productivity) และความสามารถในการดูแลรักษา (Maintainability) ด้วย

ในขณะที่แนวคิดหรือกระบวนการในการพัฒนาซอฟต์แวร์มีการพัฒนาไปอย่างต่อเนื่อง และมีความก้าวหน้าไปในระดับที่นักพัฒนาโปรแกรมสามารถสร้างแบบจำลองการออกแบบระบบเป็นแผนภาพ (Diagram) และสามารถสร้างโค้ดโปรแกรมของซอฟต์แวร์ได้แบบอัตโนมัติ ซึ่งทำให้ลดระยะ เวลาในการพัฒนาโปรแกรมลงไปอย่างมาก กระบวนการ พัฒนาซอฟต์แวร์แบบนี้เรียกว่า Model-Driven Architecture (MDA) [6] เป็นแนวคิดในการใช้แบบจำลอง เป็นศูนย์กลางในการขับเคลื่อนกระบวนการพัฒนาซอฟต์แวร์ โดยมีการแบ่งแบบจำลองออกเป็นหลายระดับ แบบจำลองในแต่ละระดับจะมีการกำหนดรายละเอียดที่ใช้อธิบายถึงระบบที่ต้องการพัฒนาแตกต่างกัน ซึ่งในการช่วงวิเคราะห์และออกแบบระบบนักวิเคราะห์และออกแบบระบบจะสร้างแบบจำลองที่อิสระจากแพลตฟอร์ม (Platform Independent Model: PIM) สำหรับใช้อธิบายถึงการทำงานของระบบซึ่งยังไม่มีการกำหนดแพลตฟอร์ม (Platform) ที่ต้องการเลือกใช้ และส่วนใหญ่แบบจำลอง PIM จะถูกอธิบายโดยใช้ภาษา UML (Unified Modelling Language) ซึ่งในขั้นตอนนี้แบบจำลอง PIM จะใช้อธิบายในส่วนของฟังก์ชันหลักของระบบ ส่วนความต้องการที่ไม่ใช่เป็นฟังก์ชันหลัก โดยเฉพาะเรื่องความปลอดภัยยังไม่ได้ถูกเพิ่มเติมเข้ามา

จากปัญหาดังกล่าวงานวิจัยนี้จึงเสนอแนวทางในการเพิ่มคุณสมบัติความปลอดภัยในแบบจำลอง PIM ในระหว่างการวิเคราะห์และออกแบบระบบ โดยกลไกในการเพิ่มคุณสมบัติความปลอดภัยในแบบจำลองทำได้โดยการเพิ่มขยาย (Extension) แบบจำลองความปลอดภัยในเมตาโมเดล (Meta model) ของ UML แล้วให้นักวิเคราะห์และออกแบบนำเมตาโมเดลดังกล่าวไปใช้สร้างแบบจำลอง PIM ที่มีคุณสมบัติความปลอดภัย และเสนอแนวทางในการแปลงแบบจำลอง (Model Transformation) จากแบบจำลอง PIM ที่มีคุณสมบัติความปลอดภัย ให้เป็นแบบจำลองที่เฉพาะเจาะจงกับแพลตฟอร์ม (Platform Specific Model: PSM) ซึ่งเมื่อนำ PSM ที่ได้มาพัฒนาเป็นโค้ดจะได้แอพพลิเคชันที่มีคุณสมบัติความปลอดภัยครบตามความต้องการที่ระบุในแบบจำลอง PIM

เนื้อหาในบทความนี้ประกอบด้วยในหัวข้อ 2 กล่าวถึงงานวิจัยที่เกี่ยวข้อง ในข้อ 3 จะอธิบายถึงภาพรวมกระบวนการเพิ่มคุณสมบัติความปลอดภัยในแบบจำลอง PIM หัวข้อ 4 อธิบายการสร้างเมตาโมเดลของแบบจำลอง PIM และ PSM หัวข้อ 5 อธิบายตัวอย่างการใช้งานแบบจำลอง ส่วนสุดท้ายของบทความนี้จะเป็นส่วนอภิปราย และสรุปผล

2. งานวิจัยที่เกี่ยวข้อง

ในหัวข้อนี้จะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับการนำคุณสมบัติด้านความปลอดภัยมาประกอบรวมกับแบบจำลองของแอพพลิเคชัน โดย Jurjens [2] ได้เสนอ UMLSec เป็น UML Profile สำหรับความปลอดภัยโดยใช้กลไกการสร้าง Stereotype และ Tagged Value ซึ่งประเด็นด้านการรักษาความปลอดภัยที่เสนอประกอบด้วย ความปลอดภัยบนเส้นทางการสื่อสาร (Data link) และการเข้าถึงข้อมูลโดยใช้ Mandatory Access Control

Lodderstedt และคณะ [3] ได้เสนอ SecureUML เป็นการกำหนดเมตาโมเดล และ UML Profile สำหรับจัดการเรื่อง Role-Based Access Control (RBAC) ให้นักออกแบบสามารถนำไปใช้งานพัฒนาแอพพลิเคชันที่รองรับการทำงานแบบ RBAC ได้ โดยได้แสดงตัวอย่างการนำ SecureUML ไปใช้ในการพัฒนาการรักษาความปลอดภัยบน Enterprise JavaBeans (EJB)

โดย Jurjens [2] และ Lodderstedt และคณะ [3] ทั้งสองงานวิจัยเน้นไปที่การกำหนดความปลอดภัยที่เป็นการควบคุมการเข้าถึง ซึ่งแตกต่างจากงานวิจัยนี้คือจะมุ่งสนใจประเด็นการรักษาความปลอดภัยในการรักษาความสมบูรณ์และการรักษาความลับด้วย

Nakamura และคณะ [4] ได้เสนอเฟรมเวิร์คในการสร้างไฟล์คอนฟิคกูเรชัน (Configuration) ที่เป็นส่วนของการรักษาความปลอดภัยบนเว็บเซอร์วิส [4] ได้เสนอวิธีการสร้างไฟล์คอนฟิคกูเรชันโดยใช้แนวคิด MDA ซึ่งกระบวนการแปลงแบบจำลองใช้วิธีการ แปลงแบบ Model-to-Code และเพิ่มคุณสมบัติด้านความปลอดภัยทำโดยใช้วิธีการเพิ่มอิลิเมนต์

พิเศษใน UML ในขั้นตอนการสร้างแบบจำลองของแอพพลิเคชัน จากนั้นก็นำเข้าสู่กระบวนการแปลงแบบจำลอง จนได้ไฟล์คอนฟิคกูเรชันสำหรับการรักษาความปลอดภัยบนเว็บเซอร์วิส แต่เครื่องมือนี้จะเฉพาะเจาะจงสำหรับแพลตฟอร์มของ WebSphere Application Server ซึ่งแตกต่างกับงานวิจัยนี้ คือ ในงานวิจัยนี้กำหนดการแปลงแบบจำลองแบบ Model-to-Model โดยแบบจำลองปลายทางที่ได้จะเป็นแบบจำลองที่ประกอบด้วยโครงสร้างแอพพลิเคชันรวมกับคุณสมบัติความปลอดภัย

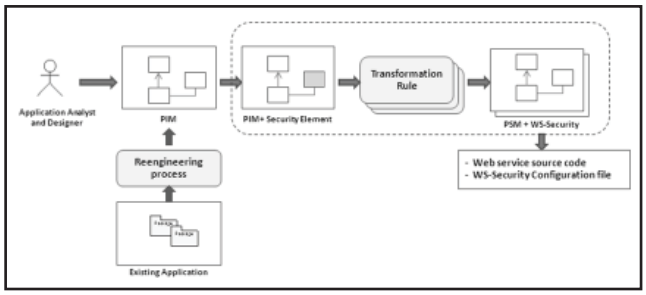
3. ภาพรวมกระบวนการเพิ่มคุณสมบัติความปลอดภัยในแบบจำลอง PIM

กระบวนการพัฒนาระบบเริ่มต้นจากนักวิเคราะห์และออกแบบระบบนำความต้องการระบบมาทำการวิเคราะห์และออกแบบ ซึ่งจะได้ผลลัพธ์เป็นแบบจำลองสำหรับใช้อธิบายการทำงานของระบบที่ต้องการ ซึ่งส่วนใหญ่แบบจำลองนี้ถูกอธิบายโดยใช้ภาษา UML ในกรณีที่เป็นแอพพลิเคชันที่มีอยู่แล้วแต่ยังไม่มีการสร้างแบบจำลอง PIM มาก่อน ในส่วนนี้สามารถสร้างแบบจำลอง PIM ได้โดยวิธีการรีเอ็นจิเนียริ่ง (Reengineering) แต่ในงานวิจัยนี้ไม่ได้กล่าวถึงกระบวนการดังกล่าวซึ่งแบบจำลอง PIM จะอธิบายถึงระบบในส่วนของความต้องการที่เป็นฟังก์ชันหลัก ส่วนความต้องการของระบบด้านความปลอดภัย ซึ่งเป็นความต้องการที่ไม่เชิงฟังก์ชันหลักนั้นจะถูกนำมาเพิ่มลงในแบบจำลอง PIM โดยหลังจากที่ผ่านขั้นตอนนี้ไปแล้วจะได้แบบจำลอง PIM ที่มีคุณสมบัติด้านปลอดภัย โดยในขั้นตอนถัดไปคือนำแบบจำลองดังกล่าวเข้าสู่กระบวนการแปลงแบบ จำลองเพื่อสร้างแบบจำลองปลายทางที่เป็นแบบจำลองที่เฉพาะเจาะจงกับแพลตฟอร์ม ซึ่งในที่นี้ได้กำหนดแอพพลิเคชันคือเว็บเซอร์วิสแอพพลิเคชันสำหรับแพลตฟอร์ม J2EE และกำหนดคุณสมบัติความปลอดภัยโดยใช้ WS-Security ซึ่งภาพรวมของกระบวนการนี้แสดงในภาพที่ 1

4. การสร้างเมตาโมเดลของแบบจำลอง

4.1 การกำหนดคุณสมบัติด้านความปลอดภัย

ในงานวิจัยนี้ได้กำหนดความต้องการด้านความปลอดภัยของแอพพลิเคชันใน 3 ด้าน โดย [7] นิยามความปลอดภัยไว้ดังนี้



ภาพที่ 1 *แผนภาพรวมการแปลงแบบจำลอง*

4.1.1 การพิสูจน์ตัวจริง (Authentication) คือ การอนุญาตให้ผู้ใช้งานเข้าถึงโปรแกรมหรือกระบวนการถูกต้องตามสิทธิ์ที่ได้รับ

4.1.2 การรักษาความลับ (Confidentiality) คือแนวคิดสำหรับการควบคุมข้อมูลที่มีความสำคัญให้อยู่ในสถานะที่ไว้ใจได้ โดยที่จะถูกจำกัดอยู่ในกลุ่มคนหรือองค์กรที่มีการจัดสรรไว้เท่านั้น

4.1.3 การรักษาความสมบูรณ์ (Integrity) คือคุณสมบัติของข้อมูลที่มีคุณภาพตามที่ได้คาดหวังไว้ โดยเหตุผลที่เลือกความต้องการด้านความปลอดภัยทั้ง 3 ด้านดังกล่าว เนื่องจากเป็นความปลอดภัยพื้นฐานของแอพพลิเคชัน และประกอบกับการรักษาความปลอดภัยบนเว็บเซอร์วิส มีเทคโนโลยีที่รองรับการทำงานตามความต้องการครอบคลุมทั้ง 3 ด้านนี้ได้อย่างสมบูรณ์แล้ว ซึ่งสามารถจับคู่ความต้องการของระบบด้านความปลอดภัยกับข้อกำหนดการรักษาความปลอดภัยบนเว็บเซอร์วิส [8] ได้ดังแสดงในตารางที่ 1 ความต้องการในเรื่องการพิสูจน์ตัวจริงพัฒนาโดยใช้ Security Token ตัวอย่างเช่นใช้ UsernameToken การรักษาความลับสามารถพัฒนาโดยใช้ XML Encryption และการรักษาความสมบูรณ์สามารถพัฒนาโดย XML Signature

ตารางที่ 1 *คุณสมบัติด้านความปลอดภัยกับเทคโนโลยี WS-Security*

Security Requirements	**WS-Security Implementation**
Authentication	UsernameToken
Confidentiality	XML Encryption
Integrity	XML Signature

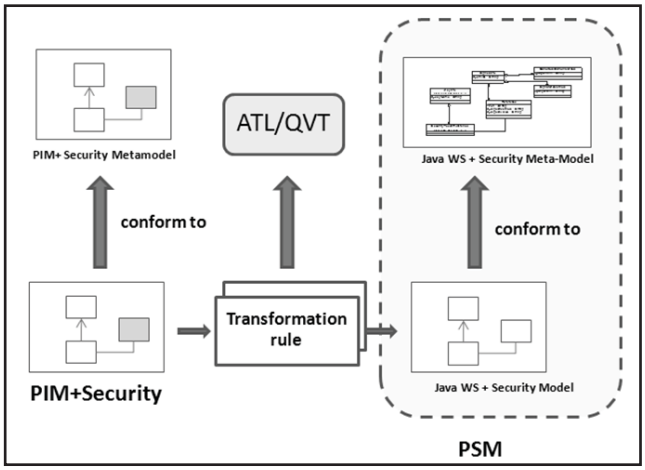
4.2 วิธีการเพิ่มคุณสมบัติความปลอดภัยในแบบจำลอง PIM

วิธีการเพิ่มความปลอดภัยลงไปในเมตาโมเดลของแบบจำลอง PIM ทำได้โดยนำความต้องการด้านความปลอดภัย

ที่กล่าวในหัวข้อ 4.1 มาเพิ่มขยายเมตาโมเดลของ UML เพื่อที่จะทำให้เมตาโมเดลของ UML นั้นสามารถรองรับความต้องการด้านคุณสมบัติของความปลอดภัยได้ วิธีการสร้างเมตาโมเดลทำได้โดยใช้ภาษา Meta-Object Facility (MOF)

4.3 แนวทางการแปลงแบบจำลอง

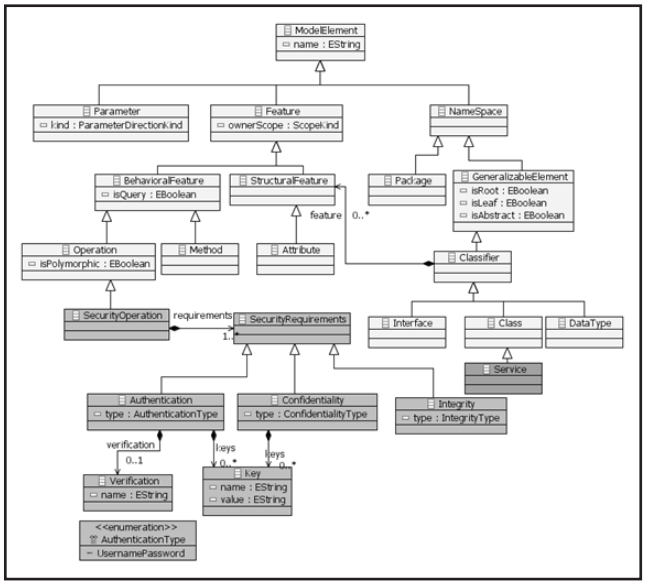
ในงานวิจัยนี้ใช้วิธีการแปลงแบบจำลองแบบ Metamodel Transformation [6] ซึ่งวิธีการนี้แบบจำลองต้นทาง (Source model) และแบบจำลองปลายทาง (Target model) ที่สร้างจะต้องสอดคล้อง (Conform) กับเมตาโมเดลของแต่ละแบบจำลอง ซึ่งกล่าวได้ว่าเมตาโมเดลคือ ภาษาที่ใช้ในการสร้างแบบจำลองต้นทางและปลายทาง โดยแบบจำลองต้นทางในที่นี้คือ แบบจำลอง PIM ที่ได้เพิ่มคุณสมบัติความปลอดภัย ซึ่งแบบจำลองนี้สร้างมาจากเมตาโมเดลของ UML ที่มีการเพิ่มการกำหนดเรื่องความปลอดภัยแล้วและแบบจำลองปลายทางคือ แบบจำลองจาวาเว็บเซอร์วิสที่รวมแบบจำลอง WS-Security เข้าไว้ด้วยกันในแบบจำลองเดียว (Java Web Service + WS-Security Metamodel) การกำหนดกฎในการแปลงแบบจำลอง (Transformation Rule) สามารถกำหนดขึ้นโดยใช้ภาษา ATL [9] หรือ QVT [10] โดยกฎในการแปลงแบบจำลองจะเป็นการจับคู่ระหว่างเมตาโมเดลของแบบจำลองต้นทาง และเมตาโมเดลของแบบจำลองปลายทาง ในงานวิจัยนี้กำหนดเมตาโมเดลของแบบจำลองต้นทางคือ เมตาโมเดลของ UML ที่รวมกับส่วนเพิ่มขยายเรื่องความปลอดภัย ส่วนเมตาโมเดลของแบบจำลองปลายทางคือ เมตาโมเดลของจาวาเว็บเซอร์วิสที่มีการเพิ่มขยายเมตาโมเดลให้รองรับโครงสร้างความปลอดภัยของ WS-Security ซึ่งกระบวนการแปลงแบบจำลองแสดงในภาพที่ 2



ภาพที่ 2 *ภาพรวมกระบวนการแปลงแบบจำลอง*

4.4 เมตาโมเดลของแบบจำลอง PIM

เมตาโมเดลของแบบจำลอง PIM คือเมตาโมเดลของ UML (ที่ไม่กำหนดแรเงา) รวมกับส่วนเพิ่มขยายเรื่องความปลอดภัยทั้ง 3 ด้าน (กำหนดเป็นคลาสที่มีแรเงา) โดยมีการเพิ่มขยาย SecurityOperation ให้เป็นซับคลาส (Subclass) ของคลาส Operation หมายความว่า ในการสร้างอิลิเมนต์คลาส (Class) สามารถมี SecurityOperation ที่รองรับความปลอดภัยได้ และจะประกอบด้วย SecurityRequirements ได้มากกว่าหนึ่ง ในที่นี้สามารถเป็นคลาส Authentication Confidentiality หรือ Integrity ก็ได้ ซึ่งในกรณีนี้ต้องการให้แบบจำลอง PIM ที่สร้างขึ้นมานั้นสามารถระบุให้แบบจำลองปลายทางทราบได้ว่าต้อง การสร้างเป็นเว็บเซอร์วิส จึงได้ทำการเพิ่มขยายในอิลิเมนต์ Class ให้มีซับคลาสเป็น Service แสดงในภาพที่ 3

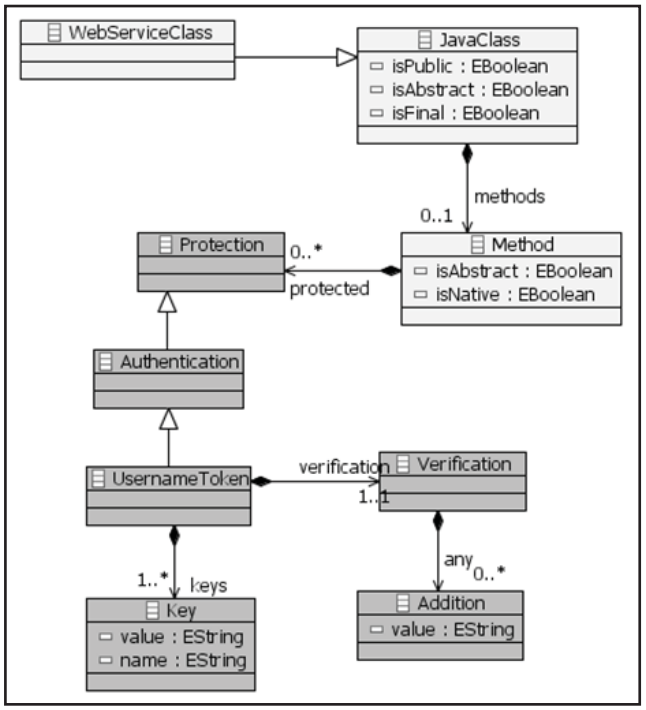


ภาพที่ 3 *เมตาโมเดลของแบบจำลอง PIM ที่เพิ่มเติมคุณสมบัติความปลอดภัยทั้ง 3 ด้าน*

4.5 เมตาโมเดลของแบบจำลอง PSM

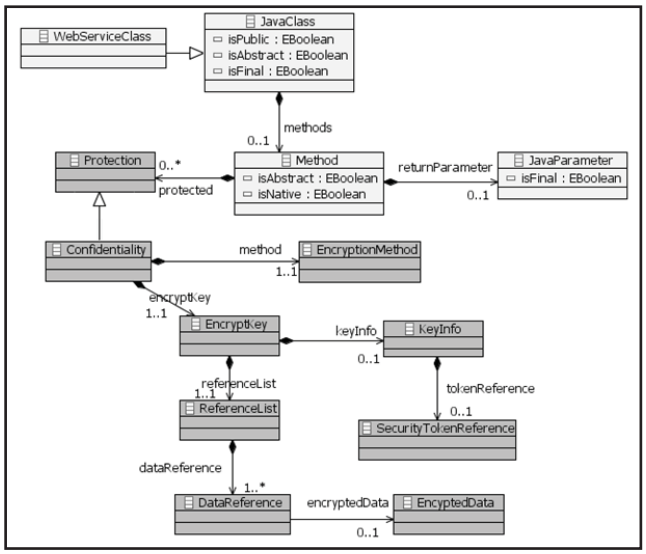
การกำหนดเมตาโมเดลของแบบจำลองปลายทาง กำหนดโดยสร้างเมตาโมเดลของจาวาเว็บเซอร์วิส โดยเพิ่มขยายในส่วนความปลอดภัยของเว็บเซอร์วิสโดยการอ้างอิงโครงสร้างจากข้อกำหนดการรักษาความปลอดภัยของ WS-Security [8] ซึ่งกลไกการรักษาความปลอดภัยบนเว็บเซอร์วิสจะกระทำในระดับของเมท็อด (Method) ดังนั้นในแบบจำลองปลายทางจึงได้กำหนดส่วนเพิ่มขยายส่วนความปลอดภัยในระดับของเมท็อดโดยคลาส Method ประกอบด้วยคลาส Protection

เป็นส่วนเพิ่มเติมความปลอดภัยของ WS-Security โดยจะมีซับคลาสเป็น Authentication Confidentiality และ Integrity ซึ่งส่วนเพิ่มขยายเมตาโมเดลของ Authentication แสดงในภาพที่ 4 โดยอ้างอิงจากโครงสร้างของ UsernameToken นอกเหนือ จากนี้ WS-Security มีข้อกำหนดอื่นๆ Binary SecurityToken สำหรับแก้ปัญหาเรื่องของการพิสูจน์ตัวจริง แต่ในงานวิจัยนี้ ยกตัวอย่างเมตาโมเดลของ UsernameToken เนื่องจากเป็นวิธีการที่ไม่ซับซ้อน โดยเมตาโมเดลของการพิสูจน์ตัวจริงประกอบอิลิเมนต์ชื่อ UsernameToken ซึ่งประกอบด้วย Username เก็บค่าผู้ใช้งานเพื่อนำไปใช้ในการตรวจสอบตัวตนกับผู้ให้บริการเว็บเซอร์วิส และมีอิลิเมนต์อื่นๆ เช่น Password เป็นประเภทของคลาสที่ชื่อ Key และส่วนคลาส Verification เป็นคลาสที่ใช้สำหรับกำหนดกลไกของการพิสูจน์ตัวจริง



ภาพที่ 4 *เมตาโมเดลของ PSM ในส่วน Authentication*

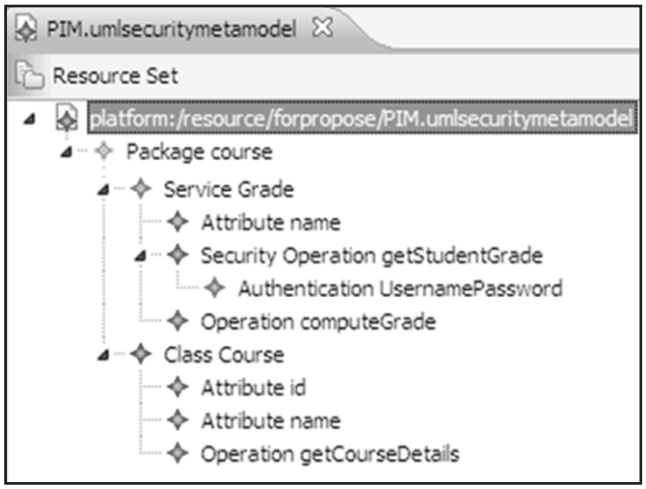
ส่วนซับคลาส Confidentiality จะถูกกำหนดโดยอ้างอิงโครงสร้างจาก XML Encryption ตามข้อกำหนดของ WS-Security ซึ่ง Confidentiality ประกอบด้วยคลาสดังนี้ EncryptedData เป็นคลาสที่ใช้ในการเก็บข้อมูลที่มีการเข้ารหัส (Encrypt) ไว้ EncryptKey เป็นคลาสที่ประกอบด้วย Shared key ที่จะถูกนำไปเข้ารหัสกับคีย์สาธารณะ (public key) ของฝั่งผู้รับ โดยคีย์ดังกล่าวจะถูกรวมไว้ใน Reference List ที่มีการกำหนด DataReference เพื่อที่สามารถย้อนกลับไปยังข้อมูลที่ถูกเข้ารหัสด้วยคีย์ดังกล่าว โครงสร้างของเมตาโมเดลของการรักษาความลับแสดงในภาพที่ 5



ภาพที่ 5 *เมตาโมเดลของ PSM ในส่วนการรักษาความลับ*

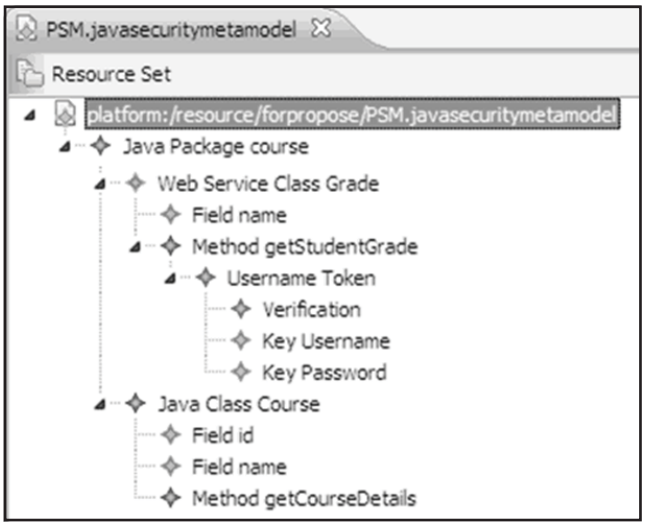
5. ตัวอย่างแบบจำลอง PIM และ PSM

ในหัวข้อนี้แสดงตัวอย่างการเพิ่มความปลอดภัยบนแบบจำลอง PIM และแสดงแบบจำลอง PSM ที่ได้จากการแปลงแบบจำลอง โดยเริ่มต้นจากต้องการให้คลาส Grade ซึ่งมีเมท็อด 2 เมท็อดเป็นคลาสเว็บเซอร์วิส ซึ่งการเรียกใช้งานเมท็อด getStudentGrade จะต้องทำการตรวจสอบสิทธิ์ก่อน ดังนั้นเมื่อสร้างแบบจำลอง PIM ต้องกำหนดคลาส Grade 2 ส่วน คือกำหนดให้คลาส Grade เป็นซับคลาสของ Service ที่ได้กำหนดไว้ในเมตาโมเดลของ PIM และเลือก Operation เป็น Authentication ซึ่งแบบจำลอง PIM แสดงในภาพที่ 6



ภาพที่ 6 *แบบจำลอง PIM ที่เพิ่มคุณสมบัติ Authentication*

เมื่อสร้างแบบจำลอง PIM เรียบร้อยแล้ว จะนำแบบจำลองเข้าสู่กระบวนการแปลงแบบจำลอง โดยใช้กฎการแปลงแบบจำลองที่มีการกำหนดไว้ก่อนหน้า ซึ่งแบบจำลองปลายทางที่ได้จะมีโครงสร้างดังภาพที่ 7 คลาส Grade ก็จะถูกสร้างขึ้นเป็นคลาสแบบ Web Service ที่มีการกำหนดส่วนเพิ่มเติมในการสร้างเป็นเว็บเซอร์วิส และเมท็อด getStudentGrade ก็จะมีการสร้างอิลิเมนต์ของ UsernameToken พร้อมทั้งโครงสร้างการเก็บ UserName และ Password ซึ่ง PSM ที่ได้นี้จะนำไปใช้เป็นแบบจำลองต้นทางในการแปลงแบบจำลองให้กลายเป็น จาวาโค้ด และไฟล์คอนฟิคกูเรชันของ WS-Security ต่อไป



ภาพที่ 7 *ตัวอย่างแบบจำลอง PSM (อยู่ในรูปแบบของ Ecore)*

6. อภิปราย และสรุปผล

งานวิจัยนี้ได้กำหนดส่วนเพิ่มขยายเมตาโมเดลของ UML เพื่อรองรับความปลอดภัยในเรื่องการพิสูจน์ตัวจริง การรักษาความลับ และการรักษาความสมบูรณ์ รวมถึงได้เสนอแบบจำลอง PSM ที่รองรับการรักษาความปลอดภัยบนเว็บเซอร์วิส เพื่อให้นักวิเคราะห์และออกแบบระบบในกระบวนการพัฒนาซอฟต์แวร์สามารถนำแบบจำลองดังกล่าวไปใช้งานได้ นอกจากนี้ยังเป็นการช่วยให้นักออกแบบที่ไม่มีความเชี่ยวชาญด้านความปลอดภัยสามารถสร้างแบบจำลองสำหรับแอพพลิเคชันเว็บเซอร์วิสที่มีคุณสมบัติด้านความปลอดภัย (WS-Security) ได้อย่างถูกต้อง ครบถ้วนมากขึ้น แต่งานวิจัยนี้ยังมีข้อจำกัดในเรื่องการนำเมตาโมเดลของแบบจำลอง PIM และกฎการแปลงแบบจำลองไปใช้งาน โดยจะสามารถนำไปใช้ได้ในเครื่องมือที่สนับสนุน MDA แต่ไม่สามารถนำไปใช้งานได้เครื่องมือ UML ทั่วไป

7. เอกสารอ้างอิง

[1] P.T. Devanbu and S.Stubbleine, “Software Engineering for Security: a Roadmap,” *ICSE* 2000.

[2] J. Jurjens, “UMLsec:Extending UML for Secure Systems Development,” *Proceedings of the 5th Conference on the Unified Modeling Language*, Dresden, Germany, 2002.

[3] T. Lodderstedt, D. Basin and J. Dorser, “Secure UML: A UML-Based Modeling Language for Model-Driven Security,” *Proceedings of the 5th Conference on the Unified Modeling Language*, Dresden, Germany, 2002.

[4] Y. Nakamura, M. Tatsubori, T. Imamura and K. Ono, “Model-Driven Security Based on a Web Services Security Architecture,” *SCC’* 05, 2005.

[5] V. Cortellessa, A. D. Marco, and P. Inverardi, “Integrating Performance and Reliability Analysis in a Non-Functional MDA Framework,” *LNCS* 4422, pp. 57-71, 2007.

[6] J.Miller and J. Mukerji, “MDA Guide Version 1.0” *OMG*, 2003.

[7] Available online at http://csrc.nist.gov.

[8] A. Nadalin, C. Kaler, P. Hallam-Baker and R. Monzillo, “Web Services Security: SOAP Message Security 1.0,” 2004.

[9] F. Jouault, I. Kurtev, “Transforming Models with ATL,” LNCS 3844, pp. 128-138, 2006.

[10] Available online at http://www.omg.org/spec/QVT/1.0.