

แนวคิดและมุมมองในการวิเคราะห์ยูสเคสโมเดล

Guideline and Perspective to Analysis Use Case Model

บุญประเสริฐ สุรภัทรตันสกุล*

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Boonprasert Surakratanasakul*

Faculty of Information Technology

King Mongkut's Institute of Technology Ladkrabang

บทคัดย่อ

ยูสเคสโมเดล (Use Case Model) หรือแบบจำลองกรณีใช้งานระบบเป็นโมเดลหนึ่งในชุดโมเดลของยูเอ็มแอลเพื่อใช้ในการพัฒนาระบบซอฟต์แวร์ เป็นโมเดลเริ่มต้นของการเข้าสู่กระบวนการวิเคราะห์และออกแบบระบบทางด้านวิศวกรรมซอฟต์แวร์มีบทบาทเป็นโมเดลเชื่อมโยงระหว่างระยะของความต้องการทางด้านซอฟต์แวร์และระยะการพัฒนาด้านอื่นๆ โดยทั่วไปแล้วยูสเคสโมเดลประกอบด้วยสองส่วนคือ ส่วนของแผนภาพและส่วนของคำอธิบาย ส่วนแผนภาพให้มุมมองในภาพรวมของระบบส่วนคำอธิบายเป็นข้อมูลสำคัญสำหรับใช้กระบวนการพัฒนา ถึงแม้ว่ายูสเคสโมเดลเป็นโมเดลที่ไม่ซับซ้อนและไม่ลงรายละเอียดเชิงเทคนิคซึ่งเหมาะสำหรับผู้อ่านทุกๆ ไป แต่ยังพบข้อผิดพลาดบ่อยครั้งในเรื่องของแนวคิดและมุมมองของนักพัฒนาระบบ ด้วยเหตุนี้ บทความนี้ จึงนำเสนอทบทวนแนวคิดและมุมมองในการวิเคราะห์ยูสเคสโมเดล ซึ่งประกอบด้วย ภาพรวมของยูสเคสโมเดล ประโยชน์ที่พึงได้รับจากยูสเคสโมเดล องค์ประกอบในยูสเคสโมเดลและแนวคิดการวิเคราะห์ในแต่ละองค์ประกอบ และส่วนสุดท้ายข้อพึงระวังในการวิเคราะห์ยูสเคสโมเดล

คำสำคัญ : ยูสเคสโมเดล ยูสเคส ผู้กระทำการ บทบรรยายยูสเคส

* ผู้ประสานงานหลัก (Corresponding Author)
e-mail: boonprasert@it.kmitl.ac.th

Abstract

Use case model is one of modeling suite of UML for software development. It acts an initial model for analysis and design in software engineering discipline and correlates software requirement phase and others phases. Generally, Use case model consists of two parts: use case diagram and their specification. Use case diagram shows an overview behavior of system. Use case specification describes specific information for software development. Though Use case model is a simple model and not deep in technical for reader, unfortunately many developers have mistake in concept and perspective. For these reasons, this article has proposes to review guideline and perspective for Use case model. It consists of: Overview of Use case model, Use case model advantage, composition of Use case model and guideline to analysis model's things, and finally hints and tips to analysis Use case model.

Keywords : Use case model, Use case, Actor, Scenario

บทนำ

ในกระบวนการทางวิศวกรรมซอฟต์แวร์โดยใช้ภาษายูเอ็มแอล (UML-Unified Modeling Language) เป็นชุดเครื่องมือในการพัฒนาระบบ จะประกอบด้วยหลายๆ โมเดลซึ่งให้มุมมองและสนับสนุนการพัฒนาระบบในบทบาทที่แตกต่างกันตามช่วงระยะของการพัฒนา (Booch, Rumbaugh & Jacobson, 1998) “ยูสเคสโมเดล (Use Case Model) หรือแบบจำลองกรณีบ่งชี้งานภายในระบบ” เป็นโมเดลหนึ่งที่มีความสำคัญและเป็นโมเดลขับเคลื่อนเริ่มต้นของการพัฒนาตามกระบวนการทางวิศวกรรมซอฟต์แวร์แบบยูนิฟายโปรเซส (Unified Process) และรูปแบบการพัฒนาโดยใช้ยูสเคสกำกับ (Use Case Driven Development) เป็นโมเดลที่ใช้แสดงมุมมองภาพรวมในพฤติกรรมของระบบเบื้องต้นและแสดงแผนภาพความต้องการทางด้านซอฟต์แวร์ที่เป็นกรณีใช้งานหรือที่รู้จักกันว่าเป็นฟังก์ชันของระบบ (Functional Software Requirements) ซึ่งจะถูกกำหนดอยู่ในรูปสัญลักษณ์ของยูสเคสนอกจากนี้ยังเป็นโมเดลที่ช่วยวางแผนวิธีการทำงานที่จะเกิดขึ้นในกระบวนการทางวิศวกรรมซอฟต์แวร์ขั้นตอนถัดไปและตลอดวงจรของการพัฒนาระบบรวมถึงกระบวนการที่เกี่ยวข้อง เช่น กระบวนการวิเคราะห์และออกแบบ กระบวนการออกแบบการทดสอบ และกระบวนการจัดทำเอกสารประกอบ (Sommerville, 2010)

โดยทั่วไปแล้ว ยูสเคสโมเดลประกอบด้วยสองส่วนหลัก คือ ส่วนของแผนภาพ (Use Case Diagram) และส่วนของคำอธิบาย (Description) ในส่วนของแผนภาพยูสเคสเป็นแผนภาพที่สนใจในพฤติกรรมของระบบที่กำลังจะพัฒนาเพื่อศึกษาว่าระบบที่จะพัฒนาขึ้นมานั้นมีพฤติกรรมอะไรและมีขอบเขต

การทำงานเท่าใดรวมถึงสิ่งแวดล้อมหรือบริบทที่ส่งผลต่อการทำงานของระบบ ซึ่งพฤติกรรมดังกล่าวจะถูกกำหนดอยู่ในรูปแบบของฟังก์ชันการทำงานของระบบ นอกจากนี้แผนภาพยูสเคสยังแสดงถึงวิธีการที่ระบบติดต่อกับสิ่งแวดล้อมโดยแสดงว่าบทบาทของใครหรือสิ่งใดมีการกระทำกับระบบ และระบบต้องตอบสนองต่อเหตุการณ์หรือสิ่งแวดล้อมภายนอกที่มากระตุ้นนั้นอย่างไร ในส่วนของคำอธิบาย จะอยู่ในรูปแบบของเอกสารประกอบคำบรรยายซึ่งสามารถแบ่งได้เป็น 2 ระดับ คือ คำอธิบายในภาพรวม (Use Case Model Survey) ซึ่งจะแสดงรายละเอียดและนิยามต่างๆ ขององค์ประกอบที่ปรากฏในภาพรวมของโมเดล และคำอธิบายตามยูสเคส (Use Case Description) ซึ่งจะแสดงรายละเอียดขั้นตอนในการทำงานตามแต่ละยูสเคส เพื่อให้บรรลุเป้าหมายที่กำหนดตามแต่ละยูสเคสนั้นๆ (Leffingwell & Widrig, 2004)

เนื่องจากยูสเคสโมเดลเป็นโมเดลที่ประกอบด้วยความต้องการของผู้ที่มีส่วนได้ส่วนเสียกับระบบ (Stakeholder) ที่ต้องการให้นักพัฒนาระบบเล็งเห็นถึงความต้องการทางด้านฟังก์ชันของระบบใหม่ รวมถึงพฤติกรรมการทำงานและการปฏิสัมพันธ์ที่พึงมีในระบบใหม่ ดังนั้นยูสเคสโมเดลจึงเป็นโมเดลที่ถูกออกแบบมาให้กับผู้อ่านในหลายๆ บทบาท ทั้งที่มาจากฝั่งทางผู้ว่าจ้าง (Client Team) และฝั่งทางผู้พัฒนาระบบ (Developer Team) ทำให้ยูสเคสโมเดลจึงเป็นโมเดลที่แสดงมุมมองในภาพกว้างทั้งระบบและไม่ลงรายละเอียดเชิงลึกด้านเทคนิค เพื่อเป้าหมายที่จะอธิบายถึงปัญหาหรือความต้องการที่ทุกๆ ฝ่ายสามารถเข้าใจตรงกันได้ และแสดงถึงจินตภาพของระบบที่จะเกิดขึ้นในอนาคตเมื่อกระบวนการพัฒนาระบบเสร็จสิ้นสมบูรณ์

ในส่วนของทางฝั่งผู้ว่าจ้าง ผู้ใช้งาน และผู้ที่มีส่วนเกี่ยวข้องกับการให้ความต้องการ จะอ่านยูสเคสที่ผู้พัฒนาระบบจัดทำขึ้น เพื่อตรวจทานความต้องการและรับรองความถูกต้องของความต้องการทางด้านฟังก์ชันในระบบใหม่ที่ได้ให้ไว้กับทีมผู้พัฒนาระบบ ในส่วนของทางฝั่งผู้พัฒนาระบบ จะใช้ยูสเคสโมเดลเป็นกรอบและแนวทางในการพัฒนาระบบซอฟต์แวร์ให้เข้าใจอย่างถูกต้องตรงกับทางฝั่งของผู้ว่าจ้างต้องการ นอกจากนี้ยังใช้คำบรรยายยูสเคสโมเดลเป็นตัวขับเคลื่อนการพัฒนาเช่น สิ่งที่คุณออกแบบระบบต้องออกแบบคืออะไร อนุญาตให้ผู้ใช้งานทำงานในขั้นตอนใด สิ่งที่คุณทดสอบต้องทดสอบคืออะไรและจะทำการทดสอบเพื่อแสดงผลสัมฤทธิ์ตามเป้าหมายของแต่ละยูสเคสนั้นอย่างไร เอกสารประกอบการใช้งานของผู้ใช้ระบบเป็นอย่างไร ขั้นตอนวิธีทำงานสามารถช่วยให้บรรลุตรงตามเป้าหมายของความต้องการที่ระบุไว้หรือไม่เป็นต้น ตัวอย่างของผู้อ่านและผู้จัดทำยูสเคสโมเดลที่มาจากทางฝั่งของผู้พัฒนาระบบ เช่น นักวิเคราะห์ระบบ (Analysts) นักออกแบบระบบ (Designer) โปรแกรมเมอร์หรือผู้ที่นำไปกระทำการ (Implementer) นักทดสอบระบบ (Tester) ผู้กำหนดความต้องการ (Requirements Specifier) ผู้จัดบันทึก (Writer) และผู้จัดการโครงการพัฒนาระบบ (Project Manager) เป็นต้น (Kotonya & Sommerville, 1998)

ประโยชน์ของยูสเคสโมเดล

การพัฒนายูสเคสโมเดลเป็นกระบวนการหนึ่งที่ทำกรรวบรวมและจัดการความต้องการเชิงซอฟต์แวร์ที่ได้มาจากผู้ที่มีส่วนเกี่ยวข้องกับระบบให้เป็นรูปธรรมโดยวิธีการนำความต้องการที่เก็บเกี่ยวได้มารวบรวมไว้อยู่ภายใต้ขอบเขตและมุมมองเดียวกันโดยใช้ยูสเคสเป็นสัญลักษณ์ภายใต้แผนภาพที่มีกฎเกณฑ์มาตรฐานและรวบรวมขั้นตอนการทำงานที่ตรงกับความต้องการให้บรรลุเป้าหมายตามฟังก์ชันที่กำหนดตามยูสเคสนั้นๆ ซึ่งประโยชน์ของยูสเคสโมเดลมีดังนี้

1. แผนภาพยูสเคสแสดงขอบเขตของระบบและการติดต่อกับสิ่งแวดล้อม หรือแสดงบริบทของระบบ
2. ยูสเคสแสดงสิ่งที่ระบบต้องพึงปฏิบัติทางด้านฟังก์ชัน เพื่อให้บรรลุเป้าหมายของผู้ที่ให้ความต้องการไว้ในระบบใหม่
3. แสดงบริบทของความต้องการทางด้านฟังก์ชันในระบบว่าความต้องการที่เก็บเกี่ยวมาได้นั้นสามารถเกิดขึ้นได้จริงในภาคปฏิบัติเมื่อระบบพัฒนาเสร็จสิ้นสมบูรณ์
4. เร่งกระบวนการเข้าใจในระบบด้วยแผนภาพที่เข้าใจง่ายและไม่ซับซ้อน เนื่องจากแผนภาพยูสเคสเป็นแผนภาพที่ประกอบด้วยสัญลักษณ์ที่ชัดเจน มีองค์ประกอบไม่ซับซ้อนและที่สำคัญคือ ไม่ลงรายละเอียดเชิงลึกทางด้านเทคนิคทั้งนี้ช่วยให้ผู้อ่านทุกๆ ฝ่ายสามารถทำความเข้าใจตรงกันและเห็นพ้องต้องกันได้เร็วขึ้น
5. การนำยูสเคสมาเป็นสัญลักษณ์ ช่วยลดการตีความที่ผิดเพี้ยนหรือการสูญหายของข้อมูลได้ เนื่องจากเป็นสัญลักษณ์ที่นิยามโดยมาตรฐานตามภาษายูเอ็มแอลและคำศัพท์ที่ปรากฏอยู่ในยูสเคสโมเดลเป็นคำศัพท์ในโดเมนของผู้ใช้ ที่สะท้อนถึงความต้องการที่ผู้ใช้คิดถึงระบบในอนาคตอย่างไรและระบบควรทำอะไร
6. มีส่วนช่วยเป็นตัวกำหนดข้อตกลงระหว่างผู้ว่าจ้างและผู้พัฒนาระบบเนื่องจากยูสเคสโมเดลเป็นโมเดลที่แสดงถึงความต้องการเชิงซอฟต์แวร์ของผู้ที่มีส่วนเกี่ยวข้อง ดังนั้นผู้พัฒนาระบบสามารถนำยูสเคสโมเดลมาใช้เป็นเอกสารประกอบหรืออ้างอิงในการกำหนดข้อตกลงและการวัดผลการบรรลุเป้าหมายของการพัฒนาระบบได้ แต่อย่างไรก็ตามยูสเคสโมเดลเป็นเพียงเอกสารประกอบหนึ่งเท่านั้น ซึ่งการกำหนดข้อตกลงร่วม จำเป็นต้องมีเอกสารอื่นประกอบด้วยเพื่อให้เกิดความรัดกุมและเข้าใจได้อย่างละเอียดครบถ้วน
7. ความต้องการของระบบที่พัฒนา จะถูกแทนที่ด้วยคำบรรยายที่ประกอบตามแต่ละยูสเคส โดยแต่ละความต้องการจะถูกปรับเปลี่ยนเป็นส่วนหนึ่งในลำดับการกระทำแบบเป็นขั้นตอนที่สามารถดำเนินการให้กระบวนการสำเร็จบรรลุเป้าหมายของที่ต้องการ
8. สามารถเป็นตัวบ่งชี้ว่า ทำไมผู้ว่าจ้างถึงต้องการระบบใหม่ในอนาคตแสดงให้เห็นถึงความสำคัญของระบบใหม่ที่จะพัฒนาขึ้นเนื่องจากยูสเคสโมเดลเป็นโมเดลของระบบใหม่ที่บรรจุความต้องการทางด้านฟังก์ชันทั้งหมดไว้ในโมเดล

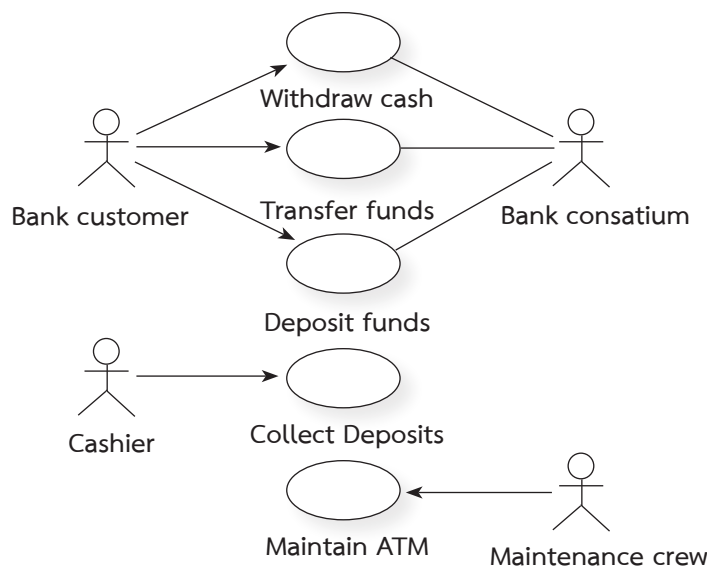
9. ช่วยในการตรวจสอบเบื้องต้น ว่าพบปัญหาอะไรหรือไม่ในกระบวนการจัดการความต้องการ (Requirements process) เนื่องจากยูสเคสโมเดลเป็นบทพิจารณาเบื้องต้นของการแก้ไขปัญหาและระบุความต้องการที่เกิดขึ้นในระบบใหม่

10. ช่วยในการตัดสินใจว่า สิ่งใดควรจะพัฒนาก่อนและมีความจำเป็น สิ่งใดเป็นสิ่งที่สามารถละเลยได้หรือเกินความจำเป็นหรือเกินขอบเขตในการพัฒนาระบบ โดยปกติแล้ว ความต้องการที่ผู้มีส่วนเกี่ยวข้องกำหนดให้ทีมพัฒนาทำการพัฒนาระบบนั้น มักมีจำนวนมากและไม่เพียงพอที่จะพัฒนาได้ทั้งหมดภายในระยะเวลาที่มีอยู่ การที่ยูสเคสโมเดลให้มุมมองภาพรวมทั้งหมดของความต้องการ ทำให้ผู้พัฒนาสามารถทำการเปรียบเทียบหรือลำดับความสำคัญของงานได้ง่ายขึ้น

ความรู้เบื้องต้นเกี่ยวกับยูสเคสโมเดล

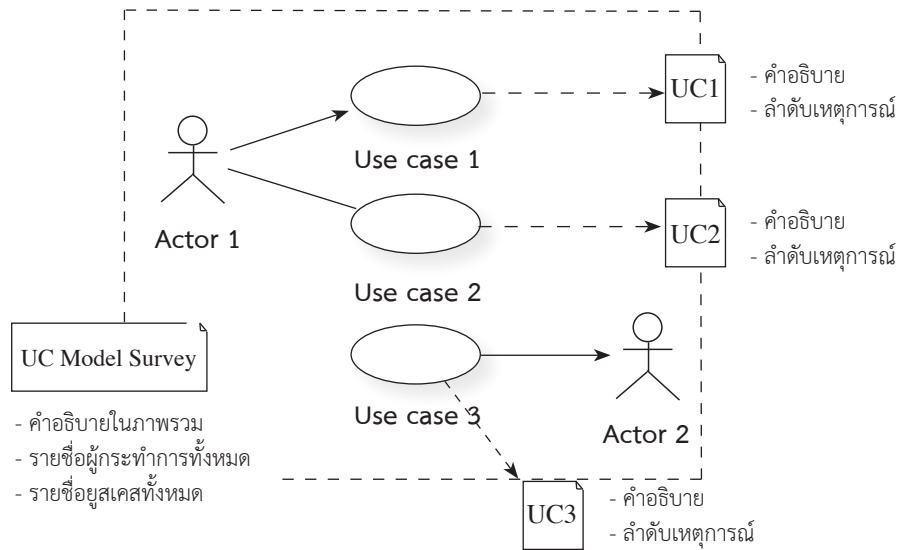
ยูสเคสโมเดลเป็นเครื่องมือที่ช่วยในการจับความต้องการที่เป็นนามธรรมให้เป็นรูปธรรมโดยใช้สัญลักษณ์ยูสเคสประกอบกับการบรรยายยูสเคสโมเดลเป็นโมเดลที่สามารถประยุกต์ได้ตามบริบทที่จะนำไปใช้ ซึ่งโมเดลที่เห็นเด่นชัดและเป็นที่ยอมรับ คือ ยูสเคสโมเดลเชิงธุรกิจ (Business Use Case Model) และ ยูสเคสโมเดลระบบ (Use Case System Model) ซึ่งทั้งสองโมเดลต่างก็มีความสัมพันธ์ในรูปแบบการพัฒนาและบริบทในการปรับเปลี่ยนมุมมอง บางครั้งนักวิเคราะห์ระบบอาจเริ่มต้นยูสเคสโมเดลโดยการวิเคราะห์เชิงธุรกิจก่อน แล้วจึงนำไปสู่การวิเคราะห์และออกแบบยูสเคสโมเดลของระบบ ซึ่งทั้งสองโมเดลต่างก็มีองค์ประกอบที่คล้ายคลึงกัน แต่ปรับเปลี่ยนตามบริบทและสิ่งแวดล้อม ซึ่งองค์ประกอบที่สำคัญในยูสเคสโมเดลประกอบด้วย ยูสเคส บทบาทของผู้กระทำการ การติดต่อสื่อสาร และบทบรรยายยูสเคส

เมื่อมีการจัดทำยูสเคสโมเดล สิ่งที่ยูสเคสโมเดลควรต้องอธิบายได้มีดังนี้ (1) ระบบ สิ่งแวดล้อมและความสัมพันธ์กับยูสเคส (2) สิ่งที่อยู่ภายนอกหรือสิ่งแวดล้อม กระทำกับระบบอย่างไร และ (3) อธิบายพฤติกรรมของระบบ ตัวอย่างยูสเคสโมเดลในภาพที่ 1 แสดงให้ผู้อ่านสามารถเข้าใจในภาพรวมของระบบตู้เอทีเอ็ม (IBM, 2012)



ภาพที่ 1 ยูสเคสโมเดลของระบบเอทีเอ็ม
(ที่มา: Rational Software, 2005)

คนส่วนใหญ่มักเข้าใจผิดว่า ยูสเคสโมเดลประกอบด้วยแผนภาพและลูกศรเท่านั้น แต่ความเป็นจริงแล้ว ยูสเคสโมเดลประกอบด้วยแผนภาพพร้อมข้อความบรรยายโดยที่แผนภาพเป็นเพียงส่วนหนึ่งเท่านั้น ซึ่ง 75% ของความต้องการทางด้านซอฟต์แวร์ในระบบใหม่ มีที่มาจากการเขียนคำบรรยายในยูสเคสในส่วนองภาษาที่ใช้ นักพัฒนาระบบสามารถใช้ภาษาไทยหรือภาษาอังกฤษก็ได้ แต่จากประสบการณ์ที่ผ่านมา การใช้ภาษาอังกฤษจะอำนวยความสะดวกในการพัฒนาระบบมากกว่า เนื่องจากศัพท์ที่ใช้เป็นศัพท์ที่อยู่ในเฉพาะโดเมนอยู่แล้ว ซึ่งผู้ที่เกี่ยวข้องสามารถเข้าใจตรงกันได้ นอกจากนี้ศัพท์ต่างๆ ยังมีส่วนช่วยในเรื่องการพัฒนาระบบในขั้นตอนถัดไป เช่น การเขียนบทวิเคราะห์และออกแบบ การกำหนดตัวแปรในการเขียนโปรแกรม เป็นต้น แต่อย่างไรก็ตาม เพื่อความสอดคล้องและเห็นพ้องต้องกัน ควรมีการอ้างอิงหรือจัดทำรูปแบบพจนานุกรมศัพท์ของระบบ (System Dictionary) เพื่อใช้ประกอบการอ้างอิงตลอดโครงการพัฒนาภาพที่ 2 แสดงแผนภาพยูสเคสและเอกสารที่ใช้ประกอบคำบรรยายยูสเคสในการพัฒนาระบบแบบยูนิฟายโปรเซส



ภาพที่ 2 แผนภาพยูสเคสและเอกสารประกอบคำบรรยาย

ขั้นตอนการพัฒนาユスเคสโมเดล

โดยทั่วไปแล้วขั้นตอนการพัฒนาユスเคสโมเดล ไม่มีกำหนดขั้นตอนพัฒนาที่แน่นอนขึ้นอยู่กับเทคนิคส่วนตัวและประสบการณ์ของผู้พัฒนาเป็นหลัก อย่างไรก็ตาม หากผู้อ่านเป็นผู้พัฒนาระบบใหม่อาจเริ่มต้นดังนี้

1. หาบทบาทผู้กระทำการและユスเคส
 - 1.1 กำหนดและอธิบายบทบาทผู้กระทำการ
 - 1.2 อธิบายユスเคสโมเดลในภาพรวม ระดับในการวิเคราะห์ ข้อมูลภาพรวม เช่น จำนวนユスเคส แพคเกจ (Use Case Package) จำนวนユスเคสที่ค้นพบในเบื้องต้น
2. เขียนユスเคส
 - 2.1 ร่างユスเคสที่ค้นหาได้ทั้งหมด
 - 2.2 จัดลำดับและบรรยายรายละเอียดในแต่ละユスเคส

ขั้นตอนแรก เริ่มจากการค้นหาผู้กระทำการซึ่งประโยชน์จากการหาผู้กระทำการก่อน จะช่วยในเรื่องการกำหนดขอบเขตของระบบและมุมมองของผู้พัฒนา หลังจากนั้นจึงทำการกำหนดユスเคสโดยการเก็บเกี่ยวความต้องการจากผู้ที่มีส่วนได้ส่วนเสียกับระบบ Surakratanasakul & Chattong (2014) แล้วจึงเริ่มทำการร่างแผนภาพอย่างคร่าวๆ แล้วค่อยเข้าสู่ขั้นตอนการลงคำอธิบายรายละเอียดของแต่ละส่วนที่พบ โดยเริ่มจากภาพรวมแล้วลงรายละเอียดจนกระทั่งถึงการสร้างลำดับเหตุการณ์ทำงานตามแต่ละ

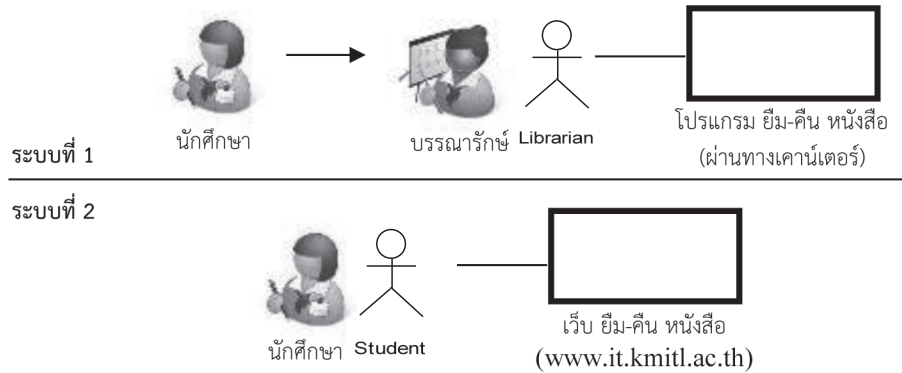
ยูสเคสซึ่งลำดับเหตุการณ์ในยูสเคสนั้นมีความสำคัญมาก เพราะเป็นข้อมูลสำคัญสำหรับการวิเคราะห์และออกแบบ โดยเฉพาะอย่างยิ่งการพัฒนาโดยใช้แนวคิดเชิงวัตถุ (Object-oriented Concept) ส่วนภาคปรับปรุงโครงสร้างหรือส่วนต่อขยายเพิ่มเติมของยูสเคส (Use Case Extension) เช่น การควมรวมขั้นตอน (Included) และการขยายทางเลือก (Extended) จะกระทำก็ต่อเมื่อมีรายละเอียดครบถ้วนและตรวจทานว่าเรียบร้อยสมบูรณ์ดีแล้วเท่านั้นโดยมากมักทำในช่วงเก็บเกี่ยวรายละเอียดภายหลังตรวจสอบการเปลี่ยนแปลงเสร็จสิ้นสมบูรณ์ ซึ่งทุกยูสเคสจะถูกทำการหยุดเปลี่ยนแปลง (Freeze) ก่อนเข้าสู่ระยะกระบวนการวิเคราะห์และออกแบบระบบ (Analysis and Design) (Borland, 2012)

ผู้กระทำการหรือแอกเตอร์ (Actor)

ผู้กระทำการหรือแอกเตอร์เป็นสัญลักษณ์แทนบทบาทของสิ่งที่อยู่ภายนอกระบบหรือสิ่งแวดล้อมที่มีบทบาทในการกระทำการหรือแลกเปลี่ยนข้อมูลกับระบบ อาจเป็นผู้กระตุ้นหรือเป็นผู้รับผลลัพธ์จากการทำงานตามขั้นตอนในยูสเคสหรือมีทั้งสองบทบาทคือ เป็นผู้ให้ข้อมูล ผู้รับข้อมูล หรือทั้งให้และรับข้อมูล

ผู้กระทำการเป็นตัวแทนบทบาท ดังนั้นการตั้งชื่อของผู้กระทำการ ควรต้องแสดงบทบาทอย่างชัดเจนให้มากที่สุดเท่าที่จะเป็นไปได้ เนื่องจากความผิดพลาดในการตั้งชื่อผู้กระทำการอาจสามารถทำให้ผู้อ่านเกิดความสับสนได้ ชื่อที่ใช้ควรเป็นคำศัพท์ที่อยู่ในโดเมนที่ผู้อ่านสามารถเข้าใจ เช่น ในระบบเรียกอย่างไร นักพัฒนาก็ควรตั้งชื่อเรียกตามนั้นซึ่งหากเป็นคำเฉพาะเจาะจง สามารถใช้การอ้างอิงพจนานุกรมศัพท์ของระบบเพื่อความสอดคล้องเข้าใจตรงกัน

การกำหนดผู้กระทำการให้กับระบบ ควรตั้งคำถามเสมอว่า ใครเป็นผู้กระทำที่แท้จริง เนื่องจากผู้กระทำการคือ บทบาทของผู้ที่ติดต่อกับระบบโดยตรงยกตัวอย่างเช่น ระบบยืม-คืนหนังสือในห้องสมุด หากนักศึกษายืม-คืนหนังสือโดยผ่านบรรณารักษ์ซึ่งเป็นผู้ใช้งานซอฟต์แวร์ของระบบ เราจะถือว่าบรรณารักษ์เป็นผู้กระทำการ โดยที่นักศึกษาเป็นเพียงผู้ที่มีส่วนได้ส่วนเสียกับระบบ (Stakeholder) แต่ในขณะเดียวกันหากในอนาคตห้องสมุดมีการพัฒนาระบบให้สามารถยืม-คืนโดยผ่านเว็บไซต์ได้โดยตรง นักศึกษาก็จะมีบทบาทเป็นผู้กระทำการแทน เพราะเป็นผู้ที่ติดต่อกับระบบโดยตรง



ภาพที่ 3 ผู้กระทำการที่แท้จริงในระบบที่ 1 คือ เจ้าหน้าที่บรรณารักษ์ ซึ่งเป็นผู้ใช้งานโปรแกรม ส่วนในระบบที่ 2 ผู้กระทำการ คือ นักศึกษา ซึ่งใช้งานระบบผ่านเว็บไซต์

นักพัฒนาระบบควรพยายามหลีกเลี่ยงการใช้ชื่อผู้กระทำว่า “User (ผู้ใช้งาน)” โดยเฉพาะอย่างยิ่งระบบที่มีผู้ใช้งานที่หลากหลาย เนื่องจากคำว่า “User (ผู้ใช้งาน)” นั้น มีบทบาทที่กว้างเกินไปและโดยมากจะนึกถึงตัวบุคคลหรือสิ่งมีชีวิต แต่ในความเป็นจริงแล้วผู้ใช้งานเป็นสับเซตหนึ่งในบทบาทของผู้กระทำเท่านั้น และผู้กระทำอาจเป็นได้ทั้งสิ่งมีชีวิตหรือไม่มีชีวิตก็ได้ อีกทั้ง “User (ผู้ใช้งาน)” ยังแสดงบทบาทที่ไม่เฉพาะชัดเจน ทำให้เข้าใจยากในระบบที่มีขนาดใหญ่ ทั้งนี้หากสังเกตความสัมพันธ์ระหว่างยูสเคสและผู้กระทำไม่ได้ ซึ่งหากมีมากเกินไปก็ความเป็นไปได้ว่ามีบทบาทมากกว่าหนึ่งบทบาทในผู้กระทำนั้น ควรต้องวิเคราะห์หาบทบาทที่แท้จริงหรือกำหนดนิยามของผู้กระทำให้ลึกซึ้งกว่าเดิม

ทุกครั้งที่ทำการกำหนดผู้กระทำสิ่งที่ควรตระหนัก คือ ต้องไม่อ้างอิงตัวบุคคล เนื่องจากในหลายสถานการณ์ บุคคลหนึ่งสามารถแสดงได้หลายบทบาทตามสถานการณ์ ตัวอย่างเช่น ที่มหาวิทยาลัย นาย ก. มีบทบาทเป็นอาจารย์ แต่ในขณะเดียวกัน นาย ก. สามารถลงเรียนที่มหาวิทยาลัยได้ ซึ่งเมื่อนาย ก. ล็อกอินในระบบลงทะเบียน นาย ก. จะแสดงบทบาทเป็นนักเรียนซึ่งเป็นบทบาทที่แตกต่างกันของผู้กระทำ การเติมชื่อควรระวังถัดมา ไม่ควรตั้งชื่อผู้กระทำตามตำแหน่ง เพราะตำแหน่งหนึ่งอาจมีหน้าที่ได้มากกว่าหนึ่งบทบาทในระบบ เช่น ตำแหน่ง “Administrator (ผู้ควบคุมระบบ)” เนื่องจากในระบบที่มีขนาดใหญ่ บทบาทของผู้ควบคุมระบบอาจมีมากกว่าหนึ่งบทบาท เช่น ทำหน้าที่กู้คืนข้อมูลในสถานการณ์ฉุกเฉิน ทำหน้าที่ตรวจสอบสิ่งผิดปกติหรือการมอนิเตอร์ ซึ่งในกรณีนี้ การตั้งชื่อผู้กระทำด้วยตำแหน่ง “Administrator (ผู้ควบคุมระบบ)” อาจไม่ใช่รูปแบบที่ดี

หลีกเลี่ยงสิ่งที่หมายถึง อินสแตนซ์ (Instance) จากผู้กระทำเดียวกันในแต่ละยูสเคสตัวอย่าง เช่น นาย ข. และ นาย ค. เป็นนักศึกษาที่ใช้ระบบลงทะเบียน เมื่อพวกเขาใช้ระบบ แต่ละคนจะเป็นอยู่ในบทบาทของอินสแตนซ์ที่มาจากตัวกระทำเดียวกัน คือ “Student (นักเรียน)”

อย่าลืมว่าผู้กระทำการบางครั้งอาจอยู่ในรูปแบบที่เป็นนามธรรม ในบางระบบจำเป็นต้องมีสิ่งกระตุ้น (Trigger) ในตอนเริ่มต้น กระบวนการของสิ่งกระตุ้นนี้อาจถูกกำหนดด้วยเวลา เช่น “At the end of the day (เมื่อสิ้นสุดวันทำการทุกวัน)” หรือ “At the end of month (ทุกวันสิ้นเดือน)” ผู้พัฒนาระบบอาจสร้างผู้กระทำการจากข้อมูลเหล่านี้ได้เช่น “Scheduler (ตัวกำหนดเวลา)” หรือ “Time (กำหนดการ)” เป็นต้น

นักพัฒนาระบบควรตระหนักถึงชื่อของผู้กระทำการในอนาคตเมื่อระบบมีการขยายหรือเพิ่มเติม ซึ่งอาจจะทำให้มีบทบาทผู้กระทำการเพิ่มเข้ามาหรืออาจมีการเน้นบทบาทของผู้กระทำการเดิมที่มีอยู่ให้มากขึ้นเมื่อมีการเชื่อมต่อกับระบบอื่น

เมื่อทำการกำหนดผู้กระทำการเสร็จสิ้นแล้ว ควรทำการตรวจสอบรายละเอียดด้วยแผนผังว่ามีผู้กระทำการใดที่ไม่มีความสัมพันธ์กับยูสเคสหรือไม่ เพื่อป้องกันการตกหล่นหรือกำหนดไว้เกิน และควรตรวจสอบว่าผู้กระทำการใดมีความสัมพันธ์กับยูสเคสแบบเดียวกันเพื่อป้องกันความซ้ำซ้อนหรือกำกวมโดยรายละเอียดต่างๆ ของผู้กระทำการ เมื่อมีความสมบูรณ์แล้ว จะถูกนิยามไว้ในเอกสารประกอบการบรรยายของยูสเคสโมเดลในภาพรวม

นักพัฒนาระบบอาจใช้คำถามเหล่านี้ในการตรวจสอบผู้กระทำการอย่างคร่าวๆ ได้ เช่น “ใครหรือสิ่งใดที่ใช้กำหนดระบบ” “ใครหรือสิ่งใดที่ได้รับข้อมูลจากระบบ” “ใครหรือสิ่งใดที่จัดหาข้อมูลให้กับระบบ” “ระบบนี้ใช้งานที่ไหน” “ใครเป็นผู้ที่สนับสนุนและบำรุงรักษา” “มีระบบอื่นๆ ที่ใช้งานระบบนี้หรือไม่” เป็นต้น ซึ่งคำถามเหล่านี้เป็นคำถามตรวจสอบและทบทวนเบื้องต้นในการกำหนดผู้กระทำการ

ยูสเคส (Use Case)

ยูสเคสคือ สัญลักษณ์ที่ใช้แทนชุดลำดับของการกระทำระหว่างระบบกับผู้กระทำการโดยมีเป้าหมายเพื่อบรรลุวัตถุประสงค์ตามความต้องการทางด้านซอฟต์แวร์ที่ผู้ให้ความต้องการระบุไว้ในระบบใหม่หรือกล่าวได้ว่าเป็นฟังก์ชันของระบบ (ให้ระบบทำเพื่อต้องการผลลัพธ์อะไร) ยูสเคสประกอบด้วยลำดับของการกระทำโดยระบบ ด้วยเหตุนี้ ผู้พัฒนาจำเป็นต้องใช้มุมมองโดยสมมติตนเองเป็นระบบหรืออยู่ฝั่งเดียวกับระบบในการเขียนขั้นตอนการบรรยายโดยวิเคราะห์ว่าถ้าจะทำให้สำเร็จระบบต้องมีขั้นตอนอย่างไรบ้าง

ในช่วงเริ่มต้น การอธิบายในรูปของลำดับการทำงานหรือการดำเนินงานตามเหตุการณ์ของระบบจะใช้รูปแบบการดำเนินงานแบบสมบูรณ์หรือการทำงานแบบปกติ (Perfect flow/Normal flow of event) คือ ปฏิบัติตามขั้นตอนแบบโดยไร้ข้อผิดพลาดแล้วได้ผลลัพธ์ที่บรรลุวัตถุประสงค์ทางฟังก์ชันของยูสเคสซึ่งรูปแบบของยูสเคสที่ดี ประกอบด้วยลักษณะดังนี้

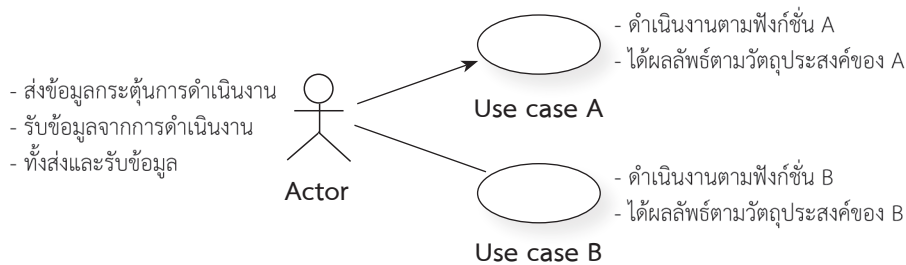
1. ลำดับของการกระทำโดยระบบต้องสามารถเสร็จสิ้นบรรลุเป้าหมายของยูสเคสได้และการกระทำที่ปฏิบัติโดยระบบนั้น เป็นความต้องการเชิงซอฟต์แวร์ทางด้านฟังก์ชันของระบบ

2. ลำดับของการกระทำในแต่ละขั้นตอน ควรมีลักษณะเป็นกิจกรรมที่เริ่มและดำเนินการไปจนถึงจุดสิ้นสุดเพื่อให้ได้ผลลัพธ์ตามที่ต้องการได้ (Atomic activity) โดยอาจประกอบด้วยขั้นตอนการตัดสินใจและขั้นตอนการร้องขอ ประกอบในการกระทำภายใต้กิจกรรม

3. ยูสเคสควรให้ผลลัพธ์ที่มั่นใจว่าสามารถสังเกตได้เพราะถ้าไม่มีใครได้รับผลลัพธ์จากยูสเคสยูสเคสที่เขียนขึ้นมานั้นอาจมีขนาดเล็กเกินไปซึ่งจำเป็นต้องทำควมรวมกับยูสเคสอื่นเพื่อให้ยูสเคสนั้นสามารถบรรลุวัตถุประสงค์ที่ผู้กระทำการต้องการได้

4. ต้องพิจารณาว่ามีผู้กระทำการเพียงหนึ่งผู้กระทำการที่ได้รับผลลัพธ์จากยูสเคสนั้น ทั้งนี้เพื่อหลีกเลี่ยงการสร้างยูสเคสที่มีขนาดใหญ่เกินไปยกตัวอย่างเช่น ระบบลงทะเบียนเรียนของนักศึกษาที่มียูสเคสชื่อว่า “Operate the registration system (ลงทะเบียนกับระบบลงทะเบียน)” ยูสเคสนี้เป็นยูสเคสที่ให้ผลลัพธ์กับหลายผู้กระทำการทำให้ยากต่อการทำความเข้าใจว่าต้องการให้ระบบทำอะไรบรรลุเป้าหมายใด ซึ่งจำเป็นต้องแก้ไขด้วยการแตกออกเป็นหลายยูสเคสเพื่อแสดงให้เห็นเป้าหมายที่ชัดเจนและชี้เฉพาะขึ้น

5. ลำดับของการกระทำในยูสเคสอาจส่งผลกระทบต่อหลายผู้กระทำการแต่มีเพียงหนึ่งผู้กระทำการเท่านั้น ที่ได้รับผลลัพธ์ซึ่งผู้กระทำการที่ได้รับผลลัพธ์นั้น เราจะเรียกว่าผู้กระทำการในระดับปฐมภูมิ (Primary actor) หรือผู้กระทำการที่สร้างขึ้นเพื่อการรับ-ส่งข้อมูลจากยูสเคส



ภาพที่ 4 ทุกยูสเคสต้องมีเป้าหมายเสมอ

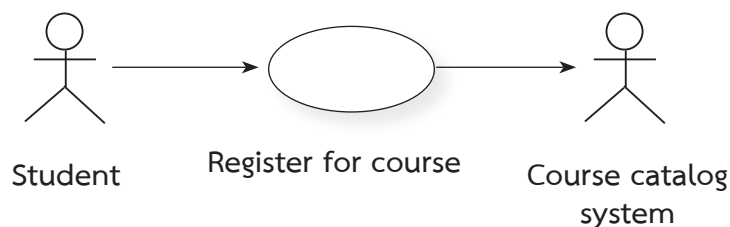
การกำหนดยูสเคสในขั้นตอนการพัฒนาโมเดลจะอยู่บนพื้นฐานแนวคิดที่ว่าผู้กระทำการต้องการให้ระบบทำอะไรหรือผู้กระทำการต้องการรับผลลัพธ์อะไรจากระบบซึ่งนักพัฒนาระบบอาจใช้คำถามเหล่านี้ในการกำหนดยูสเคสได้ เช่น “อะไรคือเป้าหมายของแต่ละผู้กระทำการ” “ทำไมผู้กระทำการจึงต้องการระบบนี้” “ผู้กระทำการสามารถสร้าง จัดเก็บเปลี่ยนแปลงลบทิ้ง หรืออ่านข้อมูลจากระบบได้หรือไม่เพราะเหตุใด” “เมื่อมีการเปลี่ยนแปลง ผู้กระทำการจำเป็นต้องแจ้งให้ทราบหรือไม่” “เมื่อมีเหตุการณ์เกิดขึ้น ผู้กระทำการจำเป็นต้องแจ้งให้ทราบหรือไม่” เป็นต้น

เมื่อกำหนดยูสเคสเสร็จสิ้นแล้วในเบื้องต้น ผู้พัฒนาควรต้องตรวจสอบอีกครั้ง เพื่อให้มั่นใจว่า ยูสเคสโมเดลที่พัฒนาขึ้นมานั้น มีฟังก์ชันครบตามที่ผู้ที่มีส่วนเกี่ยวข้องให้ความต้องการไว้ในระบบใหม่ และต้องไม่ลืมพิจารณาอุสเคสพิเศษ เช่น ยูสเคสที่เกี่ยวกับการเริ่มต้นระบบ ยูสเคสที่เกี่ยวกับการสิ้นสุดการทำงาน ของระบบ หรือยูสเคสที่เกี่ยวกับการบำรุงรักษา นอกจากนี้ต้องไม่ลืมยูสเคสที่เกิดขึ้นจากการกระตุ้นการ ทำงานของผู้กระทำการตามเหตุการณ์และงานที่ถูกกำหนดโดยอัตโนมัติตามเวลา (A time-initiated job) เช่น โปรแกรมบัญชีอาจเริ่มทำงานในตอนกลางคืนในวันสุดท้ายของแต่ละเดือน ยูสเคสที่เกี่ยวกับเหตุการณ์ อัตโนมัตินี้ถูกเริ่มต้นโดยผู้กระทำการพิเศษ

การตั้งชื่อยูสเคส ควรบ่งชี้ให้เห็นชัดว่า เมื่อดำเนินการไปแล้วได้ผลลัพธ์อะไรชื่อของแต่ละยูสเคส ควรชี้ให้เห็นถึงสิ่งที่ผู้กระทำการได้รับเมื่อกระทำกับระบบ โดยทั่วไปแล้ว ควรตั้งชื่อให้อยู่ในรูปแบบของ ผู้กระทำคือ มักขึ้นต้นด้วยคำกริยาทั้งนี้เพื่อสะท้อนเป้าหมายและผลลัพธ์ที่จะได้รับการทำงานในยูสเคส ในบางโครงการพัฒนาระบบที่ระบบมีความซับซ้อนและมีหลายทีมผู้พัฒนา ความหมายของศัพท์ต่างๆ อาจ ทำให้เกิดความผิดเพี้ยนหรือซ้ำซ้อนกันได้ง่าย ทีมผู้พัฒนาอาจต้องมีกำหนดวิธีการตั้งชื่อร่วมกัน เพื่อให้เกิด มาตรฐานเดียวและสอดคล้องกันทั้งโครงการ

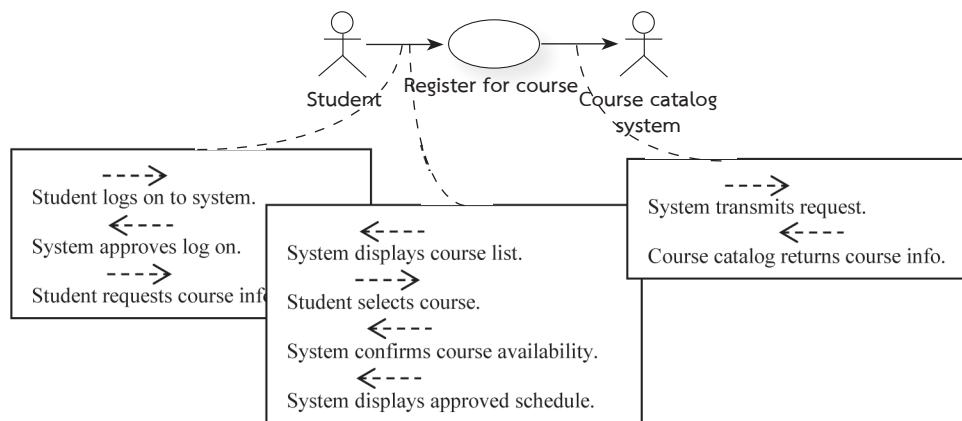
การเขียนรายละเอียดคำอธิบายในยูสเคส (Use case specification) ควรลงรายละเอียดจนกระทั่ง ทุกๆ คนสามารถเข้าใจได้โดยไม่มีการคาดเดารายละเอียด เพราะว่ามีเพียงผู้พัฒนาเท่านั้นที่รู้ว่า ผู้ว่าจ้าง ต้องการอะไรซึ่งรายละเอียดบางครั้ง อาจประกอบด้วย รูปภาพส่วนติดต่อกับผู้ใช้ (User interface) แต่ไม่ ต้องลงรายละเอียด

การติดต่อสื่อสาร



ภาพที่ 5 สัญลักษณ์ของปฏิสัมพันธ์หรือการแสดงติดต่อสื่อสารในยูสเคสโมเดล

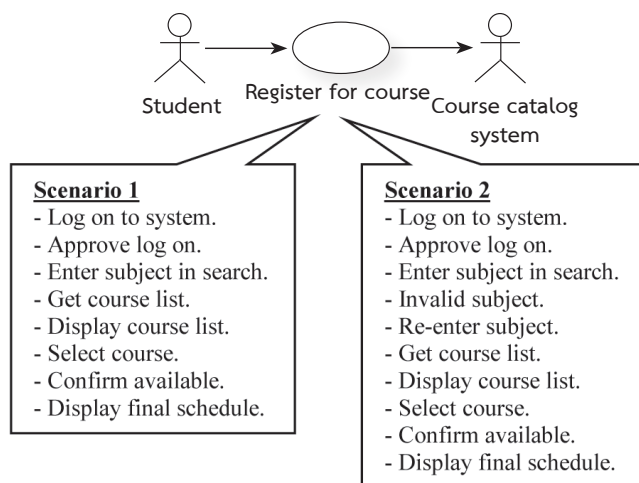
ปฏิสัมพันธ์ระหว่างผู้กระทำการและยูสเคส (Communicates-associations) เป็นการแสดงความมีส่วนร่วมในการกระทำระหว่างผู้กระทำการและยูสเคส โดยบทบาทของผู้กระทำการเป็นส่วนหนึ่งในการติดต่อกับระบบปฏิสัมพันธ์ที่แสดงในยูสเคสโมเดลตามมาตรฐานยูเอ็มแอลสัญลักษณ์ที่ใช้เป็นขีดเส้นที่ใช้หัวลูกศรแสดงทิศทางในการติดต่อซึ่งหัวลูกศรแสดงทิศทางนั้น อาจจะมีหรือไม่มีก็ได้ (Optional) วิธีการเขียนอาจสังเกตง่ายๆ ว่า หัวลูกศรอยู่ฝั่งใดท้ายของลูกศรที่อยู่ตรงข้ามจะเป็นผู้ที่เริ่มทำการติดต่อสื่อสารหรืออาจสังเกตจากผลลัพธ์ตามเป้าหมายที่ระบุในยูสเคส ว่าฝั่งใดเป็นผู้ได้รับหรือบางครั้งอาจมีความเป็นไปได้ว่าอาจเกิดทั้งสองฝั่ง



ภาพที่ 6 การติดต่อสื่อสารเป็นการบรรยายโต้ตอบกันโดยสมบูรณ์

สิ่งที่ผู้พัฒนาควรตระหนัก คือ ทิศทางของลูกศรนั้นแทนภาพรวมของขั้นตอนการทำงานทั้งหมด โดยปฏิสัมพันธ์เป็นการติดต่อสื่อสารแบบสองทางในระดับของบทบรรยายดังแสดงในภาพที่ 6 ความสัมพันธ์ระหว่างผู้กระทำการ “Student (นักเรียน)” และยูสเคส “Register for Course (ลงทะเบียนเรียนรายวิชา)” และความสัมพันธ์ระหว่าง ยูสเคส “Register for Course (ลงทะเบียนเรียนรายวิชา)” และผู้กระทำการ “Course Catalog System (ระบบข้อมูลรายวิชา)”

บทบรรยายยูสเคส (Scenario)



ภาพที่ 7 สองบทบรรยายที่มาจากยูสเคสเดียวกัน

บทบรรยาย (Scenario) เปรียบได้กับขั้นตอนการดำเนินงานในยูสเคสที่เกิดขึ้น ณ เวลาหนึ่งๆ (Use Case Instance) ซึ่งอธิบายความเป็นไปได้ของขั้นตอนที่อาจเกิดขึ้นตลอดการดำเนินงานตามเหตุการณ์ที่เกิดขึ้นในยูสเคสตัวอย่างเช่นจากภาพที่ 7 บทบรรยาย 1 สำหรับยูสเคส “Register for Course (ลงทะเบียนเรียนรายวิชา)” แสดงผู้กระทำการ “Student (นักเรียน)” กระทำกับระบบลงทะเบียนได้สำเร็จ ส่วนบทบรรยายที่ 2 ที่มาจากยูสเคสเดียวกัน แสดงผู้กระทำการ “Student (นักเรียน)” กระทำกับยูสเคส “Register for Course (ลงทะเบียนเรียนรายวิชา)” แต่เกิดเหตุการณ์ค้นหารายวิชาที่ต้องการไม่พบ ดังนั้นผู้กระทำการ “Student (นักเรียน)” ในเหตุการณ์นี้จะต้องทำการลงทะเบียนเรียนรายวิชาใหม่อีกครั้ง การลงทะเบียนเรียนรายวิชาจึงจะสำเร็จจะเห็นได้ว่า ทั้งสองบทบรรยายนั้นเป็นเหตุการณ์ที่เกิดขึ้นในยูสเคสเดียวกันมีเป้าหมายเดียวกัน แต่เกิด ณ ช่วงเวลาต่างกัน ดังนั้นการเขียนบทบรรยาย จึงจำเป็นต้องครอบคลุมทุกเหตุการณ์ที่อาจเกิดขึ้นทั้งหมดภายในยูสเคส

การเริ่มต้นเขียนบทบรรยาย อาจเริ่มจากการสังเกตการทำงานที่มีลำดับเหตุการณ์ที่สมบูรณ์ จนกระทั่งบรรลุเป้าหมายตามยูสเคสหรือที่ผู้กระทำการต้องการ ซึ่งถ้ายูสเคสมีขนาดเล็กเกินไป ให้ทำการรวมกับยูสเคสอื่น เพื่อให้ลำดับเหตุการณ์ทำงานที่มีความสมบูรณ์ ตัวอย่างเช่น ระบบลงทะเบียน มียูสเคสชื่อว่า “Enter student ID (กรอกรหัสนักเรียน)” ซึ่งยูสเคสนี้จะไม่สามารถให้ผลลัพธ์แก่ผู้กระทำการ “Student (นักเรียน)” ได้ ดังนั้นควรปรับปรุงให้อยู่ในส่วนของยูสเคสที่ใหญ่กว่า เช่น “Register for Course (ลงทะเบียนเรียนรายวิชา)” ซึ่งสามารถให้ผลลัพธ์กับผู้กระทำการได้

ข้อควรระวังในการวิเคราะห์ยูสเคสโมเดล

สิ่งที่ต้องพึงระวังในการวิเคราะห์ยูสเคสโมเดล คือ เรื่องการวิเคราะห์ปัญหาและระดับของมุมมอง เนื่องจากยูสเคสโมเดลถูกออกแบบเพื่อให้รองรับแนวคิดการออกแบบโดยการแยกระบบออกเป็นฟังก์ชันการทำงาน (Functional Decomposition) โดยส่วนที่แยกออกมาในแต่ละส่วนนั้น จะทำหน้าที่ขับเคลื่อนระบบไปพร้อมๆ กันโดยขึ้นอยู่กับเหตุการณ์ที่เกิดขึ้น (Event Driven Architecture) ซึ่งเป็นแนวคิดพื้นฐานของการวิเคราะห์และออกแบบเชิงวัตถุ แต่ปัญหาที่มักพบได้บ่อย คือ ผู้พัฒนามองยูสเคสโมเดลเป็นการแตกของปัญหา เหมือนการแตกปัญหาที่ซับซ้อนให้เป็นปัญหาเล็กๆ และทำการควบคุมวิธีการแก้ปัญหาเหล่านั้นให้เป็นผลสำเร็จ หากเป็นระบบที่มีขนาดเล็กและไม่ซับซ้อน หรือระบบที่ไม่มีแผนการต่อขยายเพิ่มเติมในอนาคต ปัญหาตรงนี้อาจมีผลกระทบไม่มากนัก ถ้าเทียบกับระบบที่มีขนาดใหญ่ ลักษณะของยูสเคสโมเดลที่มีปัญหาจากการวิเคราะห์ คือ ยูสเคสที่เกิดขึ้นมานั้นไม่แสดงผลลัพธ์ที่สังเกตได้เมื่อดำเนินงานจนเสร็จสิ้น ยูสเคสที่ได้มีขนาดเล็กและมีจำนวนมาก ที่สำคัญ มีการดำเนินการตามขั้นตอนตามยูสเคส ซึ่งในความเป็นจริง ยูสเคสแต่ละยูสเคสต้องเป็นอิสระจากกันและดำเนินการเสร็จสิ้นภายในยูสเคสเอง นอกจากนี้ ชื่อของยูสเคสที่ใช้ อยู่ในระดับของการกระทำ การคือ “กระทำ (Operation) กับอะไร (Object)” หรือ “ฟังก์ชัน (Function) กับข้อมูลอะไร (Data)” ตัวอย่างเช่น “สอดบัตรเอทีเอ็ม (Insert ATM Card)” ซึ่งจะเห็นว่าเป็นเพียงการกระทำหนึ่งเท่านั้น

ในส่วนของระดับมุมมอง เนื่องจากยูสเคสโมเดลเป็นโมเดลเริ่มต้น การวิเคราะห์ยูสเคสโมเดลจะใช้มุมมองในภาพกว้าง (Macroscopic) คือ มองทั้งระบบและพิจารณาบริบทของระบบ เช่น ถ้าสร้างระบบนี้จะสร้างอย่างไร “ผู้ใช้ต้องการได้รับอะไร” “อะไรคือเป้าหมายของยูสเคส” “ถ้าเพิ่มยูสเคสนี้ลงไปจะมีประโยชน์อย่างไร” “ยูสเคสนี้มีความหมายสื่อถึงเรื่องใด” เป็นต้น ระดับของการสร้างยูสเคส ควรอยู่ในระดับเดียวกัน คือ ระดับความเป็นฟังก์ชันที่ปฏิบัติแล้วได้หนึ่งผลลัพธ์ที่สังเกตได้ตามเป้าหมาย

สรุป

เป็นที่ทราบกันกว้างขวางวิเคราะห์และออกแบบระบบนั้น เป็นงานในเชิงลักษณะศิลปะและวิทยาศาสตร์ ศิลปะแสดงความคิดสร้างสรรค์ การพินิจวิเคราะห์ที่ได้สัดส่วนที่เหมาะสมลงตัว และมาจากการฝึกฝนเรียนรู้ หรือประสบการณ์เทคนิคส่วนบุคคล ส่วนวิทยาศาสตร์เป็นการใช้ตรรกะเพื่อการแก้ไขปัญหาที่เกิดขึ้นอย่างเป็นระบบเป็นระเบียบ และอยู่บนมาตรฐานสากลที่ยอมรับได้

ยูสเคสโมเดลเป็นแบบจำลองกรณีบังคับใช้และพฤติกรรมของระบบ โดยใช้วิธีการนำเสนอในรูปแบบจำลอง เพื่อให้ผู้อ่านเกิดจินตภาพและสามารถเข้าใจได้ง่ายขึ้นทุกยูสเคสจะต้องมีชื่อ ซึ่งชื่อที่ตั้งควรสื่อถึงพฤติกรรมในตัวยูสเคสและสามารถทำให้ผู้ที่อ่านเข้าใจได้อย่างรวดเร็วโดยหนึ่งยูสเคสจะแทนความต้องการเชิงฟังก์ชันหนึ่งฟังก์ชัน ซึ่งต้องสามารถพิสูจน์ได้ว่ามาจากความต้องการโดยยูสเคสไม่ควรมีพฤติกรรมมากเกินไป ต้องมีความเป็นหนึ่งเดียวและมีความหมายในตัวเองที่สามารถอธิบายได้ เอกสารประกอบคำบรรยาย

ยูสเคสโมเดล มีส่วนสำคัญช่วยในเรื่องของการตรวจสอบว่ามีข้อผิดพลาดหรือไม่ ถ้าข้อผิดพลาดจะต้องมีการเปลี่ยนข้ออย่างไรคำบรรยายควรอธิบายแบบกะทัดรัดแต่สะท้อนให้เห็นภาพรวมที่ถูกต้อง ควรตรวจสอบว่าแต่ละยูสเคสมีผู้กระทำการอย่างน้อยหนึ่งผู้กระทำการหรือไม่ เพื่อป้องกันยูสเคสที่ไม่ได้ใช้งานหรือไม่มีโอกาสเข้าถึง ควรตรวจสอบความซ้ำซ้อนของยูสเคสว่ามีความคล้ายคลึงกันหรือไม่ ซึ่งวิธีที่ดีที่สุด คือตรวจสอบกับพฤติกรรมหรือแผนภาพเหตุการณ์และผลลัพธ์ที่เกิดขึ้น เนื่องจากบางครั้ง ผู้พัฒนาอาจใช้กรณีของเหตุการณ์ไปกำหนดเป็นยูสเคสใหม่

ในส่วนของผู้กระทำการควรตรวจสอบว่าบทบาททั้งหมดในบริบทของระบบครบแล้วหรือยังมีคำอธิบายครบถ้วนและเข้าใจตรงกันหรือไม่ผู้กระทำการที่สร้างขึ้นมีความเกี่ยวข้องกับยูสเคสหรือไม่ สุดท้ายต้องระวังผู้กระทำการที่บางครั้งสามารถแสดงบทบาทภายใต้ชื่อมากกว่าหนึ่งชื่อ

สุดท้ายหลักในการพัฒนายูสเคสโมเดล คือ แผนภาพยูสเคสที่ดูไม่ซับซ้อนและทำความเข้าใจได้ง่าย ซึ่งผู้พัฒนาจะต้องเข้าใจระบบอย่างลึกซึ้งให้ถูกต้องและชัดเจน เพราะบางครั้งการที่ไม่เข้าใจอะไรบางอย่าง อาจทำให้พลาดข้อมูลส่วนที่สำคัญไปได้ที่สำคัญยูสเคสต้องมีเป้าหมายเสมอและลำดับขั้นตอนที่อยู่ในยูสเคสต้องละเอียดชัดเจน เพราะเป็นข้อมูลการพัฒนาในขั้นตอนวิเคราะห์และออกแบบ ซึ่งสิ่งเหล่านี้ควรให้ผู้ที่มีส่วนได้ส่วนเสียช่วยในการตรวจสอบ

References

- Booch, G., Rumbaugh, J., Jacobson, I. (1998) *The Unified Modeling Language User Guide*. Reading, Massachusetts : Addison-Wesley.
- Borland. (2012). *Caliber RM Enterprise: Software Requirements Management System*. Retrieved October 15, 2012 from <http://www.borland.com/us/products/caliber/index.html>
- IBM., *Rational Requisite Pro: A Requirements Management Tool*. Retrieved September 25, 2012 from [http://www-01.ibm.com/software/awdtools/reqpro/\(2008\)](http://www-01.ibm.com/software/awdtools/reqpro/(2008))
- Kotonya G., & Sommerville I. (1998) *Requirements Engineering Processes and Techniques*. Chichester : John Wiley & Sons.
- Leffingwell, D., & Widrig, D. (2004) *Managing Software Requirements: A Use Case Approach*. 2nd Edition. Massachusetts : Addison-Wesley Professional.
- Sommerville, I. (2010). *Software Engineering*. 9th Edition. Boston, Massachusetts : Addison-Wesley.

Rational Software. (2005). *Requirements Management with Use Case*. Boston, Massachusetts : Addison-Wesley.

Surakratanasakul B. & Chattong B. (2014). Software Requirements Eliciting Techniques. *HCU Journal*, (16)32, 169-183.

ผู้เขียน

ผู้ช่วยศาสตราจารย์ บุญประเสริฐ สุรักษรัตนสกุล

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

e-mail: boonprasert@it.kmitl.ac.th

