สำนักงานวิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้ จะกล่าวถึงทฤษฎีเกี่ยวข้องที่สำคัญในการนำมาประยุกต์ใช้ สนับสนุนและอ้างอิงในการทำงานวิทยานิพนธ์ รวมถึงรายละเอียดในส่วนของงานวิจัยต่างๆ ที่เกี่ยวข้อง โดยมีเนื้อหาดังต่อไปนี้

2.1 ทฤษฎีที่เกี่ยวข้อง

งานวิทยานิพนธ์นี้นำเสนอวิธีการจัดเก็บและค้นคืนคำอธิบายยูสเคสสำหรับสายผลิตภัณฑ์ซอฟต์แวร์ โดยมีทฤษฎีและงานวิจัยที่เกี่ยวข้องดังต่อไปนี้

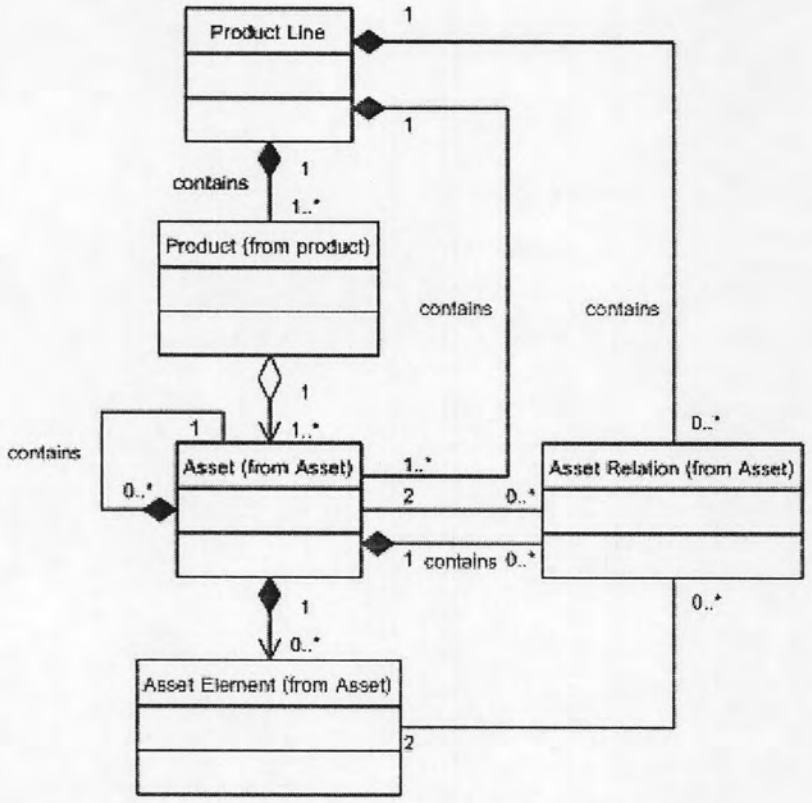
2.1.1 สายผลิตภัณฑ์ซอฟต์แวร์ (Software Product Line)

แนวคิดเรื่องสายผลิตภัณฑ์ซอฟต์แวร์มีการใช้งานอย่างกว้างขวางในวงการอุตสาหกรรมอื่นๆ เช่น บริษัทแอร์บัสในยุโรปมีการผลิตเครื่องบินหลายรุ่น ได้แก่ A-318 A-319 A-320 และ A-321 เครื่องบินแต่ละรุ่นจะมีคุณสมบัติหรือส่วนประกอบหลายอย่างร่วมกัน เช่น เก้าอี้โดยสาร อุปกรณ์นำร่อง หรืออุปกรณ์สื่อสารทางอากาศ เป็นต้น โดยแต่ละรุ่นก็จะมีคุณสมบัติอื่นที่แตกต่างกัน เช่น ขนาดของเครื่องยนต์ จำนวนที่นั่ง เป็นต้น สายผลิตภัณฑ์ซอฟต์แวร์มีชื่อเรียกอื่นๆ เช่น กลุ่มผลิตภัณฑ์ซอฟต์แวร์ (Software Product Family) กลุ่มของระบบ (Family of Systems) หรือแอพพลิเคชันโดเมน (Application Domain) เป็นต้น

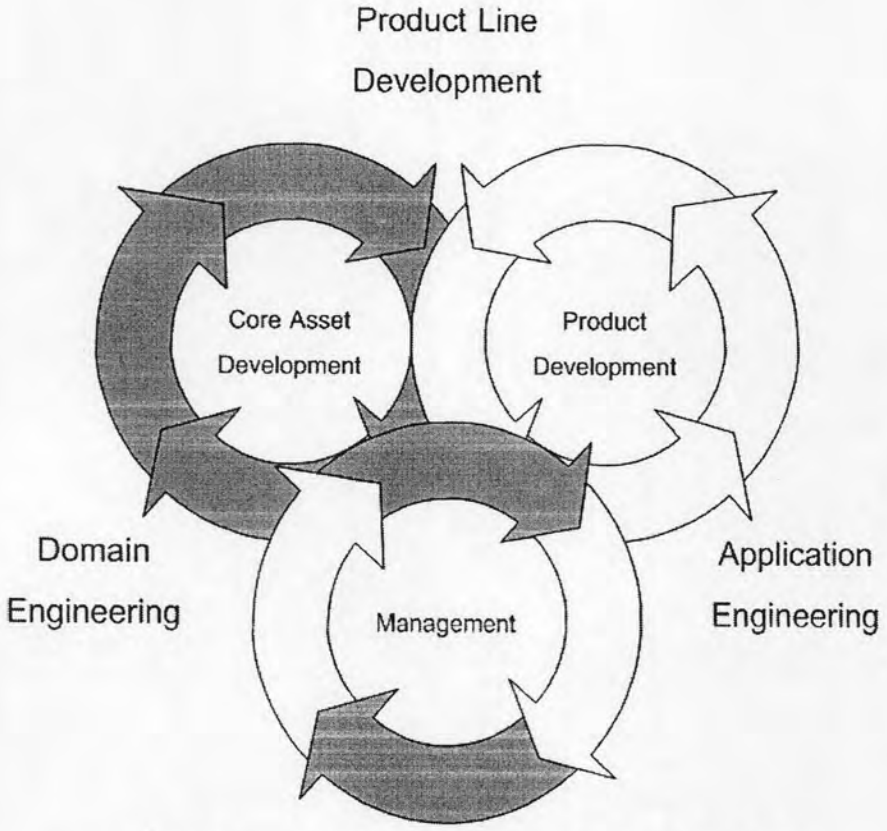
เราสามารถอธิบายสายผลิตภัณฑ์ซอฟต์แวร์ด้วยแบบจำลองดังรูปที่ 2.1 สายผลิตภัณฑ์จะประกอบไปด้วยผลิตภัณฑ์ โดยตัวผลิตภัณฑ์ถูกสร้างจากการนำสินทรัพย์มาประกอบกัน ซึ่งสินทรัพย์สามารถนำไปใช้ได้ในหลายๆ ผลิตภัณฑ์ โดยสินทรัพย์ใหม่อาจสร้างมาจากสินทรัพย์ที่มีอยู่ ภายในสินทรัพย์จะประกอบไปด้วยส่วนย่อยสินทรัพย์ (Asset Element) ที่เป็นรายละเอียดเฉพาะของแต่ละประเภทของสินทรัพย์ เช่น เอกสาร รหัสโปรแกรม และแผนภาพ เป็นต้น ความสัมพันธ์สินทรัพย์เป็นตัวกำหนดความสัมพันธ์ระหว่างสินทรัพย์ที่จะนำไปสร้างเป็นผลิตภัณฑ์ และกำหนดความสัมพันธ์ระหว่างส่วนย่อยสินทรัพย์ เพื่อจะนำไปสร้างเป็นสินทรัพย์ต่อไป

Clement และ Northrop [4] ได้อธิบายว่า กระบวนการพัฒนาสายผลิตภัณฑ์ซอฟต์แวร์ จะประกอบด้วยกิจกรรมที่สำคัญ 3 กิจกรรม คือ การพัฒนาสินทรัพย์หลัก (Core Asset Development) การพัฒนาผลิตภัณฑ์ (Product Development) และการจัดการ (Management) ทั้งทางด้านเทคนิคและด้านองค์กร โดยการพัฒนาสินทรัพย์หลักและการพัฒนาผลิตภัณฑ์จะเป็น

กิจกรรมที่เกิดขึ้นได้ในทั้งสองทิศทาง คือ การสร้างซอฟต์แวร์จากสินทรัพย์หลัก หรือการดึงเอาสินทรัพย์หลักออกมาจากซอฟต์แวร์ที่มีอยู่ การพัฒนาสินทรัพย์หลักมีชื่อเรียกอีกอย่างหนึ่งว่าวิศวกรรมโดเมน ส่วนการพัฒนาผลิตภัณฑ์มีชื่ออีกอย่างหนึ่งว่าวิศวกรรมแอพพลิเคชัน ความเกี่ยวข้องของทั้งสามกิจกรรมแสดงดังรูปที่ 2.2



รูปที่ 2.1 แบบจำลองสายผลิตภัณฑ์ซอฟต์แวร์ [4]



รูปที่ 2.2 **กิจกรรมสำคัญของสายผลิตภัณฑ์ซอฟต์แวร์** [2]

1) การพัฒนาสินทรัพย์หลัก

เป้าหมายของการสร้างสินทรัพย์หลัก คือ การสร้างความสามารถในการผลิตผลิตภัณฑ์ โดยสิ่งที่เป็นข้อมูลในการพัฒนาสินทรัพย์หลัก ประกอบด้วย

- ข้อจำกัดของผลิตภัณฑ์ (Product Constraints)
- แนวทาง แบบรูป และกรอบการทำงาน (Styles, Patterns, and Frameworks)
- ข้อจำกัดในการผลิต (Production Constraints)
- กลยุทธ์ในการผลิต (Production Strategy)
- รายการของสินทรัพย์ที่มีอยู่ (Inventory of pre-existing assets)

ข้อมูลต่างๆ ข้างต้นจะเป็นตัวกำหนดเพื่อให้ได้ผลลัพธ์ในกิจกรรมการพัฒนาสินทรัพย์หลัก ซึ่งผลลัพธ์ที่ได้ ประกอบด้วย

- ขอบเขตของสายผลิตภัณฑ์ (Product Line Scope)
- สินทรัพย์หลัก (Core Assets)
- แผนการผลิต (Production Plan)

2) การพัฒนาแอพพลิเคชัน

ข้อมูลนำเข้าของกิจกรรมการพัฒนาแอพพลิเคชัน ประกอบด้วย

- ความต้องการของผลิตภัณฑ์
- ขอบเขตของสายผลิตภัณฑ์
- สินทรัพย์หลักที่จะนำมาสร้างผลิตภัณฑ์
- แผนการผลิต

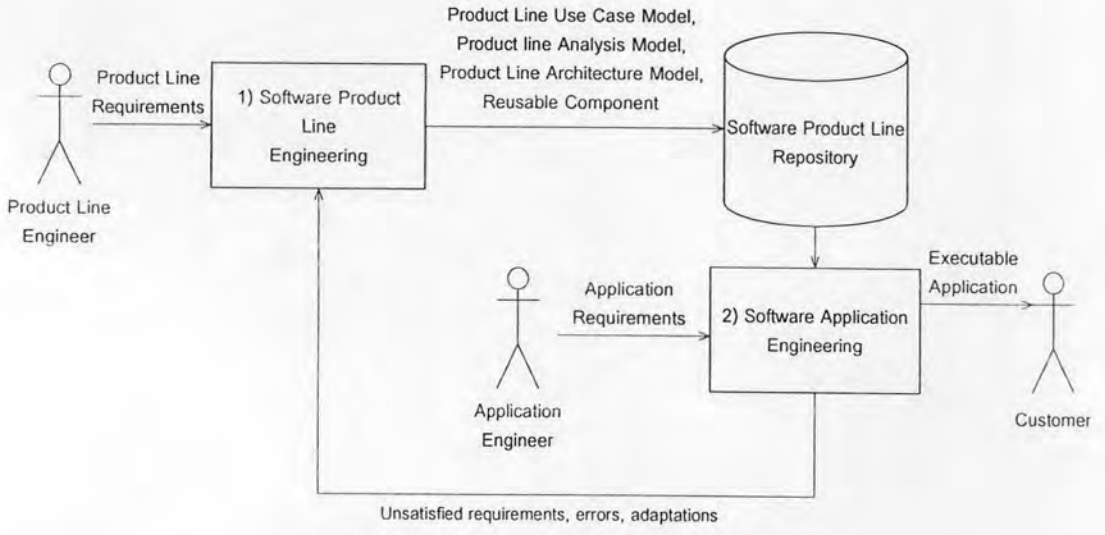
3) การจัดการ

การจัดการเป็นกิจกรรมที่มีบทบาทสำคัญในการกำหนดความสำเร็จของการสร้างสายผลิตภัณฑ์ กิจกรรมนี้ประกอบด้วย งานเชิงเทคนิคและงานบริหารจัดการเชิงองค์กร เช่น การจัดการทรัพยากร และการกำหนดแนวทางการสร้างสายผลิตภัณฑ์ เป็นต้น

2.1.2 กระบวนการวิศวกรรมซอฟต์แวร์เชิงสายการผลิตแบบวิวัฒนาการ (Evolutionary Software Product Line Engineering Process: ESPLEP)

อีเอสพีแอลอีพี[1] เป็นกระบวนการทางซอฟต์แวร์ที่มีแนวทางในการลดความความแตกต่างระหว่างการพัฒนาซอฟต์แวร์ (Software Development) กับการบำรุงรักษาซอฟต์แวร์ (Software Maintenance) โดยนำเสนอในรูปแบบของการพัฒนาแบบวนซ้ำ (Iteration) ซึ่งระบบที่พัฒนาตามกระบวนการนี้สามารถปรับปรุงให้เข้ากับการเปลี่ยนแปลงความต้องการในแต่ละรอบของการพัฒนาได้ ดังนั้นอีเอสพีแอลอีพีเป็นการพัฒนาซอฟต์แวร์ที่มีพื้นฐานอยู่บนผลิตภัณฑ์ที่มีอยู่เดิม

กระบวนการอีเอสพีแอลอีพีแบ่งออกเป็น 2 กระบวนการย่อย ดังรูปที่ 2.3 คือ



รูปที่ 2.3 กระบวนการอีเอสพีแอลอีพี [2]

1) **วิศวกรรมสายผลิตภัณฑ์ซอฟต์แวร์** (Software Product Line Engineering)

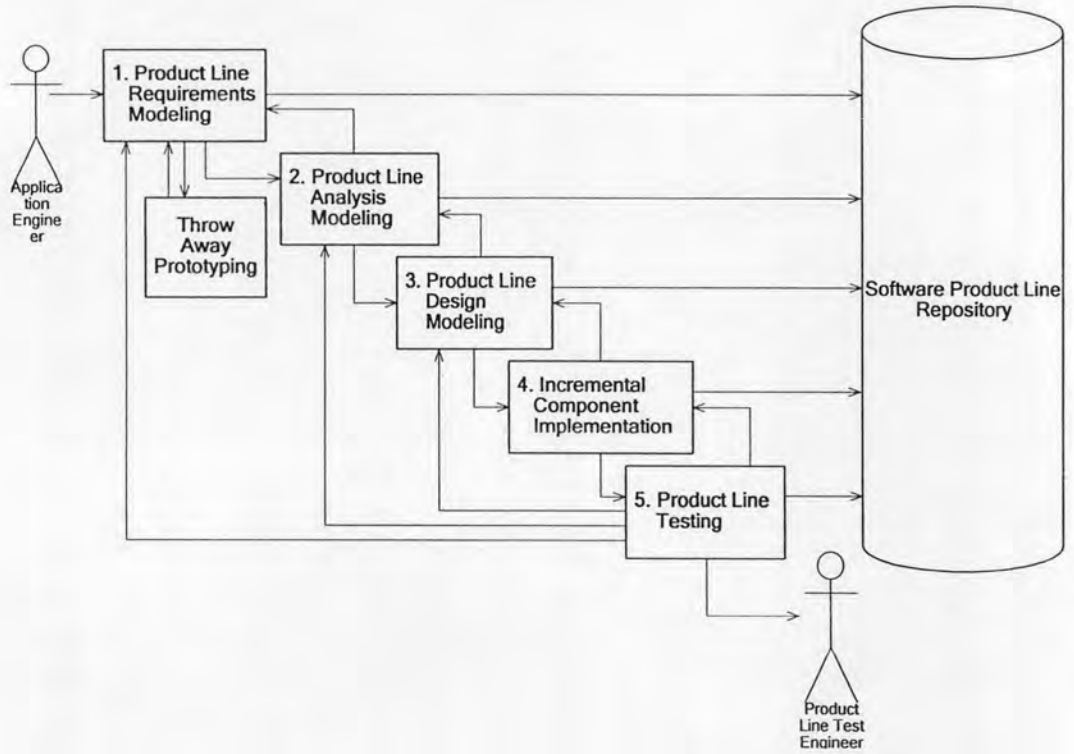
วิศวกรรมสายผลิตภัณฑ์ซอฟต์แวร์เป็นกระบวนการที่วิเคราะห์หาลักษณะทั่วไปและลักษณะแปรผันของสายผลิตภัณฑ์ซอฟต์แวร์ทั้งระบบ กิจกรรมในกระบวนการนี้ ได้แก่ การพัฒนาแบบจำลองยูสเคสสายผลิตภัณฑ์ซอฟต์แวร์ (Product Line Use Case Model) แบบจำลองการวิเคราะห์สายผลิตภัณฑ์ (Product Line Analysis Model) สถาปัตยกรรมสายผลิตภัณฑ์ซอฟต์แวร์ (Product Line Software Architecture) และการพัฒนาคอมโพเนนท์ที่สามารถนำกลับมาใช้ได้ (Reusable Component) โดยผลลัพธ์ที่ได้จากกิจกรรมต่างๆ จะถูกจัดเก็บลงในแหล่งเก็บสายผลิตภัณฑ์ซอฟต์แวร์ (Software Product Line Repository)

2) **วิศวกรรมซอฟต์แวร์แอพพลิเคชัน** (Software Application Engineering)

วิศวกรรมซอฟต์แวร์แอพพลิเคชันเป็นกระบวนการพัฒนาแอพพลิเคชันที่อยู่ในสายผลิตภัณฑ์ซอฟต์แวร์ โดยนำสินทรัพย์ในแหล่งเก็บสายผลิตภัณฑ์ซอฟต์แวร์มาใช้สร้างผลิตภัณฑ์ตามความต้องการของแต่ละแอพพลิเคชัน สร้างแผนภาพยูสเคสของแอพพลิเคชันจากแผนภาพยูสเคสสายผลิตภัณฑ์ซอฟต์แวร์ สร้างแผนภาพการวิเคราะห์แอพพลิเคชันจากแผนภาพการวิเคราะห์สายผลิตภัณฑ์ซอฟต์แวร์ สร้างสถาปัตยกรรมแอพพลิเคชันจากสายผลิตภัณฑ์ซอฟต์แวร์ และเลือกคอมโพเนนท์ที่เหมาะสมจากแหล่งเก็บสายผลิตภัณฑ์ซอฟต์แวร์มาประกอบเป็นแอพพลิเคชัน

2.1.3 วิศวกรรมสายผลิตภัณฑ์ซอฟต์แวร์บนพื้นฐานยูเอ็มแอลหรือพลัส (Product Line UML-Based Software Engineering: PLUS)

พลัส[1] เป็นหลักการทางวิศวกรรมซอฟต์แวร์ว่าด้วยกระบวนการในการวิเคราะห์ออกแบบ และพัฒนาสายผลิตภัณฑ์ซอฟต์แวร์ โดยใช้ยูเอ็มแอลเป็นภาษาในการอธิบาย ซึ่งได้มีการเพิ่มเติมสัญลักษณ์และแผนภาพต่างๆ จากมาตรฐานเดิมของยูเอ็มแอลรุ่น 1.0 โดยพลัสอ้างอิงกระบวนการจากอีเอสพีแอลอีพี ซึ่งในขั้นตอนวิศวกรรมสายผลิตภัณฑ์ซอฟต์แวร์จะมีการแบ่งออกเป็น 5 ขั้นตอนดังรูปที่ 2.4



รูปที่ 2.4 **วิศวกรรมสายผลิตภัณฑ์ซอฟต์แวร์ในกระบวนการพลัส** [2]

1) **แบบจำลองความต้องการสายผลิตภัณฑ์ซอฟต์แวร์** (Software Product Line Requirements Modeling)

ในขั้นตอนนี้จะใช้แบบจำลองยูสเคส (Use Case Model) และแบบจำลองคุณลักษณะ (Feature Model) เพื่อแสดงความต้องการของสายผลิตภัณฑ์ซอฟต์แวร์ โดยแผนภาพยูสเคสใช้อธิบายความต้องการเชิงฟังก์ชัน (Functional Requirements) โดยมีการเพิ่มเติมสัญลักษณ์เพื่อบ่งบอกว่าความต้องการดังกล่าวเป็นลักษณะทั่วไปหรือเป็นลักษณะแปรผัน ผ่านทางการกำหนดประเภทของยูสเคสแบบเคอร์เนล (Kernel) แบบเลือกได้ (Optional) หรือแบบทางเลือก (Alternative) สร้างคำอธิบายยูสเคส (Use Case Description) หรืออาจสร้างต้นแบบแบบโธรว์อะเวย์ (Throw Away Prototype) เพื่ออธิบายความต้องการให้ชัดเจนขึ้น

ในส่วนของแบบจำลองคุณลักษณะ เป็นแบบจำลองใหม่ที่กำหนดเพิ่มเติมลงไปนอกเหนือจากแบบจำลองมาตรฐานของยูเอ็มแอล เป็นแบบจำลองที่อธิบายภาพรวมของคุณลักษณะและความสัมพันธ์ทั้งหมดในสายผลิตภัณฑ์ การสร้างผลิตภัณฑ์ในสายผลิตภัณฑ์จะดึงเอาคุณลักษณะที่ต้องการจากแบบจำลองดังกล่าวนำไปสร้างเป็นซอฟต์แวร์ตามรูปแบบที่กำหนดไว้

2) **แบบจำลองการวิเคราะห์สายผลิตภัณฑ์ซอฟต์แวร์** (Software Product Line Analysis Modeling)

แบบจำลองการวิเคราะห์สายผลิตภัณฑ์ซอฟต์แวร์เป็นขั้นตอนการวิเคราะห์และสร้างแบบจำลองเชิงสถิต (Static Model) และแบบจำลองเชิงพลวัต (Dynamic Model) แบบจำลองเชิงสถิตจะแสดงโครงสร้างความสัมพันธ์ของคลาสต่างๆ ของระบบในรูปแบบของแผนภาพคลาส (Class Diagram) ที่มีการแบ่งประเภทของคลาสเป็นแบบเคอร์เนล แบบเลือกได้ และแบบแปรผัน (Variant) ส่วนแบบจำลองเชิงพลวัตจะแสดงลักษณะการโต้ตอบ (Interaction) ของวัตถุต่างๆ ในรูปแบบของแผนภาพการติดต่อ (Communication Diagram) นอกจากนั้นแต่ละคุณลักษณะของสายผลิตภัณฑ์ซอฟต์แวร์ สามารถอธิบายความสัมพันธ์ของคลาสได้ด้วยแผนภาพความสัมพันธ์คุณลักษณะ/คลาส (Feature/Class Relation Diagram)

3) **แบบจำลองการออกแบบสายผลิตภัณฑ์ซอฟต์แวร์** (Software Product Line Design Modeling)

ลักษณะการออกแบบสายผลิตภัณฑ์ซอฟต์แวร์เป็นการออกแบบเชิงคอมโพเนนท์ (Component Based Architecture) โดยใช้ข้อมูลที่ได้จากขั้นตอนการวิเคราะห์จากแบบรูป (Pattern) ทางสถาปัตยกรรมที่สอดคล้องกัน และสร้างเป็นโครงสร้างทางสถาปัตยกรรมสายผลิตภัณฑ์ซอฟต์แวร์ (Product Line Software Architecture)

4) **การเขียนโปรแกรมและการทดสอบคอมโพเนนท์ส่วนเพิ่ม** (Increment Component Implementation)

เมื่อได้สถาปัตยกรรมสายผลิตภัณฑ์ซอฟต์แวร์ ขั้นตอนต่อไปเป็นการพัฒนาคอมโพเนนท์ให้ตรงตามการออกแบบ โดยใช้หลักการพัฒนาส่วนเพิ่ม เริ่มจากการพัฒนาส่วนที่เป็นหลักก่อนแล้วจึงค่อยพัฒนาส่วนเลือกได้หรือส่วนทางเลือก การเขียนโปรแกรมต้องมีการทดสอบระดับหน่วย (Unit Testing) เพื่อให้มั่นใจว่าแต่ละส่วนของซอฟต์แวร์ทำงานได้อย่างถูกต้อง

5) **การทดสอบสายผลิตภัณฑ์** (Product Line Testing)

การทดสอบประกอบด้วยการทดสอบเบ็ดเสร็จ (Integration Testing) และการทดสอบเชิงฟังก์ชัน (Functional Testing)

ในวิทยานิพนธ์นี้มุ่งเน้นไปในขั้นตอนการวิเคราะห์ความต้องการของสายผลิตภัณฑ์ตามกระบวนการพลัส ด้วยแบบจำลองยูสเคส โดยแนวทางการออกแบบมีดังนี้

2.1.3.1 **ยูสเคสสายผลิตภัณฑ์**

การออกแบบยูสเคสสายผลิตภัณฑ์ซอฟต์แวร์นั้น มีลักษณะที่แตกต่างจากการออกแบบยูสเคสของระบบทั่วไป กระบวนการวิเคราะห์และออกแบบสายผลิตภัณฑ์ซอฟต์แวร์ต้องมีการคำนึงถึงลักษณะทั่วไป และลักษณะแปรผันของระบบ ดังนั้นจึงมีการแบ่งประเภทของยูสเคสออกเป็นประเภทต่างๆ ดังนี้

(1) ยูสเคสเคอร์เนล (Kernel Use Case)

ยูสเคสเคอร์เนลเป็นยูสเคสหลักที่ทุกผลิตภัณฑ์ในสายผลิตภัณฑ์ต้องมี

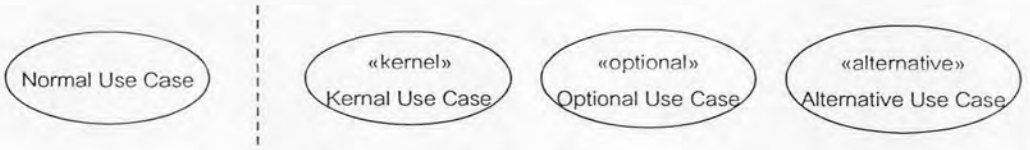
(2) ยูสเคสเลือกได้ (Optional Use Case)

ยูสเคสเลือกได้เป็นยูสเคสที่ผู้ออกแบบผลิตภัณฑ์สามารถเลือกที่จะนำมาใช้หรือไม่นำใช้ในการผลิตภัณฑ์ ขึ้นอยู่กับความต้องการหรือข้อกำหนดที่ได้ออกแบบไว้

(3) ยูสเคสทางเลือก (Alternative Use Case)

ยูสเคสทางเลือกเป็นกลุ่มของยูสเคสที่ผู้ออกแบบจะต้องทำการเลือกใช้ยูสเคสตัวใดตัวหนึ่งจากกลุ่มยูสเคสที่มี

ดังนั้นในยูสเคสสายผลิตภัณฑ์ตามกระบวนการพลัสจึงได้มีการกำหนดสเตอริโอไทป์ (Stereotype) ลงไปในแต่ละยูสเคสเพื่อบ่งบอกประเภทของยูสเคส ได้แก่ <<kernel>>, <<optional>> และ <<alternative>> ดังรูปที่ 2.5



รูปที่ 2.5 ความแตกต่างระหว่างยูสเคสทั่วไปกับยูสเคสสายผลิตภัณฑ์

สำหรับในส่วนของคำอธิบายยูสเคสสายผลิตภัณฑ์ จะมีส่วนที่แตกต่างจากยูสเคสทั่วไปตามตารางที่ 2.1

ตารางที่ 2.1 เปรียบส่วนประกอบโครงสร้างยูสเคสทั่วไปและยูสเคสสายผลิตภัณฑ์ตามกระบวนการพลัส

ส่วนประกอบ	คำอธิบาย	ยูสเคสทั่วไป	ยูสเคสสายผลิตภัณฑ์
Use Case Name:	ชื่อยูสเคส (Use Case Name) เป็นชื่อเรียกยูสเคส โดยยูสเคสแต่ละยูสเคสจะมีชื่อเรียกที่แตกต่างกัน	มี	มี
Reuse Category:	ประเภทการนำกลับมาใช้ (Reuse Category) เป็นตัวบ่งบอกประเภทของยูสเคสว่าเป็นแบบเคอร์เนล เลือกใช้ หรือทางเลือก	ไม่มี	มี
Summary:	รายละเอียดสรุป (Summary) แสดงคำอธิบายโดยย่อของยูสเคส	มี	มี
Actor:	ผู้กระทำ (Actor) แสดงรายชื่อผู้กระทำของยูสเคสในรูปแบบของผู้กระทำปฐมภูมิ (Primary Actor) หรือผู้กระทำทุติยภูมิ (Secondary Actor)	มี	มี
Dependency:	การขึ้นต่อกัน (Dependency) แสดงความสัมพันธ์กับยูสเคสอื่นๆ ในแบบการใช้ (Use) หรือการขยาย (Extend)	มี	มี
Precondition:	เงื่อนไขก่อน (Precondition) แสดงข้อกำหนดหรือเงื่อนไขที่ต้องมีหรือต้องเกิดขึ้นก่อนเริ่มยูสเคส	มี	มี
Description:	คำอธิบาย (Description) แสดงขั้นตอนการทำงานปรกติของยูสเคส	มี	มี
Alternative:	ทางเลือก (Alternative) แสดงขั้นตอนอื่นๆ ที่อาจเกิดในยูสเคสในกรณีที่เงื่อนไขการทำงานไม่ตรงกับการทำงานปรกติของยูสเคส	มี	มี
Post Condition:	เงื่อนไขสิ้นสุด (Post Condition) เป็นสถานะหรือเงื่อนไขที่จะเกิดขึ้นเมื่อยูสเคสทำสำเร็จ	มี	มี
Name:	ชื่อ (Name) เป็นชื่อเรียกส่วนแปรผัน ชื่อดังกล่าวจะใช้เป็นชื่อคุณสมบัติของระบบ	ไม่มี	มี

ตารางที่ 2.1 เปรียบส่วนประกอบโครงสร้างยูสเคสทั่วไปและยูสเคสสายผลิตภัณฑ์ตามกระบวนการพลัส (ต่อ)

ส่วนประกอบ	คำอธิบาย	ยูสเคสทั่วไป	ยูสเคสสายผลิตภัณฑ์
Type of Functionality:	ประเภทของการทำงาน (Type of Functionality) เป็นตัวบ่งบอกประเภทของส่วนแปรผันซึ่งแบ่งออกเป็น 2 ประเภทคือ 1) แบบทางเลือกจำเป็น (Mandatory Alternative) เป็นส่วนแปรผันที่ต้องเลือกใช้อย่างใดอย่างหนึ่งภายในทางเลือกที่กำหนด 2) แบบเลือกได้ เป็นส่วนแปรผันที่สามารถที่จะเลือกหรือไม่ขึ้นอยู่กับข้อกำหนดของซอฟต์แวร์ที่ต้องการ	ไม่มี	มี
Line numbers	หมายเลขบรรทัด (Line numbers) เป็นตัวบ่งบอกว่าส่วนแปรผันเกิดขึ้นที่ตำแหน่งใดในคำอธิบาย	ไม่มี	มี
Description of Functionality:	คำอธิบายการทำงาน (Description of Functionality) เป็นคำอธิบายขั้นตอนการทำงานของส่วนแปรผัน	ไม่มี	มี

2.1.3.2 **คำอธิบายยูสเคสสายผลิตภัณฑ์**

คำอธิบายเป็นคำอธิบายขั้นตอนการทำงานของแต่ละยูสเคส ซึ่งคำอธิบายยูสเคสสายผลิตภัณฑ์จะมีการเพิ่มเติมรายละเอียดที่แตกต่างจากคำอธิบายยูสเคสทั่วไปดังนี้

1) ประเภทการนำกลับมาใช้

ในคำอธิบายยูสเคสจะมีการเพิ่มหัวข้อ ประเภทการนำกลับมาใช้ เพื่อระบุว่ายูสเคสแต่ละตัวเป็นยูสเคสประเภทใด ซึ่งจะสอดคล้องกับสเตอริโอไทป์ทั้ง 3 ประเภท คือ เคอร์เนล เลือกได้ และทางเลือก

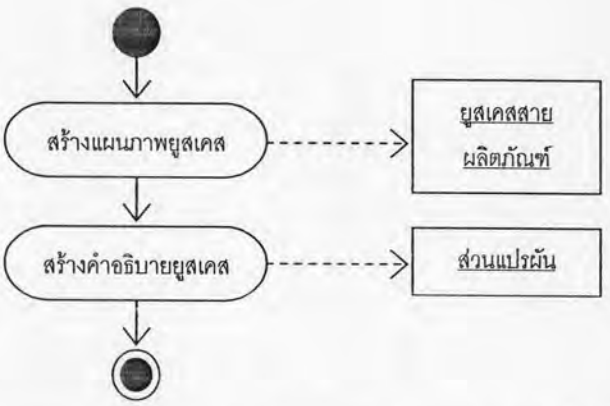
2) ส่วนแปรผัน (Variation Point)

ส่วนแปรผันเป็นคำอธิบายคุณสมบัติของระบบที่สามารถเลือกใช้ เพื่อให้เกิดลักษณะการทำงานที่แตกต่างกันออกไป ส่วนแปรผันจะมีลักษณะคล้ายเป็นยูสเคสย่อยของยูสเคสหลัก โดยจะแทรกอยู่ในแต่ละขั้นตอนการทำงานภายในคำอธิบาย (หัวข้อ 7 ของคำอธิบายยูสเคส)

ส่วนแปรผันเป็นรายละเอียดสำคัญของยูสเคสสายผลิตภัณฑ์ซอฟต์แวร์ ดังนั้นจึงนำส่วนแปรผันต่างๆ ไปสร้างเป็นคุณลักษณะของสายผลิตภัณฑ์ต่อไป

2.1.3.3 ตัวอย่างขั้นตอนการสร้างแบบจำลองความต้องการสายผลิตภัณฑ์ตามกระบวนการพลัส

ขั้นตอนการสร้างแบบจำลองสายผลิตภัณฑ์ตามกระบวนการพลัส สามารถสรุปเป็นกิจกรรมได้ดังรูปที่ 2.6 โดยเริ่มจากการออกแบบแผนภาพยูสเคส ในขั้นตอนนี้จะทำการกำหนดว่ายูสเคสแต่และตัวเป็นแบบเคอร์เนล เลือกได้ หรือทางเลือก ผลลัพธ์ของขั้นตอนนี้เราจะได้ยูสเคสทั้งหมดของสายผลิตภัณฑ์ตามกระบวนการพลัส จากนั้นทำการสร้างคำอธิบายยูสเคส ซึ่งในส่วนนี้จะสามารถกำหนดส่วนแปรผันทั้งหมดที่เกิดขึ้นจากคำอธิบายยูสเคสในแต่ละยูสเคสนั้นได้



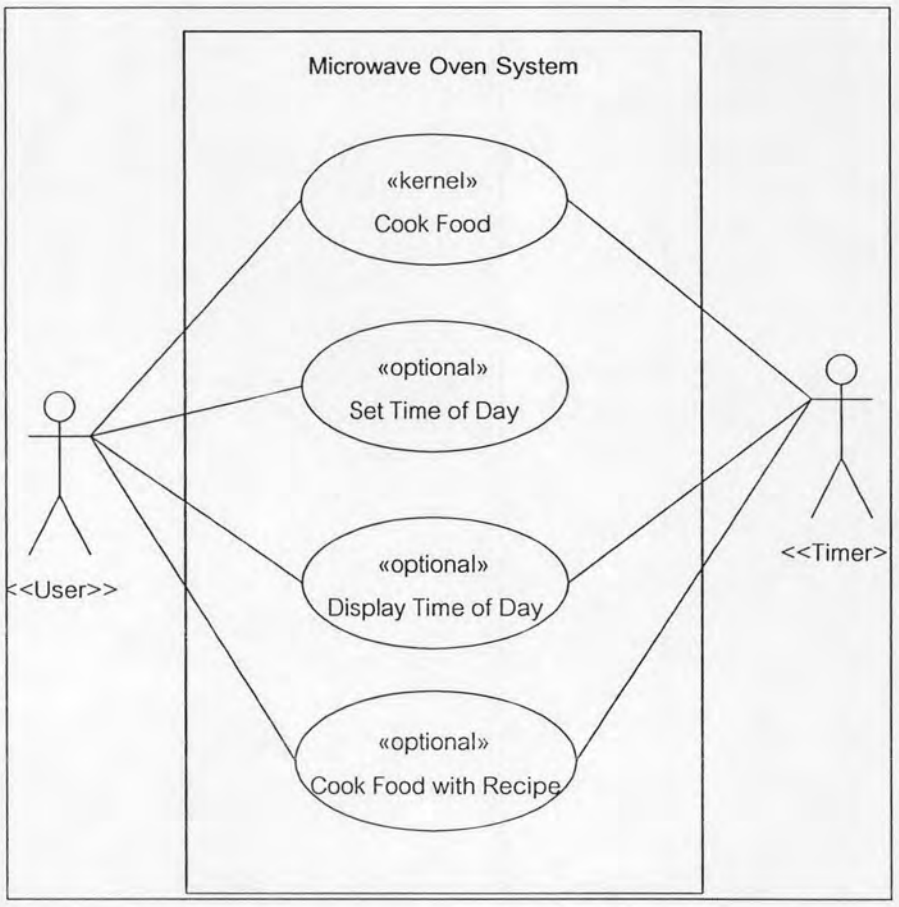
รูปที่ 2.6 ขั้นตอนการสร้างแบบจำลองความต้องการสายผลิตภัณฑ์ซอฟต์แวร์

ตัวอย่างที่จะนำเสนอมาจาก [2] เป็นตัวอย่างการผลิตเตาไมโครเวฟ ผู้ผลิตเตาไมโครเวฟต้องการสร้างสายผลิตภัณฑ์เตาไมโครเวฟเพื่อจำหน่ายทั่วโลก โดยเริ่มจากเตาไมโครเวฟที่มีฟังก์ชันการทำงานพื้นฐานไปสู่ผลิตภัณฑ์ที่มีฟังก์ชันการทำงานซับซ้อนมากขึ้น

เตาไมโครเวฟแบบเริ่มต้นประกอบไปด้วยปุ่มกำหนดเวลาทำอาหาร ปุ่มเริ่มต้นและปุ่มจบการทำงานที่มีลักษณะเป็นแป้นตัวเลข มีหน้าจอแสดงเวลาคงเหลือในการทำอาหาร มีอุปกรณ์ทำความร้อน และมีตัวรับรู้ (Sensor) ที่ฝาเตาเพื่อตรวจสอบการเปิด-ปิดเตา การทำอาหารจะทำงานก็ต่อเมื่อฝาเตาปิดและมีอาหารอยู่ในเตาเท่านั้นจากความต้องการข้างต้น

ผู้ออกแบบยูสเคสได้วิเคราะห์ว่าแบบจำลองยูสเคสเตาไมโครเวฟ มียูสเคส Cook Food เป็นยูสเคสเคอร์เนล เนื่องจากผลิตภัณฑ์เตาไมโครเวฟทุกรุ่นจะต้องมีการทำอาหาร มียูสเคส Set Time of Day, Display Time of Day เป็นยูสเคสแบบเลือกได้ ใช้กับเตาไมโครเวฟที่มีการเพิ่ม

คุณสมบัติในการกำหนดและแสดงเวลาปัจจุบัน และยูสเคส Cook Food with Recipe เป็นยูสเคสแบบเลือกได้ที่เพิ่มความสามารถของเตาให้สามารถเลือกทำอาหารได้ตามรายการอาหาร ตัวอย่างแบบจำลองยูสเคสสายผลิตภัณฑ์เตาไมโครเวฟดังรูปที่ 2.7



รูปที่ 2.7 แบบจำลองยูสเคสสายผลิตภัณฑ์เตาไมโครเวฟ [2]

ขั้นตอนการสร้างคำอธิบายยูสเคสและส่วนแปรผัน

ในแต่ละยูสเคสจะมีคำอธิบายยูสเคส (Use Case Description) เพื่อแสดงรายละเอียดและขั้นตอนการทำงาน คำอธิบายยูสเคสของสายผลิตภัณฑ์ซอฟต์แวร์มีรายละเอียดดังนี้

จากแผนภาพยูสเคสเราสามาถสร้างคำอธิบายยูสเคสและส่วนแปรผันได้ดังนี้

1) ยูสเคส Cook Food

(1) คำอธิบายยูสเคส

ยูสเคส Cook Food เป็นยูสเคสเคอร์เนลของระบบ ใช้แสดงขั้นตอนการทำอาหารของเตาไมโครเวฟ เริ่มจากผู้ใช้เปิดฝาเตา นำอาหารเข้าสู่เตา ปิดเตา กดปุ่มตั้งเวลา ป้อนเวลาโดย

กดที่แป้นตัวเลข กดปุ่มเริ่มทำงาน เมื่อครบตามเวลา เตาจะหยุดการทำงานและแสดงข้อความเสร็จการทำงาน ผู้ใช้นำอาหารที่ปรุงเสร็จออกจากเตา ปิดฝาเตา และระบบจะลบข้อความออกจากหน้าจอ

นอกจากนั้นยังมีการกำหนดทางเลือกของยูสเคสในกรณีที่การทำงานแตกต่างจากการเหตุการณ์ปรกติ ตัวอย่าง เช่น

- หากผู้ใช้กดปุ่มเริ่มต้นทำอาหารโดยที่ยังไม่ได้นำอาหารเข้าสู่เตา หรือยังไม่ได้ปิดฝา เตาจะไม่ทำงาน
- ขณะที่เตากำลังทำงาน หากผู้ใช้เปิดฝาเตา เตาจะหยุดทำงาน ผู้ใช้สามารถกดปุ่มยกเลิกเพื่อยกเลิกการทำงานและนำอาหารออกจากเตา หรือผู้ใช้สามารถกดปุ่มเริ่มทำงานเพื่อทำงานต่อ

ลักษณะของยูสเคสดังกล่าว สามารถแสดงเป็นคำอธิบายยูสเคสได้ดังนี้

ตารางที่ 2.2 **คำอธิบายยูสเคส** Cook Food [2]

Use Case Name:	Cook Food.
Reuse category:	Kernel.
Dependency:	
Summary:	Use puts food in oven, and microwave oven cooks food.
Actor:	User (primary), Timer (secondary).
Precondition:	Microwave oven is idle.
Description:	1. User opens the door, puts food in the oven, and closes the door. 2. User presses the **Cooking Time** button. 3. System prompts for cooking time. 4. User enters the cooking time on the numeric keypad and presses **Start**. 5. System starts cooking the food. 6. System continually displays the cooking time remaining. 7. Timer elapses and notifies the system. 8. System stops cooking the food and displays the end message.

ตารางที่ 2.2 **คำอธิบายยูสเคส** Cook Food [2] (**ต่อ**)

Use Case Name:	Cook Food.
Description:	9. User opens the door, removes the food from the oven, and closes the door. 10. System clears the displays.
Alternatives:	Line 1: User presses **Start** when the door is open. System does not start cooking. Line 4: User presses **Start** when the door is closed and the oven is empty. System does not start cooking. Line 4: User presses **Start** when the door is closed and the cooking time is equal to zero. System does not start cooking. Line 6: User opens door during cooking. System stops cooking. User removes food and presses **Cancel**, or user closes door and pressed **Start** to resume cooking. Line 6: User presses **Cancel**. System stops cooking. User may press **Start** to resume cooking. Alternatively, user may press **Cancel** again; system then cancels timer and clears display.

(2) ส่วนแปรผัน

ส่วนแปรผันเป็นจุดที่ระบบสามารถมีลักษณะการทำงานที่แตกต่างกันออกไป ส่วนแปรผันจะนำมาซึ่งคุณสมบัติต่างๆ ของซอฟต์แวร์ ส่วนแปรผันของยูสเคส Cook Food ประกอบด้วย

- ภาษาแสดงผล (Display Language) สามารถแสดงผลได้หลายภาษา ได้แก่ ภาษาอังกฤษ ฝรั่งเศส สเปน เยอรมัน และอิตาลี โดยมีภาษาอังกฤษเป็นค่าตั้งต้น
- ตัวรับรู้วัดน้ำหนัก (Weight Sensor) สามารถเลือกตัวรับรู้วัดน้ำหนักได้สองประเภท คือ ตัวรับรู้วัดน้ำหนักแบบบูลีนที่ตรวจจับได้เพียงมีหรือไม่มีน้ำหนัก หรือแบบแอนะล็อกที่สามารถแสดงน้ำหนักของวัตถุได้

หอสมุดกลาง สำนักงานวิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

- ตัวทำความร้อน (Heating Element) สามารถเลือกได้สองประเภท คือ ตัวทำความร้อนหนึ่งระดับ และตัวทำความร้อนหลายระดับ

- ระดับพลังความร้อน (Power Level) สามารถเลือกได้ 3 ระดับ คือ สูง (ค่าตั้งต้น) กลาง หรือต่ำ โดยสามารถเลือกระดับพลังความร้อนได้ก็ต่อเมื่อใช้ตัวทำความร้อนแบบหลายระดับ

- จอแสดงผล (Display Unit) สามารถเลือกจอแสดงผลได้สองแบบ คือ แบบหนึ่งบรรทัด และหลายบรรทัด

- ตัวเพิ่มเวลาหนึ่งนาที (Minute Plus) หากผู้ใช้งานกดปุ่มดังกล่าว ระบบจะเพิ่มเวลาในการทำอาหารอีกหนึ่งนาที

- อุปกรณ์ให้แสงสว่าง (Light) หากเลือกใช้อุปกรณ์ให้แสงสว่าง หลอดไฟจะเปิดในขณะที่มีการทำอาหารหรือเมื่อเปิดฝาเตา และหลอดไฟจะปิดเมื่อทำอาหารเสร็จหรือปิดฝาเตา

- จานหมุน (Turntable) หากใช้จานหมุน เมื่อเตาทำงานจะมีการหมุนจานหมุนเพื่อให้อาหารได้รับความร้อนอย่างทั่วถึง

- อุปกรณ์ส่งเสียง (Beeper) หากให้อุปกรณ์ส่งเสียง เมื่อทำอาหารเสร็จจะมีเสียงแจ้งการทำงาน

จากตัวอย่างข้างต้น ภาษาแสดงผล ตัวรับรู้วัดน้ำหนัก ตัวทำความร้อน จอแสดงผล เป็นส่วนแปรผันแบบทางเลือกจำเป็น ส่วนระดับพลังความร้อน ตัวเพิ่มเวลาหนึ่งนาที อุปกรณ์ให้แสงสว่าง จานหมุน อุปกรณ์ส่งเสียง เป็นส่วนแปรผันแบบเลือกใช้

ส่วนแปรผันของยูสเคส Cook Food สามารถแสดงเป็นคำอธิบายส่วนแปรผันได้ดังนี้

ตารางที่ 2.3 ส่วนแปรผันของยูสเคส Cook Food [2]

Name:	Display Language.
Type of functionality:	Mandatory alternative.
Line number(s):	3, 8.
Description of functionality:	There is a choice of language for displaying messages. The default is English. Alternative mutually exclusive languages are French, Spanish, German, or Italian.
Name:	Weight Sensor.
Type of functionality:	Mandatory alternative.
Line number(s):	1.
Description of functionality:	Cooking is prohibited if no item is present. The default is Boolean weight sensor, which indicates if item is present. Alternative mutually exclusive variation is analog weight sensor. Analog weight sensor provides weight of item.
Name:	Heating Element.
Type of functionality:	Mandatory alternative.
Line number(s):	5.
Description of functionality:	Default is a one-level heating element: high power level. Alternative is a multi-level heating element, with high, medium, and low power levels.
Name:	Power Level.
Type of functionality:	Optional.
Line number(s):	2.
Description of functionality:	Microwave oven has power level buttons for high power (default), medium, and low. User may select the power level. Requires multi-level heating element as prerequisite.

ตารางที่ 2.3 ส่วนแปรผันของยูสเคส Cook Food [2] (ต่อ)

Name:	Display Unit.
Type of functionality:	Mandatory alternative.
Line number(s):	3, 4, 6, 8, 10.
Description of functionality:	Default is one-line display unit. Alternative is multi-line display unit.
Name:	Minute Plus.
Type of functionality:	Optional.
Line number(s):	2, 6.
Description of functionality:	User may press **Minute Plus**, which results in one minute being added to the cooking time. If the cooking time was previously zero, cooking is started.
Name:	Light.
Type of functionality:	Optional.
Line number(s):	1, 5, 8, 9.
Description of functionality:	If light option is selected, lamp is switched on for duration of cooking and when the door is open. Light is switched off when door is closed and when cooking stops.
Name:	Turntable.
Type of functionality:	Optional.
Line number(s):	5, 8.
Description of functionality:	If turntable option is selected, turntable rotates for duration of cooking.
Name:	Beeper.
Type of functionality:	Optional.
Line number(s):	8.
Description of functionality:	If beeper option is selected, system activates the beeper when cooking stops.

2) ยูสเคส Set Time of Day

(1) คำอธิบายยูสเคส

ยูสเคส Set Time of Day เป็นยูสเคสเลือกใช้ ใช้เมื่อมีการเลือกจอแสดงผลแบบหลายบรรทัด ลักษณะขั้นตอนการทำงาน คือ ผู้ใช้กดปุ่มตั้งเวลา ระบบแสดงสัญลักษณ์พร้อมรับข้อมูล ผู้ใช้ป้อนเวลาในรูปแบบชั่วโมงและนาทีด้วยการกดแป้นตัวเลข ระบบจัดเก็บและแสดงผลเวลา ตัวนับเวลา (ผู้กระทำ Timer) จะเริ่มต้นนับเวลาเมื่อผู้ใช้กดปุ่มเริ่มทำงาน

ทางเลือกของยูสเคส Set Time of Day มี 2 กรณี คือ

- หากเตาไมโครเวฟกำลังทำงานอยู่ ระบบจะไม่รับคำสั่งของผู้ใช้
- ผู้ใช้สามารถกดปุ่มยกเลิกการตั้งเวลาได้หากป้อนเวลาผิด

ลักษณะของคำอธิบายยูสเคสเป็นดังนี้

ตารางที่ 2.4 คำอธิบายยูสเคส Set Time of Day [2]

Use Case Name:	Set Time of Day.
Reuse category:	Optional.
Dependency:	Variation point in **Cook Food** use case: at **Display Unit** Variation point, select **Multi-line Display.**
Summary:	User sets time-of-day clock.
Actor:	User.
Precondition:	Microwave oven is idle.
Description:	1. User presses **Time of Day (TOD)** button. 2. System prompts for time of day. 3. User enters the time of day (in hours and minutes) on the numeric keypad. 4. System stores and displays the entered time of day. 5. User press **Start.** 6. System starts the time-of-day timer.
Alternatives:	**Line 1, 3:** If the oven is busy, the system will not accept the user input. **Line 5:** The user may presses **Cancel** if the incorrect time was entered. The system clears the display.

(2) ส่วนแปรผัน

ส่วนแปรผันของยูสเคส Set Time of Day มีหนึ่งจุด คือ การเลือกแบบเวลา (12/24 Hour Clock) ผู้ใช้สามารถเลือกตั้งเวลาได้สองแบบ คือ แบบ 12 ชั่วโมง (ค่าตั้งต้น) หรือ แบบ 24 ชั่วโมง

ตารางที่ 2.5 ส่วนแปรผันของยูสเคส Set Time of Day [2]

Name:	12/24 Hour Clock.
Type of functionality:	Mandatory alternative
Line number(s):	4.
Description of functionality:	TOD display is either 12-hour clock (default) or 24-hour clock.
Post Condition:	TOD clock has been set.

3) ยูสเคส Display Time of Day

(1) คำอธิบายยูสเคส

ยูสเคส Display Time of Day เป็นยูสเคสเลือกใช้ ใช้เมื่อมีการเลือกจอแสดงผลแบบหลายบรรทัด ขั้นตอนการทำงาน คือ ตัวนับเวลา (ผู้กระทำ Timer) แจ้งระบบทุกๆ ครั้งที่เวลาเพิ่มขึ้น 1 วินาที ระบบจะทำการเพิ่มเวลาเมื่อได้รับแจ้ง โดยทำการปรับนาทีและชั่วโมงตามเวลาที่เปลี่ยนแปลง และแสดงผลเวลาทางหน้าจอ

ตารางที่ 2.6 คำอธิบายยูสเคส Display Time of Day [2]

Use Case Name:	Display Time of Day.
Reuse category:	Optional.
Dependency:	Variation point in **Cook Food** use case: at **Display Unit** Variation point, select **Multi-line Display**.
Summary:	System display time-of-day.
Actor:	Timer (primary actor), User (secondary actor).
Precondition:	TOD clock has been set.
Description:	1. Timer notifies system that one second has elapsed. 2. System increments TOD clock every second, adjusting for minutes and hours. 3. System updates the display with time of day every minute.

(2) ส่วนแปรผัน

ส่วนแปรผันของยูสเคส Display Time of Day มีจุดเดียว คือ รูปแบบการแสดงเวลา (12/24 Hour Clock) ผู้ใช้สามารถเลือกการแสดงผลแบบ 12 หรือ 24 ชั่วโมง

ตารางที่ 2.7 ส่วนแปรผันของยูเคส Display Time of Day [2]

Name:	12/24 Hour Clock.
Type of functionality:	Mandatory alternative.
Line number(s):	2.
Description of functionality:	TOD display is either 12-hour clock (default) or 24-hour clock.
Post Condition:	TOD clock has been updated (every second) and time of day displayed (every minute).

4) ยูสเคส Cook Food with Recipe

(1) คำอธิบายยูสเคส

ยูสเคส Cook Food with Recipe เป็นยูสเคสแบบเลือกใช้ แสดงขั้นตอนการทำอาหารแบบสามารถเลือกวิธีทำอาหาร ใช้งานได้ก็ต่อเมื่อมีการใช้จอแสดงผลแบบหลายบรรทัด เครื่องทำความร้อนแบบหลายระดับ และใช้ตัวรับรู้วัดน้ำหนักแบบแอนะล็อก

ขั้นตอนการทำงาน คือ ผู้ใช้เปิดฝาเตา นำอาหารเข้าสู่เตา ปิดเตา จากนั้นผู้ใช้กดเลือกวิธีทำอาหารจากปุ่มรายการวิธีทำอาหาร ระบบแสดงชื่อวิธีทำอาหาร ระดับพลังงาน เวลาทำอาหารเริ่มต้น และเวลาต่อน้ำหนักอาหาร ผู้ใช้กดปุ่มเริ่มทำงาน ระบบจะทำการคำนวณเวลาทำงานทั้งหมด โดยมีค่าเท่ากับเวลาเริ่มต้น + (เวลาต่อน้ำหนัก * น้ำหนักของอาหาร) ในขณะทำงาน ระบบจะแสดงเวลาคงเหลือจนกระทั่งสิ้นสุดเวลาทำงาน จากนั้นระบบจะแสดงข้อความเสร็จงาน ผู้ใช้เปิดฝาเตา นำอาหารออกจากเตา ปิดฝาเตา ระบบลบข้อความแสดงผลเป็นการจบขั้นตอนการทำงาน

ทางเลือกของยูสเคสมีหลายกรณีดังนี้

- หากผู้ใช้กดปุ่มเริ่มต้นทำอาหารโดยที่ยังไม่ได้นำอาหารเข้าสู่เตา หรือยังไม่ได้ปิดฝา เตาจะไม่ทำงาน
- หากผู้ใช้กดปุ่มเริ่มต้นทำอาหารโดยที่ยังไม่ได้เลือกวิธีทำอาหาร เตาจะไม่ทำงาน

- ขณะที่เตากำลังทำงาน หากผู้ใช้เปิดฝาเตา เตาจะหยุดทำงาน ผู้ใช้สามารถกดปุ่มยกเลิกเพื่อยกเลิกการทำงานและนำอาหารออกจากเตา หรือผู้ใช้สามารถกดปุ่มเริ่มทำงานเพื่อทำงานต่อ
- ขณะที่เตากำลังทำงาน หากผู้ใช้กดปุ่มยกเลิก ระบบจะหยุดทำงาน ผู้ใช้สามารถสั่งให้ทำงานต่อโดยกดปุ่มเริ่มทำงาน หรือยกเลิกการทำงานโดยกดปุ่มยกเลิกอีกครั้ง
- หากวิธีทำอาหารมีขั้นตอนมากกว่าหนึ่งขั้น เมื่อทำงานเสร็จในขั้นแรก ระบบจะทำการคำนวณเวลาและระดับพลังงานที่ต้องใช้ และเริ่มทำงานในขั้นตอนถัดไป

คำอธิบายยูสเคส Cook Food with Recipe มีลักษณะดังตารางที่ 2.8

ตารางที่ 2.8 คำอธิบายยูสเคส Cook Food with Recipe [2]

Use Case Name:	Cook Food with Recipe.
Reuse category:	Optional.
Dependency:	Variation point in **Cook Food** use case: at **Display Unit** variation point, select **Multi-line Display**; at **Heating Element** variation point, select **Multi-level Heater**; at **Weight Sensor** variation point, select **Analog Weight** Sensor.
Summary:	User puts food in microwave oven and cooks food, using recipe.
Actor:	User (primary actor), Timer (secondary actor).
Precondition:	Microwave oven is idle.
Description:	1. User opens the door, puts food in the oven, and closes the door. 2. User presses the desired recipe button from the recipe buttons on the keypad. 3. System displays the recipe name. Recipe has name, power level (p), fixed time (t1), and time per unit weight (t2). 4. User presses the **Start** button. 5. System starts cooking the food for a time given by the following equation: Cooking Time = t1 + w * t2, where t1 and t2 are times specified in the recipe and w is the weight of the time, and the power level p is specified in the recipe. 6. System continually displays the cooking time remaining. 7. Timer elapses and notifies the system. 8. System stops cooking the food and displays the end message. 9. User opens the door, removes the food from the oven, and closes the door. 10. System clears the display.

ตารางที่ 2.8 **คำอธิบายยูสเคส** Cook Food with Recipe [2] (**ต่อ**)

Use Case Name:	Cook Food with Recipe.
Alternative:	Line 1: User presses **Start** when the door is open. System does not start cooking. Line 4: User presses **Start** when the door is closed and the oven is empty. System does not start cooking. Line 4: User presses **Start** when the door is closed and a recipe has not been chosen. System does not start cooking. Line 4: Use press **Cancel**. System cancels recipe and clears display. Line 6: User opens the door during cooking. System stops cooking.. User removes food and presses **Cancel**, or user closes the door and presses **Start** to resume cooking. Line 6: User presses **Cancel**. System stops cooking. User may press **Start** to resume cooking. Alternatively, user may press **Cancel** again; system then cancels the recipe and clears the display. Line 7: If the recipe has more than one step, system completes one step, cooking food for the computed time and specified power level, and then proceeds to the next step.

(2) ส่วนแปรผัน

ส่วนแปรผันของยูสเคส Cook Food with Recipe คล้ายกันกับยูสเคส Cook Food คือ

- ภาษาแสดงผล (Display Language) สามารถแสดงผลได้หลายภาษา ได้แก่ ภาษาอังกฤษ ฝรั่งเศส สเปน เยอรมัน และอิตาลี โดยมีภาษาอังกฤษเป็นค่าตั้งต้น
- อุปกรณ์ให้แสงสว่าง (Light) หากเลือกใช้อุปกรณ์ให้แสงสว่าง หลอดไฟจะเปิดในขณะที่มีการทำอาหารหรือเมื่อเปิดฝาเตา และหลอดไฟจะปิดเมื่อทำอาหารเสร็จหรือปิดฝาเตา

- จานหมุน (Turntable) หากใช้จานหมุน เมื่อเตาทำงานจะมีการหมุนจานหมุนเพื่อให้อาหารได้รับความร้อนอย่างทั่วถึง

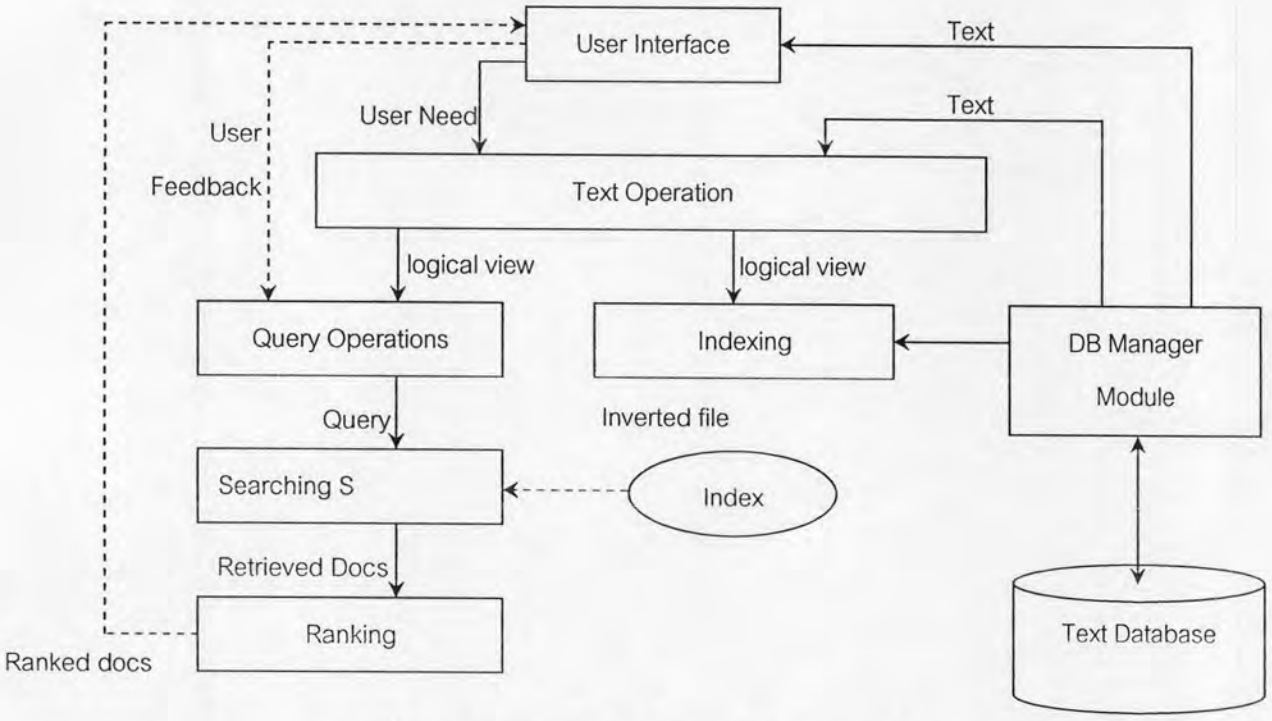
- อุปกรณ์ส่งเสียง (Beeper) หากให้อุปกรณ์ส่งเสียง เมื่อทำอาหารเสร็จจะมีการส่งเสียงแจ้ง

ตารางที่ 2.9 ส่วนแปรผันของยูสเคส Cook Food with Recipe [2]

Name:	Display Language.
Type of functionality:	Mandatory alternative.
Line number(s):	3, 8.
Description of functionality:	There is a choice of language for displaying messages. The default is English. Alternative mutually exclusive languages are French, Spanish, German, or Italian.
Name:	Light.
Type of functionality:	Optional.
Line number(s):	1, 5, 8, 9.
Description of functionality:	If light option is selected, lamp is switched on for duration of cooking and when the door is open. Light is switched off when the door is closed and when cooking stops.
Name:	Turntable.
Type of functionality:	Optional.
Line number(s):	5, 8.
Description of functionality:	If turntable option is selected, turntable rotates for duration of cooking.
Type of functionality:	Optional.
Line number(s):	8.
Description of functionality:	If beeper option is selected, system activates the beeper when cooking stops.
Post Condition:	Microwave oven has coked the food using the recipe.

2.1.4 การค้นคืนสารสนเทศ (Information Retrieval)

การค้นคืนสารสนเทศ[7] เป็นกระบวนการที่ว่าด้วยเรื่องการรวบรวม จัดเก็บ และเข้าถึงข้อมูล เพื่อให้ผู้ใช้สามารถค้นคืนเอกสารได้ตรงกับความต้องการให้มากที่สุด ในการจัดเก็บเอกสารจะมีการสร้างดรรชนีเพื่อที่จะช่วยในการค้นคืนเอกสารได้รวดเร็วยิ่งขึ้น ในการค้นคืนเอกสารเมื่อผู้ใช้ทำการป้อนข้อคำถาม (Query) จะมีการคำนวณค่าความคล้ายกัน (Similarity) ระหว่างข้อคำถามและเอกสาร เพื่อให้ได้เอกสารที่ตรงกับความต้องการของผู้ใช้มากที่สุด จากที่กล่าวมาจะแสดงกระบวนการค้นคืนสารสนเทศได้ดังรูปที่ 2.11



รูปที่ 2.8 กระบวนการจัดเก็บและค้นคืนเอกสาร [8]

2.1.4.1 การรวบรวมเอกสาร

ผู้ที่มีหน้าที่ทำการรวบรวมเอกสารที่เกี่ยวข้องจะกระทำการผ่านทางส่วนต่อประสานผู้ใช้ (User Interface) เพื่อเป็นฐานข้อมูลในการค้นคืนเอกสาร ซึ่งลักษณะการจัดเก็บจะแบ่งเป็นประเภทดังนี้

1) การจัดเก็บเชิงเส้น (Linear) เป็นลักษณะการจัดเก็บแบบเรียงกันเป็นสาย เมื่อทำการค้นคืนข้อมูลจะเริ่มจากต้นสายไปยังท้ายสุด ทำให้เกิดกรณีที่ข้อมูลอยู่ท้ายสุด จะเสียเวลาในการค้นคืนมากที่สุด

2) การจัดเก็บโดยลำดับ (Sequential) เป็นลักษณะการจัดเก็บแบบเรียงกันเป็นสาย คล้ายแบบ linear แต่จะมีการจัดลำดับข้อมูล เมื่อมีข้อมูลเข้าใหม่จะนำมาเปรียบเทียบเพื่อหาลำดับที่เหมาะสมในการนำข้อมูลใหม่เข้าไปจัดเก็บ เป็นการเพิ่มประสิทธิภาพในการค้นหาได้เร็วขึ้น

3) การจัดเก็บแบบดรรชนี (Index) ลักษณะการจัดเก็บจะมีการเลือกคำสำคัญเพื่อนำมาเป็นดรรชนีที่ช่วยในการเข้าถึงข้อมูล ลักษณะการจัดเก็บแบบ index แบ่งได้เป็น 2 แบบ คือ

(1) แบบแฟ้มตรง (Directed File) เป็นการพิจารณาตัวเอกสารเป็นหลัก โดยนำเอาเอกสารมาแยกคำสำคัญออกมา

(2) แบบแฟ้มผกผัน (Inverted File) เป็นการพิจารณาจากคำสำคัญเป็นหลัก สลับกับแบบแฟ้มตรง (directed file)

โดยเอกสารที่ถูกจัดเก็บจะอยู่ในฐานข้อมูล (Text Database) โดยมีตัวจัดการฐานข้อมูล (DB Manager Module) ช่วยในการสร้างดรรชนีของเอกสารเหล่านั้นโดยผ่านตัวดำเนินการข้อความ (Text Operation) ซึ่งทำหน้าที่ในการวิเคราะห์คำ (Lexical Analysis) หาสต็อปเวิร์ด (Stop Words) หารากศัพท์ (Stemming)

2.1.4.2 การสร้างดรรชนี

เป็นการเลือกคำสำคัญของตัวเอกสาร ซึ่งเป็นขั้นตอนสำคัญในการค้นคืนเอกสาร เนื่องจากการเลือกคำสำคัญจะมีผลโดยตรงในการค้นคืนเอกสาร ซึ่งดรรชนีที่ได้จะมีการคำนวณค่าน้ำหนักดรรชนี เพื่อเป็นการแสดงถึงความสำคัญของดรรชนีตัวนั้นๆ โดยมีการคำนวณน้ำหนักดรรชนีได้ดังนี้

$$w_{ik} = tf_{ik} * idf_k \qquad (1)$$

เมื่อ w_{ik} แทน น้ำหนักของดรรชนี k ในเอกสาร i

tf_{ik} แทน ความถี่ของดรรชนี k ที่ปรากฏทั้งหมดในเอกสาร i

idf_k แทน ค่าความถี่แบบผกผันของเอกสารของดรรชนี k

โดยที่

$$idf_k = \log\left(\frac{N}{n_k}\right) \qquad (2)$$

เมื่อ N แทน จำนวนดรรชนีทั้งหมด

n_k แทน จำนวนเอกสารที่ปรากฏดรรชนี k

2.1.4.3 **การค้นคืนเอกสาร**

กระบวนการในการค้นคืนเอกสารจะเริ่มจากผู้ใช้ทำการป้อนคำหรือข้อความที่เป็นคำสำคัญในการค้นคืนของผู้ใช้ (User Need) จากนั้นข้อมูลนำเข้าของผู้ใช้จะผ่านตัวดำเนินการข้อความ (Text Operation) ที่ทำหน้าที่ดังกล่าวข้างต้น ข้อมูลนำเข้าที่ผ่านกระบวนการดังกล่าวแล้วจะมีตัวดำเนินการข้อคำถาม (Query Operation) ในการแปลงข้อมูลนำเข้า ของผู้ใช้ให้เป็นข้อคำถาม (Query) ที่สามารถนำไปคำนวณหาค่าความคล้ายกันกับเอกสารต่างๆ ในฐานข้อมูลต่อไป เมื่อได้ผลลัพธ์ซึ่งเป็นเอกสารที่ตรงกับความต้องการแล้ว จะทำการจัดเรียงเอกสารที่ได้นั้น เพื่อให้ได้เอกสารที่ใกล้เคียงกับความต้องการของผู้ใช้อยู่ในลำดับต้น

สำหรับในส่วนการคำนวณค่าความคล้ายกัน เมื่อรับข้อคำถามที่ผู้ใช้ทำการป้อนแล้ว จะทำการคำนวณค่าความคล้ายกันระหว่างเอกสารกับข้อคำถาม ในวิทยานิพนธ์นี้จะนำหลักการของแบบจำลองเวกเตอร์สเปซมาใช้ในการคำนวณค่าความคล้ายกัน โดยแบบจำลองเวกเตอร์สเปซเป็นแบบจำลองทางคณิตศาสตร์ที่ใช้เวกเตอร์แทนตัวเอกสารและข้อคำถามของผู้ใช้ หากเวกเตอร์ของเอกสารใดมีทิศทางใกล้เคียงกับเวกเตอร์ข้อคำถาม แสดงว่าเอกสารนั้นมีเนื้อหาใกล้เคียงกับข้อคำถาม การคำนวณค่าความคล้ายกันระหว่างเอกสารและข้อคำถามจะเป็นการคูณระหว่างเวกเตอร์ ซึ่งผลลัพธ์ที่ได้จะเป็นค่าตัวเลขที่บอกว่าเวกเตอร์ของตัวเอกสารมีทิศทางใกล้เคียงกับเวกเตอร์ข้อคำถามมากน้อยแค่ไหน การหาค่าความคล้ายกันแสดงได้ดังสมการที่ 2

$$SIM(Q, DOC_i) = \frac{\sum_{k=1}^{N} (w_{q,k} \cdot w_{i,k})}{\sqrt{\sum_{k=1}^{N} (w_{q,k})^2 \cdot \sum_{k=1}^{N} (w_{i,k})^2}} \qquad (3)$$

เมื่อ $SIM(Q, DOC_i)$ แทน ค่าคะแนนความคล้ายกันของข้อคำถาม Q และเอกสาร DOC_j

$w_{i,k}$ แทน ค่าน้ำหนักของดรรชนีที่ k ในเอกสารลำดับที่ i

$w_{q,k}$ แทน ค่าน้ำหนักของดรรชนีที่ k ในข้อคำถาม

$SIM(Q, DOC_i)$ แทน ค่าความคล้ายกันระหว่างข้อคำถามและเอกสารลำดับที่ i

N แทน จำนวนเอกสารทั้งหมด

2.1.4.4 **การประเมินประสิทธิภาพ**

ภายหลังจากการค้นคืนสารสนเทศ จะทำการคำนวณค่าความแม่นยำ และค่าระลึก เพื่อประเมินประสิทธิภาพในการค้นคืนของระบบ ว่าผลลัพธ์ที่ได้จากการค้นคืนเป็นไปตามความต้องการของผู้ใช้ได้มากน้อยเพียงใด สามารถคำนวณค่าความแม่นยำ และค่าระลึกได้จากสมการต่อไปนี้

$$\text{ค่าความแม่นยำ (precision)} = \frac{|Ra|}{|A|} \quad (4)$$

เมื่อ $|Ra|$ แทน จำนวนเอกสารที่เกี่ยวข้องและค้นคืนออกมาได้

$|A|$ แทน จำนวนเอกสารที่ถูกค้นคืนทั้งหมด

$$\text{ค่าระลึก (recall)} = \frac{|Ra|}{|R|} \quad (5)$$

เมื่อ $|Ra|$ แทน จำนวนเอกสารที่เกี่ยวข้องและค้นคืนออกมาได้

$|R|$ แทน จำนวนเอกสารที่เกี่ยวข้องทั้งหมดในฐานข้อมูล

นอกจากการประเมินด้วย 2 มาตรวัดที่กล่าวมาแล้ว ค่าเฉลี่ยฮาร์โมนิคก็เป็นอีกหนึ่งมาตรวัดโดยมีการคำนวณจากค่าความแม่นยำและค่าระลึก ซึ่งค่าที่ได้จะอยู่ในช่วงตั้งแต่ 0 ถึง 1 หากค่าเป็น 0 หมายถึงเอกสารที่ค้นคืนได้ไม่ตรงกับความต้องการ หากมีค่าเป็น 1 หมายถึงเอกสารที่ค้นคืนได้ทั้งหมดตรงกับความต้องการ การคำนวณค่าเฉลี่ยฮาร์โมนิคได้จากสมการต่อไปนี้

$$\text{ค่าเฉลี่ยฮาร์โมนิค } F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}} \quad (6)$$

เมื่อ $F(j)$ แทนค่าเฉลี่ยฮาร์โมนิคของ $r(j)$ และ $P(j)$

$r(j)$ แทน ค่าระลึกของเอกสารลำดับที่ j

$P(j)$ แทน ค่าความแม่นยำของเอกสารลำดับที่ j

j แทน เอกสารลำดับที่ j

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 การค้นคืนยูสเคสโดยใช้การคำนวณความคล้ายกันของพจน์และโครงสร้างของยูสเคส (Use Case Retrieval Using Terms and Use Case Structure Similarity Computation โดย อัคเดช อุดมชัยพร)

งานวิจัยนี้ [9] นำเสนอวิธีการนำยูสเคสกลับมาใช้ใหม่ โดยในแต่ละยูสเคสจะมีเอกสารคำอธิบายยูสเคสที่เขียนด้วยภาษาธรรมชาติ ซึ่งเอกสารคำอธิบายยูสเคสนี้สามารถที่จะนำกลับมาใช้ใหม่ได้ เพื่อเป็นการช่วยลดงบประมาณและระยะเวลาในการพัฒนาซอฟต์แวร์ โดยงานวิจัยนี้เสนอระบบการจัดเก็บและค้นคืนยูสเคสโดยพิจารณาจากความคล้ายกันของเอกสารคำอธิบายยูสเคสกับชุดข้อมูลนำเข้าจากผู้ใช้งาน วิธีการนำยูสเคสกลับมาใช้ใหม่มีวิธีการหลักอยู่ 2 ส่วน คือ ส่วนของการจัดเก็บยูสเคสและส่วนของการค้นคืนยูสเคส ในส่วนของการจัดเก็บจะประกอบไปด้วย 3 ขั้นตอน คือ เริ่มจากการสร้างดรรชนี เพื่อใช้ในการระบุถึงเอกสารคำอธิบายยูสเคส ขั้นตอนต่อไปเป็นการสร้างพจนานุกรมคำพ้อง (Thesaurus) เพื่อรวบรวมและจัดหมวดหมู่ความคล้ายกันของชุดดรรชนีที่ได้ หลังจากนั้นจะทำการกำหนดน้ำหนักให้ดรรชนี เป็นการกำหนดค่าความสำคัญให้กับแต่ละดรรชนี เพื่อช่วยให้ผลลัพธ์ของการค้นคืนแม่นยำมากขึ้น ต่อไปในส่วนของการค้นคืนยูสเคส ผู้ใช้งานจะใส่ข้อคำถามเข้ามา จากนั้นจะทำการสร้างดรรชนีจากข้อคำถามที่เข้ามานั้น แล้วนำชุดของดรรชนีเหล่านั้นมาเปรียบเทียบความคล้ายกันกับดรรชนีของแต่ละเอกสารคำอธิบายยูสเคสในฐานข้อมูล ความคล้ายกันของดรรชนีจะถูกคำนวณเป็นคะแนนความคล้ายกันระหว่างดรรชนีที่สร้างจากข้อคำถามเทียบกับดรรชนีที่สร้างจากเอกสารคำอธิบายยูสเคส การค้นคืนยูสเคสจะมีการประเมินความถูกต้องแม่นยำของผลลัพธ์ที่ได้ โดยคำนวณจากค่าระลึก (Recall) ค่าความแม่นยำ (Precision) ค่าเฉลี่ยฮาร์โมนิค (Harmonic Mean)

ในงานวิจัยนี้มีข้อดี คือ วิธีการนำยูสเคสกลับมาใช้ใหม่มีขั้นตอนการทำงานแบบอัตโนมัติ ผู้ใช้สามารถใส่ข้อคำถามเพื่อทำการค้นคืนได้ง่าย โดยการค้นคืนจะมีลักษณะที่เป็นทั้งแบบโครงสร้าง และใช้คำสำคัญ ซึ่งมีข้อดีในการลดเวลาในการค้นคืนให้น้อยลงได้ นอกจากนี้ยังสามารถให้ผู้ใช้กำหนดค่าน้ำหนักเพื่อให้ความสำคัญในแต่ละส่วนของยูสเคสที่แตกต่างกัน ตามความสำคัญที่ผู้ใช้ต้องการได้

งานวิจัยนี้นำเสนอเฉพาะการจัดเก็บและค้นคืนยูสเคสเพื่อนำกลับมาใช้ใหม่ โดยผลลัพธ์ที่ได้จากการค้นคืนจะได้เอกสารคำอธิบายยูสเคสตามที่ผู้ใช้ใส่ข้อคำถามเข้าไป ซึ่งในแต่ละยูสเคสไม่ได้มีการระบุคุณลักษณะที่จะนำกลับมาใช้ใหม่ ดังนั้นในงานวิทยานิพนธ์นี้จึงจะเพิ่มเติม

I24545703

งานในการระบุคุณลักษณะในแต่ละยูสเคส เพื่อช่วยให้ผู้ใช้สามารถพิจารณาในแต่ละยูสเคส ที่จะนำกลับมาใช้ใหม่ได้อย่างมีประสิทธิภาพมากยิ่งขึ้น

2.2.2 **การค้นคืนยูสเคสด้วยการจำแนกประเภทยูสเคสและการค้นคืนย้อนกลับจากผู้ใช้** (Use Case Retrieval Using Use Case Clustering and User Relevance Feedback **โดย สุดาทิพย์ สุขสอาด**)

จากงานวิจัย [9] ที่นำเสนอวิธีการนำยูสเคสกลับมาใช้ใหม่ ซึ่งมีระบบการจัดเก็บและค้นคืนยูสเคสโดยพิจารณาจากความคล้ายกันของเอกสารคำอธิบายยูสเคสกับชุดข้อมูลนำเข้าจากผู้ใช้งาน ในส่วนของงานวิจัยนี้ [10] ได้มีการพัฒนาเพิ่มเติมในส่วนการจัดองค์ประกอบให้กับการจัดเก็บเอกสารเพื่อให้ง่ายต่อการจัดการ และการเข้าถึงตัวเอกสาร โดยใช้วิธีในการจัดกลุ่มเอกสาร (Document Clustering) ซึ่งเป็นวิธีการที่ช่วยให้ระบบการค้นคืนมีประสิทธิภาพมากขึ้น โดยข้อคำถามที่ผู้ใช้ป้อนเข้าสู่ระบบจะถูกนำมาเปรียบเทียบกับแต่ละกลุ่มของเอกสาร แทนที่จะเปรียบเทียบทีละเอกสาร ซึ่งนอกจากจะทำให้เข้าถึงกลุ่มเอกสารที่มีความใกล้เคียง หรือคล้ายกับเอกสารที่ผู้ใช้ต้องการได้อย่างรวดเร็วแล้ว ยังสามารถช่วยให้ระบบสามารถค้นหาเอกสารที่ตรงกับความต้องการของผู้ใช้เพิ่มมากขึ้นด้วย และในงานวิจัยนี้ยังมีการใช้วิธีการป้อนกลับของผู้ใช้ (User Feedback) ในการคำนวณข้อคำถามใหม่ที่มีความเหมาะสมมากยิ่งขึ้น

เช่นเดียวกันจากงานวิจัย [9] คือ มีการนำเสนอการจัดเก็บและค้นคืนยูสเคส พร้อมทั้งได้มีการเพิ่มเติมในเรื่องการจัดกลุ่มเอกสาร (Document Clustering) เพื่อให้ง่ายต่อการจัดการ และการเข้าถึงตัวเอกสารได้ดียิ่งขึ้น แต่ก็ยังไม่มีในส่วนของการระบุคุณลักษณะที่จะนำกลับมาใช้ใหม่ตามกระบวนการทางวิศวกรรมสายผลิตภัณฑ์ซอฟต์แวร์บนพื้นฐานยูเอ็มแอลหรือพลัส (Product Line UML-Based Software Engineering: PLUS) ที่กล่าวมาข้างต้น ดังนั้นในงานวิทยานิพนธ์นี้จึงจะเพิ่มเติมงานในส่วนการระบุคุณลักษณะในแต่ละยูสเคส เพื่อให้เกิดประสิทธิภาพมากยิ่งขึ้น

2.2.3 **การดึงข้อมูลเพื่อสร้างยูสเคสสำหรับสายผลิตภัณฑ์** (Elicitation of Use Case for Products Lines **โดย** A. Fantechi, S. Gnesi, I. John, G.Lami, J.Dorr)

งานวิจัยนี้ [11] นำเสนอวิธีการในการดึงข้อมูลและวางข้อกำหนดในการสร้างยูสเคสสำหรับสายผลิตภัณฑ์ซอฟต์แวร์จากเอกสารของระบบ เช่น ข้อกำหนดความต้องการ (Requirement Specification) คู่มือผู้ใช้ (User Manual) หรือเอกสารการออกแบบ (Design Document) เป็นการสร้างสายผลิตภัณฑ์จากระบบหรือซอฟต์แวร์ที่มีอยู่แล้ว โดยเสนอแนวคิดที่ว่าความต้องการของสายผลิตภัณฑ์ควรประกอบไปด้วยสองส่วนคือส่วนคงที่ (Constant Part) และ

ส่วนแปรผัน (Variable Part) ส่วนคงที่เป็นความต้องการที่ทุกผลิตภัณฑ์ในสายผลิตภัณฑ์มีเหมือนกัน ส่วนแปรผันเป็นความต้องการที่มีเฉพาะในแต่ละผลิตภัณฑ์ ส่วนแปรผันสามารถแบ่งออกได้เป็น 3 ประเภทคือแบบทางเลือก (Alternative) เป็นการเลือกความต้องการที่สนใจจากกลุ่มความต้องการ แบบกำหนดค่า (Parametric) เป็นความต้องการที่ต้องมีการกำหนดค่าให้ก่อนใช้งาน แบบเลือกได้ (Optional) เป็นความต้องการที่สามารถเลือกหรือไม่เลือกใช้ การดึงความต้องการ (Elicitation) จากเอกสารที่มีในระบบอยู่บนหลักการว่าทำอย่างไรจึงจะสามารถดึงความรู้ (Knowledge) ของโดเมนออกมาจากเอกสาร และทำอย่างไรที่จะแปลงความรู้เหล่านั้นให้อยู่ในรูปแบบของสายผลิตภัณฑ์ซอฟต์แวร์

งานวิจัยนี้เป็นการทำรีเวอร์สเอ็นจิเนียริง (Reverse Engineering) โดยนำเอกสารคู่มือผู้ใช้ระบบมาวิเคราะห์และสร้างยูสเคสสายผลิตภัณฑ์ซอฟต์แวร์ โดยกำหนดขั้นตอนในการทำงานออกเป็น 3 ขั้นคือ

1) เตรียมการ (Preparation) เป็นขั้นตอนในการรวบรวมเอกสารทั้งหมดที่เกี่ยวข้องกับระบบ

2) ค้นหา (Search) เป็นขั้นตอนในการวิเคราะห์เนื้อหาในเอกสารเพื่อค้นหาว่าคำ วลี หรือประโยคใดสามารถนำมาใช้เป็นส่วนหนึ่งของยูสเคส เช่น เป็นผู้กระทำ เป็นชื่อยูสเคส หรือเป็นเงื่อนไขเริ่มต้น เมื่อพบคำ วลี หรือประโยคดังกล่าว ให้ทำการสร้างสัญลักษณ์ (Tag) ลงไปในเอกสารและระบุรายละเอียดเพิ่มเติมลงไปด้วยว่าคำ วลี หรือประโยคนั้นใช้เป็นส่วนใดของยูสเคส และต้องแบ่งแยกต่อไปอีกว่าองค์ประกอบของยูสเคสนั้นเป็นส่วนคงหรือส่วนแปรผัน โดยผู้วิจัยได้เสนอข้อแนะนำเพื่อเป็นตัวอย่างในการเลือกคำ เช่น

- หัวข้อ (Heading) ของบท (Section) โดยทั่วไปจะเป็นชื่อของยูสเคส
- หากพบวลี "only by" "by using" "in the case of" วลีดังกล่าวเป็นเงื่อนไขเริ่มต้น
- หากพบวลี "normally" "with the exception" "except" เป็นส่วนเพิ่มขยาย (Extension) ของยูสเคส
- หากพบลำดับตัวเลข (Numbered List) หรือรายการปุ่ม (Bulleted List) โดยทั่วไปจะเป็นคำอธิบายของยูสเคส

การแบ่งแยกว่าส่วนใดเป็นส่วนคงที่หรือส่วนแปรผัน ผู้วิจัยได้กำหนดหลักการเช่น

- ส่วนใดที่พบในเอกสารคู่มือผู้ใช้เพียงเอกสารเดียว (จากเอกสารคู่มือผู้ใช้หลายๆ ระบบ) ส่วนนั้นมีแนวโน้มว่าเป็นส่วนแปรผันแบบเลือก

- หัวข้อ หรือหัวข้อย่อย (Sub heading) ที่มีเฉพาะในเอกสารฉบับเดียว ควรเป็นชื่อยูสเคสที่เป็นแบบเลือก

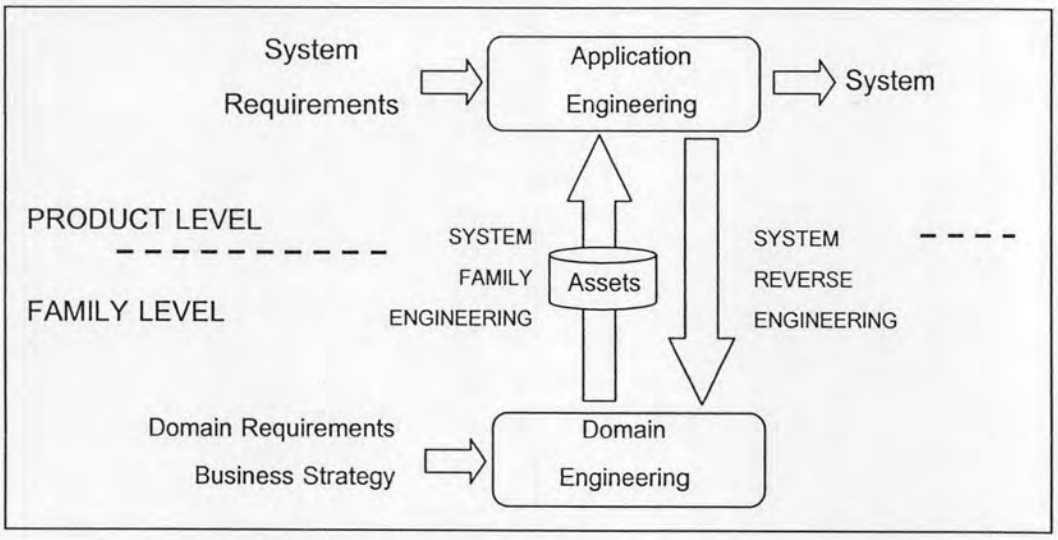
- หัวข้อ หรือหัวข้อย่อยที่มีชื่อต่างกัน แต่ปรากฏอยู่ในตำแหน่งเดียวกันในสารบัญของหลายๆ เอกสาร ควรเป็นชื่อยูสเคสแบบทางเลือก

- หากชุดตัวเลขในเอกสารทุกๆ ฉบับมีความแตกต่างกันเฉพาะค่าตัวเลข ส่วนดังกล่าวน่าจะเป็นส่วนแปรผันแบบกำหนดค่าได้

3) การเลือก การเปลี่ยนแปลง และการแก้ไขเพิ่มเติม (Selection change and modification) ขั้นตอนสุดท้ายเป็นการเลือก แยก และตรวจสอบว่าองค์ประกอบของยูสเคสจากขั้นตอนค้นหา สามารถนำมาใช้ในสายผลิตภัณฑ์ได้หรือไม่ ผู้ที่ทำหน้าที่นี้คือผู้เชี่ยวชาญระบบ (Domain Expert) นอกจากนั้นยังสามารถเปลี่ยน แก้ไขหรือปรับปรุงรายละเอียดในส่วนที่ผู้เชี่ยวชาญคิดว่าไม่เหมาะสมหรือสอดคล้องกับสายผลิตภัณฑ์

2.2.4 **เอกสารคำอธิบายยูสเคสที่แสดงถึงความต้องการในเชิงสายผลิตภัณฑ์ (Use Case Description of Requirements for Product Lines โดย A. Bertolino, A. Fantechi, S. Gnesi, G. Lami , A. Maccari)**

งานวิจัยนี้ [5] นำเสนอแนวทางในการเพิ่มขยายแผนภาพยูสเคสเพื่อใช้ในการอธิบายความต้องการของวิศวกรรมศาสตร์เชิงสายการผลิตตามกรอบการทำงานของ CAFÉ-PRM ดังรูปที่ 2.9 กรอบการทำงานดังกล่าวแบ่งกระบวนการออกเป็น 2 ระดับ คือ ระดับกลุ่ม (Family Level) และระดับผลิตภัณฑ์ (Product Level) และแบ่งความต้องการออกเป็นความต้องการในระดับโดเมน (Domain Requirements) และความต้องการในระดับระบบ (System Requirements) ความต้องการในระดับโดเมนจะใช้กระบวนการวิศวกรรมโดเมน (Domain Engineering) เพื่อสร้างสินทรัพย์และจัดเก็บลงในแหล่งเก็บสินทรัพย์ จากนั้นจะใช้กระบวนการวิศวกรรมแอพพลิเคชัน (Application Engineering) และกระบวนการวิศวกรรมเชิงกลุ่มระบบ (System Family Engineering) เพื่อสร้างระบบหรือผลิตภัณฑ์ตามความต้องการในระดับระบบจากสินทรัพย์ที่ได้จะกระบวนการวิศวกรรมโดเมน



รูปที่ 2.9 **กรอบการทำงาน** CAFÉ-PRM [5]

งานวิจัยได้นำเสนอแนวทางในการออกแบบยูสเคสความต้องการของระบบเป็น 2 แนวทาง คือ

1) ยูสเคสสองระดับ (Two Level Use Case)

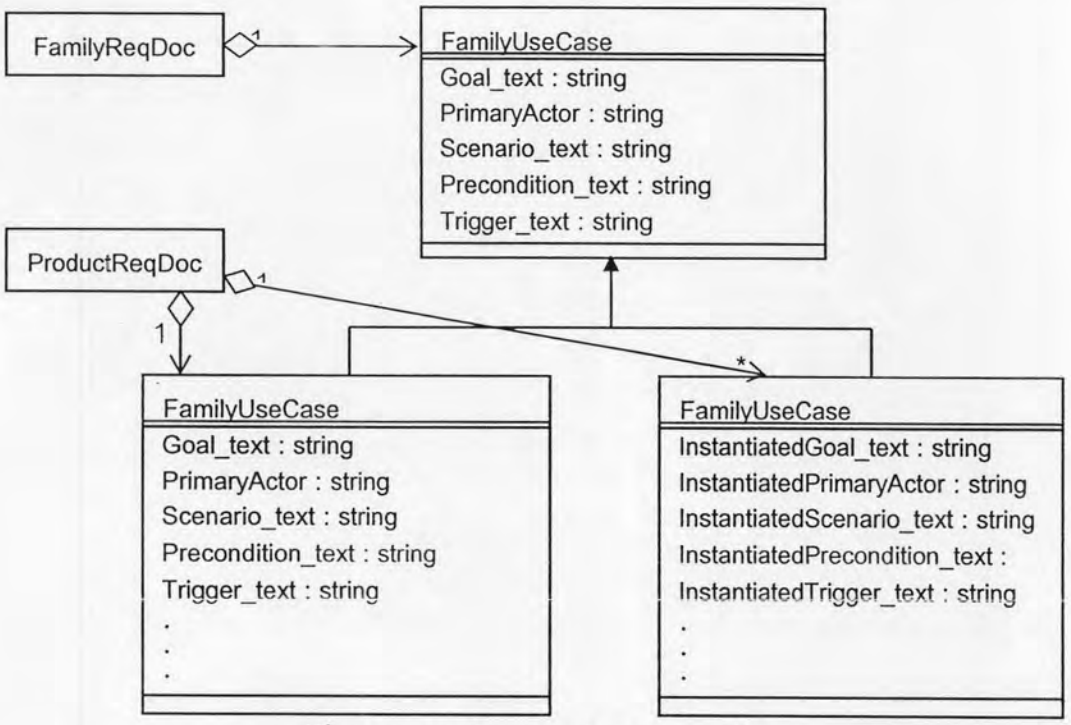
ยูสเคสสองระดับจะแบ่งยูสเคสออกเป็นระดับกลุ่มและระดับผลิตภัณฑ์ โดยยูสเคสระดับผลิตภัณฑ์จะได้มาจากยูสเคสระดับกลุ่ม และมีการกำหนดแท็ก (Tag) เพื่อใช้บ่งบอกประเภทของความแปรผันภายในยูสเคส ประเภทของแท็กมีดังต่อไปนี้

1.1) คอมโพเนนท์ทางเลือก (Alternative Component) เป็นคอมโพเนนท์ที่สามารถเลือกเอาอย่างใดอย่างหนึ่งภายในกลุ่ม ขึ้นอยู่กับความต้องการที่กำหนด

1.2) คอมโพเนนท์กำหนดค่า (Parametric Component) เป็นคอมโพเนนท์ที่สามารถกำหนดรูปแบบหรือค่าการทำงานได้

1.3) คอมโพเนนท์เลือกใช้ (Optional Component) เป็นคอมโพเนนท์ที่สามารถเลือกใช้หรือไม่เลือกใช้งานขึ้นกับความต้องการ

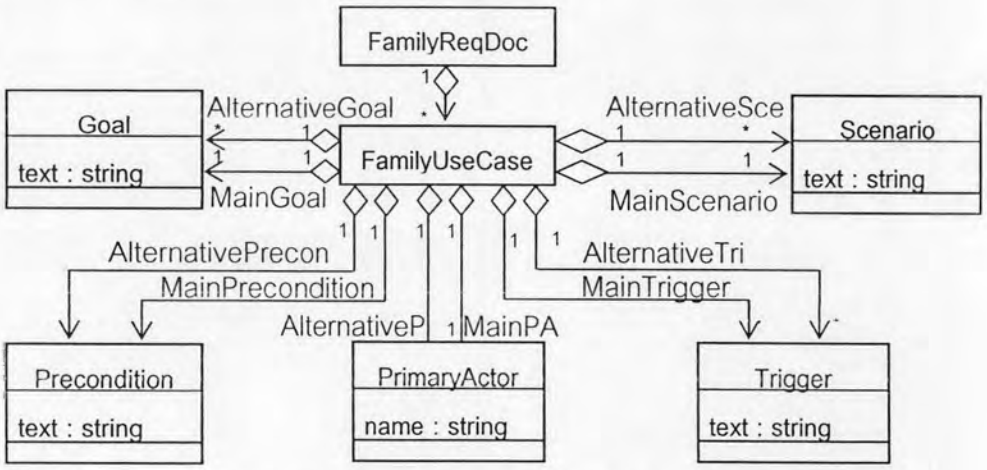
ตัวอย่างยูสเคสสองระดับแสดงในรูปที่ 2.10



รูปที่ 2.10 แบบจำลองยูสเคสสองระดับ [5]

2) ยูสเคสหนึ่งระดับ (One Level Use Cases)

ยูสเคสหนึ่งระดับจะรวบรวมความแปรผันและทางเลือกต่างๆ ทั้งหมดรวมอยู่ในยูสเคสเพื่อแสดงให้เห็นถึงคุณลักษณะทั้งหมดที่สามารถเลือกใช้เพื่อสร้างผลิตภัณฑ์ ตัวอย่างยูสเคสหนึ่งระดับแสดงในรูปที่ 2.11



รูปที่ 2.11 แบบจำลองยูสเคสหนึ่งระดับ [5]

ผู้วิจัยได้อธิบายเพิ่มเติมว่ายูสเคสทั้งสองแบบจะให้มุมมองของความต้องการที่แตกต่างกัน โดยยูสเคสสองระดับให้มุมมองแบบพลวัตเนื่องจากมีการกำหนดแท็กให้กับคอมโพเนนท์ ทำให้สามารถเพิ่มเติมหรือปรับเปลี่ยนความต้องการให้เหมาะกับสายการผลิตที่กำลังพัฒนาอยู่ ส่วนยูสเคสแบบหนึ่งระดับ ความต้องการทั้งหมดจะมองเห็นในลักษณะแบบราบ เนื่องจากความต้องการทั้งหมดจะถูกอธิบายอยู่ในยูสเคสในลักษณะแจกแจงรายละเอียด จึงเหมาะกับการอธิบายความต้องการของสายการผลิตที่ทราบความต้องการที่ชัดเจนแล้ว