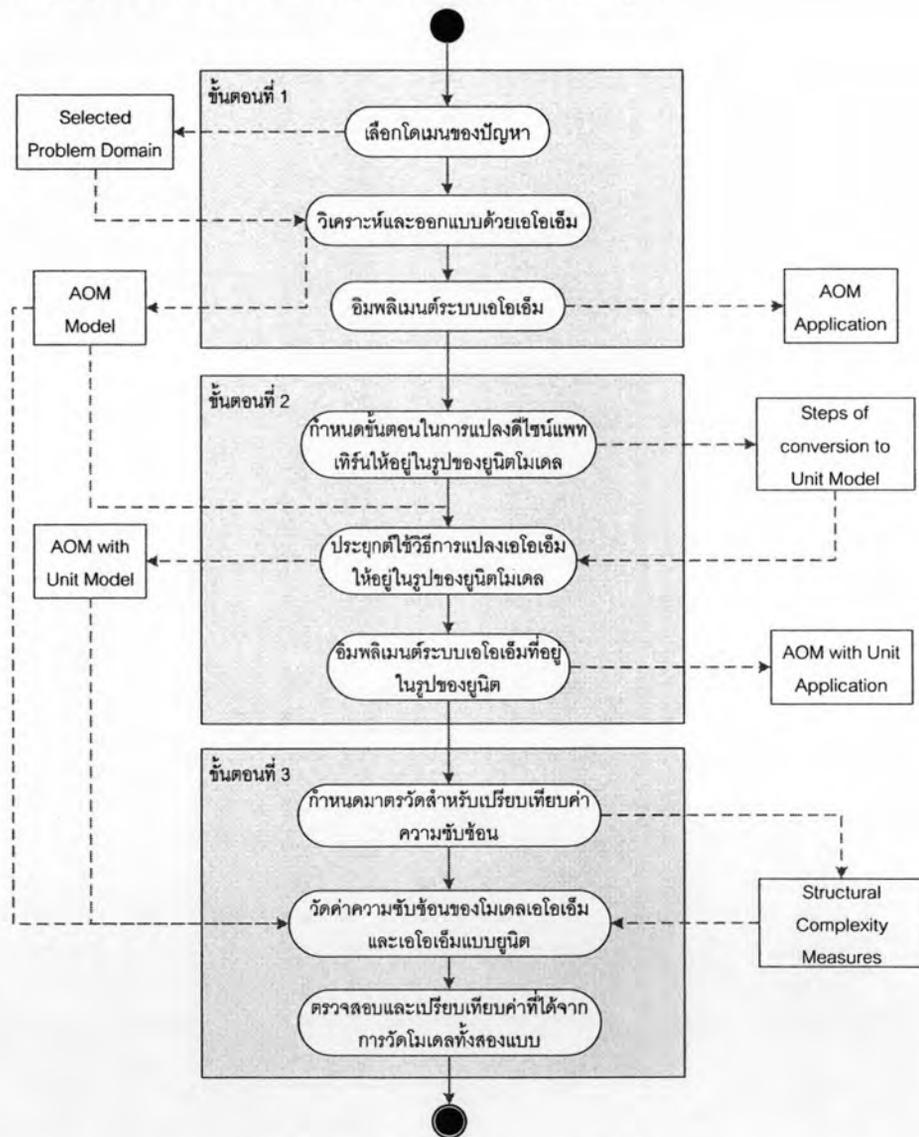


บทที่ 3

วิธีการแปลงเอไอเอ็มเป็นยูนิตโมเดล

ภาพรวมของงานวิจัยนี้จะนำเสนอวิธีการแปลงจากเอไอเอ็มโมเดลให้อยู่ในรูปของยูนิตโมเดล และจะนำเอาวิธีการแปลงดังกล่าวไปประยุกต์ใช้ในการการพัฒนาแอปพลิเคชัน รวมทั้งทำการวัดค่าความซับซ้อนของแผนภาพที่มีผลต่อความสามารถในการทำความเข้าใจและบำรุงรักษา โดยมีภาพรวมของขั้นตอนการวิจัยดังรูปที่ 3.1 ซึ่งในบทนี้จะกล่าวถึงเฉพาะวิธีการแปลงเอไอเอ็มเป็นยูนิตโมเดล ส่วนของการนำไปใช้พัฒนาแอปพลิเคชันและวัดผลนั้น จะกล่าวถึงในบทต่อไป



รูปที่ 3.1 ภาพรวมของขั้นตอนการวิจัย

3.1. ภาพรวมของขั้นตอนการวิจัย

ขั้นตอนการวิจัยนี้จะแบ่งออกเป็น 3 ส่วนหลัก คือ

1. ทำการออกแบบและพัฒนาระบบด้วยเอไอเอ็ม เพื่อใช้เป็นตัวอย่างในการเปรียบเทียบ
 - 1.1. เลือกโดเมนของปัญหา - ขั้นตอนนี้จะทำการเลือกโดเมนของปัญหาที่เหมาะสม คือ มีความต้องการที่หลากหลาย ไม่สามารถเก็บรายละเอียดของอ็อบเจกต์ในโดเมนได้อย่างสมบูรณ์ มีความต้องการที่จะเปลี่ยนแปลงอ็อบเจกต์โมเดลให้สอดคล้องกับความต้องการของธุรกิจได้ในอนาคต ซึ่งเหมาะกับการออกแบบด้วยเอไอเอ็ม
 - 1.2. วิเคราะห์และออกแบบระบบ - ขั้นตอนนี้จะทำการวิเคราะห์และออกแบบระบบโดยประยุกต์ใช้ดีไซน์แพทเทิร์นแบบต่างๆ เพื่อให้ระบบสามารถปรับเปลี่ยนได้ตามแนวทางของเอไอเอ็ม
 - 1.3. อิมพลีเมนต์ระบบเอไอเอ็ม - ทำการอิมพลีเมนต์ระบบให้สามารถทำงานได้ ตามที่ได้ออกแบบไว้เพื่อทดสอบการทำงานและวิเคราะห์ปัญหาที่อาจจะเกิดขึ้น
2. ทำการออกแบบและพัฒนาระบบเอไอเอ็มเดิมด้วยยูนิตโมเดล
 - 2.1. กำหนดขั้นตอนในการแปลงดีไซน์แพทเทิร์นให้อยู่ในรูปของยูนิตโมเดล - ขั้นตอนนี้จะทำการกำหนดวิธีในการแปลงดีไซน์แพทเทิร์นที่ไม่ปรากฏอยู่ในงานวิจัย [10] ซึ่งประกอบด้วย ไทป์อ็อบเจกต์, พรอบเพอร์ตี, ไทป์สแควร์ และ แอ็คเคานท์บิลิตี
 - 2.2. ประยุกต์ใช้วิธีการแปลงเอไอเอ็มให้อยู่ในรูปของยูนิตโมเดล - ขั้นตอนนี้จะนำเอาเอไอเอ็มที่ได้จากการออกแบบในขั้นตอนที่ 1.2 แปลงให้อยู่ในรูปของยูนิตโมเดล โดยใช้แนวทางที่ได้จากขั้นตอนที่ 2.1 และแนวทางที่ได้จากงานวิจัยของ [10]
 - 2.3. อิมพลีเมนต์ระบบเอไอเอ็มที่อยู่ในรูปของยูนิต - ทำการอิมพลีเมนต์ระบบให้สามารถทำงานได้จริงตามที่ได้ออกแบบไว้เพื่อทดสอบการทำงานและวิเคราะห์ปัญหาที่อาจจะเกิดขึ้น
3. การเปรียบเทียบและวัดผลที่ได้จากการออกแบบ
 - 3.1. กำหนดมาตรวัดสำหรับเปรียบเทียบค่าความซับซ้อน - ขั้นตอนนี้จะกำหนดมาตรวัดที่จะใช้วัดค่าความซับซ้อนของโครงสร้างของแผนภาพคลาสและแผนภาพแพ็กเกจของระบบทั้งแบบเอไอเอ็มเดิมและเอไอเอ็มที่ใช้ยูนิตโมเดล ซึ่งใช้มาตรวัดสำหรับยูนิตจะทำการเทียบเคียงความคล้ายคลึงกันระหว่างแผนภาพแพ็กเกจและยูนิตโมเดล
 - 3.2. วัดค่าความซับซ้อนของโมเดลเอไอเอ็มและเอไอเอ็มแบบยูนิต - ขั้นตอนนี้จะทำการวัดค่าความซับซ้อนของโครงสร้างของเอไอเอ็มโมเดลที่ได้จากขั้นตอนที่ 1.2 และเอไอเอ็มที่อยู่ในรูปของยูนิตที่ได้จากขั้นตอนที่ 2.2 โดยใช้มาตรวัดที่ได้จากขั้นตอนที่ 3.1 เพื่อเปรียบเทียบกันระหว่างสองโมเดล

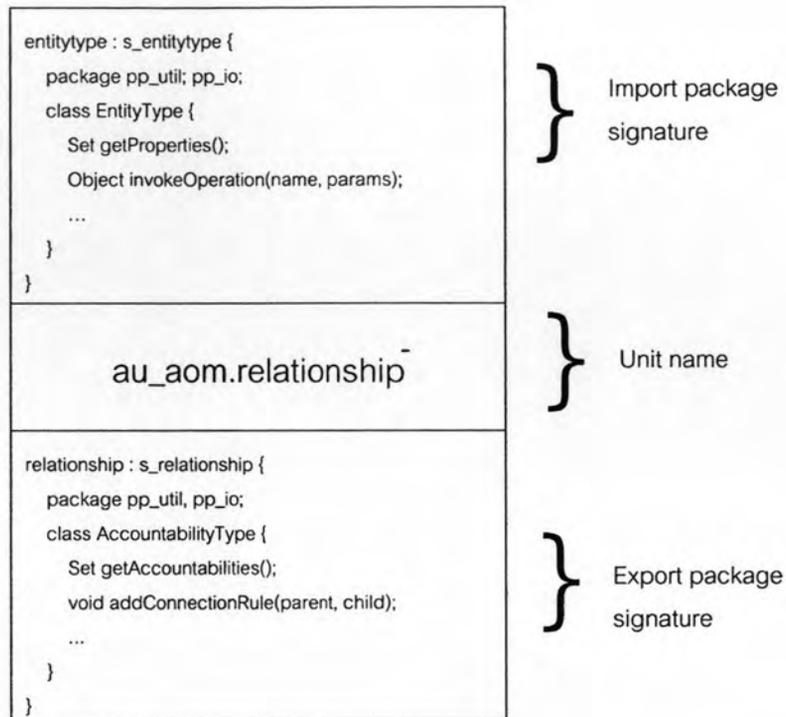
3.3. ตรวจสอบและเปรียบเทียบค่าที่ได้จากการวัดโมเดลทั้งสองแบบ - ขั้นตอนนี้จะทำการตรวจสอบค่าที่ได้จากการวัดโมเดลทั้งสองแบบและเปรียบเทียบผลว่าโมเดลเอไอเอ็มแบบยูนิต มีค่าความซับซ้อนของโครงสร้างน้อยกว่าเอไอเอ็มแบบเดิมหรือไม่ ซึ่งค่าของความซับซ้อนของโครงสร้างที่มีค่าน้อยจะสะท้อนถึงความสามารถในการทำความเข้าใจว่าเป็นโครงสร้างที่สามารถทำความเข้าใจได้ง่ายกว่า

3.2. รูปแบบของสัญลักษณ์ยูนิต

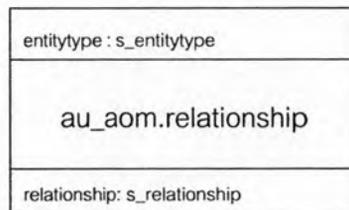
ในงานวิจัยนี้จะนำเอาภาษา Jiazzi [7] ซึ่งเป็นภาษาที่ใช้ในการสร้างยูนิตภาษาหนึ่ง โดยสัญลักษณ์แทนโครงสร้างของยูนิตที่ใช้ในงานวิจัย ผู้จัดทำวิทยานิพนธ์ได้กำหนดให้มีรูปแบบดังรูปที่ 3.2 และมีรายละเอียดดังนี้

- Import package signature – จะอยู่ในส่วนบนของสัญลักษณ์ซึ่งเป็นส่วนที่แสดงเซตของอิมพอร์ตคลาสและฟังก์ชันที่ยูนิตนั้นต้องการอ้างถึงเพื่อใช้ในการทำงาน จากรูปจะเห็นได้ว่ายูนิตอิมพอร์ตแพ็คเกจจิกเนเจอร์ที่ชื่อ s_entitytype โดยประกาศชื่อ entitytype เพื่อใช้ในการอ้างถึงในขั้นตอนอื่นๆ ภายในแพ็คเกจจิกเนเจอร์นั้นประกอบด้วยคลาส EntityType เมทอด getProperties() และ invokeOperation() ซึ่งสามารถเชื่อมต่อเข้ากับยูนิตอื่นที่เอ็กซ์พอร์ต ทำให้สามารถอ้างถึงคลาสและเรียกใช้เมทอดได้
- Unit name - ชื่อของยูนิตจะอยู่ส่วนกลางของสัญลักษณ์
- Export package signature – จะอยู่ในส่วนล่างของสัญลักษณ์ซึ่งเป็นส่วนที่แสดงเซตของเอ็กซ์พอร์ตคลาสและฟังก์ชันที่ยูนิตนั้นอนุญาตให้สามารถนำไปใช้งานได้ จากรูปจะเห็นได้ว่ายูนิตทำการเอ็กซ์พอร์ตแพ็คเกจจิกเนเจอร์ที่ชื่อ s_relationship โดยประกาศชื่อ relationship เพื่อใช้ในการอ้างถึง ภายในแพ็คเกจจิกเนเจอร์ประกอบด้วยคลาส AccountabilityType และเมทอด getAccountabilities() และ addConnectionRule() ซึ่งสามารถนำไปเชื่อมต่อกับยูนิตอื่น ที่ต้องการอิมพอร์ตแพ็คเกจจิกเนเจอร์ s_relationship ได้

สัญลักษณ์ในรูปที่ 3.2 นั้นเป็นการแสดงโครงสร้างหลักของยูนิต แต่เพื่อความเหมาะสมในการแสดงอธิบายในเล่มวิทยานิพนธ์ ผู้จัดทำวิทยานิพนธ์จึงปรับปรุงดังกล่าวให้เหลือเฉพาะแพ็คเกจจิกเนเจอร์ที่ยูนิตนั้นทำการอิมพอร์ตหรือเอ็กซ์พอร์ต ดังรูปที่ 3.3

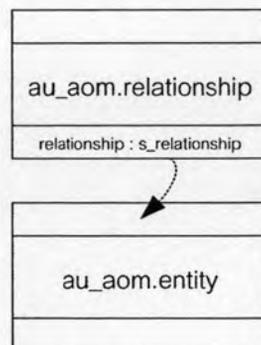


รูปที่ 3.2 ตัวอย่างสัญลักษณ์ของยูนิตที่ใช้ในงานวิจัยนี้สำหรับ Jiazzi



รูปที่ 3.3 สัญลักษณ์ยูนิตอย่างง่าย

การเชื่อมต่อระหว่างยูนิตนั้นจะแสดงด้วยเส้นประและหัวลูกศรแสดงถึงทิศทางของยูนิตที่อิมพอร์ต จากรูปที่ 3.4 อธิบายการเชื่อมต่อกันโดยยูนิต au_aom.entity ทำการอิมพอร์ต แพ็กเกจซิกเนเจอร์จากยูนิต au_aom.relationship ซึ่งเอ็กซ์พอร์ตแพ็กเกจซิกเนเจอร์ s_relationship



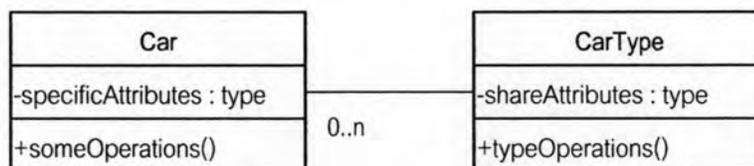
รูปที่ 3.4 เส้นการเชื่อมต่อกันระหว่างยูนิต

3.3. วิธีการแปลงจากเอไอเอ็มเป็นยูนิตโมเดล

จากปัญหาความซับซ้อนของการออกแบบและความยุ่งยากในการทำความเข้าใจอ็อบเจกต์โมเดลสำหรับเอไอเอ็มนั้น งานวิจัยนี้จึงเสนอวิธีการแปลงเอไอเอ็มให้อยู่ในรูปของยูนิตโมเดล ซึ่งเป็นการแยกระดับชั้นของการออกแบบเอไอเอ็มเป็น 2 ระดับชั้นคือ ระดับชั้นแพ็คเกจหรือคอมโพเนนต์ (Package/Component Layer) และระดับคลาส (Class Layer)

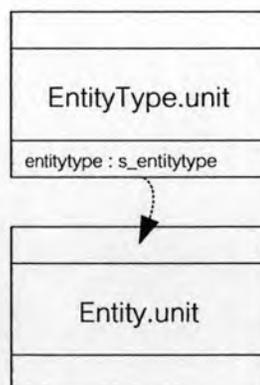
เอไอเอ็มประกอบด้วยการใช้ดีไซน์แพทเทิร์นหลายแบบคือ ทั่วไปอ็อบเจกต์, พรอบเพอร์ตี, ทั่วไปสแควร์, แอ็คเคาน์ทะบิลิตี, สตราทิจิ ดังนั้น การออกแบบยูนิตโมเดลสำหรับเอไอเอ็มนั้นแบ่งเป็นขั้นตอนสำหรับแพทเทิร์นต่างๆดังนี้

1. ทั่วไปอ็อบเจกต์



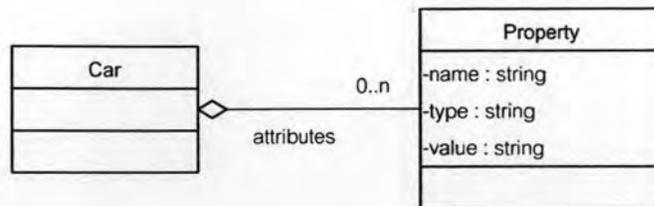
รูปที่ 3.5 ทั่วไปอ็อบเจกต์แพทเทิร์น [3]

จากรูปที่ 3.5 แพทเทิร์นนี้เป็นการแยกประเภท (Type) ของคลาสออกมา แทนการออกแบบให้เป็นคลาสแยกตามประเภท ซึ่งงานวิจัยนี้จะใช้หลักในการเปลี่ยนเอไอเอ็มให้อยู่ในรูปยูนิตคือ แยกคลาสที่เป็นเมต้าคลาสและเอนทิตีเป็นคนละยูนิต ดังนั้นเมื่อทำการแปลงแพทเทิร์นเป็นยูนิตโมเดล จะได้ยูนิต 2 ยูนิตสำหรับเอนทิตีของโดเมนและเมต้าคลาส ดังรูปที่ 3.6



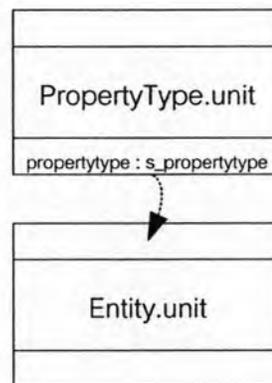
รูปที่ 3.6 ยูนิตโมเดลของทั่วไปอ็อบเจกต์แพทเทิร์น

2. พรอบเพอร์ตี



รูปที่ 3.7 พรอบเพอร์ตีแพทเทิร์น [4]

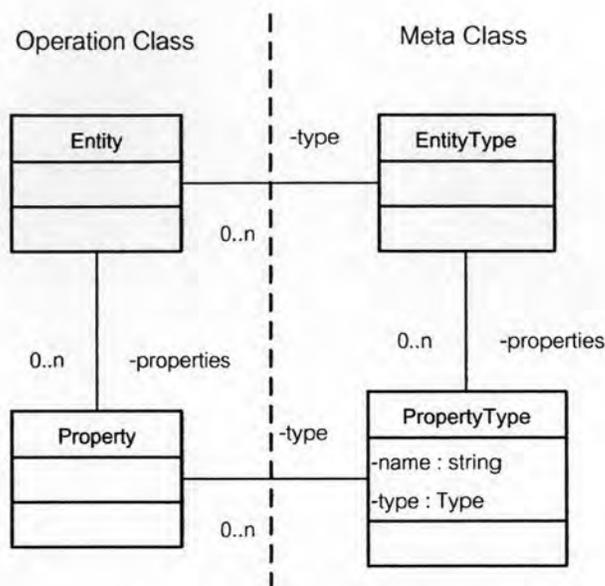
แพทเทิร์นนี้เป็นการแยกเอาส่วนของแอตทริบิวต์ของคลาสออกมาเป็นอีกหนึ่งคลาสเพื่อเป็นตัวแทนของแต่ละแอตทริบิวต์ จะเห็นได้ว่าแพทเทิร์นคล้ายกับไทป์อ็อกเจกต์แพทเทิร์น ซึ่งจะใช้ใช้วิธีการเดียวกันในการแปลงคือ แยกเมต้าคลาสและเอนทิตีออกจากกันเป็นคณลยูนิต ดังนั้นเมื่อแปลงแล้วก็จะได้ยูนิต 2 ยูนิตดังรูปที่ 3.8



รูปที่ 3.8 ยูนิตโมเดลของพรอบเพอร์ตีแพทเทิร์น

ทั้งสองแพทเทิร์นที่กล่าวมานั้นจะเห็นได้ว่ามีความสัมพันธ์ระหว่างคลาสเพียงสองคลาส เมื่อทำการแปลงให้อยู่ในรูปของยูนิตแล้วทำให้แต่ละยูนิตมีเพียงคลาสเดียว ดังนั้น ประโยชน์ที่ได้จากการแปลงเป็นยูนิต เช่น การซ่อนรายละเอียดของกลุ่มเมต้าคลาส ไม่ได้ประโยชน์เท่าที่ควร อีกทั้งโดยส่วนใหญ่การออกแบบเอไอเอ็มนั้น นอกจากต้องการให้สามารถปรับเปลี่ยนประเภทของอ็อบเจกต์ได้แล้ว ความต้องการให้สามารถปรับเปลี่ยนแอตทริบิวต์ของคลาสได้ก็มักจะปรากฏควบคู่กันมาเช่นกัน ดังนั้นการออกแบบเอไอเอ็ม จึงมักจะต้องทำการประยุกต์เข้ากับไทป์ออบเจกต์ และ พรอบเพอร์ตีแพทเทิร์นเข้าด้วยกันเป็นโดยเรียกอีกชื่อหนึ่งว่า ไทป์สแควร์

3. ไทป์สแควร์



รูปที่ 3.9 กลุ่มคลาสของไทป์สแควร์แพทเทิร์น

ไทป์สแควร์ เป็นแพทเทิร์นหลักของการนำไปใช้ในการออกแบบเอไอเอ็ม โดยการนำเอา ไทป์อ็อบเจกต์และพروبเพอร์ตีแพทเทิร์นมารวมกันทำให้ได้กลุ่มเมตาดัคลาส โดยมีเอนทิตีไทป์ (EntityType) ทำหน้าที่เก็บข้อมูลประเภทของอ็อบเจกต์ และพروبเพอร์ตีไทป์ (PropertyType) ทำหน้าที่เป็นตัวกำหนดประเภทพروبเพอร์ตี โดยมีเส้นความสัมพันธ์ระหว่างกันคือ เอนทิตีไทป์จะเป็นตัวกำหนดว่ามีพروبเพอร์ตีไทป์ใดเป็นสมาชิกได้บ้าง

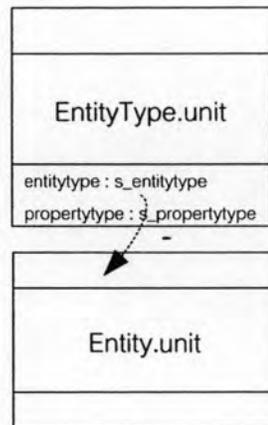
คลาสอีกกลุ่มหนึ่งคือ กลุ่มโอเปอเรชันคลาส ซึ่งเป็นคลาสที่เป็นตัวแทนของวัตถุจริงในโดเมนของปัญหานั้นๆ จากรูปที่ 3.9 คลาสเอนทิตี (Entity) จะสัมพันธ์กับคลาสเอนทิตีไทป์ และคลาสพروبเพอร์ตี (Property) เป็นจะสัมพันธ์กับคลาสพروبเพอร์ตีไทป์

ดังนั้นจะเห็นได้ว่าการที่จะให้เอนทิตีใดก็ตามสามารถเพิ่ม ลด ประเภทของคลาสและพروبเพอร์ตีของคลาสได้นั้นจะต้องสร้างความสัมพันธ์ระหว่างเอนทิตีกับเอนทิตีไทป์ และพروبเพอร์ตี กับพروبเพอร์ตีไทป์

คลาสทั้งสองกลุ่มนี้เมื่อทำการแปลงเป็นยูนิตโมเดลจะถูกแบ่งตามประเภทของกลุ่มคลาส ซึ่งจะได้เป็น 2 ยูนิต คือ

1. เอนทิตีไทป์ยูนิต (EntityType Unit) – เป็นยูนิตที่ทำหน้าที่ในการเพิ่มคุณสมบัติให้เอนทิตีสามารถเพิ่มลดประเภทของคลาสและพروبเพอร์ตีได้ โดยเอ็กซ์พอร์ตข้อกำหนด ซึ่งเป็นตัวบอกว่าเอนทิตีไทป์มีพروبเพอร์ตีไทป์ใดบ้างเป็นสมาชิกเพื่อให้ยูนิตอื่นๆสามารถนำไปใช้งานได้

2. เอนทิตียูนิต (Entity Unit) – เป็นยูนิตที่เป็นตัวแทนของเอนทิตีคลาส หรือคลาสแม่สำหรับทุกเอนทิตี

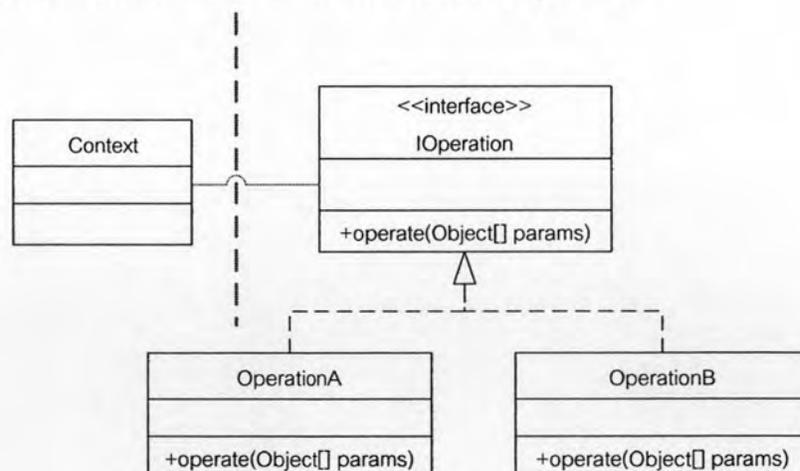


รูปที่ 3.10 เอนทิตีไทม์ยูนิตและเอนทิตียูนิต

จากรูปที่ 3.10 เมื่อนำทั้งสองยูนิตมาเชื่อมต่อกันโดยเอนทิตียูนิตทำการอิมพอร์ตเอนทิตีไทม์ยูนิตเข้ามาเพื่อใช้ในการสร้างให้เอนทิตีอ็อบเจกต์สามารถเพิ่มลดประเภทคลาสและพรอเพอร์ตี้ได้

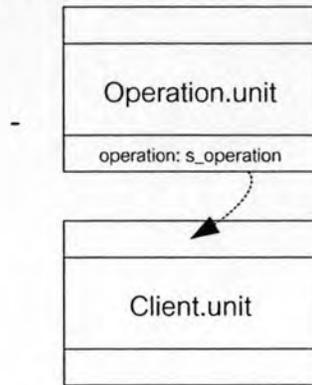
4. สตราทิจี

สตราทิจีเป็นแพทเทิร์นที่ใช้ในการเพิ่มลดเมทอดให้กับคลาส ซึ่งแพทเทิร์นนี้ประกอบด้วยอินเตอร์เฟซซึ่งเป็นตัวกำหนดมาตรฐานในการเรียกใช้เมทอดและคลาสที่อิมพลีเมนต์อินเตอร์เฟซให้ทำงานได้จริง ดังนั้น เมื่อต้องการเพิ่มเมทอดใหม่ให้กับอ็อบเจกต์หรือเอนทิตีไทม์ ก็สามารรถเพิ่มได้โดยการอิมพลีเมนต์ข้อกำหนดของอินเตอร์เฟซ คลาสที่ต้องการเรียกใช้เมทอดทำได้โดยการส่งชื่อของเมทอดเพื่อทำการค้นหาและเรียกให้เมทอดทำงาน ดังรูปที่ 3.11



รูปที่ 3.11 สตราทิจีแพทเทิร์น

สตราทิจิแพทเทิร์นนี้จะถูกแปลงเป็นยูนิตโมเดล ซึ่งใช้วิธีการของ Eric [10] คือ ทำการกำหนดให้อินเตอร์เฟสเป็นเซตของเอ็กซ์พอร์ตของยูนิต โดยซ่อนรายละเอียดของความสัมพันธ์ของคลาสอื่นๆไว้ภายใน ทำให้ได้ยูนิตที่ชื่อว่า โอเปอเรชันยูนิต (Operation Unit) และเมื่อต้องการเรียกใช้เมทอดต่างๆ ก็สามารถทำได้โดยการลิงค์หรืออิมพอร์ตโอเปอเรชันยูนิต ดังรูปที่ 3.12

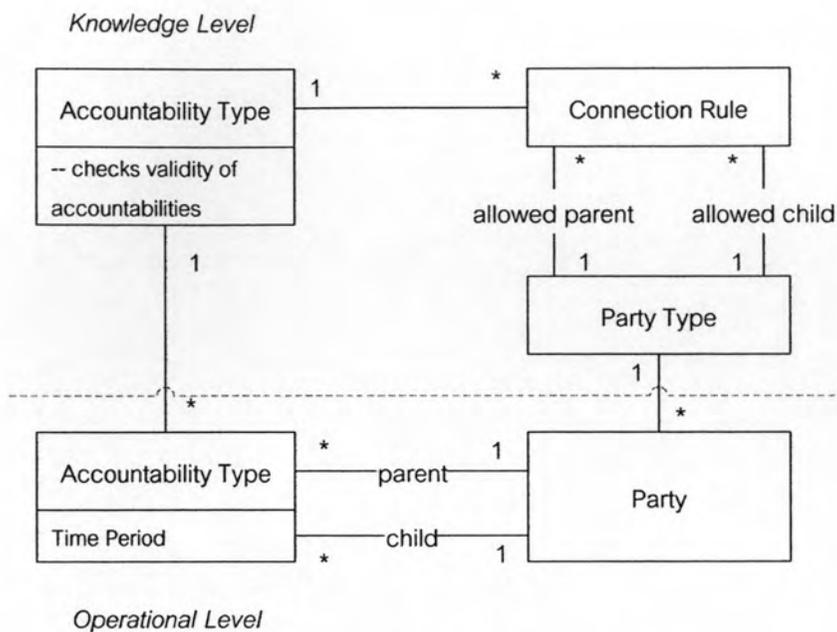


รูปที่ 3.12 โอเปอเรชันยูนิต

5. แอ็คเคาน์ทะบิลิตี

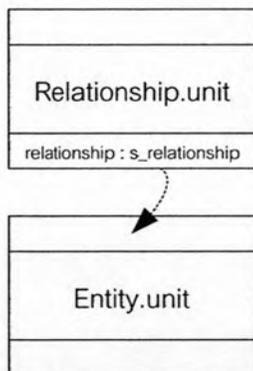
แอ็คเคาน์ทะบิลิตีเป็นแพทเทิร์นที่ใช้ในการอธิบายความสัมพันธ์ระหว่างเอนทิตีที่สนใจ โดยแบ่งกลุ่มคลาสออกเป็นสองกลุ่มคล้ายกับไทป์สแควร์แพทเทิร์น ซึ่งเรียกว่า Knowledge Level และ Operational Level [5] ดังรูปที่ 3.13 ในระดับชั้นของ Knowledge นั้นจะมีคลาส AccountabilityType เป็นตัวแทนของประเภทความสัมพันธ์ และมีคลาส Connection Rule เป็นตัวกำหนดกฎความสัมพันธ์นั้นว่ามีเอนทิตีไทป์ใดเป็นแม่หรือลูก (ในที่นี้คือคลาส PartyType)

ในส่วนของ Operational Level นั้นก็จะมีคลาส Accountability เป็นตัวแทนของความสัมพันธ์ที่เกิดขึ้นจริงระหว่างเอนทิตี เมื่อทำการสร้างความสัมพันธ์เกิดขึ้นคลาส AccountabilityType จะทำหน้าที่ในการตรวจสอบความถูกต้องของความสัมพันธ์ว่าเป็นไปตามกฎที่กำหนดไว้หรือไม่



รูปที่ 3.13 แอ็คเคาน์ทอะบิลิตีแพทเทิร์น

แอ็คเคาน์ทอะบิลิตีแพทเทิร์นนี้จะแปลงให้อยู่ในรูปของยูนิตโมเดล โดยแบ่งตามประเภทของกลุ่มคลาส ซึ่งแต่ละกลุ่มจะมีความสัมพันธ์ระหว่างกันภายในคงเดิม ส่วนเส้นความสัมพันธ์ที่เกิดขึ้นระหว่างกันนั้นจะถูกนำมากำหนดเป็นอินเตอร์เฟสของยูนิต ทำให้ได้ยูนิตจำนวน 2 ยูนิต ดังรูปที่ 3.14



รูปที่ 3.14 ยูนิตของความสัมพันธ์ระหว่างเอนทิตี

3.4. สรุปขั้นตอนในการแปลงเอไอเอ็มเป็นยูนิตโมเดล

ตารางที่ 2 ขั้นตอนการแปลงจากเอไอเอ็มเป็นยูนิตโมเดล

วัตถุประสงค์	เพื่อให้ นักพัฒนาสามารถทำความเข้าใจโมเดลของระบบแบบเอไอเอ็มได้ง่ายขึ้น
อินพุต	แผนภาพคลาสที่ออกแบบโดยใช้เทคนิคของเอไอเอ็ม
เอาต์พุต	แผนภาพ High-Level แบบยูนิตโมเดล และความสัมพันธ์ระหว่างคลาสที่ผ่านการปรับปรุงให้ตรงตามเงื่อนไขที่กำหนดไว้ในยูนิตโมเดล
เงื่อนไขก่อนการทำงาน	ดีไซน์แพทเทิร์นที่ใช้ในการออกแบบเอไอเอ็มประกอบด้วย ไทยป์อ็อบเจกต์ พรอบเพอร์ตี ไทยป์สแควร์ แอ็คเคาน์ทะเลบิลิตี สตราทิจิ
เงื่อนไขหลังการทำงาน	-
ขั้นตอนการแปลง	<ol style="list-style-type: none"> 1. บังคับกลุ่มคลาสที่เป็นเอนทิตี (หรือ กลุ่มโอเปอเรชันออลคลาส- Operational Level [5]) และกลุ่มเมต้าคลาส (หรือ Knowledge Level [5]) ของ ไทยป์สแควร์แพทเทิร์น และ แอ็คเคาน์ทะเลบิลิตีแพทเทิร์น 2. สร้างยูนิตให้กับเอนทิตีและเมต้าคลาส แยกจากกันเป็นคณละยูนิต เพื่อให้ยูนิตของกลุ่มเมต้าคลาสนั้นสามารถนำกลับมาใช้ใหม่ได้ <ol style="list-style-type: none"> 2.1. กรณีที่เอไอเอ็มโมเดลนั้นมีกลุ่มเมต้าคลาสของ ไทยป์สแควร์หรือแอ็คเคาน์ทะเลบิลิตีแพทเทิร์นหลายกลุ่ม ให้ยุบรวมกันเป็นเพียงยูนิตเดียวสำหรับแต่ละแพทเทิร์น 2.2. กรณีมีคลาสที่สืบทอดจากเอนทิตีไทยป์จะถูกตัดออกโดยแปลงให้อยู่ในรูปของอินสแตนซ์ของเอนทิตีไทยป์ 2.3. กรณีที่เอไอเอ็มไม่มีคลาส Entity หรือคลาสของโดเมนนั้นสัมพันธ์กับคลาส EntityType โดยตรง ให้ทำการรีแฟคทอริง (Refactoring) โดยกำหนดคลาสใหม่เป็นคลาส Entity และคลาสของโดเมนให้สืบทอดมาจากคลาส Entity เพื่อแยกยูนิตกัน ประโยชน์คือยูนิตของ Entity จะสามารถนำกลับมาใช้ใหม่ได้ 3. เลือกส่วนของเส้นความสัมพันธ์ระหว่างเอนทิตีของโดเมนและเมต้าคลาสนำมากำหนดเป็นเซตของการอิมพอร์ตและเซตของการเอ็กซ์พอร์ตไว้ที่แต่ละยูนิต โดยที่เส้นความสัมพันธ์ที่เกิดขึ้นภายในระหว่างคลาสของแพทเทิร์นนั้นๆ ไม่ต้องนำมากำหนดไว้ในส่วนของอิมพอร์ตหรือเอ็กซ์พอร์ตเพื่อซ่อนรายละเอียดของแพทเทิร์นนั้นไว้

ตารางที่ 2 ขั้นตอนการแปลงจากเอไอเอ็มเป็นยูนิตโมเดล (ต่อ)

	<ol style="list-style-type: none"> 4. สำหรับแอดคเอนท์อะบิลิตี้แพทเทิร์น สามารถแปลงเป็นยูนิตได้ โดยแยกแพทเทิร์นนี้ออกเป็น 1 ยูนิตสำหรับใช้จัดการความสัมพันธ์ของคลาส โดยเฉพาะ หรือ การเริ่มต้นในขั้นตอนที่ 1 เพื่อแยกกลุ่มโอเปอเรชันและเมต้าคลาสและแปลงเป็นยูนิตก่อน แล้วจึงค่อยนำมารวมเป็นยูนิตใหม่ (คอมพาวด์ยูนิต) 5. สำหรับสตราทิจีแพทเทิร์นที่จะนำไปใช้ในการเพิ่มลดเมทอดให้กับคลาสให้ทำการประยุกต์ใช้วิธีของ Eric[10] 6. เขียนรายละเอียดยูนิตของภาษายูนิตนั้นๆให้สมบูรณ์
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

จากแนวทางดังกล่าว งานวิจัยนี้จะทำการทดลองนำเอาวิธีการแปลงจากเอไอเอ็มเป็นยูนิตโมเดลไปประยุกต์ใช้ในพัฒนาแอปพลิเคชันที่ต้องการความยืดหยุ่นในการปรับเปลี่ยนอ็อบเจกต์โมเดล จากนั้นจึงเปรียบเทียบผลของค่าความซับซ้อนของโมเดลที่เป็นแบบเอไอเอ็มและแบบที่ใช้ยูนิตโมเดลเข้ามาช่วยในการออกแบบ ซึ่งจะกล่าวถึงรายละเอียดการออกแบบไว้ในบทต่อไป