บทที่ 4

การพัฒนาแอพพลิเคชันแบบเอโอเอ็มด้วยยูนิตโมเดล

ในบทนี้จะกล่าวถึงการออกแบบและพัฒนาแอพพลิเคชันแบบเอโอเอ็มจำนวน 3 ระบบ และนำขั้นตอนที่ได้เสนอไว้ในบทที่ 3 มาทำการแปลงจากเอโอเอ็มโมเดลของทั้ง 3 ระบบ ให้อยู่ในรูปของยูนิต โมเดล

ผลของการออกแบบจะถูกนำมาใช้ในการวัดผลเปรียบเทียบค่าความซับซ้อนของแผนภาพคลาส แผนภาพแพ็กเกจ และยูนิตโมเดล ซึ่งจะกล่าวถึงรายละเอียดของการวัดผลและเปรียบเทียบไว้ในบทต่อไป

4.1 เครื่องมือที่ใช้ในการพัฒนา

- Sun JDK 1.4.2 – เครื่องมือที่ใช้ในการพัฒนาซอฟต์แวร์ด้วยภาษาจาวา (Java)
- Jiazzi 2.2 – เป็นภาษาที่ใช้ในการออกแบบและพัฒนาคอมโพเนนต์หรือยูนิตโมเดลสำหรับภาษาจาวา

ในบทนี้ผู้จัดทำวิทยานิพนธ์กำหนดวิธีการตั้งชื่อของยูนิตและซิกเนเจอร์ เพื่อให้สามารถแยกแยะความแตกต่างของชื่อแพ็กเกจซิกเนเจอร์ อะตอมยูนิต และคอมพาวด์ยูนิตได้ง่าย โดยตั้งคำนำหน้าให้กับชื่อแพ็กเกจซิกเนเจอร์ด้วย s_xxx, ชื่ออะตอมยูนิตนำหน้าด้วย au_aom.xxx และ ชื่อคอมพาวด์ยูนิตนำหน้าด้วย cu_aom.xxx เช่น s_entity, au_aom.entity, cu_aom.entity เป็นต้น

4.2 ระบบคำนวณหาค่ากระแสเงินสดของหลักทรัพย์ประเภทตราสารหนี้ (Bond Cash Flow Generator - CFGen)

4.2.1 ปัญหาทางธุรกิจ (Problem Statement)

ในโลกธุรกิจทางด้านการบริหารเงินและการลงทุน ข้อมูลของกระแสเงินสด (Cash Flow) เป็นข้อมูลที่มีความสำคัญมากที่ผู้บริหารหรือนักวิเคราะห์จะต้องนำไปใช้ เช่น การคำนวณหามูลค่าของหลักทรัพย์และมูลค่าของพอร์ตลงทุน ณ ปัจจุบัน เป็นต้น

ระบบคำนวณหาค่ากระแสเงินสดของหลักทรัพย์ประเภทตราสารหนี้ (Bond Cash Flow Generator) จึงเป็นระบบที่ทำหน้าที่ในการคำนวณค่ากระแสเงินสดเพื่อให้ระบบอื่นๆสามารถนำข้อมูลไปใช้ให้เกิดประโยชน์ ความหมายของตราสารหนี้ คือ หลักทรัพย์ประเภทหนึ่งที่ผู้ออกตราสารมีข้อผูกพันตามกฎหมายที่จะชำระดอกเบี้ยและเงินต้นให้แก่ผู้ซื้อตามเวลาที่กำหนด ดอกเบี้ยหรือคูปอง (Coupon) ที่ผู้ซื้อได้รับในแต่ละงวดตามกำหนดเวลาจนถึงวันหมดอายุของตราสาร นั่นก็คือ กระแสเงิน

สดที่ผู้ซื้อจะได้รับ หรือ กระแสเงินสดที่ผู้ออกตราสารจะต้องจ่าย หลักทรัพย์แต่ละตัวจะมีการจ่ายคูปองที่แตกต่างกันไปขึ้นอยู่กับผู้ออกตราสารเป็นผู้กำหนด โดยผู้ออกตราสารจะกำหนดคุณลักษณะหรือวิธีการจ่ายดอกเบี้ยและประกาศให้นักลงทุนหรือผู้ที่สนใจทราบ

ตัวอย่างของตราสารหนี้ในตารางที่ 3 นั้นแสดงคุณลักษณะต่างๆ ซึ่งผู้ออกตราสารกำหนดวิธีการจ่ายดอกเบี้ยดังนี้

- ประเภทการจ่ายดอกเบี้ย (Coupon Type) - แบบคงที่ (Fixed Coupon)
- การนับวันเพื่อใช้ในการคำนวณมูลค่าของดอกเบี้ย(Calc Method) - แบบ 30/360
- ความถี่ในการจ่ายดอกเบี้ย (Payment Frequency) - ทุก 6 เดือน

ตารางที่ 3 ตัวอย่างตราสารหนี้ที่ออกโดยกระทรวงการคลัง

Symbol	LB06DA	Payment Frequency	Semi-Annually
Name	Thailand Government Bond	Calculation Method	30/360
Name (Thai)	พันธบัตรรัฐบาลกรณีพิเศษในปีงบประมาณ พ.ศ. 2542 ครั้งที่ 5/2 ชนิดดอกเบี้ยร้อยละ 8.00 ต่อปี	Put/Call Option	-
Issuer	Ministry of Finance	Distribution	Government Debt Security Offering
ISIN Code	TH062303GC00	Registrar	Bank of Thailand
Issue Rating	Moody's Baa1 Fitch A TRIS - S&P - Composite BBB+	Lead Underwriter(s)	-
Initial Par	THB 1,000.00	Financial Advisor(s)	-
Current Par	THB 1,000.00	Debenture Holder Representatives	-
Issue Size	THB 35,000,000.00 (M)	Covenant	-
Outstanding Size	THB 35,000,000.00 (M)	Collateral	-
Issue Date	12/8/1998	Remark	-
Maturity Date	12/8/2006	Type	Sovereign
Issue Term	8.0 Yrs.	Market of Issue	Domestic
Coupon	Fixed : 8.0000%		
Amortized Schedule	-		

ตารางที่ 4 ตัวอย่างกระแสเงินสดของตราสารหนี้

No.	Payment	XI Date	Par	Coupon (%)	Interest	Principal	Total Payment
1	8-มิ.ย.-99	29-พ.ค.-99	1000.000000	8.000000	40.000000	0.000000	40.000000
2	8-ธ.ค.-99	28-พ.ย.-99	1000.000000	8.000000	40.000000	0.000000	40.000000
3	8-มิ.ย.-00	29-พ.ค.-00	1000.000000	8.000000	40.000000	0.000000	40.000000
4	8-ธ.ค.-00	28-พ.ย.-00	1000.000000	8.000000	40.000000	0.000000	40.000000
5	8-มิ.ย.-01	29-พ.ค.-01	1000.000000	8.000000	40.000000	0.000000	40.000000
6	8-ธ.ค.-01	28-พ.ย.-01	1000.000000	8.000000	40.000000	0.000000	40.000000
7	8-มิ.ย.-02	29-พ.ค.-02	1000.000000	8.000000	40.000000	0.000000	40.000000
8	8-ธ.ค.-02	28-พ.ย.-02	1000.000000	8.000000	40.000000	0.000000	40.000000
9	8-มิ.ย.-03	29-พ.ค.-03	1000.000000	8.000000	40.000000	0.000000	40.000000
10	8-ธ.ค.-03	28-พ.ย.-03	1000.000000	8.000000	40.000000	0.000000	40.000000
11	8-มิ.ย.-04	29-พ.ค.-04	1000.000000	8.000000	40.000000	0.000000	40.000000
12	8-ธ.ค.-04	28-พ.ย.-04	1000.000000	8.000000	40.000000	0.000000	40.000000
13	8-มิ.ย.-05	29-พ.ค.-05	1000.000000	8.000000	40.000000	0.000000	40.000000
14	8-ธ.ค.-05	28-พ.ย.-05	1000.000000	8.000000	40.000000	0.000000	40.000000
15	8-มิ.ย.-06	29-พ.ค.-06	1000.000000	8.000000	40.000000	0.000000	40.000000
16	8-ธ.ค.-06	23-พ.ย.-06	1000.000000	8.000000	40.000000	1000.000000	1040.000000

จากคุณลักษณะของตราสารหนี้ที่แสดงในตารางที่ 3 นั้นสามารถสร้างเป็นกระแสเงินสดได้ดังตัวอย่างที่แสดงในตารางที่ 4

เนื่องจากการออกตราสารหนี้แต่ละครั้ง ผู้ออกตราสารจะพิจารณาถึงความจำเป็นและปัจจัยต่างๆ หลายประการ ไม่ว่าจะเป็นปัจจัยภายนอกหรือภายในบริษัทเอง ด้วยเหตุนี้ตราสารหนี้แต่ละตัวจึงมีคุณลักษณะที่หลากหลายแตกต่างๆกัน ซึ่งความหลากหลายนี้ทำให้ อ็อบเจกต์โมเดลที่มีโครงสร้างแบบคงที่ (Static Model) ไม่มีสามารถปรับเปลี่ยนให้รองรับกับการเปลี่ยนแปลงความต้องการที่มีอยู่อย่างต่อเนื่องได้ ด้วยปัญหาดังกล่าว ระบบที่พัฒนาขึ้นโดยวิธีการแบบเอโอเอ็ม จะทำให้ระบบมีความยืดหยุ่นมากขึ้น โดยมีคุณสมบัติดังนี้

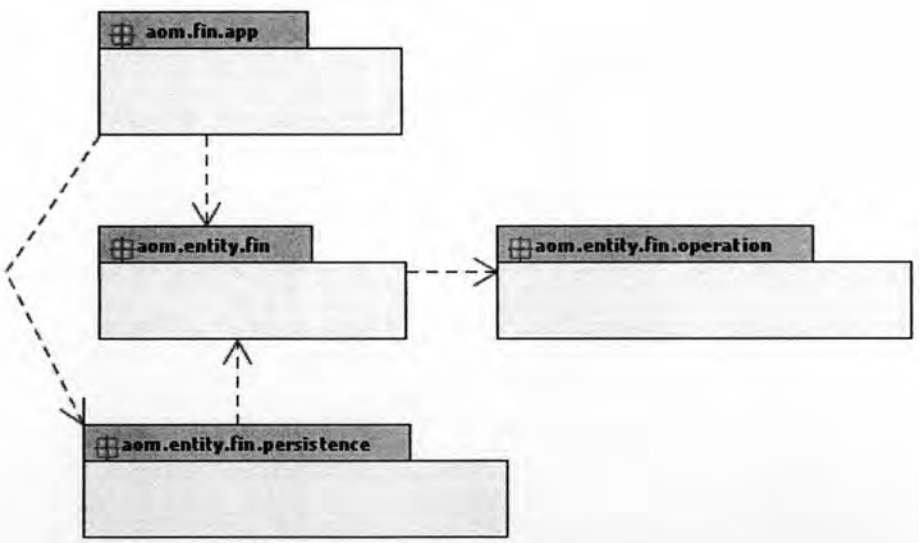
- สามารถเพิ่มลด ประเภทของการจ่ายดอกเบี้ย (Coupon Type) ประเภทของการนับวันในการคำนวณดอกเบี้ย (Days Calculation Type) ประเภทของความถี่ในการจ่ายดอกเบี้ย (Payment Frequency Type) ประเภทของกลุ่มผู้ออกตราสารหนี้ได้ (Sector Type)
- สามารถเพิ่มลด พรอบเพอร์ตีหรือแอตทริบิวต์ของประเภทการจ่ายดอกเบี้ย เช่น การจ่ายดอกเบี้ยแบบลอยตัว (Float Coupon) ซึ่งเป็นประเภทของการจ่ายดอกเบี้ยประเภทหนึ่งจะต้องสามารถเพิ่มลดพรอบเพอร์ตีได้ เนื่องจากอนาคตอาจมีการบันทึกข้อมูลอื่นๆเพิ่มขึ้นที่มีผลกระทบต่อการคำนวณดอกเบี้ย เช่น กรณีการจ่ายดอกเบี้ยแบบลอยตัวมีพรอบเพอร์ตี ได้แก่ อัตราดอกเบี้ยต่ำสุด(FloorRate), อัตราดอกเบี้ยสูงสุด (CapRate), ชื่อฟังก์ชันที่ใช้ในการเฉลี่ย

อัตราดอกเบี้ย (CalcFunction), ส่วนต่างของอัตราดอกเบี้ยที่อ้างถึง (MarginRate), สัญลักษณ์ของอัตราดอกเบี้ยที่ใช้อ้างอิง (RefRateSymbols) ต่อมาต้องการเพิ่มพรอบเพอร์ตี FactorNo เพื่อนำไปเป็นตัวคูณในการคำนวณดอกเบี้ย เป็นต้น

- สามารถเพิ่มลด เมท็อดของวิธีการนับวันในการคำนวณหาดอกเบี้ยได้ เช่น ประเภทการนับวัน 30/360 โดยมาตรฐาน ICMA หรือมาตรฐานของ US ซึ่งมีรายละเอียดต่างกัน ซึ่งแอพพลิเคชันจะต้องสามารถเปลี่ยนแปลงหรือเพิ่มลดวิธีการคำนวณได้
- สามารถเพิ่มลด ความสัมพันธ์ระหว่างบริษัทที่เกี่ยวข้องกันได้ สมมติตัวอย่างเช่น ประเภทของความสัมพันธ์แบบผู้ค้ำประกันตราสาร (Guarantor) กำหนดไว้ว่า ผู้รับค้ำประกันจะต้องเป็นธนาคาร (Bank) ผู้ได้รับการค้ำประกันซึ่งเป็นผู้ออกตราสารจะต้องเป็นบริษัทจดทะเบียน (Registered Company) ดังนั้น เมื่อบริษัท A เป็นบริษัทประเภทธนาคาร บริษัท B เป็นบริษัทจดทะเบียน ก็จะสามารถสร้างความสัมพันธ์กันในแบบผู้ค้ำประกันตราสารได้โดยบริษัท A เป็นผู้ค้ำประกัน บริษัท B เป็นผู้ได้รับการค้ำประกัน

4.2.2 การออกแบบแอพพลิเคชัน

4.2.2.1 แผนภาพแพ็กเกจ



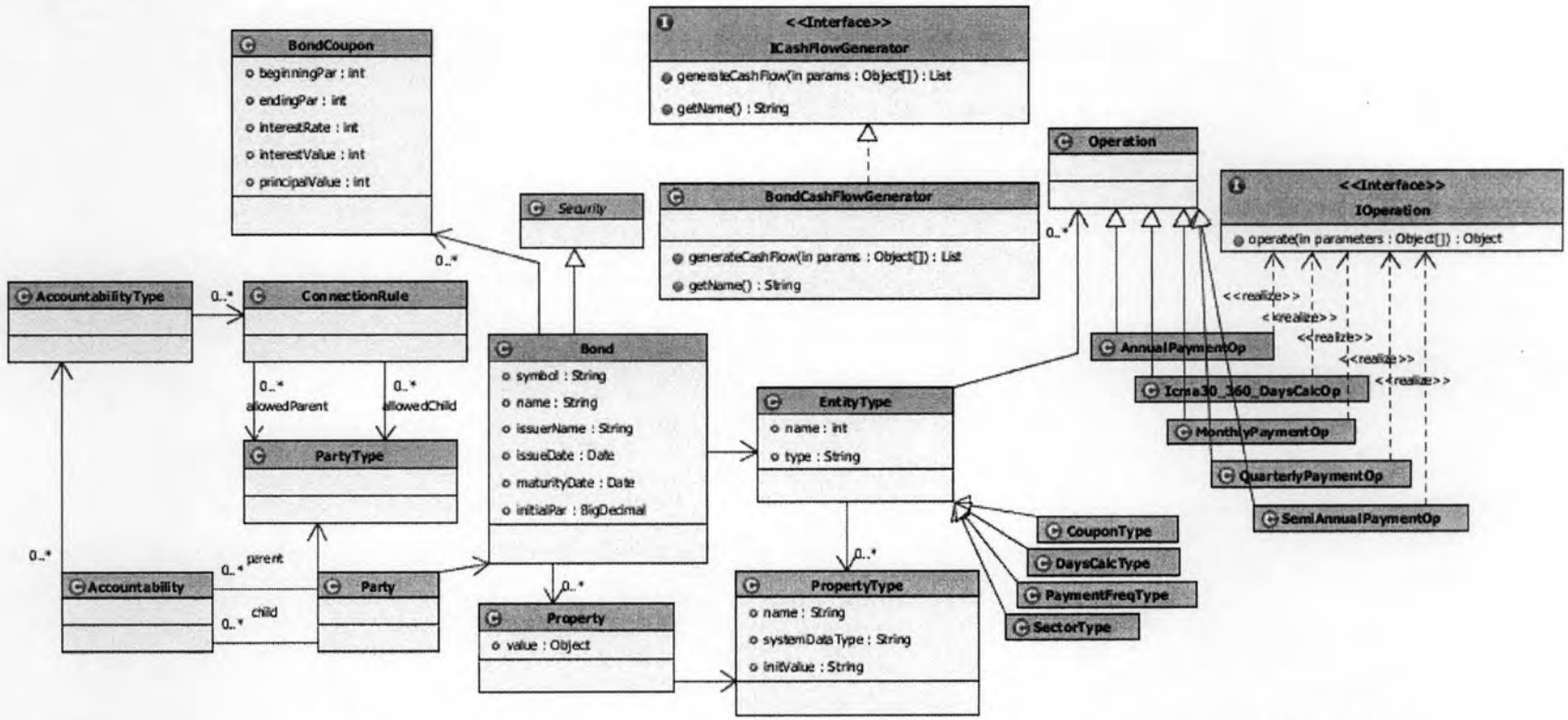
รูปที่ 4.1 แผนภาพแพ็กเกจของระบบ CFGen

จากแผนภาพแพ็กเกจในรูปที่ 4.1 ของระบบ CFGen แสดงให้เห็นถึงการจัดกลุ่มของคลาสและการขึ้นต่อกันระหว่างแพ็กเกจ ซึ่งประกอบด้วยแพ็กเกจต่างๆดังนี้

- aom.entity.fin เป็นแพ็กเกจที่รวบรวมคลาสเอนทิตีต่างๆ รวมทั้งคลาสที่เกี่ยวข้องกับเอโอเอ็มไว้ด้วยกัน

- aom.entity.fin.operation เป็นแพ็กเกจที่เกี่ยวข้องกับคลาสที่อิมพลิเมนต์อินเตอร์เฟส (IOperation) เพื่อให้สามารถแทนที่ได้ง่ายเมื่อมีการปรับปรุงเพิ่มลดฟังก์ชันหรือเมท็อดต่างๆ เช่น คลาสที่ทำหน้าที่คำนวณวันในรูปแบบมาตรฐานต่างๆ เช่น การนับวันแบบ ICMA 30/360, Actual/365, Actual/Actual เป็นต้น
- aom.entity.fin.persistence เป็นแพ็กเกจรวบรวมคลาสที่ทำหน้าที่ในการติดต่อกับระบบฐานข้อมูล
- aom.fin.app เป็นแพ็กเกจที่รวบรวมคลาสที่ทำหน้าที่ติดต่อกับผู้ใช้เพื่อเริ่มต้นและเรียกใช้ฟังก์ชันการคำนวณค่ากระแสเงินสด ติดต่อกับฐานข้อมูล นำข้อมูลเอนทิตีมาแสดงผลการทำงาน

4.2.2.2 แผนภาพคลาส

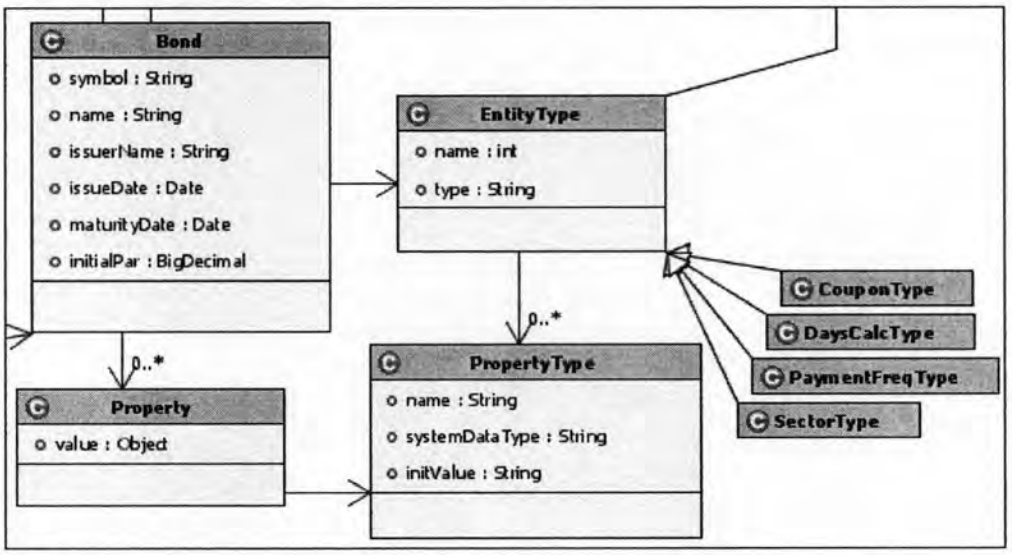


รูปที่ 4.2 แผนภาพคลาสแบบเอโอเอ็มของระบบ CFGen ของแพ็กเกจ aom.entity.fin และ aom.entity.fin.operation ก่อนแปลงเป็นยูนิตโมเดล

จากรูปที่ 4.2 แสดงแผนภาพคลาสของระบบ CFGen ซึ่งออกแบบโดยใช้ดีไซน์แพทเทิร์นแบบต่างๆ เพื่อให้อ็อบเจกต์สามารถปรับเปลี่ยนได้ตามแนวคิดของเอโอเอ็ม แผนภาพคลาสดังกล่าวจะอธิบายเฉพาะส่วนที่เกี่ยวข้องกับเอโอเอ็มเท่านั้น ไม่ได้รวมถึงคลาสอื่นๆ ซึ่งไม่เกี่ยวข้องกับการออกแบบเอโอเอ็ม เช่น คลาสที่เกี่ยวกับหน้าจอของผู้ใช้ คลาสที่ติดต่อกับระบบฐานข้อมูล เป็นต้น

4.2.3 การแปลงจากเอโอเอ็มเป็นยูนิตโมเดล

หลังจากที่ได้แผนภาพคลาสที่เป็นแบบเอโอเอ็มแล้ว ก็จะทำการแปลงให้อยู่ในรูปแบบของยูนิตโมเดล ตามวิธีการที่ได้นำเสนอไว้ในบทที่ 3 และจากแผนภาพคลาสจะพบว่ากลุ่มคลาสต่างๆที่ออกแบบด้วยแพทเทิร์นสำหรับเอโอเอ็มมีดังนี้



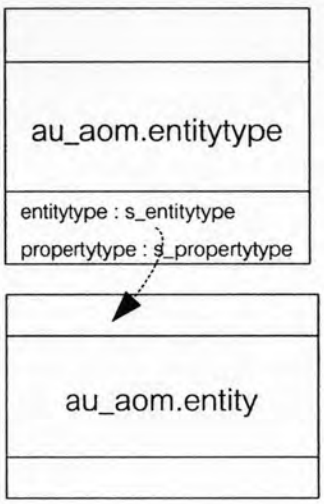
รูปที่ 4.3 แผนภาพคลาสและความสัมพันธ์ของไทป์สแควร์แพทเทิร์นระบบ CFGen

4.2.3.1 ไทป์สแควร์แพทเทิร์น

- คลาส Bond – เป็นเอนทิตีคลาสของตราสารหนี้
- คลาส EntityType – เป็นคลาสประเภทของตราสารหนี้ ซึ่งในมุมมองการจัดประเภทของตราสารหนี้นั้นมีหลายแบบ เช่น การแบ่งตามประเภทของการจ่ายคูปอง อีกทั้งประเภทของการจ่ายก็อาจจะเพิ่มแบบใหม่เข้ามาในภายหลังได้ คลาสนี้สัมพันธ์กับคลาส PropertyType เพื่อกำหนดว่าประเภทของการจ่ายคูปองแต่ละแบบนั้นมีพรอบเพอร์ตีอะไรบ้าง เช่น การจ่ายดอกเบี้ยแบบคงที่ (Fixed) ก็จะมีพรอบเพอร์ตี Rate(อัตราดอกเบี้ย) แต่หากเป็นการจ่ายแบบลอยตัว (Float) ก็จะมีพรอบเพอร์ตี Rate, CapRate, FloorRate, MarginRate, CalcFunc, FactorNo, RefSymbols
- คลาส Property – เป็นคลาสทำหน้าที่ตัวแทนของค่าพรอบเพอร์ตีแต่ละประเภท
- คลาส PropertyType – เป็นคลาสตัวแทนของพรอบเพอร์ตีประเภทต่างๆ

เมื่อประยุกต์ใช้แนวทางในการแปลงเป็นยูนิตโมเดล ผลที่ได้แสดงอยู่ในรูปของอะตอมยูนิต ดังรูปที่ 4.4 ซึ่งจะเห็นได้ว่าความสัมพันธ์ระหว่างคลาสจะถูกนำมากำหนดเป็นเซตของเอ็กพอร์ตและอิมพอร์ตของยูนิต ซึ่งอยู่ในรูปแพ็กเกจซิกเนเจอร์ของ Jiazzi

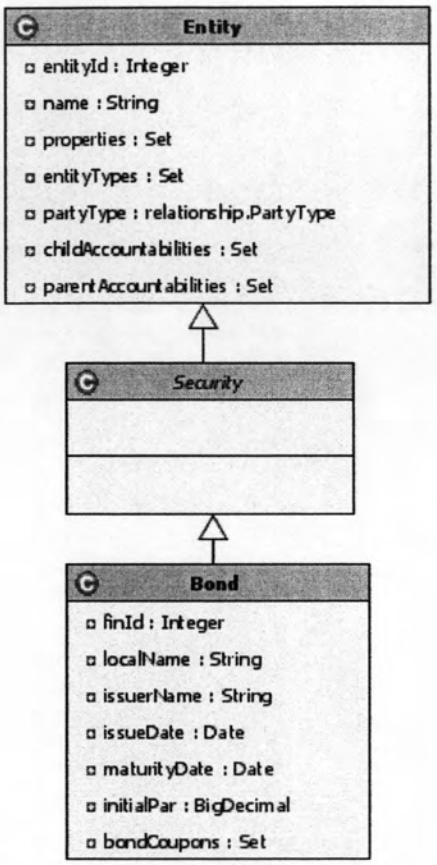
สำหรับคลาสที่สืบทอดมาจากคลาส EntityType ได้แก่ Coupon Type, DaysCalcType, PaymentFreqType, SectorType นั้นจะถูกเปลี่ยนให้เป็นอินสแตนท์ของคลาส EntityType



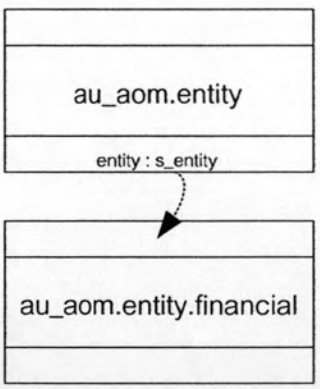
รูปที่ 4.4 อะตอมยูนิตของ au_aom.entitytype และ au_aom.entity

สำหรับคลาสที่ไม่ได้ถูกนำมากำหนดเป็นซิกเนเจอร์ของยูนิต ก็จะถูกสร้างไว้ภายในยูนิต ซึ่งยูนิตอื่นๆจะไม่สามารถเข้าถึงคลาสภายในยูนิตได้ (รายละเอียดของซิกเนเจอร์ดูได้จากภาคผนวกท้ายเล่ม)

อย่างไรก็ตามในการออกแบบและพัฒนาแอพพลิเคชันเอโอเอ็มด้วยยูนิตโมเดล พบว่าเพื่อให้ยูนิตสามารถนำกลับมาใช้ใหม่ได้ดีนั้น แทนที่จะแปลงให้คลาส Bond และคลาสอื่นๆที่ยึดติดกับโดเมนอยู่ภายในยูนิต au_aom.entity ทำให้ไม่สามารถนำไปใช้กับระบบอื่นๆได้ ดังนั้นจึงทำการเพิ่มคลาส Entity และให้คลาส Security (หลักทรัพย์) และคลาส Bond (ตราสารหนี้) ทำการสืบทอดมา ดังรูปที่ 4.5 ซึ่งทำให้สามารถแยกยูนิตสำหรับโดเมนออกจากเอนทิตียูนิต ทำให้เอนทิตียูนิตสามารถนำไปอิมพอร์ตใช้กับระบบอื่นๆได้ดังรูปที่ 4.6

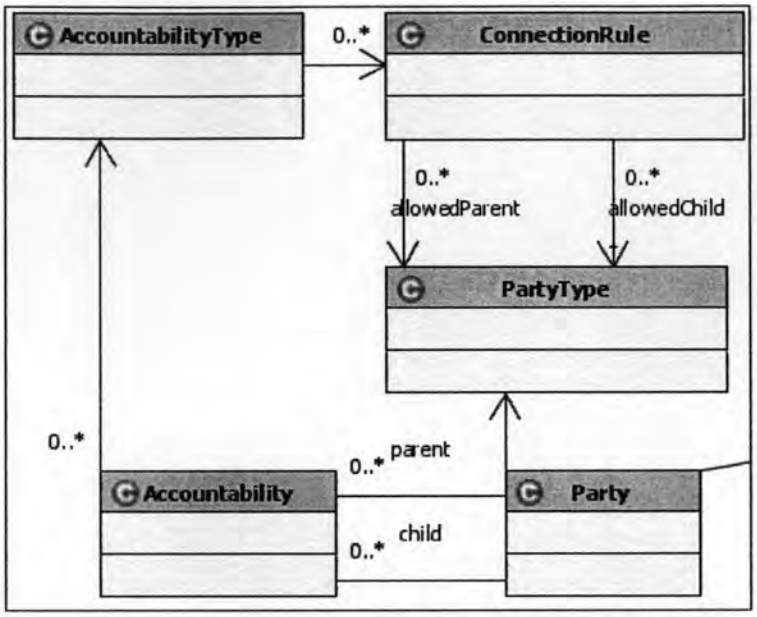


รูปที่ 4.5 แผนภาพคลาสที่เพิ่มขึ้นมาเพื่อแยกยูนิต au_aom.entity ให้นำกลับมาใช้ใหม่



รูปที่ 4.6 อะตอมยูนิตของ au_aom.entity และ au_aom.entity.financial

4.2.3.2 แอ็คเคาน์ทะบิลิตีแพทเทิร์น

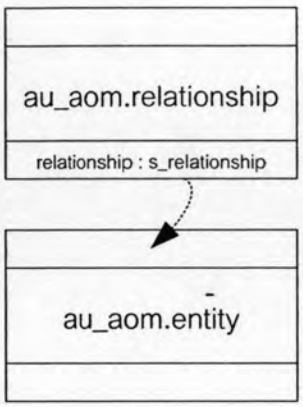


รูปที่ 4.7 แผนภาพคลาสและความสัมพันธ์ของแอ็คเคาน์ทะบิลิตีระบบ CFGen

แผนภาพคลาสในรูปที่ 4.7 แสดงการใช้แอ็คเคาน์ทะบิลิตีแพทเทิร์นในการออกแบบ โดยมีรายละเอียดดังนี้

- คลาส Party – เป็นคลาสของเอนทิตีใดๆที่สนใจ ซึ่งในระบบ CFGen เอนทิตีที่ต้องการให้สามารถเปลี่ยนแปลงความสัมพันธ์ได้คือ อ็อบเจกต์ของบริษัทหรือหน่วยงานต่างๆ เช่น ธนาคาร A เป็นผู้ค้ำประกันการออกตราสารให้กับ บริษัท B
- คลาส AccountabilityType – เป็นคลาสประเภทของความสัมพันธ์ระหว่างเอนทิตีที่สนใจ เช่น อ็อบเจกต์ Guarantor (ผู้ค้ำประกัน) FinancialAdvisor (ที่ปรึกษาทางการเงิน)
- คลาส ConnectionRule – เป็นคลาสที่กำหนดกฎความสัมพันธ์ระหว่างประเภทเอนทิตี ซึ่งอ็อบเจกต์ของคลาสนี้จะเป็นส่วนหนึ่งของอ็อบเจกต์แอ็คเคาน์ทะบิลิตีไทป์ เช่น ผู้ค้ำประกันจะต้องเป็นธนาคาร(Parent) และผู้ออกตราสารที่ได้รับการค้ำประกันต้องเป็นบริษัทที่จดทะเบียนฯ (Child)
- คลาส PartyType – เป็นคลาสประเภทของเอนทิตีที่ต้องการให้สามารถเปลี่ยนแปลงความสัมพันธ์ระหว่างเอนทิตีประเภทนั้นๆได้ เช่น ธนาคาร บริษัทที่จดทะเบียนกับสมาคมตลาดตราสารหนี้
- คลาส Accountability – เป็นคลาสของอ็อบเจกต์ความสัมพันธ์ที่เกิดขึ้นระหว่างเอนทิตีที่สนใจ

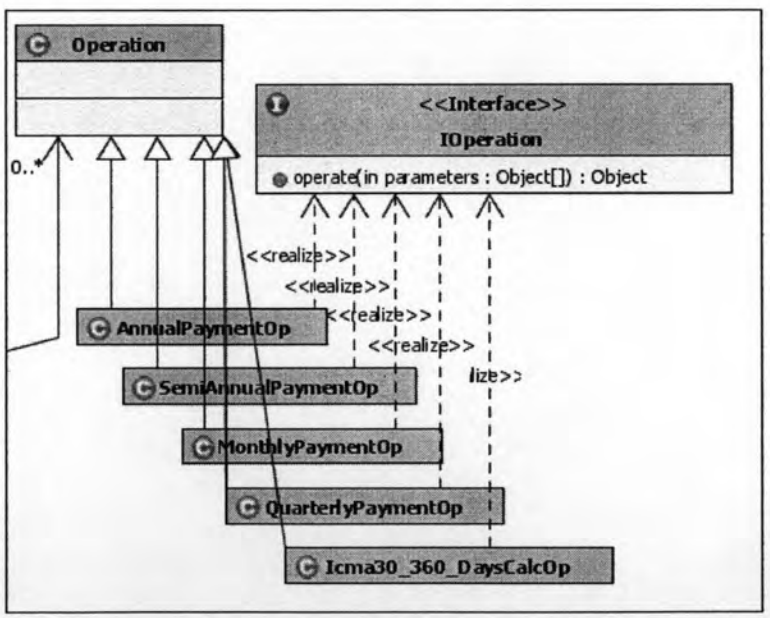
เมื่อประยุกต์วิธีการแปลงให้อยู่ในรูปของยูนิตโมเดล จะพบว่ากลุ่มคลาสที่เป็น Knowledge Level และ Operational Level ทำให้สามารถแปลงเป็นยูนิตโมเดลได้ดังรูปที่ 4.8



รูปที่ 4.8 อะตอมยูนิตของ au_aom.relationship และ au_aom.entity

ยูนิต au_aom.relationship นั้นจะเอ็กซ์พอร์ตแพ็กเกจซิกเนเจอร์ที่เพียงพอต่อการนำไปใช้ในการเพิ่มความสามารถในการปรับเปลี่ยนความสัมพันธ์ออกมาให้ยูนิตอื่นนำไปใช้ ส่วนคลาสอื่นๆจะถูกซ่อนไว้ภายในยูนิต ได้แก่ คลาส ConnectionRule

4.2.3.3 สตราทิจิแพทเทิร์น

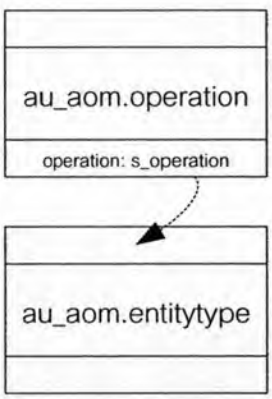


รูปที่ 4.9 แผนภาพคลาสและความสัมพันธ์ของสตราทิจิในระบบ CFGen

- อินเตอร์เฟส IOperation – เป็นอินเตอร์เฟสที่เป็นข้อกำหนดมาตรฐานให้แก่ทุกคลาสที่ทำหน้าที่เสมือนเป็นเมท็อด

- คลาส Operation – เป็นคลาสที่ทำหน้าที่แทนเมท็อดของเอนทิตี และเป็นคลาสแม่สำหรับทุกคลาสที่ต้องการให้ระบบสามารถเรียกใช้ตอนรันไทม์ได้
- คลาส AnnualPaymentOp – เป็นคลาสที่ทำหน้าที่เป็นตัวแทนของเมท็อดการคำนวณวันจ่ายคูปองแบบรายปี และสืบทอดมาจากคลาส Operation รวมทั้งอิมพลิเมนต์อินเตอร์เฟส IOperation
- คลาส SemiAnnualPaymentOp, MonthlyPaymentOp และ QuarterlyPaymentOp เป็นคลาสที่ทำหน้าที่เป็นตัวแทนของเมท็อดการคำนวณวันจ่ายคูปองแบบรายครึ่งปี รายเดือน และ รายสามเดือน ตามลำดับ และสืบทอดมาจากคลาส Operation รวมทั้งอิมพลิเมนต์อินเตอร์เฟส IOperation เช่นเดียวกัน
- คลาส Icma30_360_DaysCalcOp – เป็นคลาสที่ทำหน้าที่เป็นตัวแทนของเมท็อดการคำนวณวันที่ใช้ในการคิดดอกเบี้ยแบบ ICMA

จากแผนภาพคลาสจะเห็นว่ามีคลาส Operation เป็นเอนทิตีที่สัมพันธ์กับคลาส EntityType เนื่องจากระบบ CFGen ต้องการเอนทิตีที่เป็นตัวแทนของเมท็อดและจัดเก็บข้อมูลและความสัมพันธ์ระหว่างเอนทิตีไทป์และโอเปอเรชัน ไว้ในระบบฐานข้อมูล เมื่อประยุกต์วิธีการแปลงให้อยู่ในรูปของยูนิตโมเดล ทำให้คลาส Operation จะถูกนำมากำหนดเป็นแพ็กเกจซิกเนเจอร์ให้ยูนิตด้วย ดังรูปที่ 4.10

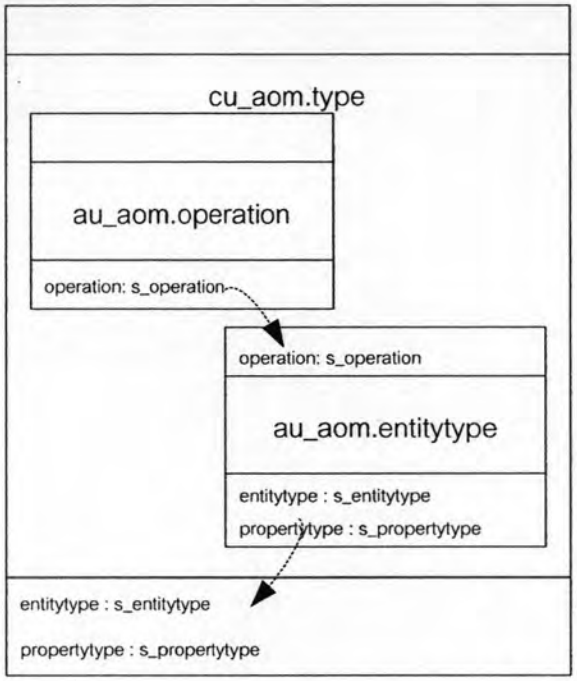


รูปที่ 4.10 อะตอมยูนิตของ au_aom.operation และ au_aom.entitytype

ยูนิตของ Operation นั้นจะเอ็กซ์พอร์ตแพ็กเกจซิกเนเจอร์ที่มีคลาสซึ่งสัมพันธ์กับคลาสที่ต้องการเรียกใช้เมท็อดและเพียงพอต่อการเรียกเมท็อดภายในยูนิต ส่วนคลาสที่อิมพลิเมนต์อินเตอร์เฟส IOperation หรือสืบทอดมาจาก Operation หรือ คลาสอื่นๆนั้นจะถูกซ่อนไว้ภายในยูนิต

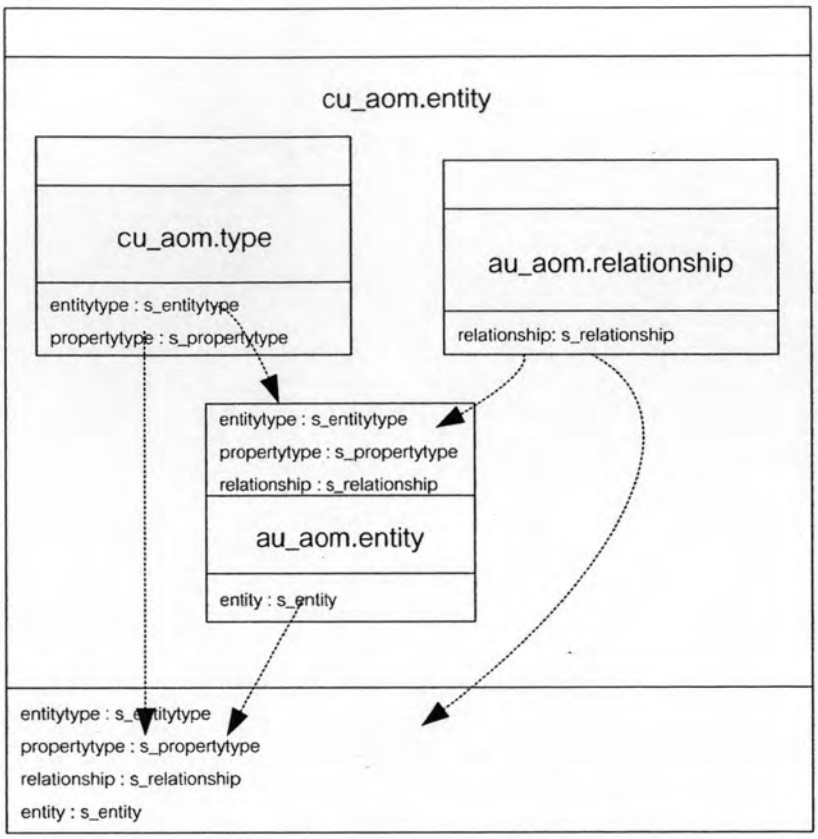
เนื่องจากยูนิตโมเดลนั้นสนับสนุนให้สามารถสร้างยูนิตใหม่ที่เกิดจากยูนิตเดิมนำมารวมกันและซ่อนรายละเอียดไว้ภายใน ทำให้ยูนิตใหม่ที่ได้มีความสามารถในการทำงานเพิ่มขึ้น

ดังนั้นเมื่อทำการแปลงดีไซน์แพทเทิร์นแต่ละให้อยู่ในรูปของยูนิตแล้ว ผู้ออกแบบสามารถนำแต่ละยูนิตมารวมกันเพื่อให้เกิดความสะดวกต่อการนำไปใช้งานและนักพัฒนาไม่จำเป็นต้องทราบรายละเอียดของการออกแบบ ทำให้สามารถพัฒนาแต่ละยูนิตได้อิสระและรวดเร็วมากขึ้น



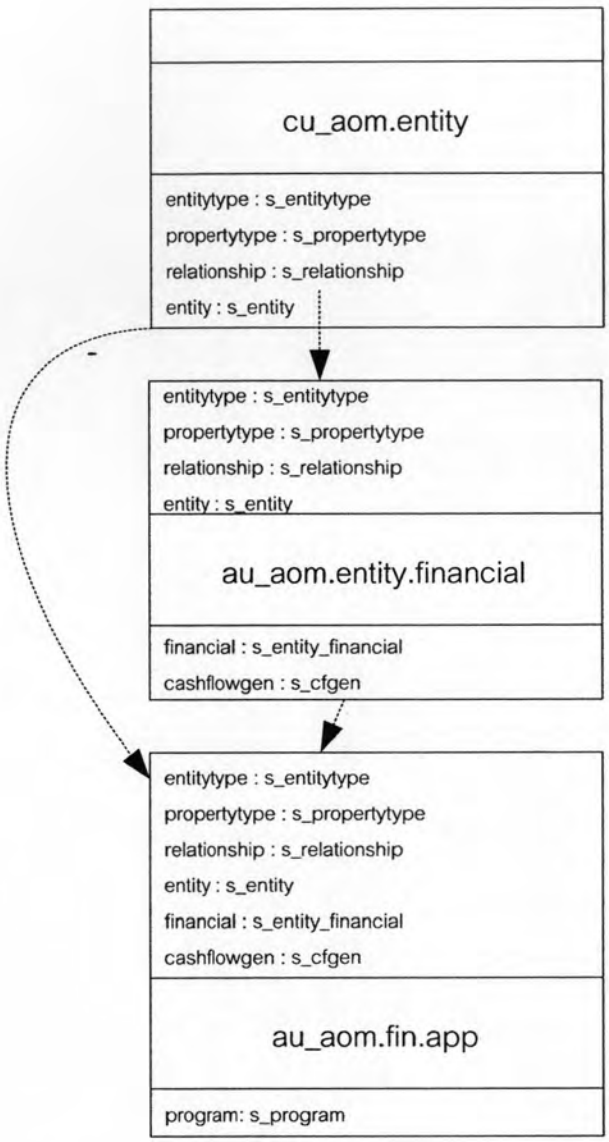
รูปที่ 4.11 คอมพาวด์ยูนิต cu_aom.type ซึ่งเกิดจากการเชื่อมต่อกันระหว่างอะตอมยูนิตของ au_aom.operation และ au_aom.entitytype

จากรูปที่ 4.11 จะเห็นได้ว่าเมื่อทำการแปลงสตราทิจิและไทป์สแควร์แพทเทิร์นให้อยู่ในรูปของยูนิตโมเดลแล้วผู้ออกแบบสามารถนำมารวมกันเป็นยูนิตใหม่ได้ จากนั้นก็นำยูนิต cu.aom.type ไปรวมกับยูนิตที่เกิดจากแปลงแอ็คเคาน์ทะบิลิตีแพทเทิร์น ทำให้ได้ยูนิต cu_aom.entity ที่สามารถนำไปใช้เพิ่มความสามารถให้แก่ยูนิตเอนทิตีของโดเมนต่างๆ ทำให้เอนทิตีนั้นสามารถปรับเปลี่ยนได้ตามความต้องการที่เปลี่ยนแปลง ดังรูปที่ 4.12



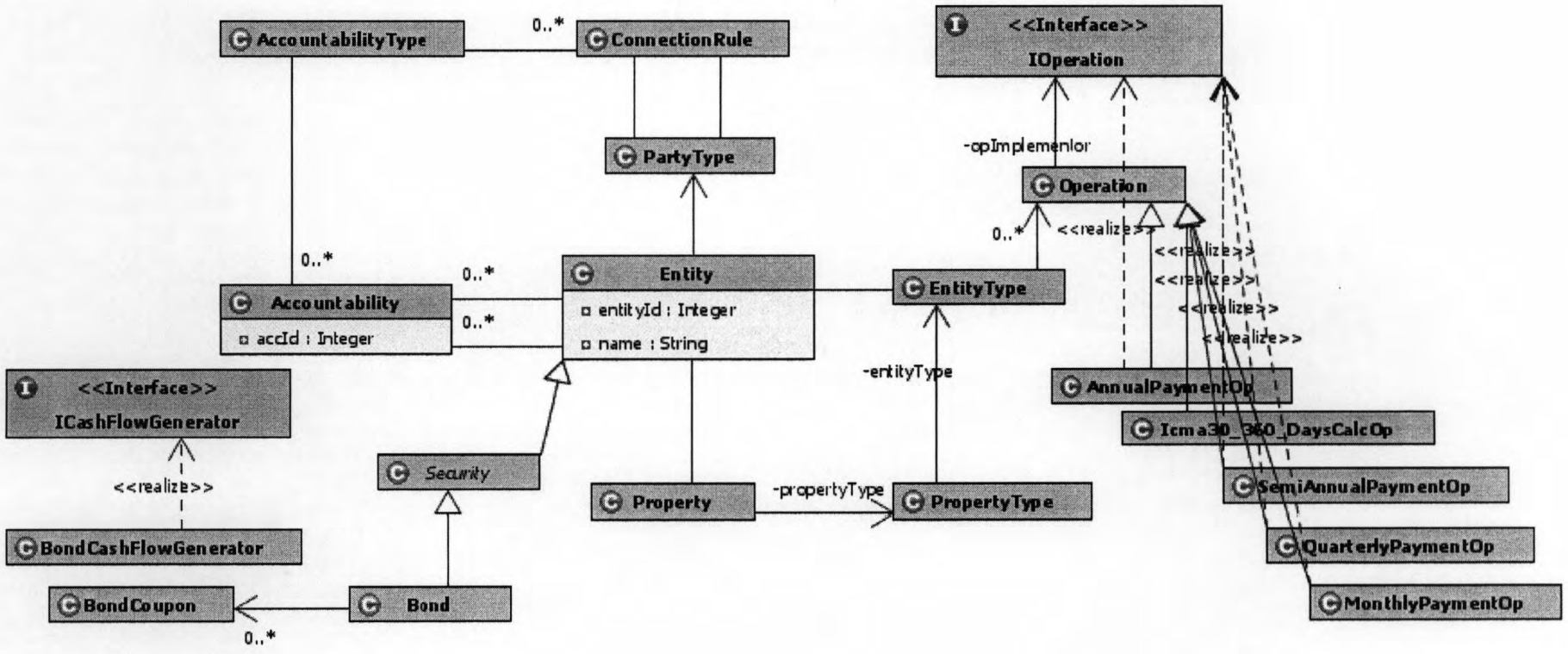
รูปที่ 4.12 คอมพาวด์ยูนิต cu_aom.entity ซึ่งเกิดจากการเชื่อมต่อกันระหว่างอะตอมยูนิตของ cu_aom.type, au_aom.relationship และ au_aom.entity

หลังจากที่ได้คอมพาวด์ยูนิต cu_aom.entity ผู้ออกแบบสามารถนำยูนิตไปใช้กับระบบ CFGen เพื่อให้เอนทิตีของระบบสามารถปรับเปลี่ยนได้ โดยอยู่ในรูปของยูนิต au_aom.entity.financial ที่อิมพอร์ต cu_aom.entity และเขียนเอนทิตีคลาสให้ทำการสืบทอดคลาส Entity ที่ยูนิตอิมพอร์ตเข้ามา ดังแสดงในรูปที่ 4.13



รูปที่ 4.13 การเชื่อมต่อกันของยูนิต cu_aom.entity, au_aom.entity.financial และ au_aom.fin.app สำหรับระบบ CFGen

เนื่องจากยูนิตโมเดลที่ได้จากการออกแบบเป็นแผนภาพของสถาปัตยกรรมที่อยู่ในระดับสูงกว่าแผนภาพคลาส ดังนั้น เพื่อให้สามารถเปรียบเทียบค่าความซับซ้อนหลังจากแปลงเป็นยูนิตโมเดลได้ จะต้องนำเอาแผนภาพคลาสของแอพพลิเคชันแบบยูนิตมาแสดงดังรูปที่ 4.14 เพื่อทำการวัดค่าความซับซ้อนนำไปเปรียบเทียบกับเอโอเอ็มแอพพลิเคชันแบบเดิม ผลการทดลองและเปรียบเทียบจะแสดงรายละเอียดอยู่ในบทต่อไป



รูปที่ 4.14 แผนภาพคลาสของเอโอเอ็มระบบ CFGen หลังจากแปลงเป็นยูนิตโมเดล

4.2.4 การตรวจสอบการทำงานของโปรแกรม

โปรแกรมที่ได้จากการแปลงให้อยู่ในรูปของยูนิตโมเดลนั้นจะต้องมีฟังก์ชันการทำงานที่เหมือนกับโปรแกรมเอโอเอ็มเดิม ดังนั้น เพื่อตรวจสอบความถูกต้องของการทำงานของโปรแกรม จึงทำการทดสอบโดยใช้ข้อมูลทดสอบชุดเดียวกัน เพื่อตรวจสอบว่าผลที่ได้จากการทำงานของทั้งสองโปรแกรมนั้นตรงกันหรือไม่ ดังตารางที่ 5

ตารางที่ 5 ผลการทดสอบการทำงานของระบบ CFGen ทั้งสองแบบ

ระบบ	วิธีทดสอบ	ข้อมูลอินพุต	ผลลัพธ์ที่ได้
ระบบ CFGen แบบเอโอเอ็มเดิม	สร้างกระแสเงินสด	ข้อมูลตราสารหนี้จำนวน 3 ชุดที่มีคุณสมบัติแตกต่างกัน	■ ชุดที่ 1 ได้งวดกระแสเงินสดจำนวน 60 งวด ■ ชุดที่ 2 ได้งวดกระแสเงินสดจำนวน 4 งวด ■ ชุดที่ 3 ได้งวดกระแสเงินสดจำนวน 10 งวด
ระบบ CFGen แบบเอโอเอ็มที่ใช้ยูนิตโมเดล			■ ชุดที่ 1 ได้งวดกระแสเงินสดจำนวน 60 งวด ■ ชุดที่ 2 ได้งวดกระแสเงินสดจำนวน 4 งวด ■ ชุดที่ 3 ได้งวดกระแสเงินสดจำนวน 10 งวด

ผลการทดสอบพบว่าข้อมูลที่ได้จากการทำงานของระบบ CFGen เดิมและ CFGen ที่ใช้ยูนิตโมเดลนั้นตรงกันทั้งหมด ไม่ว่าจะเป็นจำนวนงวดกระแสเงินสดและรายละเอียดของข้อมูลอื่นๆ เช่น วันที่จ่าย, จำนวนเงินต้น, จำนวนเงินทั้งหมดที่ต้องจ่าย เป็นต้น

4.3 ระบบงานทะเบียนข้อมูลสินค้า

4.3.1 ปัญหาทางธุรกิจ (Problem Statement)

งานทะเบียนสินค้า ในระบบทั่วๆ ไป อาทิเช่น ธุรกิจผลิตสินค้า หรือธุรกิจตัวแทนจำหน่ายสินค้า ฯลฯ มักพบกับปัญหาว่าระบบที่ซื้อมานั้น ไม่สามารถเก็บรายละเอียดของสินค้าได้ตรงกับลักษณะสินค้า (Product Characteristic) ของตน

รวมทั้งกรณีที่ในอนาคตหากปัจจัยภายนอกมีการเปลี่ยนแปลงไป อาทิเช่น การแข่งขันในอุตสาหกรรม, เทคโนโลยี, กฎหมาย ฯลฯ ส่งผลทำให้ระบบต้องการเพิ่มเติมปรับเปลี่ยนคุณลักษณะ

ของสินค้าหรือกระบวนการทำงานบางอย่าง ก็ทำได้ยากเนื่องจากคลาส Product มีแอตทริบิวต์และเมท็อดที่ไม่สามารถปรับเปลี่ยนได้

นอกจากนี้ในมุมมองของบริษัทผู้พัฒนาซอฟต์แวร์เอง ซึ่งเป็นผู้เชี่ยวชาญในการผลิตโปรแกรมในหลายโดเมนขึ้นอยู่กับธุรกิจของลูกค้านั้น ก็มักจะต้องเริ่มต้นออกแบบคลาส Product ใหม่ทั้งหมดสำหรับ รูปแบบสินค้าหรือบริการ ที่แตกต่างกัน โดยไม่สามารถนำคลาส Product กลับมาใช้ใหม่ได้ (Reusable)

โดยทั่วไปคลาส Product มีจะมีรายละเอียดของแอตทริบิวต์ซึ่งสามารถกำหนดเป็น Abstract Class ได้ดังนี้

- Product Id เลขที่สินค้า
- Product Code รหัสสินค้า
- Product Name ชื่อสินค้า
- Product Description รายละเอียด

ในส่วนของคุณสมบัติของสินค้าก็มักจะมีความแตกต่างกันไปตามแต่ละสินค้าหรือธุรกิจ คุณสมบัติที่เพิ่มเติมจาก Abstract Class ของคลาส Product สำหรับธุรกิจต่างๆ มีตัวอย่างดังนี้

Product: **เฟอร์นิเจอร์** ระบบสำหรับเว็บไซต์ขายสินค้าเฟอร์นิเจอร์

- Size ขนาด
- Color สี
- Transfer Cost ค่าขนส่ง

Product: **สีทาบ้าน** ระบบสำหรับร้านค้าปลีกผลิตภัณฑ์ก่อสร้าง

- Special Properties 1 กันน้ำได้
- Special Properties 2 ทำความสะอาดตัวเองได้

Product: **อาหาร** ระบบสำหรับร้านอาหารขายปลีก

- Expired Date วันหมดอายุ

Product: **คอมพิวเตอร์และอุปกรณ์ไอที** ระบบสำหรับร้านค้าปลีกอุปกรณ์ไอที

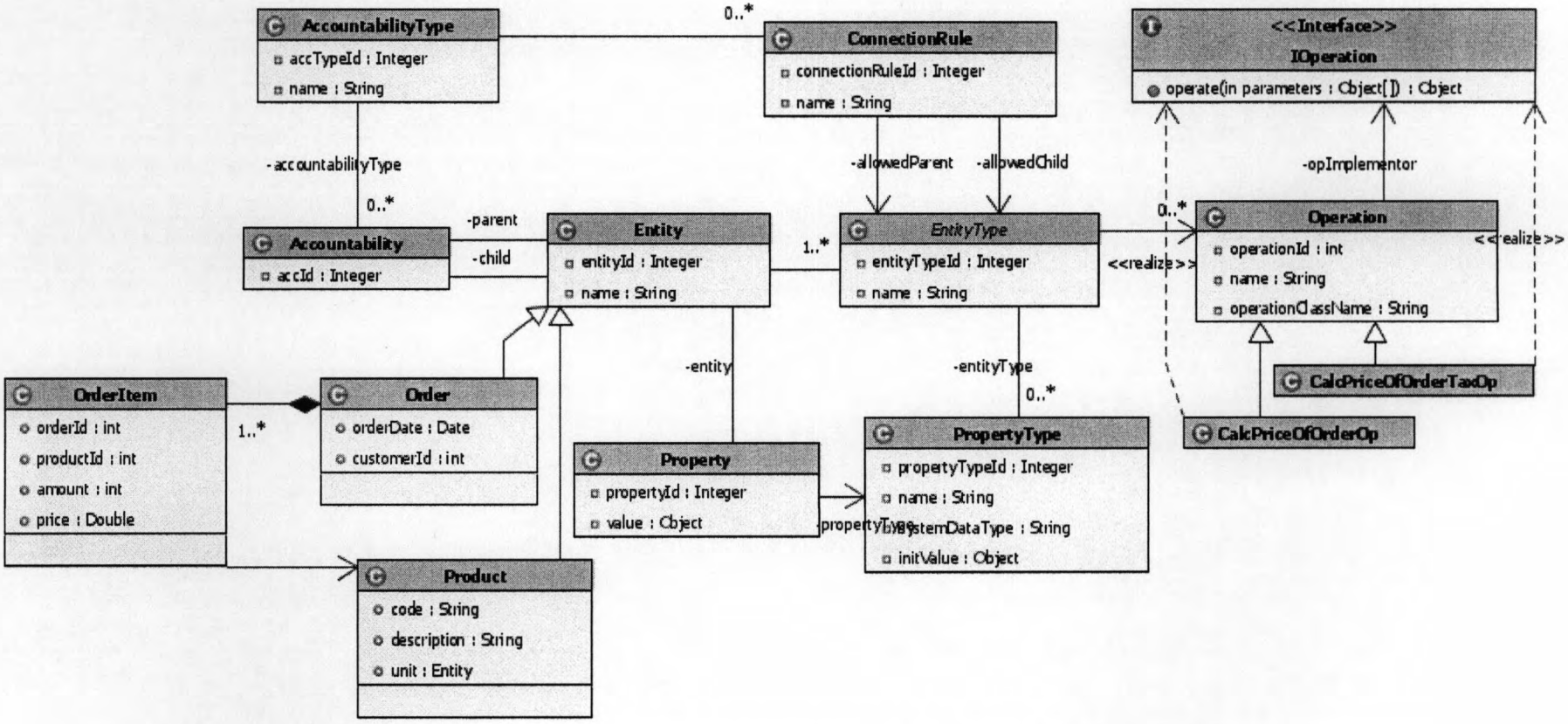
- CPU หน่วยประมวลผลกลาง
- Memory หน่วยความจำหลัก
- HDD ฮาร์ดดิสก์ไดรฟ์

ซึ่งความหลากหลายนี้ทำให้ อ็อบเจกต์โมเดลที่มีโครงสร้างแบบคงที่ไม่มีสามารถปรับเปลี่ยนให้รองรับกับการเปลี่ยนแปลงความต้องการที่มีอยู่อย่างต่อเนื่องได้ ด้วยปัญหาดังกล่าวระบบที่พัฒนาขึ้นโดยวิธีการแบบเอโอเอ็ม จะทำให้ระบบมีความยืดหยุ่นมากขึ้น โดยมีคุณสมบัติดังนี้

- สามารถเพิ่มลด ประเภทของสินค้าได้
- สามารถเพิ่มลด คุณสมบัติของสินค้าได้
- สามารถเพิ่มลด วิธีการคำนวณราคาขาย ได้แก่ การคำนวณราคาสินค้าของใบสั่งซื้อ/ขาย, การคำนวณกำไรขาดทุน ณ สิ้นเดือน (สินค้าซื้อและขายภายในเดือนเดียวกัน)
- เพิ่มลด ความสัมพันธ์ระหว่างเอนทิตี ได้แก่ ความสัมพันธ์ระหว่าง คอมพิวเตอร์ 1 เครื่อง ประกอบด้วยอุปกรณ์ต่างๆ เช่น CPU, Memory, HDD

4.3.2 การออกแบบแอพพลิเคชัน

4.3.2.1 แผนภาพคลาส



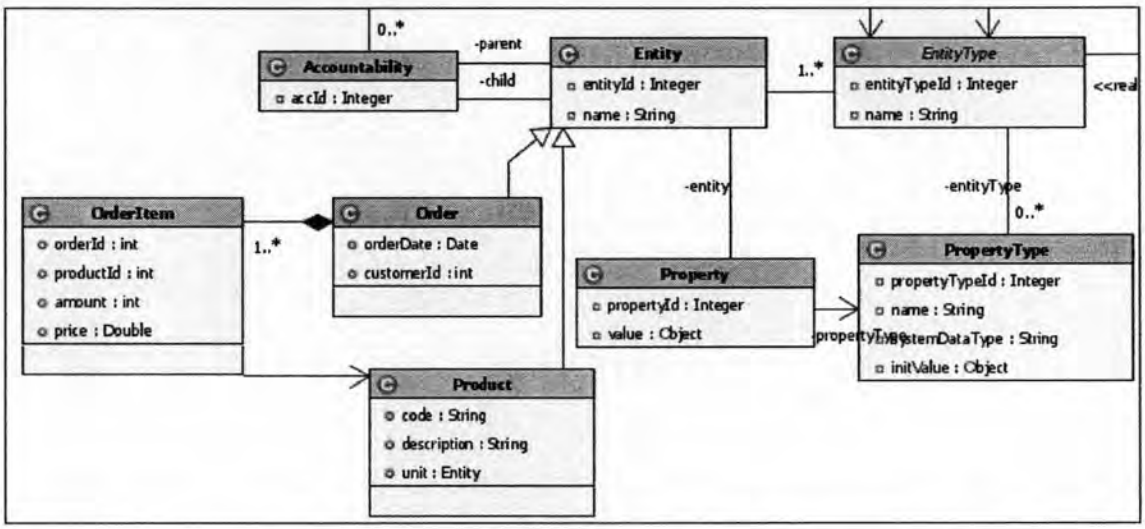
รูปที่ 4.15 แผนภาพคลาสแบบเอโอเอ็มของระบบทะเบียนสินค้าก่อนแปลงเป็นยูนิตโมเดล

รูปที่ 4.15 แสดงแผนภาพคลาสของระบบทะเบียนสินค้า ซึ่งออกแบบโดยประยุกต์ใช้ดีไซน์แพทเทิร์นแบบต่างๆเพื่อให้อ็อบเจกต์สามารถปรับเปลี่ยนได้ตามแนวคิดของเอโอเอ็ม

4.3.3 การแปลงจากเอโอเอ็มเป็นยูนิตโมเดล

จากแผนคลาสดังกล่าว จะพบว่ากลุ่มคลาสต่างๆที่ออกแบบด้วยแพทเทิร์นสำหรับเอโอเอ็มมีดังนี้

4.3.3.1 ไทป์สแควร์แพทเทิร์น

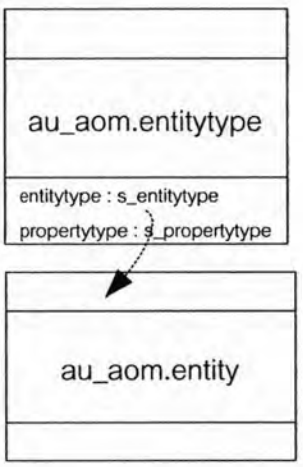


รูปที่ 4.16 แผนภาพคลาสและความสัมพันธ์ของไทป์สแควร์แพทเทิร์นระบบทะเบียนสินค้า

- คลาส Entity – เป็นเอนทิตีคลาสของระบบ โดยมีเอนทิตีไทป์เป็นคลาสที่กำหนดประเภทให้กับเอนทิตี
- คลาส EntityType – เป็นคลาสของประเภทของเอนทิตี ซึ่งในระบบงานทะเบียนสินค้านั้น ขึ้นอยู่กับประเภทสินค้าที่ต้องการกำหนด ตัวอย่างเช่น สินค้าประเภทสี, คอมพิวเตอร์, อุปกรณ์ชิ้นส่วนคอมพิวเตอร์ เป็นต้น สินค้าแต่ละประเภทก็สามารถเพิ่มลดพรอบเพอร์ตีของสินค้าได้ เช่น ปริมาตรของภาชนะบรรจุสี, สี, ความสามารถในการกันน้ำ, ความสามารถในการทำความสะอาดตนเอง ซึ่งคุณสมบัติเหล่านี้จะถูกกำหนดเป็นอ็อบเจกต์ของพรอบเพอร์ตีไว้ในคลาส PropertyType
- คลาส Property – เป็นคลาสทำหน้าที่ตัวแทนของค่าพรอบเพอร์ตีแต่ละประเภท
- คลาส PropertyType – เป็นคลาสตัวแทนของพรอบเพอร์ตีประเภทต่างๆ
- คลาส Product – เป็นคลาสของสินค้า คลาสนี้ถูกสร้างขึ้นมาเนื่องจากเป็นเอนทิตีหลักที่เกี่ยวของกับโดเมนนี้โดยตรง การกำหนดคลาสนี้ขึ้นมาทำให้สะดวกต่อการพัฒนา นอกจากนั้น คลาส Product นี้ยังสืบทอดมาจากคลาสเอนทิตี ก็เพื่อให้สามารถเพิ่มลดประเภทของคลาส Product ได้ลดการกำเนิดคลาสใหม่แทนการสร้างคลาสลูกของ Product

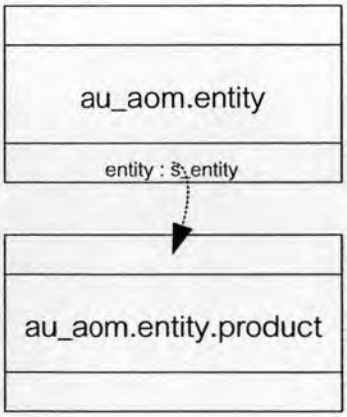
- คลาส Order – เป็นคลาสของใบสั่งเกี่ยวกับสินค้า เช่น ใบสั่งซื้อ, ใบสั่งขาย, ใบสั่งขนย้าย เป็นต้น ประเภทของใบสั่งรวมถึงพรอบเพอร์ตีต่างๆ เช่น ใบสั่งซื้อ หรือ ขาย ก็อาจแตกต่างกันไป สิ่งเหล่านี้สามารถเพิ่มลดได้ภายหลัง เนื่องจากคลาสนี้สืบทอดมาจากคลาส Entity
- คลาส OrderItem – เป็นคลาสที่เกี่ยวกับรายการสินค้าของใบสั่งต่างๆ เนื่องจากคลาสนี้เป็นคลาสที่มีพรอบเพอร์ตีที่แน่นอน ทำให้คลาสนี้ไม่จำเป็นต้องสืบทอดความสามารถของคลาส Entity

จากแผนภาพคลาสสามารถแปลงเป็นยูนิต ตามแนวทางที่นำเสนอ โดยเลือกและแบ่งคลาสที่เป็นเมต้าคลาสและเอนทิตีคลาสของไทป์สแควร์แพทเทิร์น ซึ่งคลาสที่อยู่ในกลุ่มของเมต้าคลาสคือ EntityType, PropertyType นำมาสร้างเป็นยูนิตที่ชื่อ au_aom.entitytype และกลุ่มเอนทิตีคลาสได้แก่ Entity, Property, Order, OrderItem, Product นำกำหนดเป็นยูนิตชื่อ au_aom.entity ดังรูปที่ 4.17



รูปที่ 4.17 อะตอมยูนิตของ au_aom.entitytype และ au_aom.entity

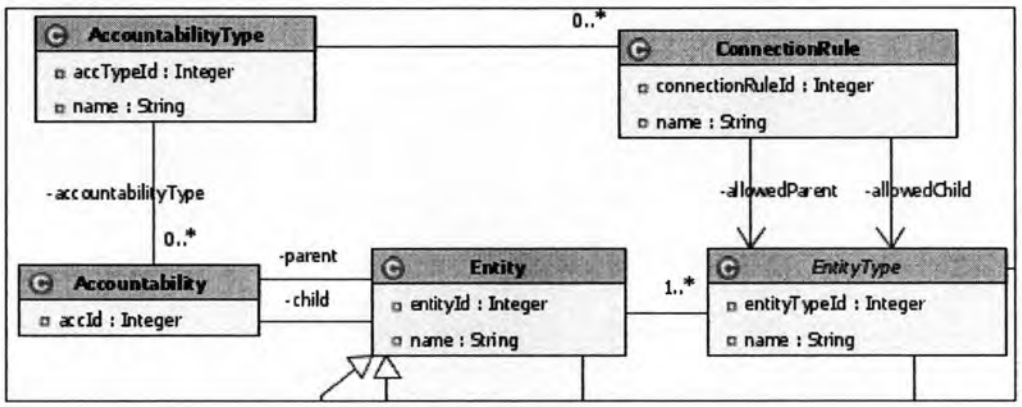
เพื่อให้ยูนิตสามารถนำกลับมาใช้ใหม่ได้ จึงต้องแยกยูนิตของที่เกี่ยวข้องกับเฉพาะโดเมนออกจากกัน ทำให้ได้ยูนิตสำหรับโดเมนอีกหนึ่งยูนิตคือ au_aom.entity.product ดังรูปที่ 4.18



รูปที่ 4.18 อะตอมยูนิตของ au_aom.entity และ au_aom.entity.product

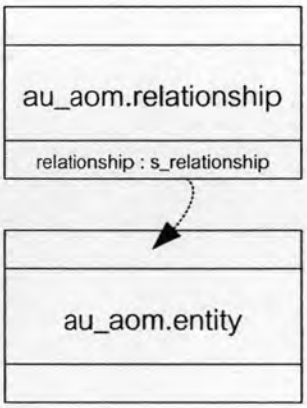
4.3.3.2 แอ็คเคาน์ทะบิลิตีแพทเทิร์น

- คลาส AccountabilityType – เป็นคลาสประเภทของความสัมพันธ์ระหว่างเอนทิตีที่สนใจ
- คลาส ConnectionRule – เป็นคลาสที่กำหนดกฎความสัมพันธ์ระหว่างประเภทเอนทิตี (EntityType) ซึ่งอ็อบเจกต์ของคลาสนี้จะเป็นส่วนหนึ่งของอ็อบเจกต์แอ็คเคาน์ทะบิลิตีไทป์ เช่น ผู้ค้ำประกันจะต้องเป็นธนาคาร(Parent) และผู้ออกตราสารที่ได้รับการค้ำประกันต้องเป็นบริษัทที่จดทะเบียนฯ (Child)
- คลาส EntityType – เป็นคลาสของประเภทของเอนทิตี
- คลาส Accountability - เป็นคลาสของอ็อบเจกต์ความสัมพันธ์ที่เกิดขึ้นระหว่างเอนทิตี
- คลาส Entity – เป็นเอนทิตีคลาสของระบบ ซึ่งในที่นี้เป็นคลาสแม่ของเอนทิตีสำหรับโดเมนนี้



รูปที่ 4.19 แผนภาพคลาสและความสัมพันธ์ของแอ็คเคาน์ทะบิลิตีแพทเทิร์นระบบทะเบียนสินค้า

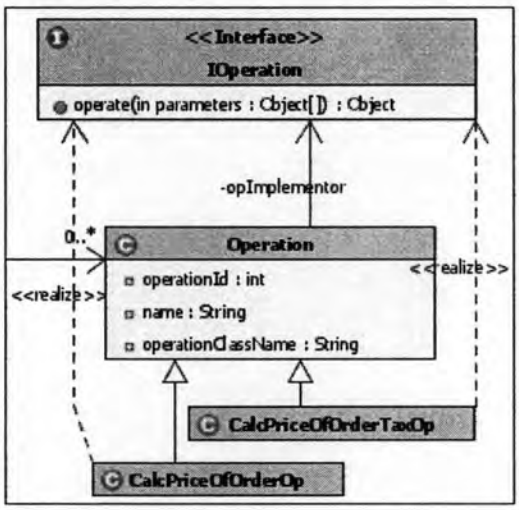
เมื่อประยุกต์วิธีการแปลงให้อยู่ในรูปของยูนิตโมเดล จะพบว่ากลุ่มคลาสที่เป็น Knowledge Level ได้แก่ AccountabilityType, ConnectionRule และ EntityType จะถูกกำหนดเป็นยูนิต au_aom.relationship สำหรับ Operational Level ได้แก่ Entity และ Accountability จะถูกสร้างเป็นยูนิต au_aom.entity ดังรูปที่ 4.20



รูปที่ 4.20 อะตอมยูนิตของ au_aom.relationship และ au_aom.entity

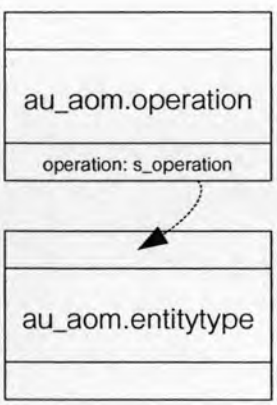
4.3.3.3 สตราทิจิแพทเทิร์น

- อินเตอร์เฟส IOperation – เป็นอินเตอร์เฟสที่เป็นข้อกำหนดมาตรฐานให้แก่ทุกคลาสที่ทำหน้าที่เสมือนเป็นเมท็อด
- คลาส Operation – เป็นคลาสที่ทำหน้าที่แทนเอนทิตีของเมท็อด และเป็นคลาสแม่สำหรับทุกคลาสที่ต้องการให้ระบบสามารถเรียกใช้ตอนรันไทม์ได้
- คลาส CalcPriceOfOrderOp – เป็นคลาสตัวแทนของเมท็อดการคำนวณราคาของใบสั่งซื้อขาย โดยสืบทอดมาจากคลาส Operation พร้อมทั้งอิมพลิเมนต์อินเตอร์เฟส IOperation ด้วย
- คลาส CalcPriceOfOrderTaxOp – เป็นคลาสตัวแทนของเมท็อดการคำนวณราคาของใบสั่งซื้อขายรวมถึงคิดภาษี ณ ที่จ่าย



รูปที่ 4.21 แผนภาพคลาสและความสัมพันธ์ของสตราทิจิระบบทะเบียนสินค้า

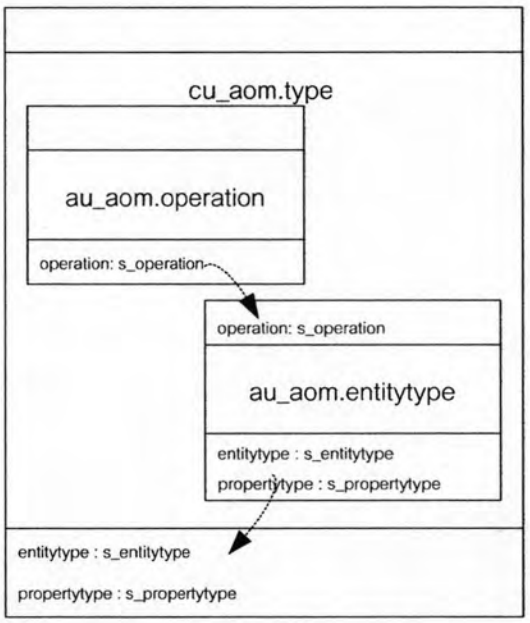
จากแผนภาพคลาสจะเห็นว่ามีคลาส Operation เป็นเอนทิตีที่สัมพันธ์กับคลาส EntityType เมื่อประยุกต์วิธีการแปลงให้อยู่ในรูปของยูนิตโมเดล ทำให้ได้ยูนิตสำหรับจัดการเรื่องที่เกี่ยวข้องกับการจัดการฟังก์ชันต่างๆ ทำให้คลาส Operation จะถูกนำมากำหนดเป็นแพ็กเกจซิกเนเจอร์ของยูนิต au_aom.operation ดังรูปที่ 4.22



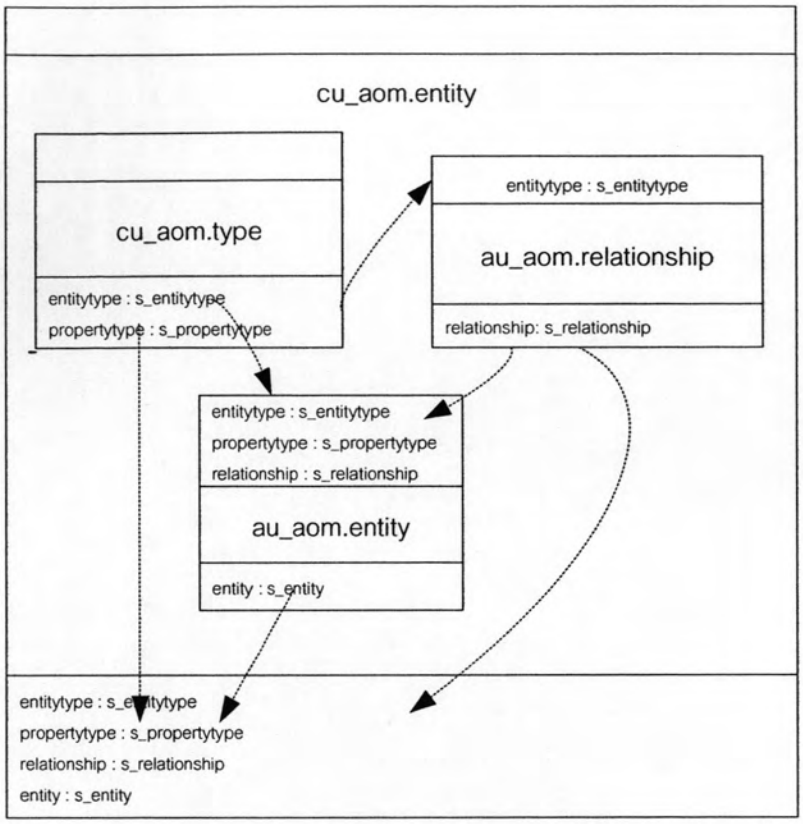
รูปที่ 4.22 อะตอมยูนิตของ au_aom.operation และ au_aom.entitytype

ยูนิต au_aom.operation นั้นจะเอ็กซ์พอร์ตแพ็กเกจซิกเนเจอร์ที่มีคลาสซึ่งสัมพันธ์กับคลาสที่ต้องการเรียกใช้เมท็อดและเพียงพอต่อการเรียกเมท็อดภายในยูนิต ส่วนคลาสที่อิมพลิเมนต์อินเตอร์เฟส IOperation หรือสืบทอดมาจาก Operation หรือ คลาสอื่นๆนั้นจะถูกซ่อนไว้ภายในยูนิต

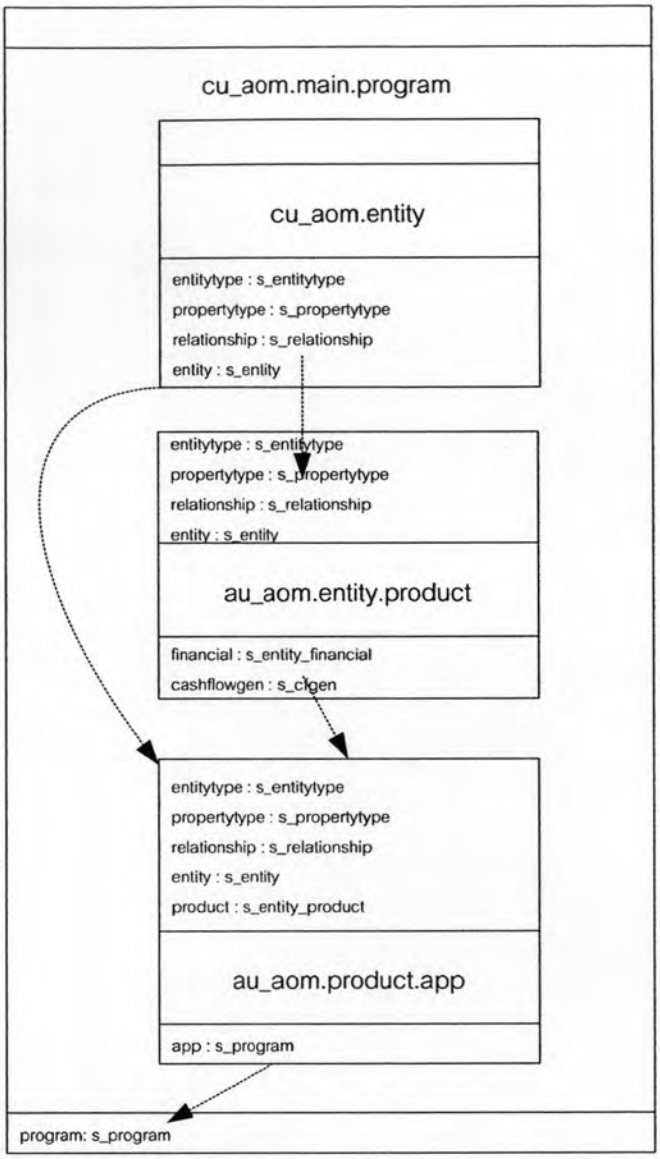
ตามที่ได้กล่าวไปแล้วว่า ยูนิตโมเดลนั้นสนับสนุนให้สามารถสร้างยูนิตใหม่ที่เกิดจากยูนิตเดิมนำมารวมกันและซ่อนรายละเอียดไว้ภายใน ทำให้ยูนิตใหม่ที่ได้มีความสามารถในการทำงานเพิ่มขึ้น เกิดความสะดวกต่อการนำไปใช้งานและนักพัฒนาไม่จำเป็นต้องทราบรายละเอียดภายใน ทำให้สามารถพัฒนาแต่ละยูนิตได้อิสระและรวดเร็วมากขึ้น ซึ่งคล้ายคลึงกับระบบ CFGen ดังแสดงในรูปที่ 4.23



รูปที่ 4.23 คอมพาวด์ยูนิต cu_aom.type ซึ่งเกิดจากการเชื่อมต่อกันระหว่างอะตอมยูนิตของ au_aom.operation และ au_aom.entitytype

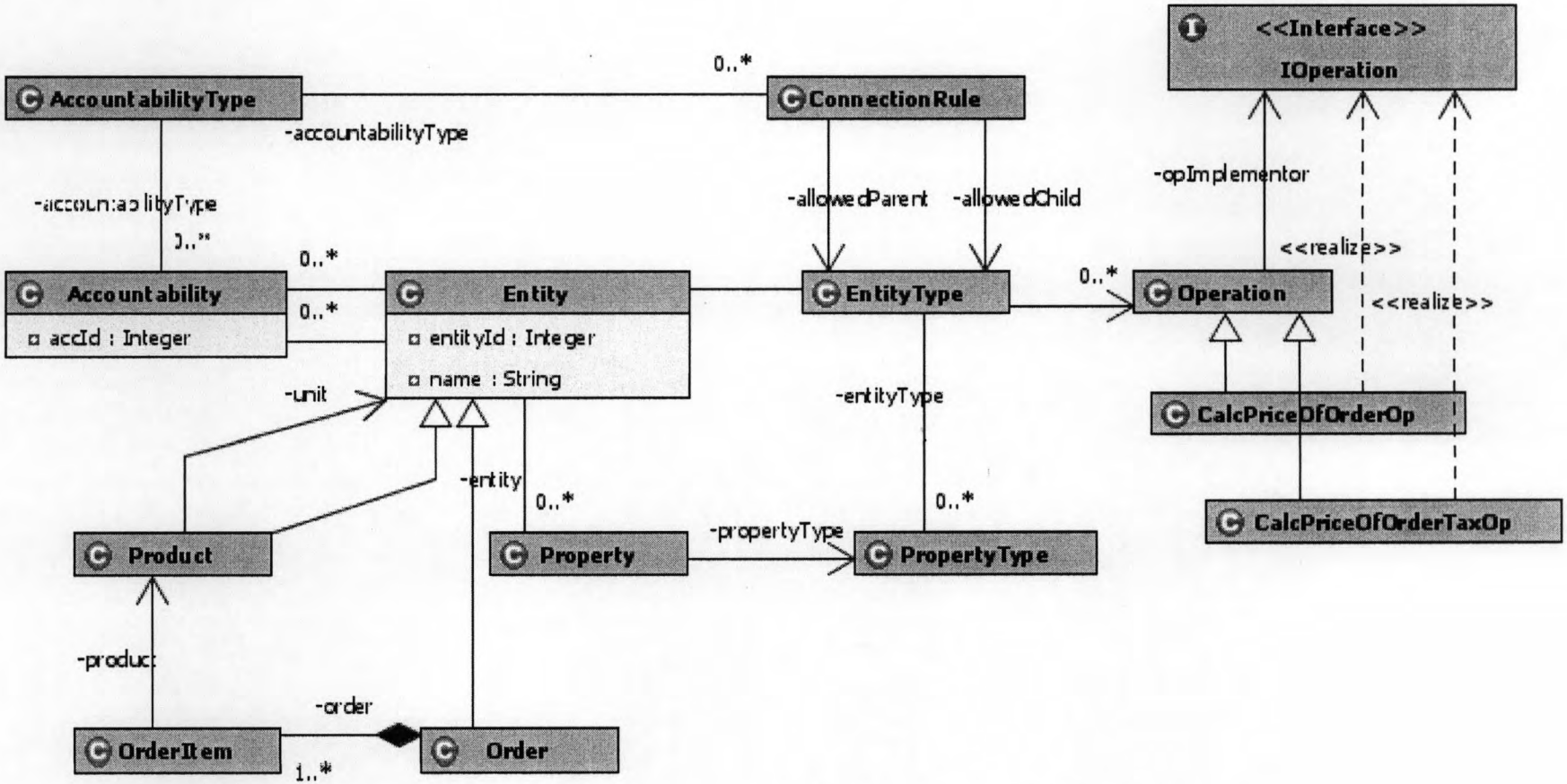


รูปที่ 4.24 คอมพาวด์ยูนิต cu_aom.entity ซึ่งเกิดจากการเชื่อมต่อกันระหว่างอะตอมยูนิตของ cu_aom.type, au_aom.relationship และ au_aom.entity



รูปที่ 4.25 การเชื่อมต่อกันของยูนิต cu_aom.entity, au_aom.entity.product และ au_aom.product.app สำหรับระบบงานทะเบียนสินค้า

เนื่องจากยูนิตโมเดลที่ได้จากการออกแบบเป็นแผนภาพของสถาปัตยกรรมที่อยู่ในระดับสูงกว่าแผนภาพคลาส ดังนั้น เพื่อให้สามารถเปรียบเทียบค่าความซับซ้อนหลังจากแปลงเป็นยูนิตโมเดลได้ จะต้องนำเอาแผนภาพคลาสของแอพพลิเคชันแบบยูนิตมาแสดงดังรูปที่ 4.26 เพื่อทำการวัดค่าความซับซ้อนนำไปเปรียบเทียบกับเอโอเอ็มแอพพลิเคชันแบบเดิม



รูปที่ 4.26 แผนภาพคลาสของเอโอเอ็มระบบทะเบียนสินค้าหลังจากแปลงเป็นยูนิตโมเดล

4.3.4 การตรวจสอบการทำงานของโปรแกรม

โปรแกรมที่ได้จากการแปลงให้อยู่ในรูปของยูนิตโมเดลนั้นจะต้องมีฟังก์ชันการทำงานที่เหมือนกับโปรแกรมเอโอเอ็มเดิม ดังนั้น เพื่อตรวจสอบความถูกต้องของการทำงานของโปรแกรม จึงทำการทดสอบโดยการใช้ข้อมูลทดสอบชุดเดียวกัน เพื่อตรวจสอบว่าผลที่ได้จากการทำงานของทั้งสองโปรแกรมนั้นตรงกันหรือไม่ ดังตารางที่ 6

ตารางที่ 6 ผลการทดสอบการทำงานของระบบทะเบียนสินค้าทั้งสองแบบ

ระบบ	วิธีทดสอบ	ข้อมูลอินพุต	ผลลัพธ์ที่ได้
ระบบทะเบียนสินค้าแบบเอโอเอ็มเดิม	คำนวณกำไรขาดทุนของรายการสั่งซื้อและขายสินค้าออกไปโดยไม่คำนวณสินค้าคงเหลือในคลัง	คำสั่งซื้อและขายสินค้าอย่างละ 1 ชุด	▪ ตัวเลขขาดทุนเท่ากับ -13,000 บาท
ระบบทะเบียนสินค้าแบบเอโอเอ็มที่ใช้ยูนิตโมเดล			▪ ตัวเลขขาดทุนเท่ากับ -13,000 บาท

ผลการทดสอบพบว่าข้อมูลที่ได้จากการทำงานของระบบทะเบียนสินค้าแบบเอโอเอ็มเดิมและแบบที่ใช้ยูนิตโมเดลนั้นตรงกัน

4.4 ระบบวัดประสิทธิภาพโฆษณา

4.4.1 ปัญหาทางธุรกิจ (Problem Statement)

ในการวางแผนสื่อโฆษณาทั่วไป ใช้วิธีวัดตามอัตราการเข้าถึงผู้บริโภคแยกตามประเภทสื่อ ได้แก่ โทรทัศน์ 60% สื่อวิทยุ 20% สื่อสิ่งพิมพ์ 15% อื่นๆ 5% ผู้บริโภค มีการระบุว่าลูกค้าแต่ละกลุ่มมีความแตกต่างกันในด้านการรับรู้และเข้าถึงของสื่อโฆษณา ผลนั้นยังไม่มีการวัดออกมาชัดเจนว่า สื่อที่วางไว้มี ประสิทธิภาพมากน้อยแค่ไหน การวัดประสิทธิภาพของสื่อโฆษณาสามารถบอกถึงความคุ้มค่าของเม็ดเงินที่ลงทุนในการซื้อสื่อ ว่าโดยแท้จริงจะทำให้เข้าถึงผู้บริโภคมากน้อยแค่ไหน พฤติกรรมของ เอเยนซี่โฆษณาปัจจุบัน นิยมให้เจ้าของสินค้าแต่ละชนิดใช้สื่อโฆษณาที่มีลักษณะเป็นแคมเปญพิเศษเชิญชวนผู้ซื้อ แต่ยังขาดการวัดผล จึงต้องมีการพัฒนาวิธีการวัดผลการเข้าถึงของสื่อโฆษณาในกลุ่มลูกค้าของแต่ละแคมเปญ เพื่อให้การวางแผนงานมีประสิทธิภาพสูงกว่าการวัดเพียงสื่อใดสื่อหนึ่งเท่านั้น

ระบบวัดประสิทธิภาพโฆษณา มีวัตถุประสงค์เพื่อประมวลผลสื่อโฆษณาที่ใช้ในแต่ละแคมเปญ โดยนำข้อมูลที่ได้จากการสัมภาษณ์กลุ่มเป้าหมาย (ทัศนคติ ความนิยมในสินค้าแต่ละชนิด ระยะเวลาการจดจำโฆษณาแต่ละชนิดของแบรนด์สินค้า) มาวิเคราะห์ร่วมกับผลการวิจัยทางด้านสื่อเฉพาะแต่ละด้าน แล้วจึงนำข้อมูลที่ได้ทั้งหมดเข้าประมวลผลด้วยโปรแกรม ซึ่งสามารถครอบคลุมได้ทั้ง

สื่อโทรทัศน์, สื่อสิ่งพิมพ์, สื่อวิทยุ, สื่อโฆษณากลางแจ้ง, สื่อภาพยนตร์ และสื่ออินเทอร์เน็ต โดยตัวอย่างรายละเอียดของการวัดประสิทธิภาพของสื่อแต่ละประเภทแสดงในตารางที่ 7 ดังนี้

ตารางที่ 7 การวัดประสิทธิภาพของโฆษณาด้วยวิธีการต่างๆ

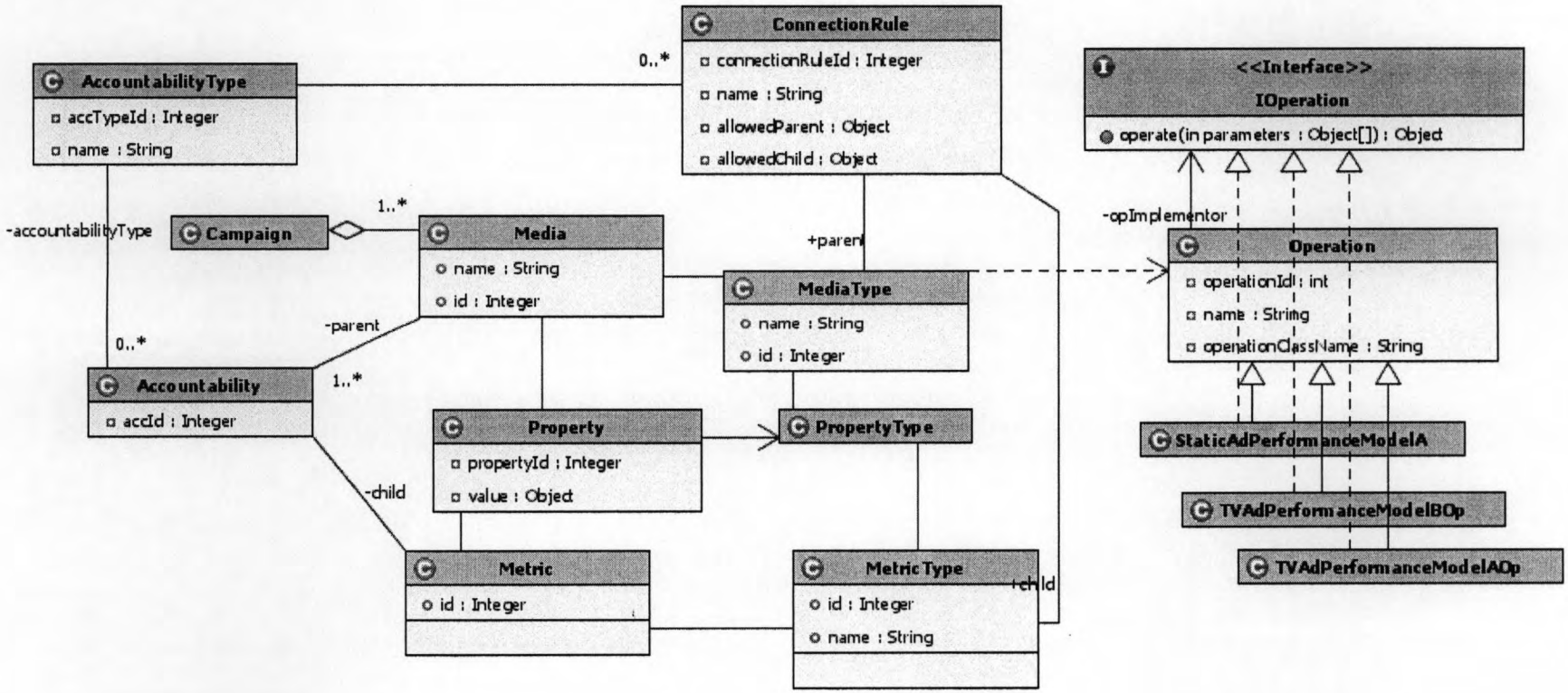
	โทรทัศน์	วิทยุ	หนังสือพิมพ์	นิตยสาร	สื่อกลางแจ้ง	Banner Ads	Interactive	SEO	Direct Mail
RATING (**หรือความนิยมรายการ**) – จำนวนตัวเลขที่บอกความนิยมที่ประชาชนมีต่อรายการ วัดจาก % ของบ้านที่เปิดชมรายการเปรียบเทียบกับ % ของครัวเรือนที่มีเครื่องรับโทรทัศน์ทั้งหมด									
	√	√							
TARP (AUDIENCE RATING POINT **หรือความนิยมรายการจากกลุ่มเป้าหมาย**) – ความนิยมรายการเฉพาะกลุ่มเป้าหมายที่เราต้องการ									
	√	√							
REACH (**การเข้าถึง**) – จำนวนคนที่ได้เห็นโฆษณานั้น ๆ อย่างน้อย 1 ครั้ง ใน 4 สัปดาห์ (%)									
	√								
FREQUENCY (**ความถี่เฉลี่ย**) – ความถี่เฉลี่ยของการได้เห็นโฆษณาของกลุ่มเป้าหมายใน 4 สัปดาห์(ครั้ง) จะได้ REACH * FREQUENCY = GRP (GROSS RATING POINT) * ครั้ง									
	√								
CPM (**ต้นทุนต่อพัน หรือ** COST PER THOUSAND) - ตัววัดประสิทธิภาพของการใช้เงินซื้อสื่อโฆษณาโดยเทียบเป็นหน่วยต่อพัน									
	√	√	√	√	√				
CPM (Cost Per Thousand **หรืออัตราค่าโฆษณา ต่อการขึ้นป้ายโฆษณา 1,000 ครั้ง**) – การซื้อพื้นที่โฆษณาตามจำนวนป้ายแบนเนอร์โฆษณาที่ถูกเรียกให้แสดงบนหน้าจอคอมพิวเตอร์ หรือเรียกว่าจำนวน Impression									
						√	√	√	
- CTR (Click Through Rate **หรืออัตราการคลิกที่ป้ายโฆษณา**) – เป็นหลักเกณฑ์การวัดจำนวนผู้เข้าชมเว็บไซต์ที่คลิกบน แถบป้ายโฆษณา เพื่อเชื่อมต่อไปยังเว็บไซต์ของโฆษณานั้นๆ									
						√	√	√	
CPA (Cost Per Action **หรืออัตราค่าโฆษณาต่อการ ตอบรับหรือการสั่งซื้อ**) – การซื้อพื้นที่โฆษณาตามจำนวนการตอบสนองเงื่อนไขของ									
						√	√	√	
Bounce rate - การรวมจำนวนอีเมล์ทั้งหมด ที่ส่งย้อนกลับมาที่เซิร์ฟเวอร์									
									√
Open rate - คิดในลักษณะเปอร์เซ็นต์ของอีเมล์ที่ผู้รับเปิดดู									
									√

ความหลากหลายของสื่อโฆษณาและข้อมูลที่ใช้ในการวัดอาจเปลี่ยนแปลงไป ทำให้ระบบไม่มีความยืดหยุ่น ด้วยปัญหาดังกล่าวระบบที่พัฒนาขึ้นโดยวิธีการแบบเอโอเอ็ม จะทำให้ระบบมีความยืดหยุ่นมากขึ้น โดยมีคุณสมบัติดังนี้

- สามารถเพิ่มลด ประเภทของสื่อโฆษณาได้
- สามารถเพิ่มลด คุณสมบัติของสื่อโฆษณาได้
- สามารถเพิ่มลด วิธีการคำนวณประสิทธิภาพของสื่อโฆษณา
- สามารถปรับเปลี่ยน ความสัมพันธ์ของสื่อโฆษณากับวิธีการวัดที่ใช้กับสื่อโฆษณานั้น เพื่อนำไปคำนวณหาประสิทธิภาพของโฆษณานั้นได้

4.4.2 การออกแบบแอพพลิเคชัน

4.4.2.1 แผนภาพคลาส



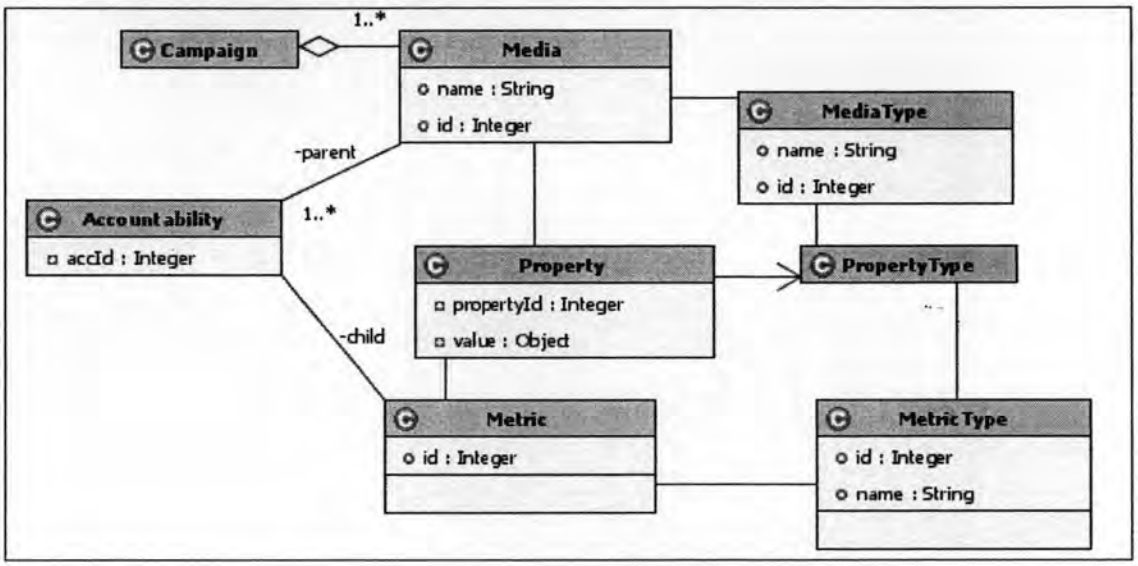
รูปที่ 4.27 แผนภาพคลาสแบบเอโอเอ็มของระบบวัดประสิทธิภาพโฆษณาก่อนแปลงเป็นยูนิตโมเดล

รูปที่ 4.27 แสดงแผนภาพคลาสของระบบวัดประสิทธิภาพโฆษณา ซึ่งออกแบบโดยประยุกต์ดีไซน์แพทเทิร์นแบบต่างๆเพื่อให้อ็อบเจกต์สามารถปรับเปลี่ยนได้

4.4.3 การแปลงจากเอโอเอ็มเป็นยูนิตโมเดล

จากแผนคลาสดังกล่าว จะพบว่ากลุ่มคลาสต่างๆที่ออกแบบด้วยแพทเทิร์นสำหรับเอโอเอ็มมีดังนี้

4.4.3.1 ไทป์สแควร์แพทเทิร์น



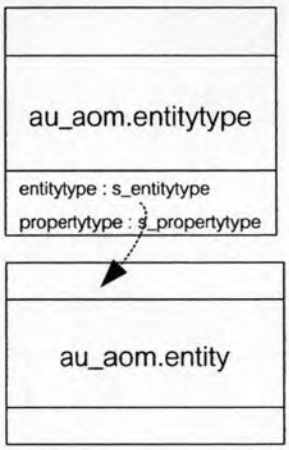
รูปที่ 4.28 แผนภาพคลาสและความสัมพันธ์ของไทป์สแควร์แพทเทิร์นระบบวัดประสิทธิภาพโฆษณา

- คลาส Campaign – เป็นเอนทิตีคลาสตัวแทนของแคมเปญ ซึ่งแต่ละแคมเปญจะประกอบด้วยสื่อโฆษณาไม่ว่าจะเป็น การโฆษณาทางโทรทัศน์ วิทยุ สิ่งพิมพ์ต่างๆ
- คลาส Media – เป็นเอนทิตีคลาสตัวแทนของสื่อโฆษณา
- คลาส MediaType – เป็นเอนทิตีไทป์คลาสที่กำหนดประเภทของสื่อโฆษณาให้กับคลาส Media เช่น สื่อโทรทัศน์, สื่อสิ่งพิมพ์, สื่อวิทยุ, สื่อโฆษณากลางแจ้ง เป็นต้น
- คลาส Metric – เป็นเอนทิตีคลาสตัวแทนของมาตรวัดต่างๆที่ใช้เพื่อการนำไปคำนวณประสิทธิภาพของโฆษณา
- คลาส MetricType – เป็นเอนทิตีไทป์คลาสที่กำหนดประเภทของมาตรวัดที่ใช้กับสื่อโฆษณา มาตรวัดแต่ละประเภทสามารถนำไปใช้กับสื่อที่ต่างชนิดกันได้ดังตารางที่ 7
- คลาส Property – เป็นคลาสตัวแทนของค่าพรอบเพอร์ตีแต่ละประเภท
- คลาส PropertyType – เป็นคลาสตัวแทนของพรอบเพอร์ตีประเภทต่างๆ

จากแผนภาพคลาสในส่วนของไทป์สแควร์แพทเทิร์นนี้ จะถูกแปลงเป็นยูนิตด้วยวิธีการนำเสนอไว้ในบทที่ 3 โดยเริ่มจากแบ่งคลาสที่เป็นเมต้าคลาสและเอนทีตีคลาสออกจากกัน โดยในที่นี้คลาสที่เป็นเมต้าคลาส ได้แก่ MediaType, MetricType, PropertyType จะถูกนำมาสร้างเป็นยูนิตชื่อ

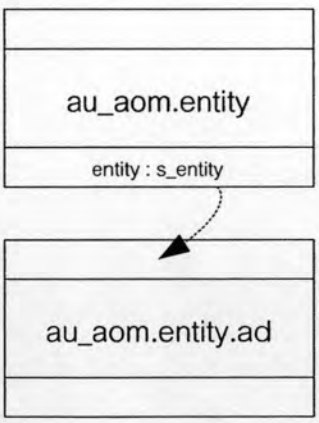
au_aom.entitytype และเอนทิตีคลาสได้แก่ Campaign, Media, Metric, Property และนำมาสร้างเป็นยูนิตชื่อ au_aom.entity

เนื่องจากคลาส MediaType และ MetricType ทำหน้าที่เดียวกันคือ เป็นตัวแทนของประเภทต่างๆ และมีแอตทริบิวต์เหมือนกัน จึงสามารถรวมกันเป็นเพียงคลาสเดียวได้ โดยตั้งชื่อใหม่เป็นคลาสชื่อ EntityType ซึ่งจะถูกนำมากำหนดเป็นเซตของเอ็กซ์พอร์ต (คลาส EntityType ถูกกำหนดไว้ในแพ็กเกจซิกเนเจอร์ชื่อ s_entitytype) ไว้ที่ยูนิต au_aom.entitytype ดังรูปที่ 4.29



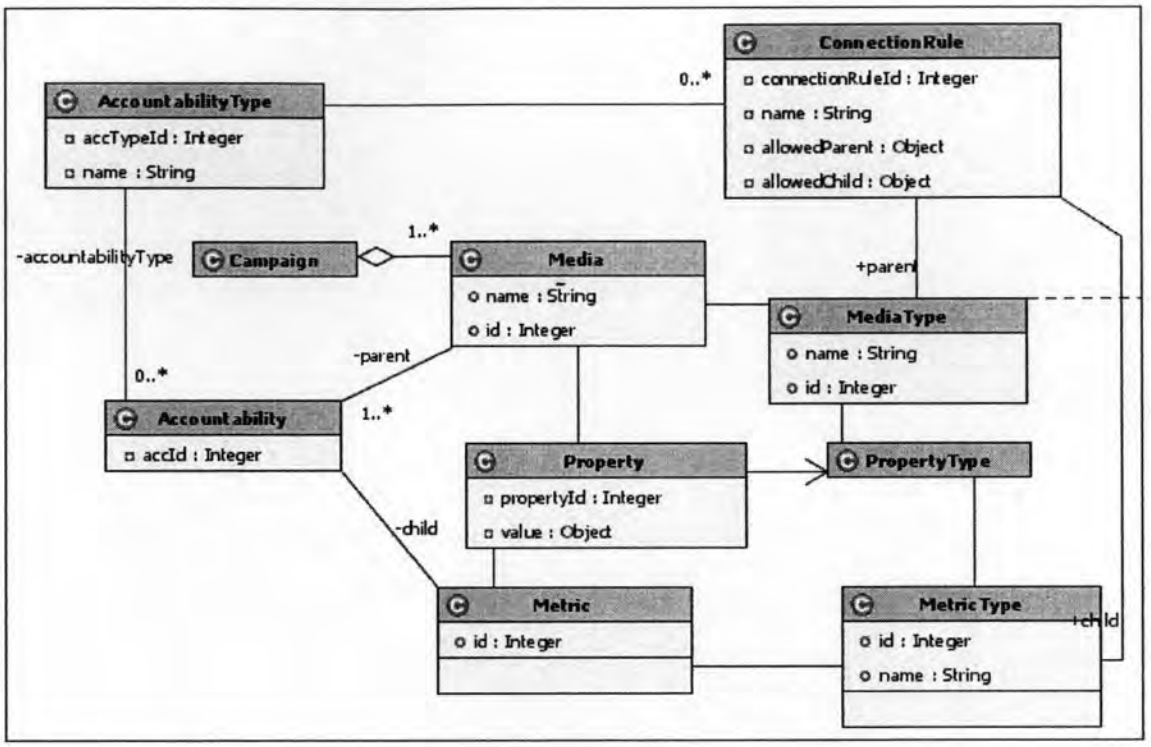
รูปที่ 4.29 อะตอมยูนิตของ au_aom.entitytype และ au_aom.entity

ด้วยเหตุผลที่ต้องการให้ยูนิต au_aom.entity สามารถนำกลับมาใช้ใหม่ได้ จึงแบ่งแยกยูนิตที่เกี่ยวข้องเฉพาะโดเมนออกเป็นยูนิตใหม่คือ au_aom.entity.ad ดังรูปที่ 4.30 และเพิ่มคลาส Entity (ถูกกำหนดไว้ในแพ็กเกจซิกเนเจอร์ชื่อ s_entity) ซึ่งจะถูกนำมากำหนดเป็นเซตของเอ็กพอร์ตไว้ที่ยูนิต au_aom.entity



รูปที่ 4.30 อะตอมยูนิตของ au_aom.entity และ au_aom.entity.ad

4.4.3.2 แอ็คเคาน์ทะบิลิตีแพทเทิร์น



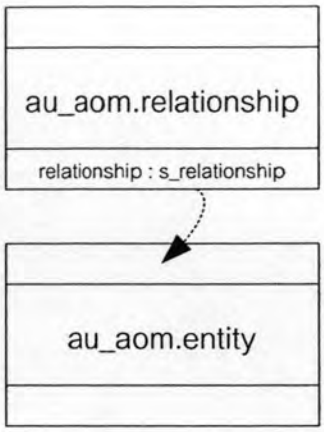
รูปที่ 4.31 แผนภาพคลาสของแอ็คเคาน์ทะบิลิตีระบบวัดประสิทธิภาพโฆษณา

- คลาส AccountabilityType – เป็นคลาสประเภทของความสัมพันธ์ระหว่างเอนทิตีที่สนใจ
- คลาส ConnectionRule – เป็นคลาสที่กำหนดกฎความสัมพันธ์ระหว่างประเภทของเอนทิตี (EntityType) โดยมีคลาส MediaType เป็น Parent และ MetricType เป็น Child
- คลาส Accountability – เป็นคลาสของอ็อบเจกต์ความสัมพันธ์ที่เกิดขึ้นระหว่างเอนทิตี
- คลาส Media, MediaType, Metric, MetricType จะสัมพันธ์กับคลาสในแพทเทิร์นนี้ทั้งหมด

จากแผนภาพคลาสและความสัมพันธ์ระหว่างคลาสที่เกิดขึ้นของแพทเทิร์นนี้ สามารถแปลงเป็นยูนิตโมเดลได้โดยใช้วิธีการที่ได้นำเสนอไว้ คือ กลุ่มคลาสที่เป็น Knowledge Level นั้นได้แก่ AccountabilityType, ConnectionRule, EntityType (MediaType และ MetricType รวมเป็นคลาสเดียวกัน) ซึ่งถูกนำมาสร้างเป็นยูนิตชื่อ au_aom.relationship ดังรูปที่ 4.32

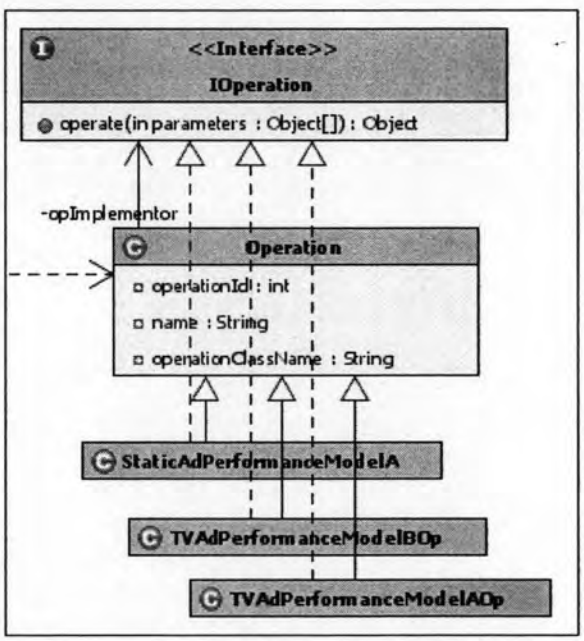
ในส่วนของ Operational Level ได้แก่ Media, Metric จะถูกนำไปสร้างเป็นยูนิต Entity ดังที่กล่าวไปแล้วของไทป์สแควร์แพทเทิร์น

เส้นของความสัมพันธ์ระหว่าง Knowledge Levelและ Operational Level จะนำมากำหนดเป็นเซตของเอ็กซ์พอร์ตให้แก่ยูนิต au_aom.relationship (คลาส AccountabilityType ถูกกำหนดไว้ในแพ็กเกจซิกเนเจอร์ชื่อ s_relationship)



รูปที่ 4.32 อะตอมยูนิตของ au_aom.relationship และ au_aom.entity

4.4.3.3 สตราทิจิแพทเทิร์น



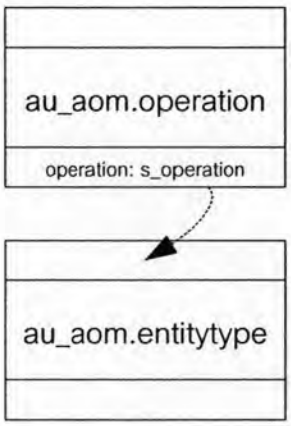
รูปที่ 4.33 แผนภาพคลาสของสตราทิจิระบบวัดประสิทธิภาพโฆษณา

- อินเตอร์เฟส IOperation – เป็นอินเตอร์เฟสที่เป็นข้อกำหนดมาตรฐานให้แก่ทุกคลาสที่ทำหน้าที่เสมือนเป็นเมท็อด
- คลาส Operation – เป็นคลาสที่ทำหน้าที่แทนเอนทิตีของเมท็อด และเป็นคลาสแม่สำหรับทุกคลาสที่ต้องการให้ระบบสามารถเรียกใช้ตอนรันไทม์ได้
- คลาส StaticAdPerformanceModelA – เป็นคลาสที่ทำหน้าที่คำนวณค่าประสิทธิภาพของโฆษณา โดยนำเอาข้อมูลที่ได้จากมาตรวัดต่างๆมาประมวลผลตามโมเดลการคำนวณ
- คลาส TVAdPerformanceModelBOp – ทำหน้าที่คำนวณประสิทธิภาพของโฆษณาเช่นเดียวกัน แต่เป็นโมเดลการคำนวณสำหรับการโฆษณาทางโทรทัศน์

- คลาส TVAdPerformanceModelAOp – เช่นเดียวกับคลาส TVAdPerformanceModelBOp แต่โมเดลการคำนวณนั้นแตกต่างกัน

จากแผนภาพคลาส สตราทิจิแพทเทิร์นจะถูกแปลงให้อยู่ในรูปของยูนิตโมเดลได้ โดยสร้างยูนิตชื่อ au_aom.operation ซึ่งเป็นยูนิตที่เกี่ยวข้องกับการคำนวณประสิทธิภาพของโฆษณาในรูปแบบต่างๆ คลาส Operation ก็จะถูกนำมากำหนดเป็นเซตของเอ็กซ์พอร์ตให้กับยูนิต (คลาส Operation ถูกกำหนดไว้ในแพ็กเกจซิกเนเจอร์ชื่อ s_operation) เพื่อให้ยูนิต au_aom.entitytype สามารถนำไปเรียกใช้เมท็อดได้ ดังรูปที่ 4.34

ส่วนคลาสอื่นๆจะถูกซ่อนไว้ภายในยูนิต ซึ่งช่วยป้องกันไม่ให้นักพัฒนาไปเรียกใช้คลาสเหล่านั้นได้โดยตรง



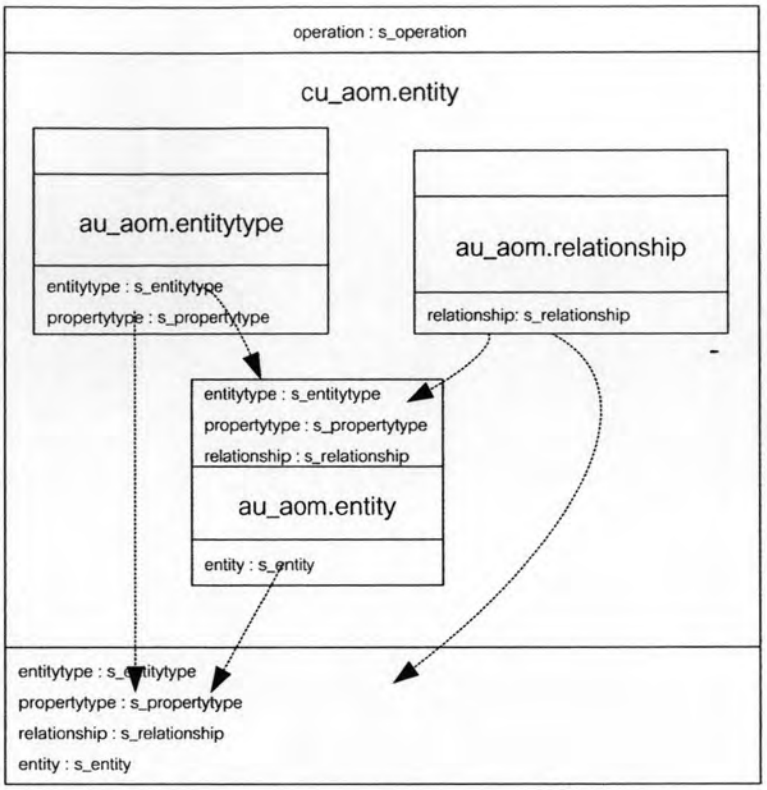
รูปที่ 4.34 อะตอมยูนิตของ au_aom.operation และ au_aom.entitytype

หลังจากที่ได้แปลงดีไซน์แพทเทิร์นต่างๆให้อยู่ในรูปของยูนิตแล้ว นักพัฒนาสามารถนำเอายูนิตมาประกอบกันเป็นยูนิตใหม่จากยูนิตเดิมที่มีอยู่ เพื่อซ่อนรายละเอียดความยุ่งยากซับซ้อนของการออกแบบไว้ภายใน

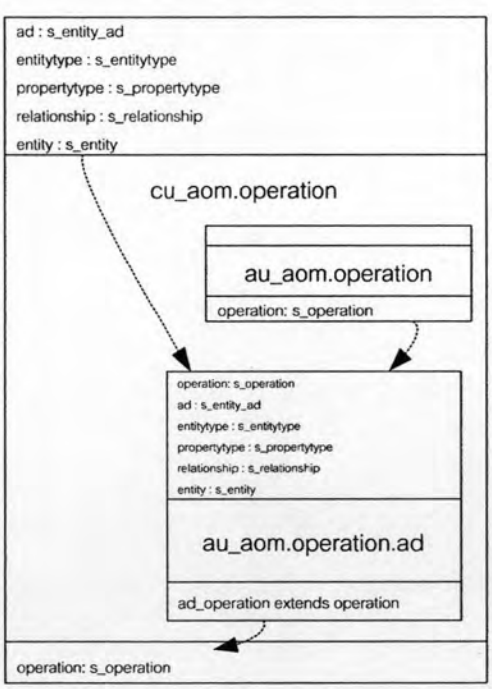
เมื่อนำยูนิตที่ได้จากการแปลงของแต่ละแพทเทิร์นมาเชื่อมต่อกันเป็นยูนิตใหม่ทำให้ได้ยูนิตใหม่ชื่อ cu_aom.entity ดังรูปที่ 4.35

เนื่องจากในแพ็กเกจซิกเนเจอร์ของยูนิต au_aom.operation มีอินเตอร์เฟสของ IOperation ซึ่งรับพารามิเตอร์เป็นอาร์เรย์ของคลาส Object ทำให้อาจเกิดความไม่สะดวกต่อการพัฒนา เนื่องจากไม่สามารถเข้าถึงเอนทิตีคลาสได้โดยตรง ดังนั้นผู้ออกแบบจึงสร้างยูนิตใหม่ที่เรียกใช้เอนทิตีของโดเมนโดยตรงด้วยการอิมพอร์ตซิกเนเจอร์ของ s_operation เข้ามาดังรูปที่ 4.36

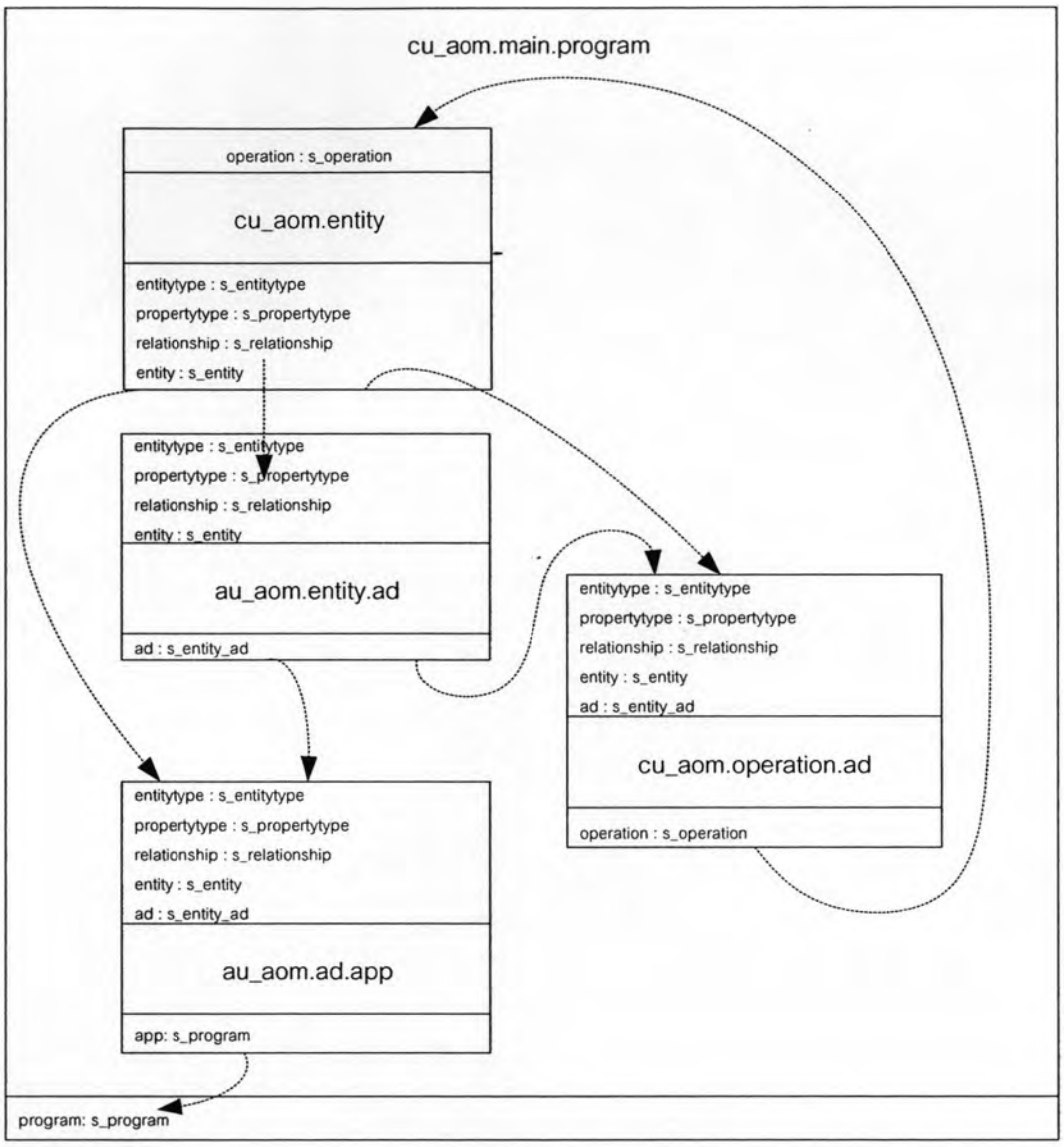
เมื่อนำยูนิตทั้งหมดมาเชื่อมโยงกันคอมพาวด์ยูนิต ดังรูปที่ 4.37 ก็จะได้โปรแกรมที่สามารถนำไปใช้งานได้



รูปที่ 4.35 คอมพาวด์ยูนิต cu_aom.entity ซึ่งเกิดจากการเชื่อมต่อกันระหว่างอะตอมยูนิต au_aom.entitytype, au_aom.relationship และ au_aom.entity

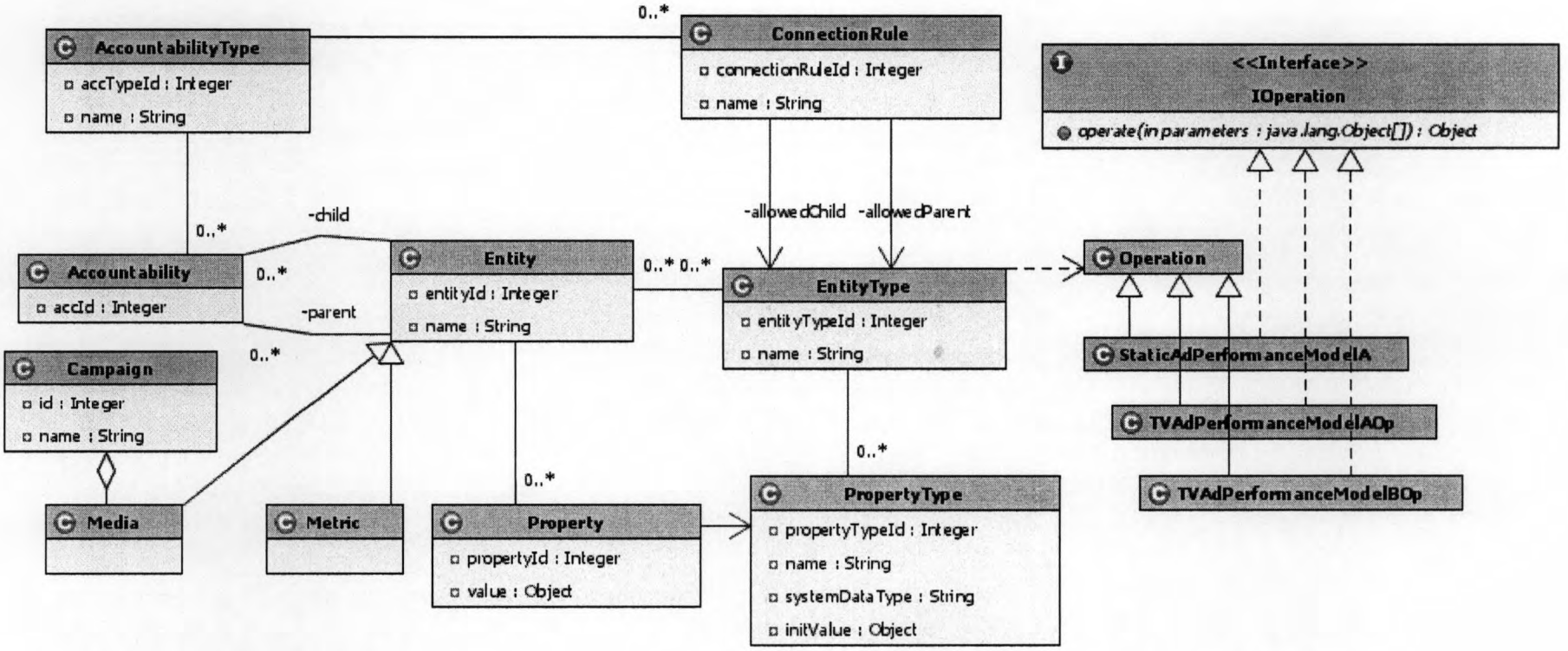


รูปที่ 4.36 คอมพาวด์ยูนิตซึ่งเกิดจากการเชื่อมต่อกันของยูนิต au_aom.operation และ au_aom.operation.ad



รูปที่ 4.37 การเชื่อมต่อกันของยูนิตทั้งหมดเป็นโปรแกรมที่สามารถทำงานได้ของระบบวัดประสิทธิภาพโฆษณา

เช่นเดียวกับระบบ CFGen และระบบงานทะเบียนข้อมูลสินค้า ที่จะต้องแสดงแผนภาพคลาสของแอพพลิเคชันแบบยูนิตดังรูปที่ 4.38 เพื่อนำไปใช้ในการวัดค่าความซับซ้อนเปรียบเทียบกับเอโอเอ็มแอพพลิเคชันแบบเดิม ผลของค่าที่วัดได้จะแสดงรายละเอียดอยู่ในบทต่อไป



รูปที่ 4.38 แผนภาพคลาสของเอโอเอ็มระบบวัดประสิทธิภาพโฆษณาหลังจากแปลงเป็นยูนิตโมเดล

4.4.4 การตรวจสอบการทำงานของโปรแกรม

โปรแกรมที่ได้จากการแปลงให้อยู่ในรูปของยูนิตโมเดลนั้นจะต้องมีฟังก์ชันการทำงานที่เหมือนกับโปรแกรมเอโอเอ็มเดิม ดังนั้น เพื่อตรวจสอบความถูกต้องของการทำงานของโปรแกรม จึงทำการทดสอบโดยการใช้ข้อมูลทดสอบชุดเดียวกัน เพื่อตรวจสอบว่าผลที่ได้จากการทำงานของทั้งสองโปรแกรมนั้นตรงกันหรือไม่ ดังตารางที่ 8

ตารางที่ 8 ผลการทดสอบการทำงานของระบบวัดประสิทธิภาพโฆษณาทั้งสองแบบ

ระบบ	วิธีทดสอบ	ข้อมูลอินพุต	ผลลัพธ์ที่ได้
ระบบวัดประสิทธิภาพโฆษณาแบบเอโอเอ็มเดิม	คำนวณประสิทธิภาพของโฆษณา	ข้อมูลสื่อที่ใช้ในการโฆษณาทางโทรทัศน์และป้ายโฆษณาในหน้าเว็บและค่ามาตรวัดที่ใช้กับสื่อแต่ละประเภท	■ ตัวเลขประสิทธิภาพของโฆษณาทางโทรทัศน์ได้เท่ากับ 680 และป้ายโฆษณาในหน้าเว็บเท่ากับ 140
ระบบวัดประสิทธิภาพโฆษณาแบบเอโอเอ็มที่ใช้ยูนิตโมเดล			■ ตัวเลขประสิทธิภาพของโฆษณาทางโทรทัศน์ได้เท่ากับ 680 และป้ายโฆษณาในหน้าเว็บเท่ากับ 140

ผลการทดสอบพบว่าข้อมูลที่ได้จากการทำงานของระบบวัดประสิทธิภาพโฆษณาแบบเอโอเอ็มเดิมและแบบที่ใช้ยูนิตโมเดลนั้นตรงกัน