บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

ในงานวิจัยนี้มีทฤษฎีที่เกี่ยวข้อง ได้แก่ คลาส เมตาคลาส (Metaclass) อ็อบเจกต์ (Object) แผนภาพคลาส และสัญกรณ์เซด โดยมีรายละเอียดดังต่อไปนี้

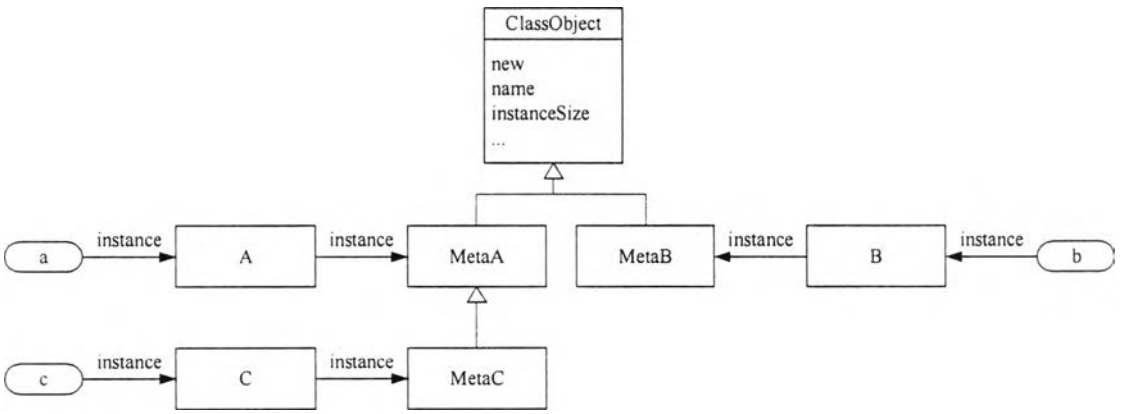
2.1.1 คลาส เมตาคลาส และอ็อบเจกต์ [9]

คลาสคือคำอธิบายที่เป็นนามธรรมของข้อมูล (Data) และพฤติกรรม (Behavior) ของเซตของอ็อบเจกต์ ซึ่งทุก ๆ อ็อบเจกต์ต้องมีตัวระบุอ็อบเจกต์ (Object Identity) เพื่อเป็นการแบ่งแยกอ็อบเจกต์แต่ละตัวให้เห็นความแตกต่างระหว่างแต่ละอ็อบเจกต์

ภาษาเชิงวัตถุ (Object-Oriented Language) จะมีข้อสนับสนุนหลัก 2 อย่างในเรื่องที่เกี่ยวกับคลาสและอ็อบเจ็กต์ ดังนี้

- ทุก ๆ อ็อบเจกต์เป็นอินสแทนซ์ (Instance) ของคลาสใดคลาสหนึ่ง
- คลาสคืออ็อบเจกต์

ดังนั้นในระบบจะต้องมีคลาสอ็อบเจกต์ (Class Object) ที่เป็นเสมือนคลาสทั่วไป (General Class) ซึ่งก็คือคลาสชื่อ *ClassObject* ในรูปที่ 2.1 โดยข้อมูลและพฤติกรรมที่มีอยู่ในคลาสอ็อบเจกต์จะประกอบไปด้วยชื่อของคลาส จำนวนหน่วยความจำที่ถูกใช้โดยอินสแทนซ์ของคลาส และการสร้างอินสแทนซ์ใหม่ เป็นต้น



รูปที่ 2.1 คลาส เมตาคลาส และอ็อบเจกต์

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

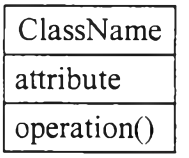
อินสแทนซ์ของคลาสชื่อ *ClassObject* เรียกว่าเมตาคลาส โดยเมตาคลาสจะสืบทอดคุณสมบัติและพฤติกรรมมาจากคลาสชื่อ *ClassObject* ซึ่งลำดับชั้นของเมตาคลาสจะมีจำนวนเท่ากับลำดับชั้นของคลาส ตัวอย่างเช่น ถ้ามี 3 ระดับของคลาสที่สืบทอดคุณสมบัติกัน ก็จะมี 3 ระดับของเมตาคลาสที่สืบทอดคุณสมบัติกันด้วย ดังรูปที่ 2.1 โดยทุก ๆ คลาสก็จะมีเมตาคลาสสำหรับคลาสนั้น ๆ ซึ่งแต่ละคลาสจะเป็นอินสแทนซ์ของเมตาคลาสของคลาสนั้น แต่ในการเขียนโปรแกรม โปรแกรมเมอร์จะไม่เห็นในส่วนของเมตาคลาส เพราะได้ถูกซ่อนไว้โดยคลาสเบราว์เซอร์ (Class Browser) ส่วนอ็อบเจกต์ a อ็อบเจกต์ b และอ็อบเจกต์ c จะเป็นอินสแทนซ์ของคลาส A คลาส B และคลาส C ตามลำดับ ดังรูปที่ 2.1

2.1.2 แผนภาพคลาส [7,8]

แผนภาพคลาสของยูเอ็มแอล บรรยายถึงโครงสร้างของระบบในเทอมของคลาสและความสัมพันธ์ โดยมีส่วนประกอบสำคัญ คือ คลาสและความสัมพันธ์ ดังนี้

2.1.2.1 คลาส

ในยูเอ็มแอล คลาสจะแสดงเป็นรูปสี่เหลี่ยมซึ่งแบ่งออกเป็นสามส่วน ส่วนบนแสดงชื่อคลาส ส่วนกลางแสดงลักษณะประจำ และส่วนล่างแสดงการดำเนินการ ดังรูปที่ 2.2 ซึ่งในส่วนของลักษณะประจำและการดำเนินการสามารถละไว้ได้



รูปที่ 2.2 คลาสของยูเอ็มแอล

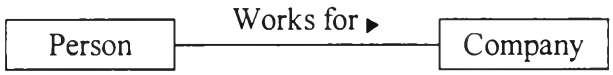
ในส่วนของลักษณะประจำและการดำเนินการ สามารถระบุแบบชนิดข้อมูล เช่น จำนวนเต็ม (Integer) สายอักขระ (String) หรือบูลีน (Boolean) เป็นต้น เพื่อกำหนดรายละเอียดเพิ่มขึ้น เฉพาะในส่วนลักษณะประจำจะสามารถกำหนดค่าโดยปริยาย (Default value) เพื่อให้เป็นค่าของลักษณะประจำนั้นถ้าไม่มีการกำหนดเป็นอย่างอื่น และในการดำเนินการสามารถมีเครื่องชี้ที่มีลักษณะคล้ายกับพารามิเตอร์ (Parameter) ของฟังก์ชันได้

ในกรณีที่ต้องการใช้แผนภาพคลาสในแบบจำลองต่าง ๆ โดยไม่ระบุรายละเอียดของคลาส สามารถทำได้โดยเขียนเฉพาะชื่อคลาสเท่านั้น

2.1.2.2 ความสัมพันธ์

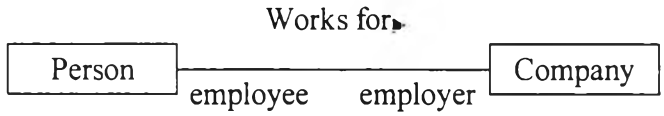
คลาสในระบบจะมีความความสัมพันธ์กันในรูปแบบต่าง ๆ ซึ่งจะต้องแสดงไว้ในแผนภาพคลาสเพื่อเป็นการแสดงข้อมูลของระบบ ความสัมพันธ์หลักในแผนภาพคลาสมี 3 ชนิด ดังนี้

1) ความสัมพันธ์แอสโซซิเอชัน (Association relationship) เป็นความสัมพันธ์ที่แสดงถึงความเกี่ยวข้องระหว่างคลาส หรือสิ่งที่คลาสหนึ่งกระทำกับอีกคลาสหนึ่ง เช่น ในระบบบุคลากร มีความสัมพันธ์ระหว่างคลาส Person กับคลาส Company คือ "a Person works for Company" ดังรูปที่ 2.3



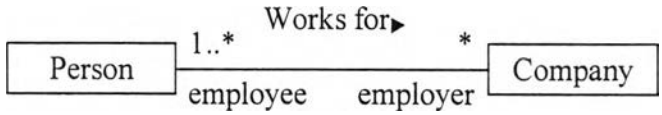
รูปที่ 2.3 ความสัมพันธ์แอสโซซิเอชัน

เมื่อคลาสหนึ่งมีความสัมพันธ์กับอีกคลาสหนึ่ง แต่ละคลาสมักจะเกิดบทบาท (Role) ระหว่างกันขึ้น ซึ่งคลาสหนึ่งอาจจะมีบทบาทที่แตกต่างกันเมื่อมีความสัมพันธ์กับคลาสที่ต่างกัน เช่น ดังรูปที่ 2.4 คลาส Person มีบทบาทเป็น employee ซึ่งมีความสัมพันธ์กับ Company ซึ่งมีบทบาทเป็น employer



รูปที่ 2.4 การแสดงบทบาทของความสัมพันธ์แอสโซซิเอชัน

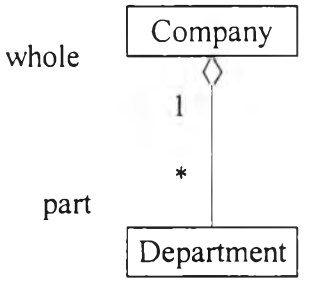
มัลติพลิซิตี (Multiplicity) คือการระบุจำนวนของอ็อบเจกต์จากคลาสหนึ่ง ที่มีความสัมพันธ์กับหนึ่งอ็อบเจกต์ของอีกคลาสหนึ่ง โดยมัลติพลิซิตีมีหลายรูปแบบ คลาสหนึ่งสามารถมีความสัมพันธ์กับคลาสอื่นแบบหนึ่งต่อหนึ่ง (One to one) หนึ่งต่อหลาย (One to many) หลายต่อหนึ่ง (Many to one) หนึ่งต่อหนึ่งหรือมากกว่า (One to one or more) หนึ่งต่อศูนย์หรือหนึ่ง (One to zero or one) เป็นต้น โดยสัญลักษณ์ * (Asterisk) มีความหมายเช่นเดียวกับ 0..* ซึ่งมัลติพลิซิตีสามารถกำหนดเป็นช่วงได้ คือ 5..10 หมายถึงจำนวนใด ๆ ตั้งแต่ 5 ถึง 10 แต่ถ้าเป็นค่าใดค่าหนึ่ง เช่น 5,10 จะหมายถึง 5 หรือ 10 ดังรูปที่ 2.5



รูปที่ 2.5 มัลติพลิซิตี

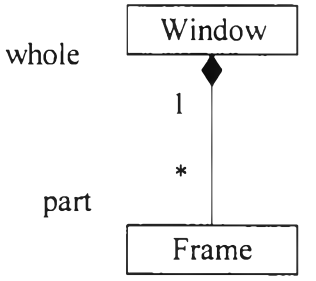
นอกจากนี้ ยังมีแบบพิเศษของความสัมพันธ์แอสโซซิเอชันอีก 2 ชนิด ดังนี้

- ความสัมพันธ์แอกริเกชัน (Aggregation relationship) เป็นแบบพิเศษของความสัมพันธ์แอสโซซิเอชันชนิดหนึ่ง ซึ่งเป็นการรวมกันขึ้นของคลาสอื่นหรือคอมโพเนนท์คลาส (Component class) จำนวนหนึ่ง ดังรูปที่ 2.6 โดยคลาสที่รวมคลาสต่าง ๆ จะเรียกว่าโฮลคลาส (Whole class)



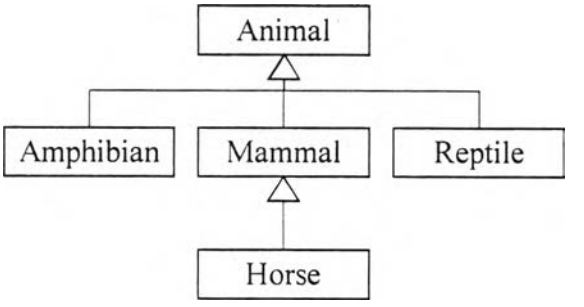
รูปที่ 2.6 ความสัมพันธ์แอกริเกชัน

- ความสัมพันธ์คอมโพสิชัน (Composition relationship) เป็นแบบพิเศษของความสัมพันธ์แอสโซซิเอชันชนิดหนึ่ง ซึ่งเป็นแบบที่แข็งแรงของความสัมพันธ์แอกริเกชัน แต่ละคอมโพเนนท์คลาสที่เป็นความสัมพันธ์คอมโพสิชันจะต้องมีในโฮลคลาส ในขณะที่ความสัมพันธ์แอกริเกชันอาจไม่มีก็ได้ และต้องเป็นคอมโพเนนท์ของโฮลคลาสเพียงคลาสเดียว ดังรูปที่ 2.7



รูปที่ 2.7 ความสัมพันธ์คอมโพสิชัน

2) ความสัมพันธ์เจเนอรัลไลเซชัน (Generalization relationship) เป็นความสัมพันธ์ที่แสดงถึงการที่คลาสหนึ่งสืบทอดคุณสมบัติของลักษณะประจำ และการดำเนินการจากอีกคลาสหนึ่ง คลาสที่ถูกสืบทอดคุณสมบัติจะเรียกว่าแม่ (Parent) หรือซูเปอร์คลาส (Super class) ส่วนคลาสที่สืบทอดคุณสมบัติจะเรียกว่าลูก (Child) หรือซับคลาส (Subclass) ดังรูปที่ 2.8



รูปที่ 2.8 ความสัมพันธ์เจเนอรัลไลเซชัน

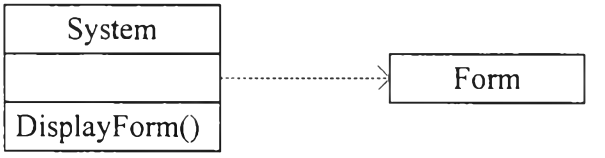
จากรูปที่ 2.8 คลาส Mammal จะมีความสัมพันธ์แบบเป็นชนิดหนึ่งของ (Is a kind of) กับคลาส Animal (Mammal is a kind of Animal) จะสังเกตว่าสามารถรวมความสัมพันธ์เจเนอรัลไลเซชันเป็นเส้นเชื่อมเดียวกันได้

ในการเขียนลักษณะประจำและการดำเนินการในแผนภาพคลาส ลักษณะประจำและการดำเนินการที่ถูกสืบทอดคุณสมบัติจะไม่แสดงในซับคลาส เนื่องจากได้แสดงไว้แล้วในซูเปอร์คลาส

ในซับคลาสจะมีลักษณะประจำหรือการดำเนินการเพิ่มขึ้น เพราะถ้าไม่มีจะถือว่าเป็นคลาสเดียวกันกับซูเปอร์คลาส เช่น Mammal จะมีขนและเลี้ยงลูกด้วยนม ซึ่งเป็นลักษณะประจำที่ไม่มีใน Animal เป็นต้น

คลาสที่ไม่มีแม่จะเรียกว่าคลาสพื้นฐาน (Base class) หรือคลาสราก (Root class) และถ้าคลาสใดมีคลาสแม่เพียงคลาสเดียวจะเรียกว่าการสืบทอดคุณสมบัติเดี่ยว (Single inheritance) ถ้ามีแม่มากกว่าหนึ่งจะเรียกว่าหลายการสืบทอดคุณสมบัติ (Multiple inheritance)

3) ความสัมพันธ์ดิเพนเด็นซี (Dependency relationship) เป็นความสัมพันธ์ที่แสดงถึงการดำเนินการของคลาสหนึ่งใช้อีกคลาสหนึ่งเป็นพารามิเตอร์ เช่น ในการแสดงผลดังรูปที่ 2.9 คลาส System ทำหน้าที่ในการแสดง Form โดยใช้การดำเนินการ displayForm(f:Form) ซึ่ง Form ที่แสดงจะขึ้นอยู่กับ Form ที่ผู้ใช้เลือก



รูปที่ 2.9 ความสัมพันธ์ดิเพนเด็นซี

2.1.3 สัญกรณ์เซด [2,6]

เซดเป็นภาษาข้อกำหนด (Specification language) ถูกสร้างขึ้นที่ห้องปฏิบัติการการคำนวณมหาวิทยาลัยอ๊อกซ์ฟอร์ด (Oxford University Computing Laboratory: OUCL) ในช่วงปลายของยุคปี ค.ศ.1970 ซึ่งเป็นภาษาข้อกำหนดที่มีหลักมาจากแนวคิดทางคณิตศาสตร์ที่แจ่มชัดของทฤษฎีเซตและตรรกเพรดิเคต

ในหัวข้อนี้ จะอธิบายเฉพาะบางส่วนของสัญกรณ์เซดที่จำเป็นเพื่อทำความเข้าใจกับข้อกำหนดในงานวิจัยนี้

1) เค้าร่าง (Schema) เป็นโครงสร้างหลักของสัญกรณ์เซด โดยเค้าร่างแบ่งเป็น 2 ส่วน คือ

- ส่วนการประกาศ (Declaration part) ประกอบไปด้วยการประกาศตัวแปรซึ่งอยู่ในรูปแบบของ $v : Type$ โดยที่ v เป็นชื่อตัวแปร (Variable name) และ *Type* เป็นชื่อแบบชนิดข้อมูล (Type name) ซึ่งความหมายของการประกาศนี้ คือ v เป็นสมาชิกของเซต *Type* เพราะแบบชนิดข้อมูลในสัญกรณ์เซดเป็นเซต

- ส่วนเพรดิเคต (Predicate part) ประกอบด้วยนิพจน์ตรรกเพรดิเคตที่นิยามความสัมพันธ์ระหว่างตัวแปรที่ได้ประกาศ

เค้าร่างเซดมีรูปแบบดังนี้

2) แบบชนิดข้อมูลในสัญกรณ์เซด เป็นได้ทั้งแบบพื้นฐาน (Basic) หรือแบบประกอบ (Composite) โดยสมาชิกของแบบชนิดข้อมูลพื้นฐานหรือกีเวนเซต (Given set) ถูกใช้เป็นพื้นฐานของการสร้างสมาชิกแบบประกอบ ซึ่งแบบชนิดข้อมูลประกอบในสัญกรณ์เซดมี 3 ชนิด คือ

- แบบชนิดข้อมูลเซต (Set type) คือมีสมาชิกเป็นเซต
- แบบชนิดข้อมูลคาร์ทีเซียนโปรดัก (Cartesian product type) คือมีสมาชิกที่เป็นค่าคู่ลำดับ
- แบบชนิดข้อมูลเค้าร่าง (Schema type) คือมีสมาชิกผูก (Binding) ด้วยค่าที่เป็นตัวแปรที่ประกาศในเค้าร่างเซด

3) เค้าร่างเซดจะถูกใช้ในการจำลองคุณสมบัติโครงสร้าง (Structural property) และคุณสมบัติเชิงพลวัตของระบบ (Dynamic property) ดังนี้

- เค้าร่างเซดในการจำลองคุณสมบัติโครงสร้าง จะถูกอ้างถึงเหมือนกับเป็นเค้าร่างสถานะ (State schema) โดยคอมโพเนนท์ของระบบถูกประกาศในส่วนการประกาศ และคุณสมบัติเชิงเสถียร (Static property) ของระบบถูกนิยามในส่วนเพรดิเคต
- เค้าร่างเซดในการจำลองคุณสมบัติเชิงพลวัต จะถูกอ้างถึงเหมือนกับเป็นเค้าร่างการดำเนินการ (Operation schema) โดยส่วนประกาศมีการประกาศตัวแปรที่แสดงสถานะก่อนและสถานะหลัง ตัวแปรเข้า ตัวแปรออก และตัวแปรอื่น ๆ ที่จำเป็นในการนิยามเงื่อนไขก่อน (Precondition) และเงื่อนไขหลัง (Postcondition) และในส่วนเพรดิเคตได้นิยามถึงความสัมพันธ์ระหว่างสถานะก่อน (Pre state) และสถานะหลัง (Post state)

4) การตกแต่ง (Decoration) คือการใส่ตัวอักษรพิเศษท้ายชื่อตัวแปร เพื่อสื่อความหมายบางอย่างดังนี้

- ตัวแปรที่ไม่มี ′ (Unprimed variable) คือ สถานะก่อนการดำเนินการ เช่น w
- ตัวแปรที่มี ′ (Primed variable) คือ สถานะหลังการดำเนินการ เช่น w'
- ตัวแปรที่มี ? คือ ตัวแปรเข้าของการดำเนินการ เช่น $w?$
- ตัวแปรที่มี ! คือ ตัวแปรออกของการดำเนินการ เช่น $w!$

5) การประกาศเค้าร่างเพื่อใช้ในเค้าร่างอื่น (Schema Inclusion)

- ΔS เป็นการระบุว่ามีการเปลี่ยนแปลงในสถานะ S
- ΞS เป็นการระบุว่าสถานะ S ไม่มีการเปลี่ยนแปลงหลังการดำเนินการ

6) การดำเนินการทางเซต (Set operation) เช่น เซตย่อย (Subset) เซตยูเนียน (Set union) เซตอินเตอร์เซคชัน (Set intersection) เซตดิฟเฟอเรนซ์ (Set difference) สามารถกำหนดได้ในสัญกรณ์เซด และจำนวนสมาชิกในเซตจำกัด S ใช้แสดงโดย $\#S$

7) ความสัมพันธ์ สามารถนิยามได้โดยใช้แบบชนิดข้อมูลพื้นฐานและแบบประกอบที่เป็นเซตของคู่ลำดับ โดยที่สมาชิกลำดับแรกมาจากโดเมน (Domain) และสมาชิกลำดับที่สองมาจากพิสัย (Range) ของความสัมพันธ์ เช่น $R : A \longleftrightarrow B$ และ $R = \{(1,a),(1,b),(2,b),(3,a),(3,b)\}$

- dom R ส่งคืนเซตที่มีสมาชิกใน A ที่สัมพันธ์กับสมาชิกใน B จากข้างบน dom $R = \{1,2,3\}$

- ran R ส่งคืนเซตที่มีสมาชิกใน B ที่สัมพันธ์กับสมาชิกใน A จากข้างบน ran $R = \{a,b\}$

- $S \lhd R$ (Domain anti-restriction) โดยที่ S เป็นเซตย่อยของ A ส่งคืนคู่ลำดับใน R ที่สมาชิกตัวแรกของคู่ลำดับ (Order pair) ไม่อยู่ใน S เช่น $\{1,2\} \lhd R = \{(3,a),(3,b)\}$

- $R\sim$ (Relational inverse) จากข้างบน $R\sim = \{(a,1),(b,1),(b,2),(a,3),(b,3)\}$

- $R\ (|\{S\}|)$ (Relational image) โดยที่ S เป็นเซตย่อยของ A ส่งคืนสมาชิกลำดับที่สองของคู่ลำดับใน R ที่สมาชิกลำดับที่หนึ่งอยู่ใน S เช่น $R\ (|\{2,3\}|) = \{a,b\}$

- $R \oplus S$ (Overriding) โดยที่ S เป็นเซตของคู่ลำดับ ส่งคืนทุก ๆ คู่ลำดับใน R หรือ S ยกเว้นคู่ลำดับใน R ที่สมาชิกลำดับที่หนึ่งอยู่ใน S เช่น $R \oplus \{(1,c)\} = \{(1,c),(2,b),(3,a),(3,b)\}$

8) ฟังก์ชัน (Function) ในสัญกรณ์เซดฟังก์ชันเป็นความสัมพันธ์ที่แต่ละสมาชิกในโดเมนสัมพันธ์กับสมาชิกในพิสัย เช่น $F : A \nrightarrow B$ ซึ่งเป็นฟังก์ชันบางส่วน (Partial function)

2.2 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องที่ใช้สัญกรณ์ภาพกราฟิคเพื่อสร้างข้อกำหนดรูปนัย หรือสร้างภาพนามธรรมของข้อกำหนดรูปนัย มีดังนี้

2.2.1 **การรู้เหตุผลด้วยแผนภาพคลาสของยูเอ็มแอล** (Reasoning with UML Class Diagram) [10] **โดย** Andy S. Evans

ในงานวิจัยนี้ได้เสนอเทคนิคการวิเคราะห์แบบรีกเออะรัซ (Rigorous Analysis) สำหรับยูเอ็มแอล โดยใช้การวิเคราะห์แผนภาพคลาสจากข้อกำหนดเซดซึ่งได้นิยามไว้อย่างเที่ยงตรงในรูปแบบของกฎที่แจ่มชัด (Well-formedness rule) โดยผลจากการวิเคราะห์สามารถแปลงแผนภาพคลาสให้มีความเที่ยงตรงได้ตามกฎที่ได้นิยามไว้นั้น

2.2.2 **การวิเคราะห์เชิงปัญหาของแนวคิดแบบจำลองความต้องการเชิงเสถียร ของยูเอ็มแอลพื้นฐาน** (A Problem-Oriented Analysis of Basic UML Static Requirements Modeling Concepts) [11] **โดย** Robert B. France

ในงานวิจัยนี้ได้นิยามความหมายที่เที่ยงตรงของสัญกรณ์ยูเอ็มแอล โดยใช้เทคนิครูปนัยในการนิยามความหมายที่เป็นความหมายพื้นฐานสำหรับแนวคิดเชิงวัตถุและสัญกรณ์ยูเอ็มแอล และได้ใช้ความหมายพื้นฐานนี้ในการพัฒนากฎการแปลง (Transformation rule) สำหรับใช้ในการแปลงแบบจำลองให้เป็นแบบจำลองที่มีความเที่ยงตรงตามความหมายพื้นฐาน และได้พัฒนาวิธีที่ทำให้ผู้พัฒนา

ระบบสามารถทำการวิเคราะห์แบบรีกเออะรัชในแบบจำลองยูเอ็มแอลได้ โดยใช้กฎการแปลงมาทำการวิเคราะห์แผนภาพคลาส

2.2.3 **การกำหนดโครงสร้างคลาสของยูเอ็มแอลด้วยเซด** (Towards a Formalization of UML Class Structures in Z) [12] **โดย** Malcolm Shroff **และ** Robert B. France

ในงานวิจัยนี้ได้แสดงถึงการรวมวิธีรูปนัยและวิธีกึ่งรูปนัย ซึ่งแสดงด้วยแผนภาพคลาสของยูเอ็มแอลเข้าด้วยกันโดยใช้สัญกรณ์เซด เพื่อแสดงความเที่ยงตรงของแผนภาพคลาสของยูเอ็มแอล โดยการพัฒนากฎการแปลงเพื่อใช้ในการแปลงแผนภาพคลาสให้เป็นข้อกำหนดเซด ซึ่งในงานวิจัยนี้ได้แสดงเฉพาะคุณสมบัติเชิงเสถียรเท่านั้น

2.2.4 **การสร้างภาพนามธรรมของข้อกำหนดรูปนัย** (Visualization of Formal Specifications) [13] **โดย** Soon-Kyeong Kim **และ** David Carrington

งานวิจัยนี้เป็นการแปลงข้อกำหนดเซดให้เป็นแผนภาพหลายชนิด ซึ่งแผนภาพเหล่านี้ใช้ในการแสดงข้อกำหนดเซดที่มีคุณสมบัติที่แตกต่างกัน ได้แก่

- คุณสมบัติเชิงเสถียร แสดงโดยการใช้แผนภาพคลาสของยูเอ็มแอลในการสร้างภาพนามธรรมของโครงสร้างระบบที่ระบุในเค้าร่างสถานะ ซึ่งการแปลงข้อกำหนดเซดเป็นแผนภาพคลาสใช้การวิเคราะห์ความหมายของความสัมพันธ์ระหว่างตัวแปร ที่ได้ประกาศไว้ในเค้าร่างสถานะและโครงสร้างคลาสของยูเอ็มแอล

- เงื่อนไขที่ซับซ้อน (Complex constraint) ซึ่งแผนภาพคลาสไม่สามารถกำหนดได้ แสดงโดยการใช้แผนภาพเงื่อนไข (Constraint diagram)

- คุณสมบัติเชิงพลวัตของเค้าร่างการดำเนินการ แสดงโดยการใช้กล่องคอนแทร็ค (Contract box) ในการแสดงสถานะก่อนและสถานะหลังของแต่ละการดำเนินการในเซด

2.2.5 **เครื่องมือแปลงแบบจำลองเอนทิตีและความสัมพันธ์ เป็นข้อกำหนดรูปนัยในสัญกรณ์เซด** (A Tool for Translating Entity Relationship Model to Formal Specification in Z) [14] **โดย วัชรวรรณ ศรีรัตนาลัย**

งานวิจัยนี้ได้นำเสนอวิธีการแปลงแผนภาพเอนทีติและความสัมพันธ์ เป็นข้อกำหนดรูปนัยในรูปสัญกรณ์เซด และสร้างชุดคำสั่งพื้นฐาน โดยการสังเคราะห์คุณสมบัติต่างๆ ดังนี้

- การสังเคราะห์คุณสมบัติโครงสร้าง โดยการนิยามเค้าร่างเอนทีตี ซึ่งเป็นส่วนที่นิยามถึงโครงสร้างของเอนทิตี ได้แก่ ลักษณะประจำและเงื่อนไขของลักษณะประจำ การนิยามเค้าร่างเพิ่มเติม ได้แก่ การนิยามข้อกำหนดของคีย์หลักและคีย์นอก เป็นต้น และการนิยามเค้าร่างความสัมพันธ์

- การสังเคราะห์คุณสมบัติพฤติกรรม เป็นส่วนการนิยามถึงชุดคำสั่งพื้นฐานที่สามารถสร้างได้จากพจนานุกรมข้อมูล ได้แก่ การเพิ่ม การแก้ไข และการลบข้อมูล