บทที่ 2

งานวิจัยและทฤษฎีที่เกี่ยวข้อง

2.1 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องที่ใช้สัญกรณ์เซดเพื่อสร้างข้อกำหนดรูปนัย มีดังนี้

2.1.1 เครื่องมือแปลงแบบจำลองเอนทิตีและความสัมพันธ์ เป็นข้อกำหนดรูปนัยในรูปสัญกรณ์เซด (A Tool for Translating The Entity Relationship Model to Formal Specification in Z Notation) [4] โดย วัชรวรรณ ศรีรัตนาลัย

วิทยานิพนธ์นี้นำเสนอขั้นตอนการออกแบบ และพัฒนาเครื่องมือแปลงแบบจำลองเอนทิตีและความสัมพันธ์เป็นข้อกำหนดรูปนัยในรูปสัญกรณ์เซด โดยใช้ข้อมูลจากพจนานุกรมข้อมูลและพจนานุกรมความสัมพันธ์ของแบบจำลองเอนทิตีและความสัมพันธ์เป็นข้อมูลนำเข้า

ขั้นตอนวิธีที่ใช้ในการแปลงแบ่งได้เป็น 2 กลุ่ม คือ กลุ่มขั้นตอนการสังเคราะห์คุณสมบัติโครงสร้าง และกลุ่มขั้นตอนการสังเคราะห์คุณสมบัติพฤติกรรม โดยผลของขั้นตอนการสังเคราะห์คุณสมบัติโครงสร้างจะได้สัญกรณ์เซด 3 เค้าร่างคือ เค้าร่างเอนทิตี เค้าร่างเพิ่มเติม และเค้าร่างความสัมพันธ์ ในขณะที่ผลของขั้นตอนการสังเคราะห์คุณสมบัติพฤติกรรมจะได้สัญกรณ์เซด 3 เค้าร่างเช่นกันคือ เค้าร่างการเพิ่มข้อมูล เค้าร่างการแก้ไขข้อมูล และเค้าร่างการลบข้อมูล ซึ่งคำนึงถึง บูรณภาพของการอ้างอิง (Referential Integrity) นอกจากนี้ยังได้มีการนิยามชนิดข้อมูลเพิ่มเติม คือ ชนิดข้อมูลตัวอักษร ชนิดข้อมูลสายอักษร และชนิดข้อมูลบูลลีน เพื่อให้สัญกรณ์เซดซึ่งเป็นผลของการแปลงที่ได้มีความสมบูรณ์ยิ่งขึ้น

แต่เนื่องจากคุณสมบัติพฤติกรรมที่ได้มีเฉพาะส่วนการเพิ่มและเปลี่ยนแปลงข้อมูลเท่านั้น ยังขาดส่วนของการค้นคืนข้อมูล รวมทั้งไม่สามารถนำเค้าร่างที่ได้นำกลับไปใช้ใหม่ (Reuseable)

2.1.2 การสังเคราะห์ข้อกำหนดรูปนัยสำหรับแบบจำลองฐานข้อมูลเชิงสัมพันธ์ (Formal Specification Synthesis for Relational Data Model) [5] โดย วิวัฒน์ วัฒนาวุฒิ และวันชัย ริ้วไพบูลย์

งานวิจัยนี้เสนอแนวคิดในการสร้างเค้าร่างอัตโนมัติในการสังเคราะห์ข้อกำหนดรูปนัยจาก แผนภาพความสัมพันธ์ข้อมูล โดยใช้แผนภาพเอนทิตีและความสัมพันธ์ซึ่งอยู่ในรูปของพจนานุกรมข้อมูลและพจนานุกรมความสัมพันธ์

ข้อกำหนดรูปนัยที่ได้จะอยู่ในรูปของเค้าร่างภาษาเซด ซึ่งประกอบด้วยส่วนโครงสร้างและส่วนคุณสมบัติเชิงพฤติกรรมของแผนภาพความสัมพันธ์ข้อมูล โดยครอบคลุมถึงข้อกำหนดเรื่องโครงสร้างข้อมูล เช่น

คีย์หลัก (Primary Key) จะต้องไม่ซ้ำกัน รวมถึงบูรณภาพของการอ้างอิงระหว่างความสัมพันธ์ต่างๆ มีการกำหนดกฎเกณฑ์ในการสร้างเค้าร่างเซตของสถานะและการดำเนินการพื้นฐาน ได้แก่ การเพิ่มข้อมูล การลบข้อมูล และการแก้ไขข้อมูล และเสนอวิธีการสร้างการดำเนินการใหม่ที่ประกอบขึ้นจากชุดการดำเนินการพื้นฐานที่มีอยู่ โดยข้อกำหนดรูปนัยที่สร้างขึ้นใหม่นี้สามารถพิสูจน์ความถูกต้องได้ด้วยวิธีการทางคณิตศาสตร์จากแผนภาพข้อมูลที่มีอยู่

2.2 ทฤษฎีที่เกี่ยวข้อง

ในงานวิทยานิพนธ์นี้ มีทฤษฎีต่างๆ ที่เกี่ยวข้องดังนี้

2.2.1 สัญกรณ์เซด (Z Notation) [1], [6]

สัญกรณ์เซดเป็นส่วนประกอบภาษารูปนัย (Formal Language) ภาษาหนึ่งซึ่งข้อกำหนดที่ได้จากสัญกรณ์เซดมีโครงสร้างที่คล้ายคลึงกับภาษาโปรแกรมเชิงคำสั่ง มีการนำสัญกรณ์และทฤษฎีทางคณิตศาสตร์ที่ได้มีการกำหนดความหมายไว้อย่างดีแล้ว (well-form) มาเป็นเครื่องมือในการอธิบายถึงพฤติกรรมของระบบ

สัญกรณ์เซดเขียนอยู่ในรูปของเค้าร่าง (Schema) ซึ่งสามารถแบ่งได้เป็น 2 ชนิด คือ

1) เค้าร่างการกำหนดข้อมูล เป็นส่วนที่ใช้ในการกำหนดข้อมูลที่มีการใช้งานร่วมกันภายในข้อกำหนด ซึ่งสามารถเขียนอยู่ในรูปของ

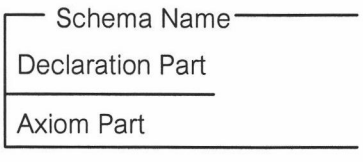
- เค้าร่างสถานะ (State Schema)
- เค้าร่างทั่วไป (Generic Schema)
- นิยามสัจพจน์ (Axiom Definition)
- นิยามชนิดอิสระ (Free Type Definition)
- กีเวนเซต (Given Set)

โดยส่วนการประกาศสามารถเขียนได้ 2 ชนิดคือ การประกาศส่วนกลาง (Global Declaration Part) และการประกาศส่วนท้องถิ่น (Local Declaration Part)

2) เค้าร่างการทำงาน (Operation Schema) เป็นส่วนที่จะอธิบายถึงการทำงานต่างๆ ภายในข้อกำหนด ซึ่งภายในเค้าร่างการทำงานหนึ่งๆ ประกอบด้วย 2 ส่วน คือ

- ส่วนของการประกาศตัวแปร (Declaration Part) ที่ต้องใช้ภายในเค้าร่างนั้นๆ
- ส่วนของสัจพจน์ (Axiom Part) ซึ่งจะเป็นส่วนที่กำหนดเงื่อนไขการทำงานในรูปของเงื่อนไขก่อนและหลังการทำงาน ส่วนสัจพจน์นี้จะใช้ทฤษฎีและสัญกรณ์ทางคณิตศาสตร์เป็นหลัก

การเขียนเค้าร่างสามารถเขียนได้ 2 แบบ คือ เค้าร่างแบบกล่อง (Box Like Schema)



และเค้าร่างตามขวาง (Horizontal Schema) ซึ่งมีรูปแบบดังนี้

SchemaName $\widehat{=}$ [Declaration Part | Axiom Part]

2.2.2 ทฤษฎีทางคณิตศาสตร์ที่เกี่ยวข้อง [7]

2.2.2.1 ตรรกศาสตร์ (Logic)

ตรรกศาสตร์ เป็นการกำหนดกฎและวิธีในการพิจารณาว่าอาร์กิวเมนต์นั้นถูกต้อง ในทางคณิตศาสตร์ ตรรกศาสตร์ถูกใช้ในการพิสูจน์ทฤษฎีต่างๆ ส่วนในทางคอมพิวเตอร์ ตรรกศาสตร์ถูกใช้ในการตรวจสอบความถูกต้องของโปรแกรม และใช้ในการพิสูจน์ทฤษฎี

ตรรกศาสตร์สามารถแบ่งได้เป็น 2 ชนิด คือ

1) ตรรกศาสตร์ประพจน์ (Propositional Logic) เป็นตรรกะซึ่งไม่มีตัวแปร เช่น $P \vee Q$ โดยที่ P และ Q เป็นประพจน์ เป็นต้น สัญลักษณ์สำหรับประพจน์ ประกอบด้วยสัญลักษณ์และ $\wedge$ สัญลักษณ์หรือ $\vee$ สัญลักษณ์ถ้า...แล้ว $\Rightarrow$ สัญลักษณ์ก็ต่อเมื่อ $\Leftrightarrow$ และสัญลักษณ์นิเสธ $\neg$
2) ตรรกศาสตร์ภาคแสดง (Predicate Logic) เป็นตรรกะซึ่งมีตัวแปรทำให้ไม่สามารถหาค่าความจริงได้ถ้าหากไม่รู้ค่าของตัวแปรนั้น เช่น $x > 5$ เป็นต้น ซึ่งสามารถเขียนภาคแสดงให้เป็นประพจน์ได้ โดยการเพิ่มตัวบ่งปริมาณทั้งหมดหรือตัวบ่งปริมาณบางส่วนเข้าไปข้างหน้าของนิพจน์ (Expression) เพื่อให้สามารถหาค่าความจริงได้ เช่น

$$\exists x : \mathbb{N} \bullet x > 5$$

เป็นการระบุว่า "บางค่าของ x ซึ่งมีชนิดเป็นจำนวนเต็ม มีค่ามากกว่า 5"

ในตรรกศาสตร์มีตัวบ่งปริมาณ (Quantifiers) ซึ่งเป็นสัญลักษณ์ที่ใช้ในการบอกจำนวนของสิ่งที่สนใจ ประกอบด้วยตัวบ่งปริมาณ 2 ชนิด คือ

1) ตัวบ่งปริมาณทั้งหมด (Universal Quantification) แทนด้วยสัญลักษณ์ $\forall$ หมายถึงต้องใช้ทุกสิ่งทุกอย่างในเอกภพสัมพัทธ์
2) ตัวบ่งปริมาณบางส่วน (Existential Quantification) แทนด้วยสัญลักษณ์ $\exists$ หมายถึงอย่างน้อย 1 สมาชิกในเอกภพสัมพัทธ์

2.2.2.2 เซตและความสัมพันธ์ (Set and Relation)

เซตคือกลุ่มของข้อมูลที่มีชนิดเดียวกัน ใช้สัญลักษณ์ { } แสดงความเป็นเซต โดยข้อมูลที่อยู่ภายในเครื่องหมายปีกกาหมายถึงอยู่ภายในเซตเดียวกัน และเรียกข้อมูลที่อยู่ภายในว่าเป็นสมาชิก เช่น {1,2,3,4} เป็นต้น

ตัวดำเนินการที่เกี่ยวกับเซต เช่น สัญลักษณ์เท่ากับ == สัญลักษณ์เป็นสมาชิก $\in$ สัญลักษณ์ยูเนียน $\cup$ สัญลักษณ์อินเตอร์เซค $\cap$ และสัญลักษณ์เซตว่าง $\varnothing$ เป็นต้น

ในการเขียนข้อกำหนดของระบบ จำเป็นอย่างยิ่งที่จะต้องกล่าวอ้างถึงความสัมพันธ์ของข้อมูล โดยทฤษฎีความสัมพันธ์จะแสดงให้เห็นถึงการที่จับคู่ค่าของเซตต้นทาง (Source Set) และเซตปลายทาง (Target Set) ในลักษณะของหลายต่อหลาย (Many to Many)

ภายในความสัมพันธ์หนึ่งๆ จะประกอบไปด้วยข้อมูล 2 ส่วนคือ โดเมน และพิสัย (Range) โดยโดเมนจะเป็นเซตย่อย (Subset) ของเซตต้นทาง ส่วนพิสัยจะเป็นเซตย่อยของเซตปลายทาง

2.2.2.3 **ฟังก์ชัน** (Function)

ฟังก์ชันเป็นลักษณะหนึ่งของความสัมพันธ์ แสดงถึงความสัมพันธ์ของโดเมนกับพิสัย โดยสมาชิกในโดเมน 1 ตัวจะมีความสัมพันธ์กับสมาชิกของพิสัยได้เพียง 1 ตัวเท่านั้น แต่สมาชิกของพิสัย 1 ตัวจะมีความสัมพันธ์กับสมาชิกของโดเมนกี่ตัวก็ได้

2.2.2.4 **จำนวนและลำดับ** (Numbers and Sequences)

ในภาษาเซตมีชนิดของข้อมูลที่เป็นจำนวนอยู่ 2 ชนิดคือ จำนวนเต็ม (Integers) และ จำนวนธรรมชาติ (Natural Numbers) รวมทั้งโอเปอร์เรชันมาตรฐานสำหรับจำนวนเหล่านี้ ลำดับคือการที่ข้อมูลเรียงต่อกันไปโดยข้อมูลเหล่านั้นจะต้องมีชนิดเดียวกัน และลำดับของข้อมูลมีความสำคัญ

2.2.3 **พีชคณิตความสัมพันธ์** (Relational Algebra) [8]

พีชคณิตความสัมพันธ์ ใช้ในการจัดการข้อมูลของแบบจำลองความสัมพันธ์ (Relational model) โดยระบุตัวกระทำ (Operator) และความสัมพันธ์ที่ต้องการจัดการ ผลลัพธ์ที่ได้จะเป็นความสัมพันธ์ใหม่

ในพีชคณิตความสัมพันธ์ ประกอบด้วยตัวกระทำ 9 ชนิด ได้แก่

1) การเลือกแถว (Selection หรือ Restriction) ซึ่งใช้ตัวกระทำคือ where เป็นการทำงานบนความสัมพันธ์เดียว (Unary operation) โดยการเลือกข้อมูลความสัมพันธ์จากเงื่อนไขที่กำหนด ผลลัพธ์ที่ได้คือเซตของสมาชิกในความสัมพันธ์นั้นๆ ที่ตรงตามเงื่อนไขที่กำหนด

2) การเลือกสดมภ์ (Projection) ซึ่งใช้ตัวกระทำคือ project เป็นการทำงานบนความสัมพันธ์เดียว โดยเลือกเฉพาะบางสดมภ์ของความสัมพันธ์ ผลลัพธ์ที่ได้คือเซตของสมาชิกในความสัมพันธ์นั้นๆ โดยตัดสดมภ์ที่ไม่ต้องการออกไป

3) การหาผลคูณ (Cartesian product) ซึ่งใช้ตัวกระทำคือ times เป็นการทำงานบน 2 ความสัมพันธ์ (Binary operation) โดยการรวมข้อมูลจาก 2 ความสัมพันธ์เข้าด้วยกัน ผลลัพธ์ที่ได้คือเซตของแถวทั้งหมดจากทั้ง 2 ความสัมพันธ์

4) การเชื่อมความสัมพันธ์ (Union) ซึ่งใช้ตัวกระทำคือ union เป็นการทำงานบน 2 ความสัมพันธ์ ผลลัพธ์ที่ได้คือเซตของสมาชิกจากทั้ง 2 ความสัมพันธ์รวมกัน โดยทั้ง 2 ความสัมพันธ์จะต้องมีจำนวนลักษณะประจำเท่ากัน และต้องมีโดเมนเหมือนกันทุกประการ

5) การหาผลต่าง (Difference) ซึ่งใช้ตัวกระทำคือ minus เป็นการทำงานบน 2 ความสัมพันธ์ ผลลัพธ์ที่ได้คือเซตของสมาชิกที่อยู่ในความสัมพันธ์แรกแต่ไม่อยู่ในความสัมพันธ์หลัง โดยทั้ง 2 ความสัมพันธ์จะต้องมีจำนวนลักษณะประจำเท่ากัน และต้องมีโดเมนเหมือนกันทุกประการ

6) การหาส่วนที่ซ้ำ (Intersection) ซึ่งใช้ตัวกระทำคือ intersect เป็นการทำงานบน 2 ความสัมพันธ์ ผลลัพธ์ที่ได้คือเซตของสมาชิกที่มีอยู่ในทั้ง 2 ความสัมพันธ์ โดยทั้ง 2 ความสัมพันธ์จะต้องมีจำนวนลักษณะประจำเท่ากัน และต้องมีโดเมนเหมือนกันทุกประการ

7) การรวมความสัมพันธ์ (Join) ซึ่งใช้ตัวกระทำคือ join เป็นการทำงานบน 2 ความสัมพันธ์ ผลลัพธ์ที่ได้คือเซตของสมาชิกจากทั้ง 2 ความสัมพันธ์ซึ่งมีค่าของลักษณะประจำที่เป็นเงื่อนไขในการจอยตรงกัน

8) การหาผลหาร (Divide) ซึ่งใช้ตัวกระทำคือ divideby เป็นการทำงานบน 2 ความสัมพันธ์ โดยเลือกแถวของข้อมูลในความสัมพันธ์หนึ่งที่ตรงกับสมาชิกทั้งหมดในอีกความสัมพันธ์หนึ่ง

9) การเปลี่ยนชื่อ (Renaming) ซึ่งใช้ตัวกระทำคือ rename เป็นการทำงานบนความสัมพันธ์เดียว โดยการเปลี่ยนชื่อความสัมพันธ์หรือชื่อลักษณะประจำ

2.2.4 **ภาษาเอสคิวแอล** (Structured Query Language) [8]

ภาษาเอสคิวแอล คือ ภาษาที่ใช้ในการกำหนดและการจัดการข้อมูลในรูปแบบความสัมพันธ์ โดยการกำหนดและการจัดการข้อมูลในความสัมพันธ์เชิงคณิตศาสตร์ จำเป็นต้องใช้พีชคณิตความสัมพันธ์ (Relational Algebra) แต่เนื่องจากพีชคณิตความสัมพันธ์อยู่ในรูปแบบที่เข้าใจได้ยาก จึงได้เกิดภาษาเอสคิวแอลขึ้นมา เพื่อใช้ในการแสดงการกำหนดและการจัดการข้อมูลในความสัมพันธ์ขึ้น ภาษาเอสคิวแอลประกอบด้วย 2 ส่วนคือ

2.2.4.1 **ภาษาในการกำหนดโครงสร้างข้อมูล** (Data Definition Language - DDL)

ภาษาในการกำหนดโครงสร้างข้อมูล เพื่อใช้ในการจัดการโครงสร้างของความสัมพันธ์ เช่น สร้างความสัมพันธ์ (Create Table) สร้างวิว (Create View) สร้างดัชนี (Create Index) แก้ไขความสัมพันธ์ (Ulter Table) และลบความสัมพันธ์ (Drop Table) เป็นต้น

2.2.4.2 **ภาษาในการจัดการข้อมูล** (Data Manipulation Language - DML)

ภาษาในการจัดการข้อมูล เพื่อใช้ในการจัดการข้อมูลในความสัมพันธ์ แบ่งออกเป็น 2 ชนิด คือ

การปรับปรุงข้อมูล (Data Updates) เป็นการเปลี่ยนแปลงข้อมูลในความสัมพันธ์ ประกอบด้วยการเพิ่มข้อมูล (Insert) แก้ไขข้อมูล (Update) และลบข้อมูล (Delete)

การค้นคืนข้อมูล (Data Retrievals) เป็นการเลือกเซตย่อยของข้อมูลในความสัมพันธ์ด้วยเงื่อนไขที่กำหนด เช่น การเลือกแถวของข้อมูลด้วยเงื่อนไขที่กำหนด การเลือกสดมภ์ หรือการจอย นอกจากนี้ยังประกอบด้วยฟังก์ชันมาตรฐาน เช่น การนับจำนวนสมาชิกในความสัมพันธ์ (Count) การหาค่าผลรวม (Sum) การหาค่าสูงสุดและต่ำสุด (Max, Min) เป็นต้น

วิทยานิพนธ์ฉบับนี้ ได้นำมาตรฐานภาษาเอสคิวแอลในเรื่องการจัดการข้อมูลมาเป็นต้นแบบในการสร้างการดำเนินการพื้นฐาน

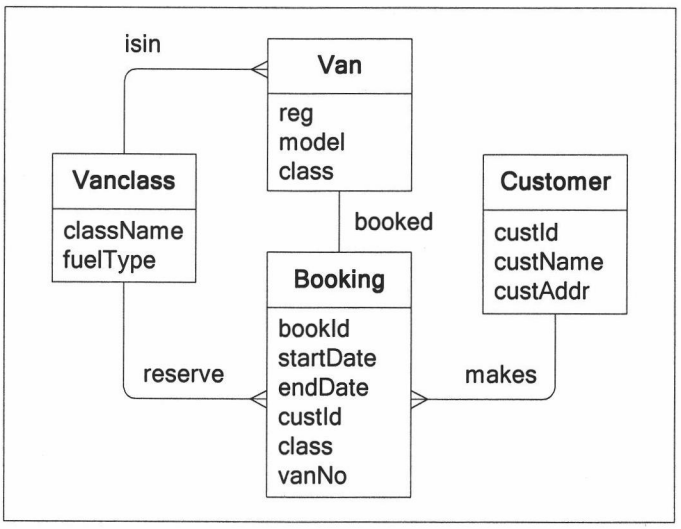
2.2.5 **แผนภาพเอนทิตีและความสัมพันธ์** (Entity-Relationship Diagram) [9],[10]

แผนภาพเอนทิตีและความสัมพันธ์เป็นแบบจำลองข้อมูล (Data model) ชนิดหนึ่งนำเสนอโดย Peter Codd ในปี 1976 ใช้แสดงความสัมพันธ์ระหว่างข้อมูลในตารางต่างๆ ภายในฐานข้อมูล โครงสร้างพื้นฐานของแผนภาพเอนทิตีและความสัมพันธ์ประกอบด้วย

1) เอนทิตี คือ สิ่งที่สนใจในระบบ แทนด้วยสัญลักษณ์สี่เหลี่ยมผืนผ้าและระบุชื่อเอนทิตี ดังรูปที่ 2.1 ประกอบด้วย 4 เอนทิตี คือ Van, Vanclass, Booking และ Customer

2) ความสัมพันธ์ (Relationship) คือ คู่ลำดับของสมาชิกตั้งแต่ 2 เอนทิตีขึ้นไปที่มีความสัมพันธ์กัน แทนด้วยรูปสี่เหลี่ยมข้าวหลามตัดและระบุชื่อความสัมพันธ์ และมีเส้นเชื่อมระหว่างเอนทิตี โดยมีชนิดของความสัมพันธ์ 4 ชนิด คือ แบบหนึ่งต่อหนึ่ง (1-1) แบบหนึ่งต่อหลาย (1-m) แบบหลายต่อหนึ่ง (m-1) และแบบหลายต่อหลาย (m-m) สัญลักษณ์ใช้แทนความสัมพันธ์เขียนได้หลายแบบ ในงานวิจัยนี้จะใช้เส้นตรง โดยสัญลักษณ์ —— แทนความสัมพันธ์แบบหนึ่งต่อหนึ่ง สัญลักษณ์ —≺ แทนความสัมพันธ์แบบหนึ่งต่อหลาย และสัญลักษณ์ ≻— แทนความสัมพันธ์แบบหลายต่อหนึ่ง ดังรูปที่ 2.1

3) ลักษณะประจำ (Attribute) คือ คุณสมบัติของเอนทิตี ลักษณะประจำแบ่งเป็น 2 ประเภท คือ ลักษณะประจำที่ไม่ใช่คีย์ และลักษณะประจำที่เป็นคีย์ ลักษณะประจำสามารถเขียนได้ 2 แบบ คือ วงรีและมีเส้นเชื่อมต่อกับเอนทิตี และการระบุลักษณะประจำลงในเอนทิตีโดยเส้นคั่นระหว่างชื่อเอนทิตีและลักษณะประจำ โดยในงานวิจัยนี้ใช้การเขียนแบบหลัง ดังรูปที่ 2.1



รูปที่ 2.1 แผนภาพเอนทิตีและความสัมพันธ์

2.2.6 **แผนภาพเชิงลำดับชั้น** (Hierarchical Diagram)

แผนภาพเชิงลำดับชั้น เป็นแผนภาพอย่างหนึ่งซึ่งใช้อธิบายลำดับของเหตุการณ์ใดๆ ซึ่งแต่ละเหตุการณ์สามารถแตกเป็นเหตุการณ์ย่อยได้อีก โดยแผนภาพเชิงลำดับชั้นไม่ได้กำหนดรูปแบบของสัญลักษณ์ที่ใช้ในแผนภาพ เพียงแต่ต้องมีเส้นเชื่อมระหว่างเหตุการณ์ต่างๆ เพื่อแสดงถึงลำดับของเหตุการณ์ แผนภาพเชิงลำดับชั้นที่พบบ่อย ได้แก่ แผนภาพองค์กร (Organization chart) แผนภาพโครงสร้าง (Structure Chart) เป็นต้น