บทที่ 4

คลาสที่เกี่ยวกับการจำลองการทำงานของโปรแกรมจำลองการทำงานวงจรไฟฟ้า

บทนี้กล่าวถึงคลาสที่เกี่ยวกับการจำลองการทำงานของโปรแกรมจำลองการทำงานวงจรไฟฟ้า โดยจะแยกกล่าวเป็นหัวข้อดังนี้

4.1 คลาสจำนวนเชิงซ้อนและคลาสเมตริกซ์ทั่วไป (Complex Number Class and Generic Matrix Class)

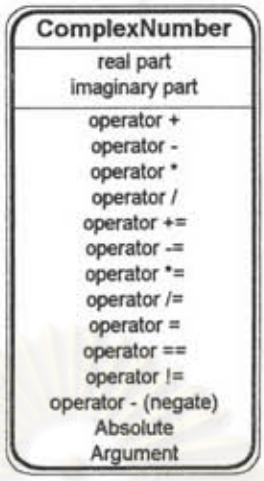
เนื่องจากในการจำลองการทำงานวงจรไฟฟ้านั้นจะต้องสร้างสมการเมตริกซ์ของวงจรไฟฟ้า ดังนั้นก่อนที่จะกล่าวถึงคลาสสมการเมตริกซ์ของวงจรไฟฟ้าจะกล่าวถึงคลาสจำนวนเชิงซ้อน, คลาสเมตริกซ์และคลาสเวกเตอร์ก่อน เพราะว่าคลาสเหล่านี้เป็นส่วนประกอบของคลาสสมการเมตริกซ์ของวงจรไฟฟ้า และจะกล่าวถึงคลาสสมการเมตริกซ์ของวงจรไฟฟ้าในหัวข้อ 4.2

4.1.1 ComplexNumber

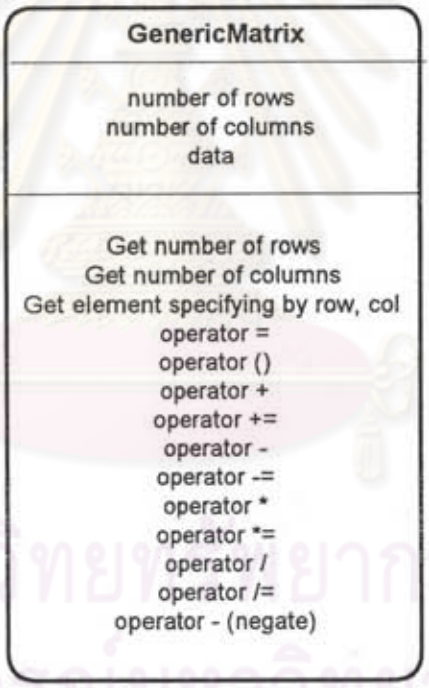
เป็นคลาสที่ครอบคลุมข้อมูลและการจัดการกับจำนวนเชิงซ้อน โดยที่ ComplexNumber มี real part และ imaginary part ซึ่งเป็นจำนวนจริงเป็นคุณสมบัติของวัตถุ และมีวิธีในการจัดการกับคุณสมบัติทั้งสองนี้ให้ทำงานร่วมกันเป็นจำนวนเชิงซ้อน เช่น บวก, ลบจำนวนเชิงซ้อน เป็นต้น คุณสมบัติและวิธีของ ComplexNumber แสดงดังรูปที่ 4.1

4.1.2 GenericMatrix

ทำหน้าที่จัดการกับข้อมูลที่เรียงกันอยู่ในรูปเมตริกซ์ เช่น การใส่ค่าลงในเมตริกซ์ในตำแหน่งแถวและหลักที่กำหนด และการกระทำกันทางคณิตศาสตร์ของเมตริกซ์ เป็นต้น ดังแสดงคุณสมบัติและวิธีของ GenericMatrix ในรูปที่ 4.2



รูปที่ 4.1 คลาสและวัตถุ ComplexNumber



รูปที่ 4.2 คลาส GenericMatrix

จากรูปที่ 4.2 คุณสมบัติ data เป็นรายการแบบเชื่อมโยงของข้อมูลที่เป็นสมาชิกในเมตริกซ์ ซึ่งมีจำนวนเท่ากับผลคูณของ number of rows กับ number of columns ซึ่ง GenericMatrix นี้แบ่งเป็น 2 ชนิด ตามจำนวนของแถวและหลัก คือ

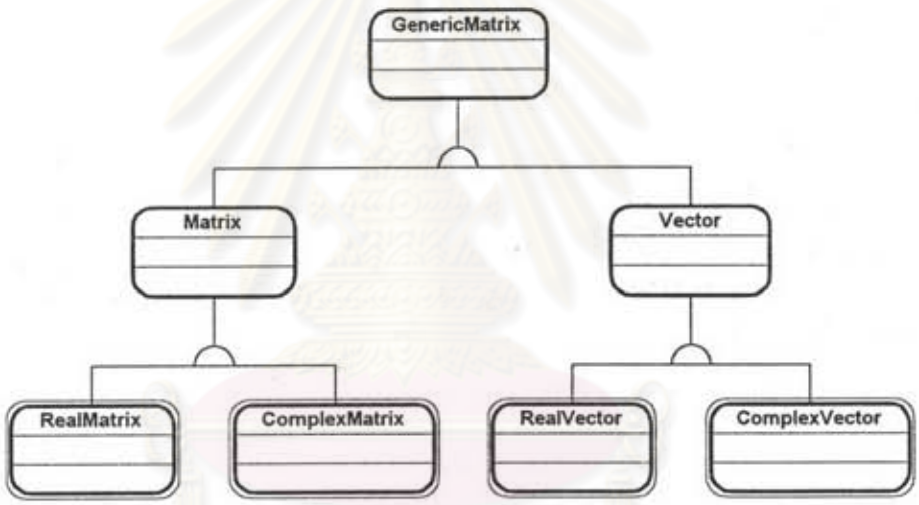
1. Matrix มีคุณสมบัติ number of rows เท่ากับ number of columns

2. Vector มีคุณสมบัติ number of columns เท่ากับ 1

เมื่อมาพิจารณาถึงชนิดของคุณสมบัติ data ของ GenericMatrix ในโปรแกรมที่ต้องการจะออกแบบนั้น มี 2 ชนิดคือ จำนวนจริงและจำนวนเชิงซ้อน ขึ้นกับชนิดของการจำลองการทำงาน ทำให้สามารถแบ่ง Matrix และ Vector ออกเป็นชนิดย่อย ๆ ตามชนิดของคุณสมบัติ data ดังแสดงในรูปที่ 4.3 โดยที่

1. คลาสที่มีคุณสมบัติ data เป็นข้อมูลชนิดจำนวนจริง ได้แก่ RealMatrix, RealVector สำหรับใช้ในการจำลองเพื่อหาจุดทำงานสงบและการจำลองการทำงานทางเวลา

2. คลาสที่คุณสมบัติ data เป็นข้อมูลชนิดจำนวนเชิงซ้อน ได้แก่ ComplexMatrix และ ComplexVector สำหรับใช้ในการจำลองการทำงานทางความถี่



รูปที่ 4.3 ลำดับชั้นของคลาสที่สืบทอดจากคลาส GenericMatrix

จากรูปที่ 4.3 พบว่า ลำดับชั้นของคลาสที่สืบทอดมาจากคลาส Generic-Matrix นั้น สำหรับคลาสที่อยู่ระดับล่างสุดของลำดับชั้นมีความแตกต่างกันเพียงแค่ชนิดของคุณสมบัติ data เท่านั้น ดังนั้นในสำหรับคลาส RealMatrix กับ ComplexMatrix และ RealVector กับ ComplexVector สามารถนำคลาสเท็มเพลทมาช่วยในการเขียนโปรแกรม

4.2 คลาสสมการวงจรไฟฟ้า (Circuit Equation Class)

ในการจำลองการทำงานวงจรไฟฟ้าทั้ง 3 แบบนั้น สามารถเขียนสมการวงจรไฟฟ้าให้อยู่ในรูปของสมการเมตริกซ์

$$Ax = b$$

โดยที่ A เป็นเมตริกซ์ของสัมประสิทธิ์ในสมการวงจรไฟฟ้า ขนาด n x n, n คือจำนวนตัวแปรวงจรไฟฟ้า

b เป็นเวกเตอร์ของค่าคงที่ในสมการวงจรไฟฟ้า ขนาด n

x เป็นเวกเตอร์ของตัวแปรวงจรไฟฟ้าในสมการวงจรไฟฟ้า ขนาด n

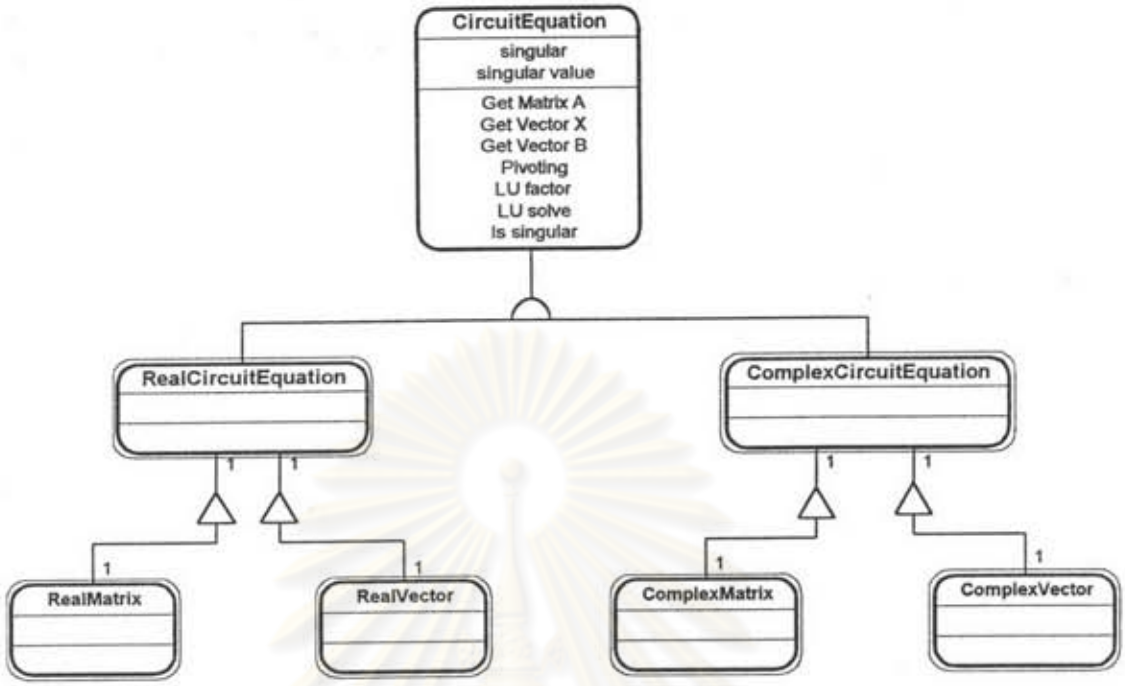
วิธีการแก้สมการเมตริกซ์ของวงจรไฟฟ้าที่ใช้ในโปรแกรมที่จะออกแบบนั้น จะใช้วิธี LU Decomposition with Complete Pivoting และการแก้สมการเมตริกซ์นี้สามารถใช้เมตริกซ์ A และเวกเตอร์ b เพียงสองตัวเท่านั้นในการแก้สมการ โดยไม่จำเป็นต้องใช้เวกเตอร์ x ซึ่งคำตอบของสมการเมตริกซ์นี้จะเก็บไว้ในเวกเตอร์ b แทนที่จะเก็บไว้ที่เวกเตอร์ x

ดังนั้นในโปรแกรมที่ต้องการจะออกแบบนี้จึงมีคลาส CircuitEquation ที่ทำหน้าที่จัดการกับเมตริกซ์ A และเวกเตอร์ b เพื่อแก้สมการเมตริกซ์ของวงจรไฟฟ้า และเนื่องจากในการจำลองเพื่อหาจุดทำงานสงบและการจำลองการทำงานทางเวลา สัมประสิทธิ์ของตัวแปรวงจรไฟฟ้าและค่าคงที่ของสมการวงจรไฟฟ้าจะเป็นจำนวนจริง ส่วนการจำลองการทำงานทางความถี่ สัมประสิทธิ์ของตัวแปรวงจรไฟฟ้าและค่าคงที่ของสมการวงจรไฟฟ้าเป็นจำนวนเชิงซ้อน ทำให้สามารถแบ่ง CircuitEquation ออกเป็น 2 ชนิด คือ

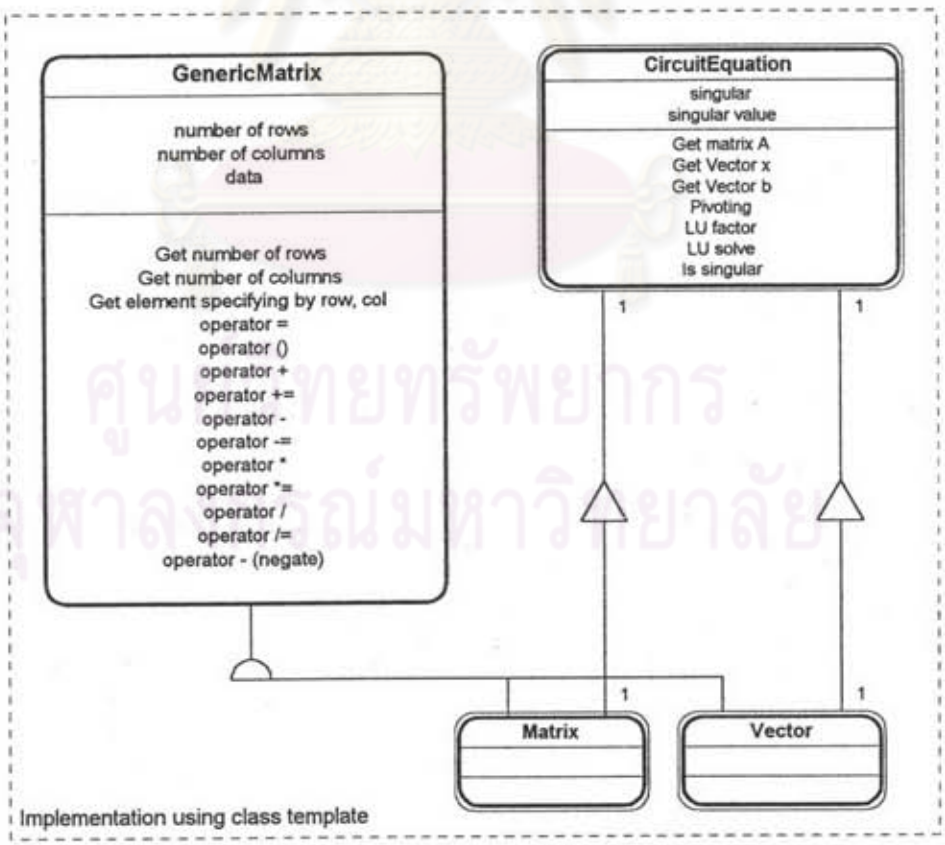
1. RealCircuitEquation ประกอบด้วย RealMatrix และ RealVector ซึ่ง RealCircuitEquation ใช้ในการจำลองเพื่อหาจุดทำงานสงบและการจำลองการทำงานทางเวลา

2. ComplexCircuitEquation ประกอบด้วย ComplexMatrix, และ ComplexVector ใช้ในการจำลองการทำงานทางความถี่

ลำดับชั้นของคลาสที่สืบทอดมาจากคลาส CircuitEquation แสดงไว้ในรูปที่ 4.4 และจากรูปที่ 4.4 นี้จะเห็นว่า RealCircuitEquation และ ComplexCircuitEquation มีส่วนประกอบเท่านั้นที่ต่างกัน ส่วนวิธีจัดการกับส่วนประกอบเหมือนกัน ดังนั้นสามารถใช้คลาสเท็มเพลทมาช่วยในการเขียนโปรแกรม ดังแสดงไว้ในรูปที่ 4.5 โดย CircuitEquation ประกอบด้วย Matrix และ Vector เป็นคลาสและวัตถุที่ทำหน้าที่จัดการกับเมตริกซ์ A และเวกเตอร์ b



รูปที่ 4.4 ลำดับชั้นของคลาสที่สืบทอดจากคลาส CircuitEquation



รูปที่ 4.5 ลำดับชั้นของ CircuitEquation และส่วนประกอบของ CircuitEquation โดยใช้คลาสเท็มเพลท

4.3 คลาสแบบจำลองของอุปกรณ์

ในหัวข้อนี้กล่าวถึง รายละเอียดของส่วนประกอบของ Component ที่ทำหน้าที่ใส่ตรา ประจำอุปกรณ์ของในสมการเมตริกซ์ของวงจรไฟฟ้า ที่เคยอ้างถึงแล้วในหัวข้อที่ 3.4 "รูปแบบทั่วไปของคลาส Component" ในรูปที่ 3.11

4.3.1 Model (รูปที่ 3.11)

เป็นคลาสที่กำหนดวิธีใส่ตราประจำอุปกรณ์ของ Component ชนิดที่เป็นอุปกรณ์อิเล็กทรอนิกส์ลงที่ตำแหน่งแถวและหลักของเมตริกซ์ A และเวกเตอร์ b ของสมการเมตริกซ์ของวงจรไฟฟ้า โดยตำแหน่งแถวและหลักนี้ได้จากคุณสมบัติ node number ของ NodeHandle แต่ละตัวที่เป็นส่วนประกอบของ Component วิธีใส่ตราประจำอุปกรณ์ที่กำหนดไว้มี 3 แบบ คือ Stamp DC, Stamp TR และ Stamp AC สำหรับการจำลองเพื่อหาจุดทำงานสงบ, การจำลองทำงานทางเวลา และการจำลองการทำงานทางความถี่ ตามลำดับ

คลาส Model นี้สืบทอดจากคลาส CObject ของคลาสไลบรารี่ MFC เพราะต้องการใช้วิธีการเก็บคุณสมบัติของ Model ลงไฟล์และอ่านคุณสมบัติของ Model จากไฟล์ ทำให้คลาสที่สืบทอดต่อจากคลาส Model นี้มีวิธีนี้ด้วย

4.3.2 SpecificModel (รูปที่ 3.11)

สืบทอดจากคลาส Model และกำหนดคุณสมบัติเฉพาะของแบบจำลองของอุปกรณ์อิเล็กทรอนิกส์ชนิดนั้นเพิ่มเติมลงไป ซึ่งคลาส SpecificModel นี้หมายถึงคลาสแบบจำลองของอุปกรณ์อิเล็กทรอนิกส์ SpecificComponent ในโปรแกรมต้นแบบ ดังนี้

4.3.1 ResistorModel ในรูปที่ 3.13

4.3.2 CapacitorModel ในรูปที่ 3.15

4.3.3 InductorModel ในรูปที่ 3.17

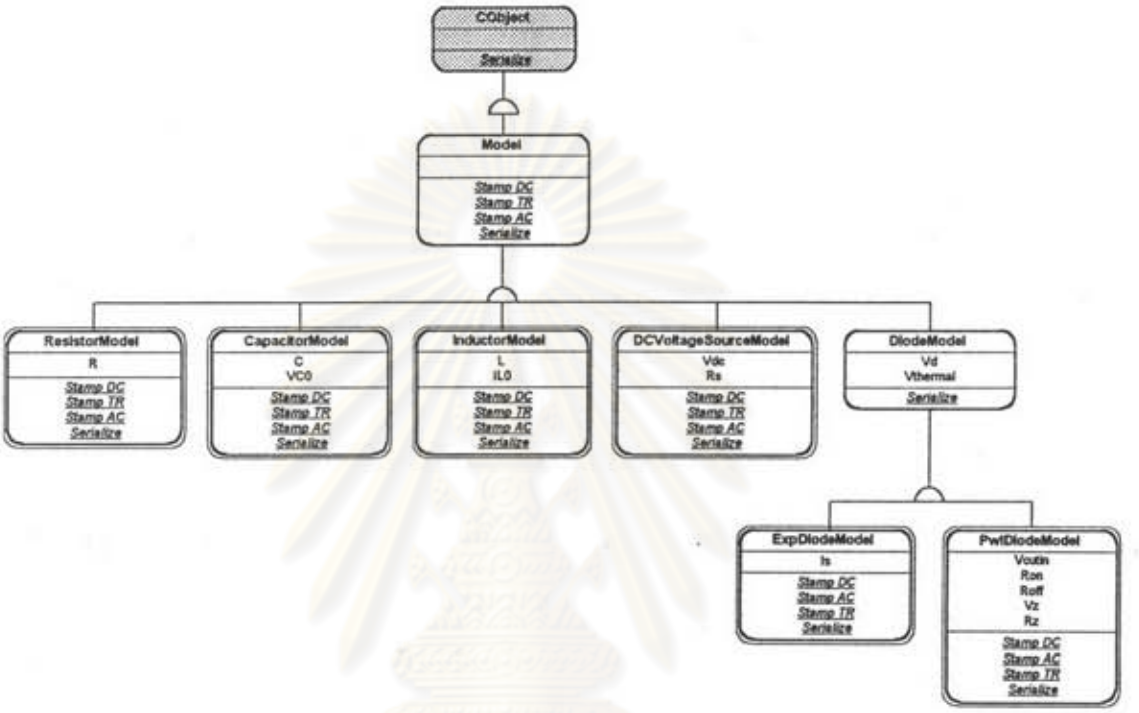
4.3.4 DCVoltageSourceModel ในรูปที่ 3.19

4.3.5 DiodeModel ในรูปที่ 3.21 ซึ่งแบ่งออกเป็น 2 ชนิด ตามชนิดของสมการของไดโอดที่ใช้ในการจำลองการทำงาน คือ

4.3.5.1 ExpDiodeModel สมการของไดโอดเป็น Exponential Equation

4.3.5.2 PwlDiodeModel สมการของไดโอดเป็น Piecewise Linear Equation

ลำดับชั้นของคลาสที่สืบทอดจากคลาส Model แสดงไว้ในรูปที่ 4.6



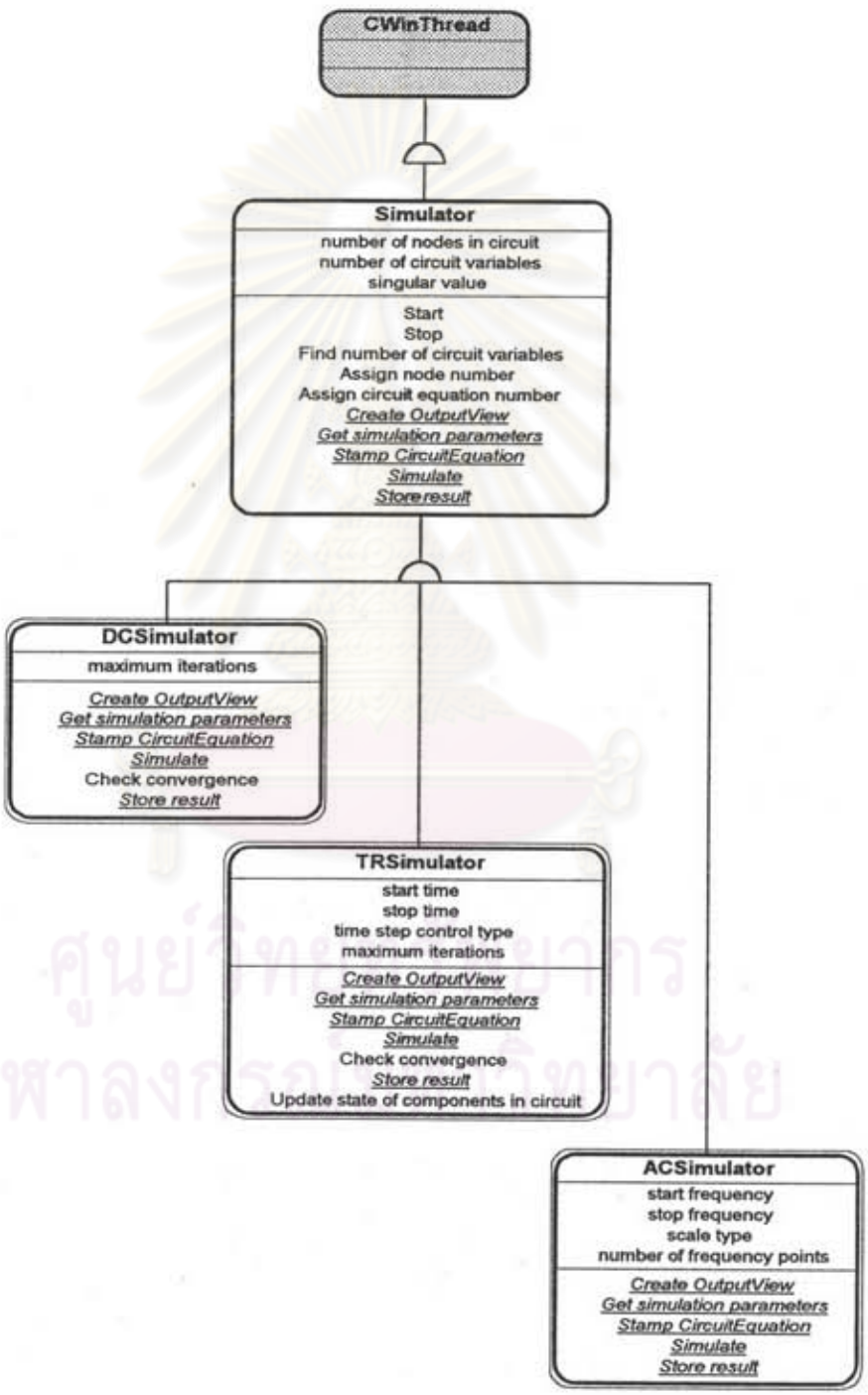
รูปที่ 4.6 ลำดับชั้นของคลาสที่สืบทอดจากคลาส Model

4.4 คลาสตัวควบคุมการแก้สมการวงจรไฟฟ้า (Simulator Class)

คลาสที่กำหนดทำหน้าที่ควบคุมการหาคำตอบของ CircuitEquation คือ Simulator ซึ่งเป็นคลาสที่สืบทอดมาจากคลาส CWinThread ของคลาสไลบรารี่ MFC ซึ่งคลาส CWinThread เป็นคลาสที่สนับสนุนการทำงานแบบหลายเส้นทาง (Multithreading) ทำให้ในระหว่างที่จำลองการทำงานอยู่ผู้ใช้ยังสามารถติดต่อกับโปรแกรมได้ เช่น สามารถสั่งยกเลิกการจำลองการทำงานได้โดยเลือกคำสั่งจากเมนู หรือ สามารถเลื่อนหน้าต่างที่อยู่บนหน้าต่างหลักได้ในขณะจำลองการทำงานอยู่ เป็นต้น

คลาส Simulator จะเป็นเบสคลาสของคลาส DCSimulator, TRSimulator และ ACSimulator ซึ่งคลาส Simulator จะครอบคลุมการจัดการด้านการสร้างเส้นทาง (Thread) ใหม่เมื่อผู้ใช้สั่งเริ่มต้นการจำลองการทำงานและการทำลายเส้นทางเมื่อการจำลองการทำงานสิ้นสุด

หรือผู้ใช้สั่งหยุดการจำลองการทำงานวงจรไฟฟ้า นอกจากนั้นยังประกาศฟังก์ชันเสมือนที่จำเป็นสำหรับการจำลองการทำงานวงจรไฟฟ้าแต่ละแบบ ซึ่งคลาสที่สืบทอดนั้นเพียงแต่เขียนวิธีการจำลองการทำงานแบบต่าง ๆ ลงไปในฟังก์ชันเหล่านั้น ดังแสดงในรูปที่ 4.7



รูปที่ 4.7 ลำดับชั้นของ Simulator

4.5 คลาสผลการจำลองการทำงานวงจรไฟฟ้า

คลาสที่เกี่ยวข้องกับผลของการจำลองการทำงานมี 3 ชนิด คือ

4.5.1 DCOutput

เป็นผลของการจำลองเพื่อหาจุดทำงานสงบ ซึ่งก็คือผลลัพธ์ของสมการเมตริกซ์ และ DCOutput นี้จะถูกนำไปเก็บที่ DCOutputView โดย DCSimulator

4.5.2 TROutput

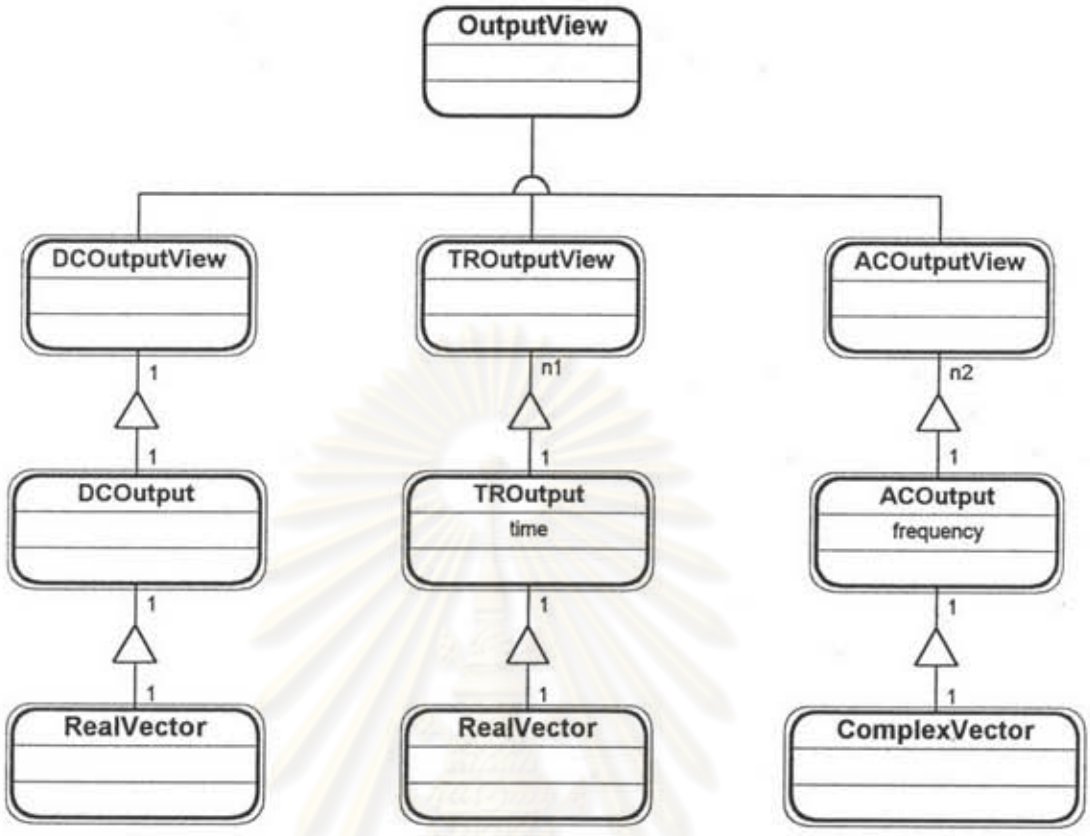
เป็นผลของการจำลองการทำงานทางเวลา ประกอบด้วยเวลาและผลลัพธ์ของสมการเมตริกซ์ที่เวลานั้น ซึ่ง TROutput จะถูกนำไปเก็บที่ TROutputView โดย TRSimulator

4.5.3 ACOutput

เป็นผลของการจำลองการทำงานทางความถี่ ประกอบด้วยความถี่และผลลัพธ์ของสมการเมตริกซ์ที่ความถี่นั้น ซึ่ง ACOutput จะถูกนำไปเก็บที่ ACOutputView โดย ACSimulator

รูปที่ 4.8 แสดงส่วนประกอบของ DCOutput, TROutput และ ACOutput และส่วนประกอบของ DCOutputView, TROutputView และ ACOutputView

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



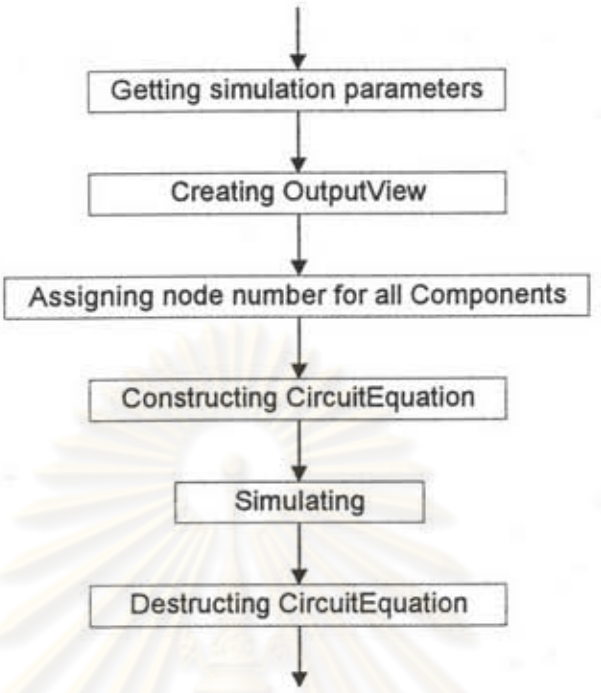
n1 = จำนวนผลลัพธ์ที่คำนวณโดยใช้วิธีการจำลองการทำงานทางเวลา (Transient Simulation)

n2 = จำนวนผลลัพธ์ที่คำนวณโดยใช้วิธีการจำลองการทำงานทางความถี่ (AC Simulation)

รูปที่ 4.8 ส่วนประกอบ DCOutput, TROutput และ ACOutput ของ DCOutputView, TROutputView และ ACOutputView ตามลำดับ

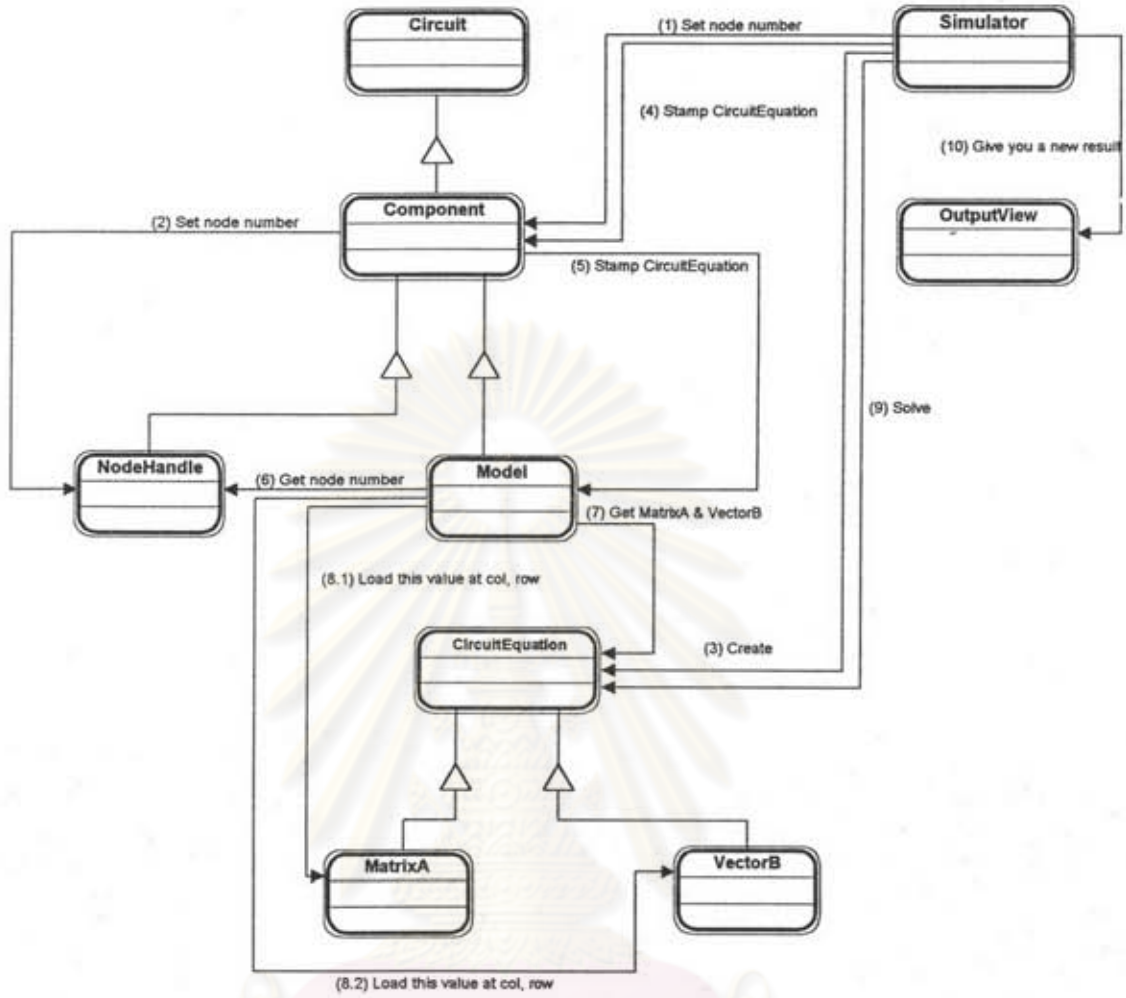
4.6 พฤติกรรมของ Simulator และการติดต่อส่งข่าวสารระหว่างวัตถุที่เกี่ยวข้องกับการจำลองการทำงาน

ในรูปที่ 4.9 แสดงพฤติกรรมของ Simulator เมื่อผู้ใช้สั่งให้ Simulator เริ่มจำลองการทำงานไม่ว่าจะเป็นแบบใดใน 3 แบบ Simulator จะแสดงกรอบข้อความเพื่อรับค่าพารามิเตอร์ของการจำลองการทำงาน, สร้างหน้าต่าง Output View, กำหนดหมายเลขประจำปมให้กับทุก NodeHandle ของทุก Component ใน Circuit แล้วสร้าง CircuitEquation เพื่อใช้ในการคำนวณหาผลของการจำลองการทำงานตามวิธีการจำลองการทำงานแต่ละแบบ



รูปที่ 4.9 พฤติกรรมโดยทั่วไปของ Simulator

จากรูปที่ 4.10 เป็นการติดต่อระหว่างวัตถุต่าง ๆ ที่เกี่ยวข้องกับการจำลองการทำงานวงจรไฟฟ้า ซึ่งในรูปนี้จะอธิบายถึงในกรณีทั่ว ๆ ไปของ Simulator แต่ละชนิดว่า Simulator จะเริ่มการจำลองการทำงานโดยการกำหนดหมายเลขประจำปมของ Component ทุกตัวใน Circuit เมื่อกำหนดหมายเลขประจำปมให้กับทุก Component เรียบร้อยแล้ว ก็จะรู้ว่า Circuit ที่ผู้ใช้วาดนี้มีจำนวนตัวแปรวงจรไฟฟ้ากี่ตัว เพื่อนำไปใช้ในการสร้าง CircuitEquation เมื่อสร้าง CircuitEquation เรียบร้อยแล้ว Simulator ก็จะส่ง CircuitEquation นั้นไปให้ Component ทุกตัวใน Circuit ที่มี Model ใส่ตราประจำอุปกรณ์ของ Component ลงใน CircuitEquation ซึ่งวิธีตราประจำอุปกรณ์ขึ้นกับชนิดของ Component และชนิดของการจำลองการทำงานวงจรไฟฟ้า เมื่อ Component ทุกตัวประทับค่าลงใน CircuitEquation เรียบร้อยแล้ว Simulator จะสั่งให้ CircuitEquation หาคำตอบ จากนั้น Simulator จะนำคำตอบที่ได้จาก CircuitEquation ไปเก็บที่ OutputView



รูปที่ 4.10 การติดต่อระหว่างวัตถุขณะจำลองการทำงานวงจรไฟฟ้า

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย