

ส่วนประกอบกราฟสำหรับสร้างแฟ้มข้อมูลเอสวีจี



นางสาววิมลศรี หอมหวล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2547

ISBN 974-17-6833-8

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

A GRAPHIC COMPONENT FOR GENERATING SVG FILE



Miss Wimonsri Homhual

สถาบันวิทยบริการ

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2004

ISBN 974-17-6833-8

หัวข้อวิทยานิพนธ์ ส่วนประกอบกราฟสำหรับสร้างแฟ้มข้อมูลเอสวีจี
โดย นางสาววิมลศรี หอมหวล
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ
อาจารย์ที่ปรึกษาร่วม ผู้ช่วยศาสตราจารย์ ดร.ธราทิพย์ สุวรรณศาสตร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(อาจารย์ ดร.อาทิตย์ ทองทักษ์)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

..... อาจารย์ที่ปรึกษาร่วม
(ผู้ช่วยศาสตราจารย์ ดร.ธราทิพย์ สุวรรณศาสตร์)

..... กรรมการ
(อาจารย์ ดร.เสรษฐา ปานงาม)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิมลศรี หอมหวล : ส่วนประกอบกราฟสำหรับสร้างแฟ้มข้อมูลเอสวีจี. (A GRAPHIC COMPONENT FOR GENERATING SVG FILE) อ.ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ, อ.ที่ปรึกษาร่วม : ผู้ช่วยศาสตราจารย์ ดร.ธราทิพย์ สุวรรณศาสตร์, 64 หน้า ISBN 974-17-6833-8.

การแสดงสารสนเทศในรูปแบบของกราฟช่วยให้ผู้ใช้งานสามารถรับรู้ข้อมูลอย่างเป็นรูปธรรม การวิเคราะห์ข้อมูลและการตัดสินใจสามารถทำได้ง่ายขึ้น โดยเฉพาะการแสดงกราฟในรูปแบบเอสวีจี ซึ่งจะสามารถแสดงผลบนเว็บได้ดี เพราะขนาดของแฟ้มข้อมูลประเภทนี้จะมีขนาดเล็กทำให้การแสดงผลทำได้อย่างรวดเร็ว ความคมชัดสูง และสามารถปรับเปลี่ยนขนาดของรูปได้ตามต้องการ และเนื่องด้วยคุณสมบัติการพัฒนาซอฟต์แวร์แบบส่วนประกอบซึ่งทำให้ส่วนที่ได้รับการพัฒนานั้นๆสามารถนำกลับมาใช้ใหม่ได้ นอกจากนี้ยังมีความถูกต้องน่าเชื่อถือและความยืดหยุ่นสูง

งานวิจัยได้ทำการออกแบบและพัฒนาส่วนประกอบกราฟสำหรับสร้างแฟ้มข้อมูลเอสวีจี โดยนำเข้าข้อมูลจากฐานข้อมูลไมโครซอฟต์เอสคิวแอลเซิร์ฟเวอร์ และให้ผู้ใช้สามารถเลือกแสดงผลเป็นกราฟได้ 3 รูปแบบ ได้แก่ กราฟแท่ง กราฟเส้น และ กราฟวงกลม เนื่องด้วยมีการออกแบบและพัฒนาในแบบส่วนประกอบ ทำให้ส่วนประกอบกราฟนี้มีคุณสมบัติในการนำกลับมาใช้ใหม่ จากการทดสอบการใช้งานผ่านภาษาซีชาร์ป และ ภาษาวิซวลเบสิกดอทเน็ต พบว่า ส่วนประกอบกราฟสามารถทำงานได้ถูกต้องและสามารถสร้างกราฟได้ตามที่ต้องการ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชาวิศวกรรมคอมพิวเตอร์.....
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์.....
ปีการศึกษา2547.....

ลายมือชื่อผู้ผลิต
ลายมือชื่ออาจารย์ที่ปรึกษา
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....

4471446421: MAJOR COMPUTER SCIENCE

KEY WORD: GRAPH, COMPONENT, SVG

WIMONSRI HOMHUAL : A GRAPHIC COMPONENT FOR GENERATING SVG FILE.

THESIS ADVISOR: ASST.PROF.WIWAT VATANAWOOD,PH.D., THESIS CO-

ADVISOR: ASST.PROF.TARATIP SUWANNASART,PH.D.,64pp. ISBN 974-17-6833-8

Information presented in graph helps user to understand, analyze, and make decision more easily. A graphic file in SVG format is suitable for displaying on the web because it has small size, high resolution, and scalable. Besides the advantages of designing and developing software in component based approach are reusability, reliability, and high flexibility.

The objective of this research is to design and develop a graph component that will generate SVG file by importing data from Microsoft SQL Server. Three different views of graph can be generated; which are bar, line, and pie graph. Thus, through using component based software development, source code is reusable. According to the testing of conformance development with C# and Visual Basic .Net, this component works correctly and meets all requirements.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Department Computer Engineering.....

Field of study Computer Science.....

Academic year 2004.....

Student's Signature.....

Advisor's Signature.....

Co-Advisor's Signature.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงลงได้ โดยได้รับความช่วยเหลืออย่างมากจาก ผู้ช่วยศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ อาจารย์ที่ปรึกษาวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร.ธราทิพย์ สุวรรณศาสตร์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ซึ่งได้กรุณาให้คำแนะนำ ข้อคิดเห็นและแนวทางในการค้นคว้าและแก้ปัญหาต่างๆ ด้วยดีมาตลอด ผู้วิจัยขอขอบพระคุณอาจารย์เป็นอย่างสูง มา ณ ที่นี้

นอกเหนือจากนี้ ผู้วิจัยขอขอบพระคุณกรรมการวิทยานิพนธ์ทุกท่านที่กรุณาเสียสละเวลาให้คำแนะนำ และตรวจสอบวิทยานิพนธ์ฉบับนี้

ขอขอบคุณพี่ๆ เพื่อนๆ ร่วมงานที่เครือข่ายเจริญโภคภัณฑ์ที่ให้ความสนับสนุนช่วยเหลือผู้วิจัย ทั้งทางด้านความคิดเห็น เครื่องมืออุปกรณ์และกำลังใจ นอกจากนี้ยังพี่ๆ เพื่อนๆ ร่วมรุ่นที่คอยให้กำลังใจผู้วิจัยเสมอมา

สุดท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดา มารดา และครอบครัวที่ให้ความเข้าใจและสนับสนุนผู้วิจัยทางการศึกษาตลอดมา

วิมลศรี หอมหวล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ง
สารบัญ.....	จ
สารบัญตาราง.....	ช
สารบัญภาพ.....	ณ
บทที่	
1.บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 ขั้นตอนของการดำเนินการวิจัย.....	3
2.ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 การพัฒนาซอฟต์แวร์แบบส่วนประกอบ.....	4
2.1.1 วงจรการพัฒนาซอฟต์แวร์แบบส่วนประกอบ.....	4
2.1.2 ประโยชน์ของการพัฒนาซอฟต์แวร์แบบส่วนประกอบ.....	5
2.1.3 เทคโนโลยีของการพัฒนาซอฟต์แวร์แบบส่วนประกอบ.....	6
2.2 เอสวีจี.....	7
2.2.1 ข้อดีของเอสวีจี.....	7
2.2.2 ความแตกต่างระหว่างรูปแบบเอสวีจีกับรูปแบบบิทแมพหรือรูปแบบจีฟ.....	8
2.2.3 โครงสร้างพื้นฐานของเอสวีจี.....	9
2.2.4 การแสดงผลของแฟ้มเอสวีจี.....	10
2.2.5 รูปทรงพื้นฐาน.....	12
2.3 ซีแอลอาร์.....	14
2.3.1 การทำงานของซีแอลอาร์.....	14
2.4 งานวิจัยที่เกี่ยวข้อง.....	15

2.4.1 การใช้เครื่องมือพาวเวอร์บีวเดอร์พัฒนาส่วนประกอบกราฟบนอีเอเชิร์ฟเวอร์ แอปพลิเคชันเซิร์ฟเวอร์ในรูปแบบเอสวีจี	15
2.4.2 บาดิกเอสวีจีทูลคิด	15
3.การออกแบบและพัฒนาส่วนประกอบกราฟสำหรับสร้างเพิ่มข้อมูลเอสวีจี	17
3.1 แผนภาพยูสเคส	17
3.1.1 ยูสเคส Create GraphConfiguration	17
3.1.2 ยูสเคส Edit GraphConfiguration	19
3.1.3 ยูสเคส Create SVG	20
3.2 การออกแบบคลาส	22
3.2.1 ความสัมพันธ์ระหว่างคลาส	22
3.2.2 รายละเอียดคลาส	22
3.3 แผนภาพซีเควนซ์	28
3.3.1 แผนภาพซีเควนซ์ของการตั้งค่าข้อกำหนด	28
3.3.2 แผนภาพซีเควนซ์ของการแก้ไขข้อกำหนด	28
3.3.3 แผนภาพซีเควนซ์ของการสร้างเพิ่มข้อมูลเอสวีจีของกราฟแท่ง	29
3.3.4 แผนภาพซีเควนซ์ของการสร้างเพิ่มข้อมูลเอสวีจีของกราฟเส้น	31
3.3.5 แผนภาพซีเควนซ์ของการสร้างเพิ่มข้อมูลเอสวีจีของกราฟวงกลม	33
3.4 แผนภาพกิจกรรม	34
3.4.1 แผนภาพกิจกรรมแสดงการตั้งค่าข้อกำหนดลงในเพิ่มข้อกำหนด	34
3.4.2 แผนภาพกิจกรรมแสดงการแก้ไขข้อกำหนดลงในเพิ่มข้อกำหนด	35
3.4.3 แผนภาพกิจกรรมแสดงการสร้างเพิ่มข้อมูลเอสวีจี	36
3.5 แผนภาพดีฟลอยเมนต์	36
3.6 ลักษณะข้อมูลนำเข้า	39
4.การทดสอบการใช้งานส่วนประกอบกราฟ	43
4.1 การทดสอบส่วนประกอบกราฟด้วยภาาษาวิซวลเบสิกคอตเน็ต	43
4.1.1 การแสดงผลกราฟแท่ง	44

4.1.2 การแสดงผลกราฟเส้น.....	45
4.1.3 การแสดงผลกราฟวงกลม.....	47
4.2 การทดสอบส่วนประกอบกราฟด้วยภาษาซีชาร์ป.....	49
4.3 การวิเคราะห์ผลการทดสอบ.....	50
5.บทสรุปปัญหาและข้อเสนอแนะ.....	54
5.1 สรุปผลการวิจัย.....	54
5.2 ประโยชน์ที่คาดว่าจะได้รับ.....	54
5.3 ข้อจำกัดของส่วนประกอบกราฟสำหรับสร้างแฟ้มข้อมูลเอชวีจี.....	55
5.4 ข้อเสนอแนะ.....	55
รายการอ้างอิง.....	56
ภาคผนวก.....	57
ภาคผนวก ก. ตัวอย่างแฟ้มข้อมูลเอชวีจีของกราฟในรูปแบบต่างๆ.....	58
ประวัติผู้เขียนวิทยานิพนธ์.....	64

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงคำสั่งที่ใช้สำหรับการสร้างพาธ.....	13
ตารางที่ 3.1 รายละเอียดของคลาส Create GraphConfiguration	18
ตารางที่ 3.2 รายละเอียดของคลาส Edit GraphConfiguration	19
ตารางที่ 3.3 รายละเอียดของคลาส Create SVG.....	21
ตารางที่ 3.4 รายละเอียดของคลาส GraphCtrl	23
ตารางที่ 3.5 รายละเอียดของคลาส GraphConfiguration.....	24
ตารางที่ 3.6 รายละเอียดของคลาส Graph	25
ตารางที่ 3.7 รายละเอียดของคลาส BarGraph	26
ตารางที่ 3.8 รายละเอียดของคลาส LineGraph	27
ตารางที่ 3.9 รายละเอียดของคลาส PieGraph	27
ตารางที่ 3.10 ตัวอย่างรูปแบบข้อมูลนำเข้าที่ต้องผ่านการแปลงก่อนนำไปสร้างกราฟ.....	41
ตารางที่ 3.11 ตัวอย่างรูปแบบข้อมูลนำเข้าที่แปลงให้อยู่ในรูปแบบพิกัดเทเบิล	42
ตารางที่ 4.1 สรุปขนาดเพิ่มรูปภาพระหว่างรูปแบบเอสวีจีกับรูปแบบเจพีค.....	51
ตารางที่ 4.2 เปรียบเทียบคุณสมบัติของส่วนประกอบกราฟ	52

สารบัญภาพ

	หน้า
รูปที่ 2.1 วงจรการพัฒนาซอฟต์แวร์แบบส่วนประกอบ	4
รูปที่ 2.2 อีลีเมนต์ ซีโอแอดทริบิวต์ และค่าแอดทริบิวต์	9
รูปที่ 2.3 แผนภาพแสดงประเภทโครงสร้างของแฟ้มเอสวีจี	11
รูปที่ 2.4 การเรียกใช้งานแฟ้มเอสวีจีด้วยแท็กออบเจกต์	12
รูปที่ 2.5 การใช้แท็กสีเหลี่ยม	12
รูปที่ 2.6 การใช้แท็กเส้น	13
รูปที่ 2.7 การใช้แท็กพาท	14
รูปที่ 2.8 หลักการทำงานของซีแอลอาร์	15
รูปที่ 3.1 แผนภาพยูสเคสของส่วนประกอบกราฟสำหรับสร้างแฟ้มข้อมูลเอสวีจี	17
รูปที่ 3.2 แผนภาพแสดงความสัมพันธ์ระหว่างคลาส	23
รูปที่ 3.3 แผนภาพซีเควนซ์ของการตั้งค่าข้อกำหนด	28
รูปที่ 3.4 แผนภาพซีเควนซ์ของการแก้ไขแฟ้มข้อกำหนด	29
รูปที่ 3.5 แผนภาพซีเควนซ์ของการสร้างแฟ้มข้อมูลเอสวีจีของกราฟแท่ง	31
รูปที่ 3.6 แผนภาพซีเควนซ์ของการสร้างแฟ้มข้อมูลเอสวีจีของกราฟเส้น	33
รูปที่ 3.7 แผนภาพซีเควนซ์ของการสร้างแฟ้มข้อมูลเอสวีจีของกราฟวงกลม	35
รูปที่ 3.8 แผนภาพกิจกรรมการกำหนดข้อกำหนดของกราฟ	36
รูปที่ 3.9 แผนภาพกิจกรรมการแก้ไขแฟ้มข้อกำหนดของกราฟ	37
รูปที่ 3.10 แผนภาพกิจกรรมการสร้างแฟ้มข้อมูลเอสวีจีของกราฟ	38
รูปที่ 3.11 แผนภาพดีพลอยเมนต์	38
รูปที่ 3.12 รูปแบบของข้อความสั่งเอสคิวแอลที่กำหนดไว้	39
รูปที่ 3.13 ตัวอย่างของข้อความสั่งเอสคิวแอล	39
รูปที่ 3.14 แสดงตัวอย่างผลข้อมูลที่ได้จากข้อความสั่งเอสคิวแอล	40
รูปที่ 3.15 ตัวอย่างรูปภาพที่ได้จากข้อความสั่งเอสคิวแอล	40
รูปที่ 4.1 หน้าจอเริ่มต้นการแสดงผล	43
รูปที่ 4.2 แฟ้มข้อกำหนดสำหรับแสดงกราฟแท่ง	44
รูปที่ 4.3 การเรียกใช้ส่วนประกอบกราฟสำหรับกราฟแท่งโดยภาษาวิชวลเบสิกดอทเน็ต	44

รูปที่ 4.4 ข้อมูลและรูปภาพแท่ง	45
รูปที่ 4.5 เพิ่มข้อกำหนดสำหรับแสดงกราฟเส้น	46
รูปที่ 4.6 การเรียกใช้ส่วนประกอบกราฟสำหรับกราฟเส้นโดยภาษาวิซวลเบสิกดอทเน็ต	46
รูปที่ 4.7 ข้อมูลและรูปภาพเส้น	47
รูปที่ 4.8 เพิ่มข้อกำหนดสำหรับแสดงกราฟวงกลม	47
รูปที่ 4.9 การเรียกใช้ส่วนประกอบกราฟสำหรับกราฟวงกลมโดยภาษาวิซวลเบสิกดอทเน็ต	48
รูปที่ 4.10 ข้อมูลและรูปภาพวงกลม	48
รูปที่ 4.11 หน้าจอสำหรับแสดงความคิดเห็น	49
รูปที่ 4.12 เพิ่มข้อกำหนดสำหรับแสดงกราฟแท่ง	50
รูปที่ 4.13 การเรียกใช้ส่วนประกอบกราฟสำหรับการสำรวจความนิยมโดยภาษาซีชาร์ป	50
รูปที่ 4.14 กราฟคะแนนความนิยมของพรรคการเมืองต่างๆ	51

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการพัฒนาโปรแกรมเชิงพาณิชย์นิยมพัฒนาในรูปแบบของโปรแกรมประยุกต์บนเว็บ (Web Application) ซึ่งการแสดงผลกราฟบนเว็บนั้นจะสามารถจัดทำได้ในหลายรูปแบบ ทั้งในรูปแบบของรูปภาพ (Images) รูปแบบเอสวีจี (SVG: Scalable Vector Graphic) และรูปแบบอื่นๆ

การแสดงผลกราฟในรูปแบบของรูปภาพ เป็นวิธีที่นิยมทำกันมากที่สุดวิธีหนึ่ง แต่เนื่องจากการแสดงผลรูปภาพที่มีความละเอียดสูงจะทำให้ขนาดของแฟ้มข้อมูลมีขนาดใหญ่ การส่งข้อมูลมาแสดงผลบนเว็บเป็นไปได้ช้า ในทางกลับกันถ้าต้องการเพิ่มความเร็วในส่งข้อมูลมาแสดงผลจะต้องทำการลดขนาดของแฟ้มข้อมูลรูปภาพนั้นๆ รูปที่ได้มีความละเอียดต่ำทำให้ได้รูปภาพที่ไม่คมชัด และรูปภาพในลักษณะนี้ยังมีข้อเสียที่การแสดงผลบนเว็บนั้นไม่สามารถเปลี่ยนแปลงขนาดของรูปได้ ในขณะที่การแสดงผลกราฟด้วยรูปแบบเอสวีจีขนาดของแฟ้มข้อมูลจะมีขนาดเล็กทำให้การส่งข้อมูลมาแสดงผลบนเว็บได้รวดเร็ว นอกจากนี้ยังสามารถแสดงรูปภาพบนเว็บได้อย่างละเอียด รวมทั้งพิมพ์รูปภาพออกทางเครื่องพิมพ์ได้อย่างมีคุณภาพ ซึ่งในปัจจุบันการแสดงผลกราฟนิยมใช้ภาษาพีเอชพี (PHP : Professional Home Pages) โดยใช้ไลบรารี (Library) สำหรับการแสดงผลรูปภาพ ผลลัพธ์ที่ได้มีลักษณะเป็นภาพ แต่เนื่องจากไลบรารีของพีเอชพีไม่ได้พัฒนาอยู่ในรูปแบบส่วนประกอบ ทำให้สามารถนำมาใช้ได้เฉพาะกับโปรแกรมภาษาพีเอชพีเท่านั้น

ในปัจจุบันวิธีการพัฒนาซอฟต์แวร์มีแนวโน้มไปสู่การพัฒนาซอฟต์แวร์แบบส่วนประกอบ (Component-Based Software Development) [1] เนื่องจากเป็นการพัฒนาซอฟต์แวร์ที่ช่วยในการลดต้นทุนและเวลาของการพัฒนาซอฟต์แวร์ในระบบที่มีความซับซ้อน นอกจากนี้ซอฟต์แวร์ที่พัฒนาแบบส่วนประกอบมีความน่าเชื่อถือมากกว่าซอฟต์แวร์ที่พัฒนาด้วยวิธีการแบบปกติ ซึ่งส่งผลให้ระบบที่พัฒนาในรูปแบบส่วนประกอบมีความถูกต้องและน่าเชื่อถือ การพัฒนาแบบนี้ยังมีความยืดหยุ่นสูงในการเปลี่ยนแปลงส่วนประกอบใหม่แทนที่ส่วนประกอบเดิม ผู้พัฒนาสามารถนำส่วนประกอบจากระบบประยุกต์หนึ่งมาใช้ยังอีกระบบประยุกต์หนึ่งได้ เนื่องจากซอฟต์แวร์ที่พัฒนาแบบส่วนประกอบมีความสามารถนำกลับมาใช้ใหม่ได้ (Reusable)

1.2 วัตถุประสงค์ของการวิจัย

เพื่อออกแบบและพัฒนาส่วนประกอบที่แสดงเป็นแผนภาพกราฟโดยดึงข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ แล้วนำข้อมูลที่ได้มาสร้างอยู่ในรูปของแฟ้มเอสวีจี

1.3 ขอบเขตของการวิจัย

- 1) ใช้แฟ้มเอสวีจี สำหรับแสดงกราฟฟิกทางอินเทอร์เน็ต
- 2) ข้อกำหนดเอสวีจีที่ใช้จะเป็นไปตามมาตรฐานรุ่น 1.0 เป็นอย่างน้อย ซึ่งกำหนดไว้ในดีที่สุด
- 3) ใช้แฟ้มเอ็กซ์เอ็มแอลในการกำหนดโครงสร้างขอบเขตของข้อมูลในการแสดงผลในรูปกราฟ
- 4) ข้อกำหนดเอ็กซ์เอ็มแอลที่ใช้จะเป็นไปตามมาตรฐานรุ่น 1.0 เป็นอย่างน้อย
- 5) ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ที่ใช้ทดลองในการวิจัยคือไมโครซอฟต์เอสคิวแอลเซิร์ฟเวอร์
- 6) เครื่องมือที่พัฒนาขึ้นจะถูกพัฒนาด้วยภาษาซีชาร์ป บนเครื่องมืออวิซวลสตูดิโอเดสทอปเน็ต ภายใต้ระบบปฏิบัติการวินโดวส์
- 7) ส่วนประกอบสามารถสร้างกราฟแท่ง กราฟเส้น กราฟวงกลม โดยกราฟแต่ละแบบมีคุณสมบัติดังนี้
 - กราฟแท่ง ประกอบด้วย ชื่อกราฟ ชื่อแกน X,Y คำอธิบายข้อมูล เส้นตารางหลัก (Grid)
 - กราฟเส้น ประกอบด้วย ชื่อกราฟ ชื่อแกน X,Y คำอธิบายข้อมูล เส้นตารางหลัก (Grid)
 - กราฟวงกลม ประกอบด้วย ชื่อกราฟ คำอธิบายข้อมูล
- 8) ภาษาที่ใช้ทดสอบส่วนประกอบต้องสนับสนุนดอตเน็ตเฟรมเวิร์ค ได้แก่ ภาษาซีชาร์ป ภาษาอวิซวลเบสิกดอตเน็ต
- 9) สามารถแสดงกราฟจากเอกสารเอสวีจี โดยใช้ อโดบีเอสวีจีวิวเวอร์รุ่น 3.0 เป็นอย่างน้อย

1.4 ขั้นตอนของการดำเนินการวิจัย

- 1) ศึกษาวิชาพลศึกษาอินเทอร์เน็ต
- 2) ศึกษาภาษาซีชาร์ป
- 3) ศึกษาเอสวีจีและโครงสร้างเอกสารดีทีดี
- 4) ศึกษาวิธีการเขียนโปรแกรมในการติดต่อกับระบบฐานข้อมูลเชิงสัมพันธ์
- 5) ศึกษาวิธีการเขียนโปรแกรมในการวาดกราฟ
- 6) ออกแบบและพัฒนาส่วนประกอบกราฟสำหรับสร้างแฟ้มข้อมูลเอสวีจี
- 7) ทดสอบส่วนประกอบกราฟสำหรับสร้างแฟ้มข้อมูลเอสวีจี
- 8) สรุปผลและจัดทำรูปเล่มวิทยานิพนธ์



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

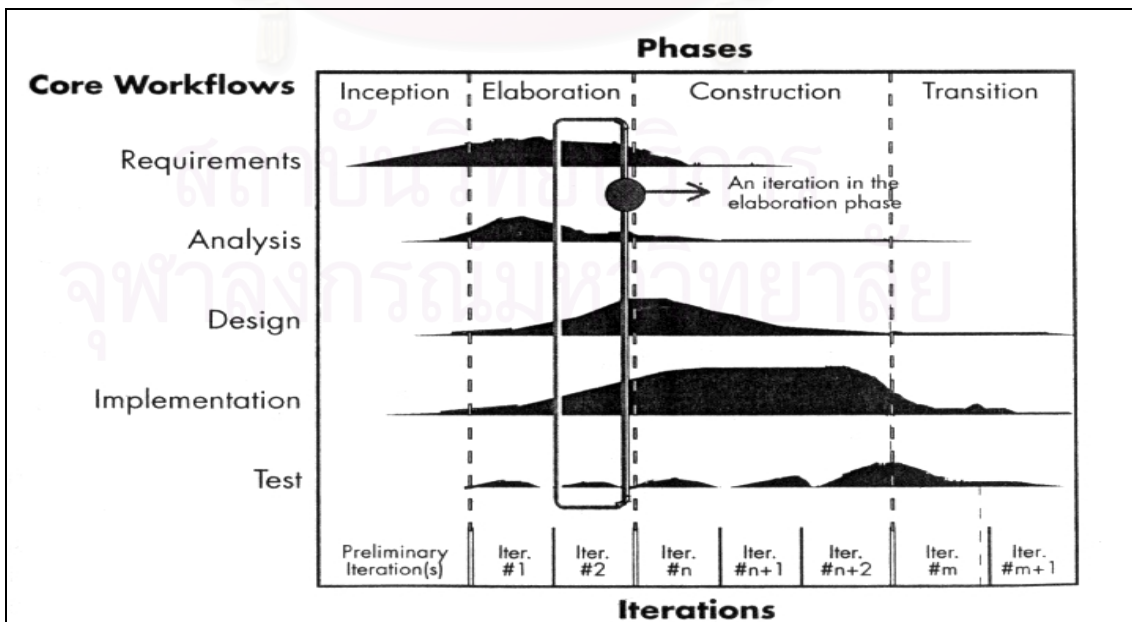
2.1 การพัฒนาซอฟต์แวร์แบบส่วนประกอบ

ส่วนประกอบ คือ หน่วยของการพัฒนาซอฟต์แวร์ที่พัฒนาขึ้นให้สามารถเรียกใช้งานได้อย่างอิสระ ซึ่งส่วนประกอบจะประกอบด้วยฟังก์ชันโดยสามารถเรียกใช้งานฟังก์ชันที่อยู่ในส่วนประกอบผ่านทางส่วนต่อประสาน (Interface)

การพัฒนาซอฟต์แวร์แบบส่วนประกอบเป็นวิธีการพัฒนาซอฟต์แวร์ที่ทำการสร้างฟังก์ชันการทำงานเป็นส่วนย่อยๆ และนำมาวมกันอยู่เป็นซอฟต์แวร์ขนาดใหญ่อีกทีหนึ่ง ทำให้ต้นทุนในการพัฒนาต่ำ และได้ปริมาณซอฟต์แวร์จำนวนมาก นอกจากนี้สามารถควบคุมคุณภาพ และลดความเสี่ยงเมื่อมีการเปลี่ยนแปลงเทคโนโลยี เช่น ส่วนประกอบเกี่ยวกับไปรษณีย์อิเล็กทรอนิกส์ (Electronic Mail) ประกอบด้วยส่วนประกอบที่ทำหน้าที่ส่งไปรษณีย์อิเล็กทรอนิกส์ และส่วนประกอบที่ทำหน้าที่รับไปรษณีย์อิเล็กทรอนิกส์ ซึ่งส่วนประกอบเหล่านี้สามารถนำกลับมาใช้ใหม่ได้ (Reusable Software Component) ด้วยโปรแกรมประยุกต์อื่น ๆ

2.1.1 วงจรการพัฒนาซอฟต์แวร์แบบส่วนประกอบ [1,2]

วงจรการพัฒนาซอฟต์แวร์แบบส่วนประกอบ สามารถแสดงได้ด้วยรูปที่ 2.1



รูปที่ 2.1 วงจรการพัฒนาซอฟต์แวร์แบบส่วนประกอบ [1]

จากรูปที่ 2.1 วงจรการพัฒนาซอฟต์แวร์แบบส่วนประกอบ จะมีความแตกต่างจากการพัฒนาซอฟต์แวร์แบบปกติ คือ เป็นการพัฒนาที่ใช้แนวคิดการวนซ้ำ (Iteration) หลายรอบและการพัฒนาแบบเพิ่มพูน (Incremental) จะประกอบไปด้วย 4 ระยะ ดังนี้

- 1) ระยะอินเซ็ปชัน (Inception Phase) เป็นระยะในการกำหนดปัญหา โครงสร้างของผลิตภัณฑ์ตลอดทั้งหน้าที่และความสามารถของผลิตภัณฑ์ เพื่อแบ่งส่วนของซอฟต์แวร์ออกเป็นส่วนๆ ศึกษาข้อจำกัดต่างๆของระบบ รวมทั้งศึกษาแนวทางการนำเทคโนโลยีต่างๆมาใช้งาน ศึกษาถึงความสัมพันธ์ที่มีต่อระบบอื่นๆ
- 2) ระยะอีแลบอเรชัน (Elaboration Phase) เป็นระยะการออกแบบสถาปัตยกรรมเบื้องต้นของซอฟต์แวร์ โดยศึกษาและวิเคราะห์จากขอบเขตของปัญหาและโครงสร้างของซอฟต์แวร์เบื้องต้นที่ได้ทำการศึกษาไว้จากระยะอินเซ็ปชัน
- 3) ระยะคอนสตรัคชัน (Construction Phase) เป็นระยะการพัฒนาต้นแบบของผลิตภัณฑ์ คือเป็นช่วงของการเขียนโปรแกรม ให้มีความสามารถทำงานได้ตามฟังก์ชันที่ถูกออกแบบมาจากระยะอีแลบอเรชัน
- 4) ระยะทรานสิชัน (Transition Phase) เป็นระยะทำการปรับปรุงต้นแบบที่ได้จากการพัฒนาในระยะคอนสตรัคชัน เพื่อนำมาปรับปรุงแก้ไขข้อผิดพลาดต่างๆ และออกแบบวิธีการติดตั้ง วิธีการบำรุงรักษาเพื่อส่งมอบโปรแกรม

ในแต่ละระยะ จะประกอบด้วยขบวนการที่คล้ายกับการพัฒนาระบบตั้งแต่การศึกษาความต้องการ (Requirements) วิเคราะห์ (Analysis) ออกแบบ (Design) พัฒนาโปรแกรม (Implementation) และทดสอบ (Test)

2.1.2 ประโยชน์ของการพัฒนาซอฟต์แวร์แบบส่วนประกอบ

การพัฒนาซอฟต์แวร์แบบส่วนประกอบมีประโยชน์ ดังนี้

- 1) ความเชื่อถือได้ ส่วนประกอบถูกใช้ในหลายๆ ระบบประยุกต์ ทำให้ถูกทดสอบการทำงานหลายครั้งจนน่าจะมีความถูกต้องสูง

- 2) ลดต้นทุน ส่วนประกอบถูกใช้ได้หลายๆ ครั้ง ดังนั้นต้นทุนในการพัฒนาโปรแกรมจะลดลง
- 3) ลดวงจรในการพัฒนาระบบประยุกต์ ระบบประยุกต์เกิดจากการนำส่วนประกอบมารวมกันทำให้ไม่เสียเวลาในการพัฒนาโปรแกรม
- 4) เปลี่ยนแปลงง่าย เมื่อมีการเปลี่ยนแปลงส่วนประกอบสามารถทำได้ง่าย เนื่องจากส่วนประกอบแต่ละส่วนประกอบมีความเป็นอิสระต่อกัน

2.1.3 เทคโนโลยีของการพัฒนาซอฟต์แวร์แบบส่วนประกอบ

ในเชิงพาณิชย์สามารถจำแนกหมวดหมู่ของส่วนประกอบ ได้ดังนี้

2.1.3.1 ส่วนประกอบซอฟต์แวร์แบบคอม, ดีคอม, คอมพลัส, ดอทเน็ต

- ส่วนประกอบซอฟต์แวร์แบบคอม (COM: Component Object Model) [3] คือ ส่วนประกอบซอฟต์แวร์ที่พัฒนาโดยใช้แบบจำลองของโปรแกรมแบบส่วนประกอบ โดยขึ้นอยู่กับแพลตฟอร์มวินโดวส์ (Windows) และ วินโดวส์เอ็นที (Windows NT) ทำหน้าที่สร้างซอฟต์แวร์ให้ทำงานร่วมกันได้ ซึ่งเป็นเทคโนโลยีที่คิดค้นโดยบริษัทไมโครซอฟต์จำกัด
- ส่วนประกอบซอฟต์แวร์แบบดีคอม (DCOM: Distributed Component Object Model) [3] สนับสนุนการใช้ส่วนประกอบซอฟต์แวร์แบบคอมบนคอมพิวเตอร์ที่แตกต่างกัน เช่น ในเครือข่าย (LAN: Local Area Network) การใช้ส่วนประกอบซอฟต์แวร์แบบดีคอมทำให้โปรแกรมประยุกต์สามารถกระจายจากส่วนกลาง แล้วส่งคำสั่งเพื่อเรียกใช้ส่วนประกอบซอฟต์แวร์แบบคอมบนเซิร์ฟเวอร์
- ส่วนประกอบซอฟต์แวร์แบบคอมพลัส (COM+) [3] คือ เฟรมเวิร์คที่ออกแบบเพื่อสนับสนุนการพัฒนาโปรแกรมประยุกต์แบบกระจาย (Distributed Application) บนแพลตฟอร์มวินโดวส์ โดยเพิ่มความสามารถในการจัดการรายการ (Transaction) เมื่อมีระบบประยุกต์มาเรียกใช้ส่วนประกอบซอฟต์แวร์แบบคอมในหลายๆด้าน ทำให้ทรัพยากรในระบบไม่ลดลง

- ส่วนประกอบซอฟต์แวร์แบบดอทเน็ต (.NET) [4] มีลักษณะเหมือนส่วนประกอบซอฟต์แวร์แบบคอม แต่มีคุณสมบัติเพิ่มเติม คือ สามารถสร้างส่วนประกอบซอฟต์แวร์แบบดอทเน็ต โดยใช้ภาษาใดก็ได้ที่ซีแอลอาร์ (CLR: Common Language Runtime) รองรับ

2.1.3.2 CORBA (Common Object Request Broker Architecture) [5]

เป็นเทคโนโลยีที่นำเสนอโดยโอเอ็มจี (OMG: Object Management Group) โดยส่วนประกอบซอฟต์แวร์นั้นถูกสร้างด้วยภาษาใดก็ได้ และสามารถสื่อสารข้ามแพลตฟอร์มได้โดยผ่านโพรโตคอลไอไอโอพี (IIOP: Internet Interoperability Protocol)

2.1.3.3 จาวาบีน (Java Bean) และ เอนเตอร์ไพร์สจาวาบีน (Enterprise Java Bean) [6]

การพัฒนาส่วนประกอบซอฟต์แวร์ของบริษัทชั้นไม่โครซิสเต็มจะแบ่งเป็น 2 อย่าง คือ จาวาบีน ใช้สำหรับพัฒนาส่วนประกอบซอฟต์แวร์สำหรับฝั่งไคลเอนต์ (Client Side) และเอนเตอร์ไพร์สจาวาบีนสำหรับพัฒนาส่วนประกอบซอฟต์แวร์ในฝั่งเซิร์ฟเวอร์ (Server Side) ส่วนประกอบซอฟต์แวร์ที่สร้างจากจาวาบีน สามารถใช้ได้ในทุกๆ แพลตฟอร์ม

2.2 เอสวีจี

เอสวีจี เป็นภาษาที่ถูกพัฒนาโดยดับเบิลยูทีซี (W3C: World Wide Web Consortium) สำหรับแสดงรูปภาพแบบเวกเตอร์ (Vector Graphic) และภาพเคลื่อนไหว (Animation) โดยนำรูปแบบเอ็กซ์เอ็มแอล (XML: Extensible Markup Language) มาใช้ประโยชน์ และถือเป็นมาตรฐานเปิดแบบใหม่สำหรับรูปภาพแบบเวกเตอร์

2.2.1 ข้อดีของเอสวีจี

เอสวีจีมีข้อดีดังนี้

- 1) เปลี่ยนแปลงง่าย เนื่องจากแฟ้มเอสวีจีเป็นภาษาที่อยู่ในรูปข้อความ (Text) ดังนั้นสามารถเพิ่มหรือเปลี่ยนแปลงโดยใช้โปรแกรมเท็กซ์อีดิเตอร์ (Text

Editor) โครงสร้างของเอสวีจีจะเป็นป้ายอิลิเมนต์ (Element Tags) ซึ่งจะเป็นคำภาษาอังกฤษที่สามารถเข้าใจได้ง่าย เช่น <circle/> สำหรับวาดรูปวงกลม

- 2) สามารถค้นหาเนื้อหาในแฟ้มเอสวีจี ส่วนใหญ่เนื้อหาในรูปแบบกราฟิกจะอยู่ในรูปแบบไบนารี แต่สำหรับแฟ้มเอสวีจีจะเหมือนกับแฟ้มเอชทีเอ็มแอล (HTML: Hyper Text Markup Language) ซึ่งมีลักษณะเป็นข้อความจึงสามารถค้นหาคำที่ต้องการได้
- 3) เอสวีจีเป็นมาตรฐานเปิด สามารถเพิ่มฟังก์ชันลงในเอกสารได้ เช่น สามารถเพิ่มฟังก์ชันการหมุนภาพไว้ในเอกสารเอสวีจี
- 4) การจัดการกับข้อมูล (Data Handling) สามารถแก้ไขตำแหน่งของข้อมูล ทำให้สร้างไดนามิกกราฟิก (Dynamic Graphics) ได้
- 5) ควบคุมรูปแบบอักษร (Font) ได้ สามารถใช้รูปแบบอักษรจริงๆกับเอสวีจีได้
- 6) วิซิวิก (WYSIWYG: What You See Is What You Get) สามารถออกแบบกราฟิกได้โดยตรงเช่นกำหนดรูปภาพอย่างไร เมื่อนำไปแสดงผลบนเบราว์เซอร์จะได้ภาพตามที่ออกแบบไว้
- 7) โปรแกรมตรวจแก้จุดบกพร่อง (Debugger) เมื่อมีข้อผิดพลาดในแฟ้มเอสวีจี จะแสดงจุดที่ผิดพลาดนั้น

2.2.2 ความแตกต่างระหว่างรูปแบบเอสวีจีกับรูปแบบบิตแมพหรือรูปแบบจีพ

แฟ้มรูปภาพแบบเอสวีจีมีความแตกต่างกับแฟ้มรูปภาพแบบบิตแมพหรือแบบจีพที่นิยมใช้กันในปัจจุบันดังนี้คือ

- 1) สามารถดึงภาพ โดยที่ภาพยังคงรักษารายละเอียดและความคมชัดของภาพได้
- 2) ไม่มีการจำกัดรูปแบบอักษร สามารถแก้ไขและค้นหาในรูปแบบอักษรได้
- 3) แฟ้มข้อมูลมีขนาดเล็ก แฟ้มข้อมูลในรูปแบบเอสวีจีจะมีขนาดเล็กกว่าเมื่อเทียบกับรูปแบบเว็บกราฟิกอื่นๆ ทำให้การส่งแฟ้มเอสวีจีสามารถทำได้เร็วกว่าเมื่อนำมาแสดงผลบนเว็บ

- 4) การแสดงผลเป็นอิสระ ดังนั้นรูปภาพที่เกิดจากแฟ้มเอสวีจีจึงแสดงผลได้อย่างชัดเจนบนหน้าจอ และสามารถพิมพ์รูปภาพออกทางเครื่องพิมพ์ได้อย่างมีประสิทธิภาพ
- 5) การควบคุมสี เอสวีจีแสดงได้ 16 ล้านสี และสนับสนุนไอซีซี (ICC) เอสอาร์จีบี (sRGB) แกรเดรียนท์
- 6) สามารถตอบสนองในการเน้นข้อความ การแสดงทูลทิป และการเคลื่อนไหว

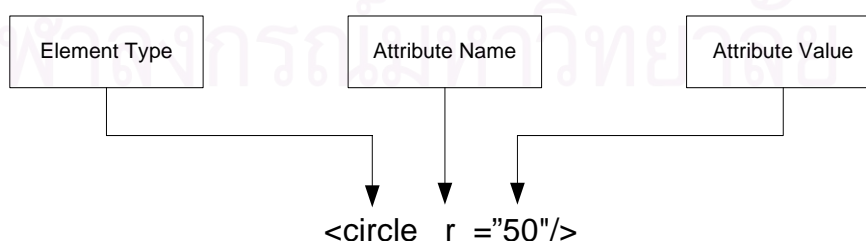
2.2.3 โครงสร้างพื้นฐานของเอสวีจี

2.2.3.1 วากยสัมพันธ์ (Syntax)

วากยสัมพันธ์ของเอสวีจีเป็นไปตามหลักการของภาษาเอ็กซ์เอ็มแอล ซึ่งมีลักษณะดังนี้

- 1) แท็ก (Tag) ของเอสวีจี ตัวอักษรตัวใหญ่และเล็กมีผลแตกต่างกัน (Case Sensitive) เช่น <aa> กับ <aA> หมายถึง มีประเภทอีลีเมนต์ (Element Type) ที่แตกต่างกัน
- 2) เมื่อมีแท็กเปิด ต้องมีแท็กปิดเสมอ
- 3) แอตทริบิวต์ (Attribute) ประกอบด้วย ชื่อแอตทริบิวต์ (Attribute Name) และค่าของแอตทริบิวต์ (Attribute Value) โดยที่ค่าของแอตทริบิวต์ต้องอยู่ในเครื่องหมายคำพูด

วากยสัมพันธ์ของเอสวีจีแสดงได้ดังรูปที่ 2.2



รูปที่ 2.2 อีลีเมนต์ ชื่อแอตทริบิวต์ และค่าแอตทริบิวต์ [7]

2.2.3.2 ดีทีดี (DTD: Document Type Definition)

คือโครงสร้างและไวยากรณ์ของเอสวีจี ซึ่งเป็นเอกสารที่กำหนดรายละเอียดของอีลีเมนต์

2.2.3.3 โครงสร้างของแฟ้มเอสวีจี

ในทุกๆ แฟ้มเอสวีจีเนื้อหาทั้งหมดจะอยู่ในแท็กระบุเอสวีจี: `<svg></svg>` ซึ่งประกอบด้วย 6 ประเภท คือ

- 1) ข้อมูลเอกสาร (Document data) เป็นส่วนที่กำหนดเอกสารอ้างอิงดีทีดีของเอสวีจี (SVG DTD) รวมทั้งการกำหนดรูปแบบสีสันด้วยสไตล์ชีต (Stylesheets)
- 2) บรรณนิทัศน์ (Annotations) เป็นส่วนที่กำหนดคำอธิบายเอกสารเอสวีจี
- 3) ข้อกำหนดของวัตถุที่แสดงผล (Reference Material) เป็นส่วนที่ใช้กำหนดรายละเอียดของวัตถุให้มีส่วนประกอบอย่างไร เช่น สีเหลี่ยมให้มีสีน้ำเงิน เส้นมีสีเหลือง เป็นต้น
- 4) กราฟฟิกคอนเทนต์ (Graphic Content) เป็นส่วนที่ใช้แสดงเนื้อหากราฟฟิก ได้แก่ สีเหลี่ยมและรูปภาพที่สร้างขึ้นจากแท็กต่าง ๆ
- 5) รูปแบบการแสดงผล (Manipulation) ใช้กำหนดการสร้างภาพเคลื่อนไหวต่างๆ
- 6) ข้อความ (Text) เป็นส่วนที่ใช้แสดงข้อความบนเอกสารเอสวีจี

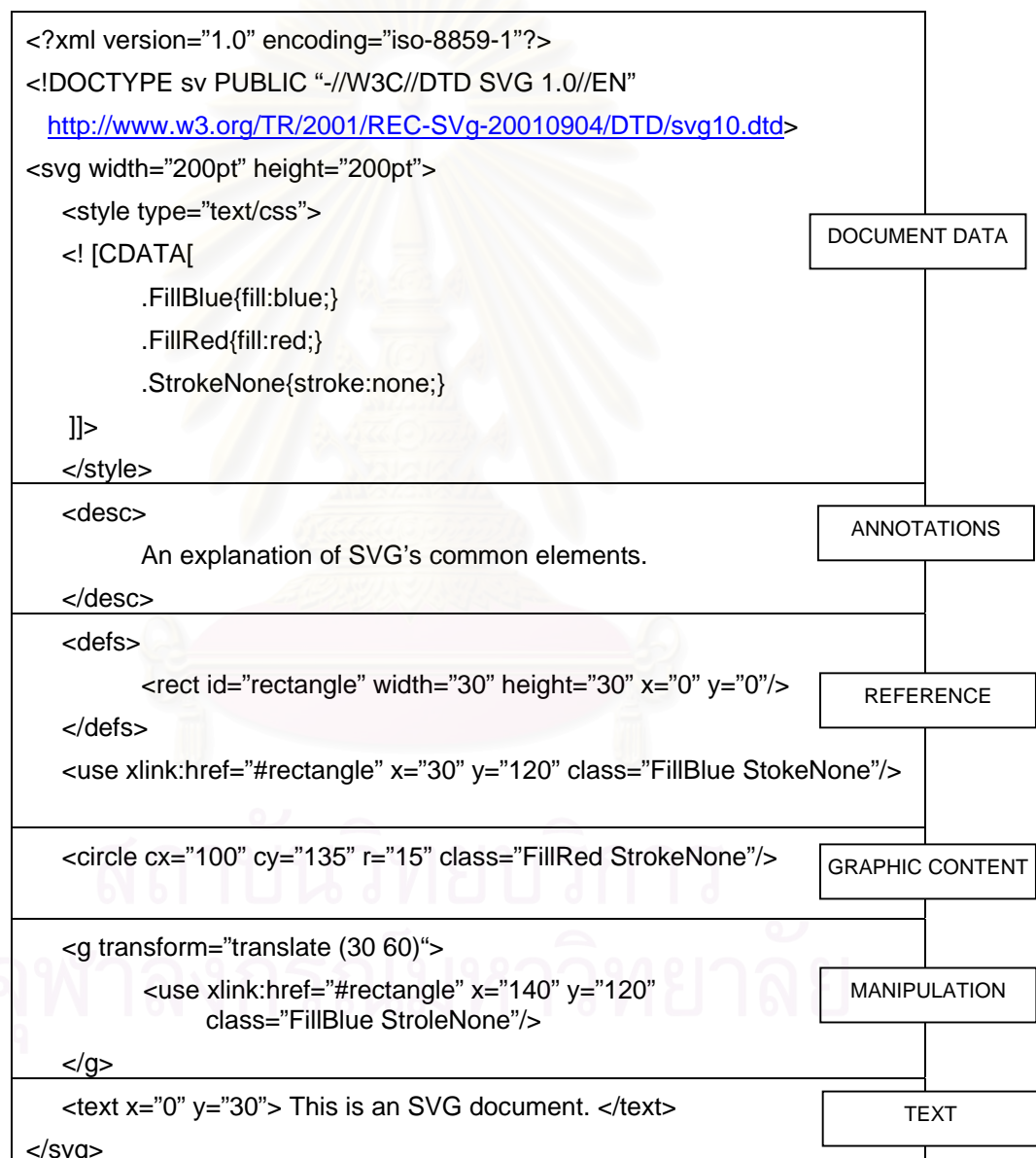
โครงสร้างของแฟ้มเอสวีจีจะแสดงได้ดังรูปที่ 2.3

2.2.4 การแสดงผลของแฟ้มเอสวีจี

การแสดงผลของแฟ้มเอสวีจี สามารถแสดงได้บนบราวเซอร์ที่เป็นบราวเซอร์สำหรับแสดงเอกซ์เอ็มแอล เช่น X-Smiles (<http://www.xsmiles.org/>) แต่สำหรับบราวเซอร์ที่เป็นที่นิยมอย่างไออี (IE : Internet Explorer) และเนสเคป (Netscape

Navigator) ต้องมีการติดตั้งตัวเอสวีจีวิวเวอร์ ซึ่งพัฒนาโดย อโดบี (Adobe) [8] เพื่อใช้ในการแสดงผล

นอกจากนี้ยังสามารถนำแฟ้มเอสวีจีไปไว้ในหน้าเว็บที่เอ็มแอล (HTML Page) ซึ่งจะใช้อีลีเมนต์เอ็มเบด (Embed Element) แสดงดังรูปที่ 2.4



รูปที่ 2.3 แผนภาพแสดงประเภทโครงสร้างของแฟ้มเอสวีจี [7]

```

<html>
  <head>
    <title>Embedding SVG</title>
  </head>
  <body bgcolor="white">
    <embed src="filepath/filename.svg"/>
  </body>
</html>

```

รูปที่ 2.4 การเรียกใช้งานแฟ้มเอสวีจีด้วยแท็กเอมเบด [7]

2.2.5 รูปทรงพื้นฐาน

การแสดงผลเอสวีจี มีรูปทรงพื้นฐาน ดังต่อไปนี้

- 1) สี่เหลี่ยม (Rectangles) เป็นรูปทรงที่ใช้ส่วนใหญ่ โดยคำสั่งที่ใช้สำหรับวาดรูปสี่เหลี่ยม คือ แท็กชื่อ rect ซึ่งประกอบด้วยแอตทริบิวต์ดังนี้
 - X และ Y: เป็นแอตทริบิวต์ที่จะแสดงถึงจุดเริ่มต้นของมุมสี่เหลี่ยม
 - Width: บอกความยาวทางแนวนอน คือ ขนาดความยาวไปทางขวาจากจุด X
 - Height: บอกความยาวทางแนวตั้ง คือ ขนาดความยาวไปทางใต้ของจุด Y

ตัวอย่างการใช้แท็กสี่เหลี่ยม แสดงดังในรูปที่ 2.5

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE sv PUBLIC "-//W3C//DTD SVG 1.0//EN"
  http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd>
<svg width="500" height="300">
  <rect id="Outline" x="10" y="10" width="480" height="280"/>
</svg>

```

รูปที่ 2.5 การใช้แท็กสี่เหลี่ยม [7]

- 2) เส้น (Lines) คำสั่งสำหรับวาดเส้น คือ แท็กชื่อ line ซึ่งประกอบด้วยแอตทริบิวต์ดังนี้

- X1 และ Y1: เป็นจุดเริ่มต้นของเส้น
- X2 และ Y2: เป็นจุดสิ้นสุดของเส้น

ตัวอย่างการใช้แท็กเส้น แสดงดังในรูปที่ 2.6

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE sv PUBLIC "-//W3C//DTD SVG 1.0//EN"
  http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd>
<svg width="500" height="300">
  <line x1="105" y1="160" x2="410" y2="100"/>
</svg>
```

รูปที่ 2.6 การใช้แท็กเส้น [7]

- 3) พาท (Path) คือ การเขียนเส้นต่อเนื่อง คำสั่งสำหรับวาดพาทคือแท็กชื่อ path ซึ่งมีแอตทริบิวต์คือ d ใช้สำหรับเก็บค่าคำสั่งต่างๆได้แก่ M L C A Z ซึ่งอธิบายในตารางที่ 2.1 ตัวอย่างการใช้แท็กพาท แสดงดังในรูปที่ 2.7

ตารางที่ 2.1 แสดงคำสั่งที่ใช้สำหรับการสร้างพาท

คำสั่ง	ความหมาย	คำอธิบาย
M	Moveto	จุดเริ่มต้นพาท เช่น <path d="M x1 y1.... "/>
L	Lineto	คำสั่ง L สำหรับวาดเส้นตรงจากจุดเริ่มต้น เช่น <path d="M x1,y1 Lx2,y2"/>
C	Curveto	ใช้วาดรูปเส้นโค้ง เช่น <path d="M x1y1 C x2,y2 x3,y3 x4,y4"/>
A	Elliptical Arc	ใช้วาด ส่วนโค้ง เช่น <path d="M xy-radius x-axis-rotation large-arc-flag sweep-flag xy-coordinate of ending point
Z	Closepath	จุดสิ้นสุดพาท เช่น "<path d=M x1y1 L x2,y2 x3,y3 Z"/>

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE sv PUBLIC "-//W3C//DTD SVG 1.0//EN"
  http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd>
<svg width="500" height="300">
  <path d="M16,0
    C 16,0,0,24,0,33
    C 0,42,7,49,15,49
    C 25,49,32,42,32,33
    C 32,24,16,0,16,0" />
</svg>

```

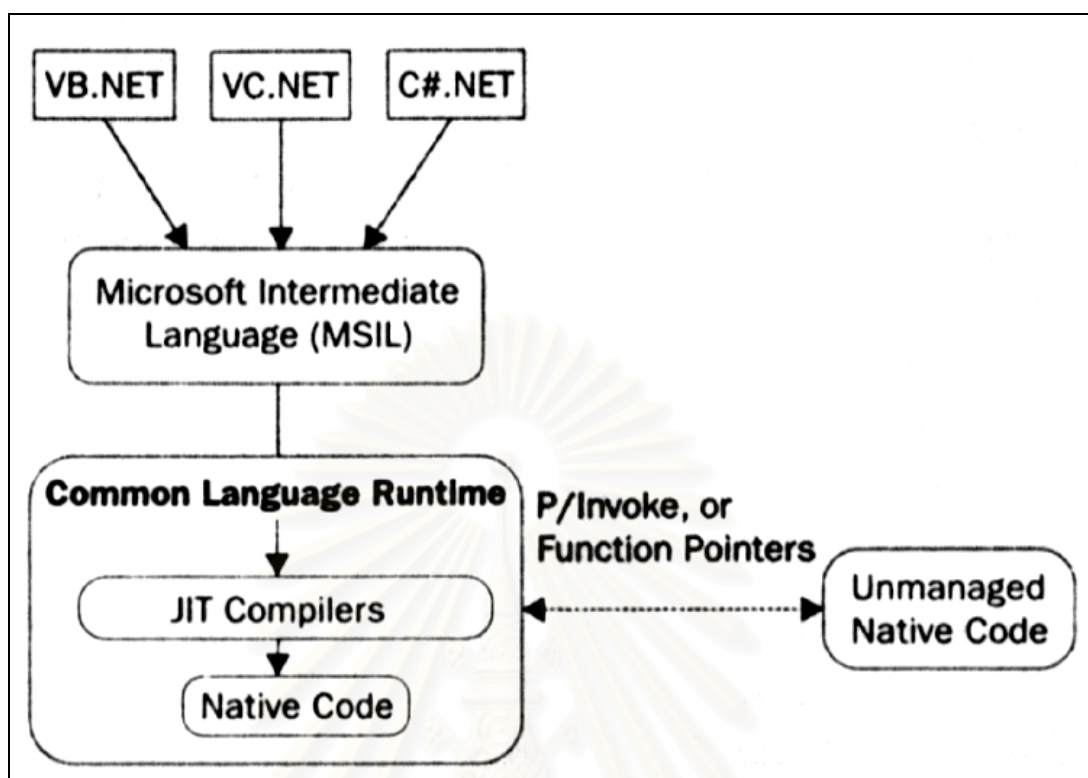
รูปที่ 2.7 การใช้แท็กพาท [7]

2.3 ซีแอลอาร์ (CLR: Common Language Runtime)

ซีแอลอาร์ทำหน้าที่เป็นตัวกลางในการจัดเตรียมบริการและทรัพยากรสำหรับรองรับการประมวลผลและการทำงานของโปรแกรมประยุกต์ ที่ทำงานอยู่บนสภาพแวดล้อมดอทเน็ต เช่น การจัดการหน่วยความจำที่แอปพลิเคชันใช้งาน การรักษาความปลอดภัยในการรันโค้ดโปรแกรมต่าง ๆ และการติดต่อระหว่างโปรแกรม อีกทั้งยังก่อให้เกิดสภาพแวดล้อมการทำงานที่ไม่ยึดติดกับระบบปฏิบัติการ สามารถใช้ได้กับภาษาโปรแกรมหลายๆ ภาษา ทำให้ส่วนประกอบต่างๆ เรียกใช้งานกันได้อย่างสมบูรณ์

2.3.1 การทำงานของซีแอลอาร์

ซีแอลอาร์ ทำหน้าที่จัดการการประมวลผลคำสั่งของดอทเน็ต โดยคำสั่งเหล่านั้น จะถูกทำเป็นคำสั่งในภาษาเอ็มเอสไอแอล (MSIL: Microsoft Intermediate Language) ซึ่งไม่ขึ้นกับฮาร์ดแวร์ และไม่ขึ้นกับระบบปฏิบัติการหลังจากนั้นเอ็มเอสไอแอลจะถูกเปลี่ยนเป็นภาษาเครื่อง (Machine Code) ในซีแอลอาร์ ซึ่งมีพื้นฐานการทำงานเหมือนเจวีเอ็ม (JVM: Java Virtual Machine) บนแพลตฟอร์มจาวา โดยที่ซีแอลอาร์จะใช้คอมไพเลอร์เจไอที (JIT: Just in Time) ซึ่งจะแปลโปรแกรม (Compile) เฉพาะส่วนที่ใช้งานก่อน จากนั้นเมื่อนำไปประมวลผลบนระบบปฏิบัติการใดจะทำการแปลโปรแกรมให้เข้ากับระบบปฏิบัติการนั้นๆ แสดงดังรูปที่ 2.8



รูปที่ 2.8 หลักการทำงานของซีแอลอาร์ [9]

2.4 งานวิจัยที่เกี่ยวข้อง

2.4.1 การใช้เครื่องมือพาวเวอร์บิวเดอร์ (PowerBuilder) พัฒนาส่วนประกอบกราฟบนอีเอเซอร์ฟเวอร์ (EAServer) แอปพลิเคชันเซิร์ฟเวอร์ในรูปแบบเอสวีจี [10]

เครื่องมือพาวเวอร์บิวเดอร์สามารถพัฒนาส่วนประกอบกราฟที่แสดงผลในรูปแบบเอสวีจี โดยส่วนประกอบจะอยู่บนอีเอเซอร์ฟเวอร์ (EAServer) ซึ่งเป็นแอปพลิเคชันเซิร์ฟเวอร์ คอมโพเนนต์จะทำการติดต่อและดึงข้อมูลจากฐานข้อมูลมาแสดงกราฟแบบซีวีจี หรือแบบวงกลมบนบราวเซอร์ผ่านทางเอสวีจีวีเวอร์ แต่ส่วนประกอบกราฟที่สร้างขึ้นไม่สามารถนำไปใช้กับภาษาอื่นได้

2.4.2 บาดิกเอสวีจีทูลคิต (Batik SVG Toolkit) [11]

บาดิกเอสวีจีทูลคิต เป็นไลบรารีที่พัฒนาจากภาษาจาวาซึ่งเกิดจากโครงการอปาเซ่ เอ็กซ์เอ็มแอล เพื่อสนับสนุนจาวาแอปพลิเคชัน และจาวาแอปเพล็ต ให้สามารถเรียกฟังก์ชันเพื่อนำเอกสารรูปเอสวีจีแสดงในรูปแบบกราฟฟิบบนบราวเซอร์ได้ รวมทั้งมีฟังก์ชัน

เพื่อแปลงเอกสารเอกสารวีซีดีไปเป็นรูปภาพในรูปแบบจีฟ (GIF : Graphics Interchange Format) เจเพ็ท ก JPEG (Joint Photographic Experts Group) และ ทีฟพี (TIFF : Tagged Image File Format)



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

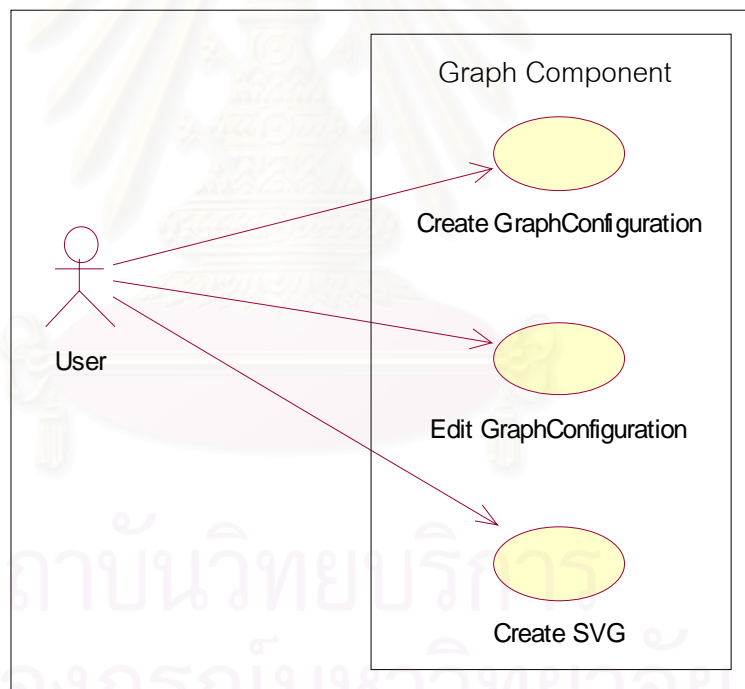
บทที่ 3

การออกแบบและพัฒนาส่วนประกอบกราฟสำหรับสร้างเพิ่มข้อมูลเอสวีจี

ในบทนี้จะกล่าวถึงการออกแบบและพัฒนาส่วนประกอบกราฟสำหรับสร้างเพิ่มข้อมูลเอสวีจีซึ่งสามารถอธิบายโครงสร้างต่างๆโดยใช้ภาษายูเอ็มแอลได้ดังต่อไปนี้

3.1 แผนภาพยูสเคส (Use Case Diagram)

จากการพิจารณาลักษณะการใช้งานของผู้ใช้งานสามารถแบ่งการใช้งานออกได้เป็น 3 ส่วน คือ การตั้งค่าข้อกำหนดของกราฟ (Create GraphConfiguration) การแก้ไขข้อกำหนดกราฟ (Edit GraphConfiguration) และการสร้างเพิ่มข้อมูลเอสวีจีของกราฟ (Create SVG) ดังแสดงในรูปที่ 3.1 ซึ่งสามารถอธิบายได้ดังนี้



รูปที่ 3.1 แผนภาพยูสเคสของส่วนประกอบกราฟสำหรับสร้างเพิ่มข้อมูลเอสวีจี

3.1.1 ยูสเคส Create GraphConfiguration

ยูสเคสนี้อธิบายถึงการใช้งานกรณีที่ผู้ใช้งานต้องการสร้างเพิ่มข้อกำหนดของกราฟโดยจะรับข้อกำหนดต่างๆจากผู้ใช้งาน เช่น ประเภทของกราฟ เป็นต้น และบันทึก

ข้อกำหนดนั้นลงในแฟ้มข้อกำหนดซึ่งอยู่ในรูปแฟ้มเอ็กซ์เอ็มแอลเพื่อนำไปใช้ในการสร้างกราฟ แสดงรายละเอียดได้ดังตารางที่ 3.1

ตารางที่ 3.1 รายละเอียดยูสเคส Create GraphConfiguration

Use Case:	Create GraphConfiguration	
Actor:	User	
Purpose:	เพื่อดีงค่าข้อกำหนดของกราฟ	
Overview:	รับข้อมูลข้อกำหนดเกี่ยวกับรูปแบบของกราฟที่ต้องการ (Configuration) จากผู้ใช้งาน ได้แก่ ตำแหน่งเริ่มต้นแกนตั้งและแกนนอน ขนาดพื้นที่แสดงกราฟ ชื่อกราฟ คำอธิบายแกนนอน คำอธิบายแกนตั้ง ประเภทของกราฟ ชื่อฐานข้อมูล ข้อความสั่งเอสคิวแอล (SQL Statement) ตำแหน่งแฟ้มข้อมูลเอสวิจี ชื่อแฟ้มข้อมูลเอสวิจี และบันทึกข้อกำหนดนั้นเป็นแฟ้มข้อกำหนดซึ่งอยู่ในรูปแบบเอ็กซ์เอ็มแอล	
Typical Course of Event:	Actor Action	System Response
	1. ระบบเริ่มต้นเมื่อผู้ใช้งานต้องการตั้งค่าข้อกำหนดของกราฟ เรียกใช้ฟังก์ชันการทำงาน	2. ระบบแสดงค่าเบื้องต้น (Default) ของประเภทกราฟ
	3. ผู้ใช้งานใส่ข้อมูลข้อกำหนดของกราฟที่ต้องการ	4. ระบบตรวจสอบความถูกต้องข้อมูลดังนี้ <ul style="list-style-type: none"> ▪ พื้นที่แสดงกราฟต้องมากกว่าศูนย์ ▪ ชนิดของกราฟต้องมีค่า ▪ ฐานข้อมูลต้องมีค่า ▪ ข้อความสั่งเอสคิวแอลถูกต้องตามหลักไวยากรณ์
		5. ระบบจะบันทึกข้อกำหนดลงแฟ้มข้อกำหนด โดยมีชื่อตามที่ใช้กำหนดไว้

ตารางที่ 3.1 รายละเอียดยูสเคส Create GraphConfiguration (ต่อ)

Alternative Courses:	4a) ถ้าข้อมูลที่ผู้ใช้งานใส่ไม่ถูกต้อง : ระบบจะทำการแจ้งเป็นข้อผิดพลาด (Error) 4b) ถ้าไม่สามารถบันทึกเพิ่มข้อกำหนดได้ : ระบบจะทำการแจ้งเป็นข้อผิดพลาด
Relate Use Case:	-
Operations:	เรียกใช้เมทอด CreateConfiguration

3.1.2 ยูสเคส Edit GraphConfiguration

ยูสเคสนี้อธิบายถึงการใช้งานกรณีที่ผู้ใช้งานต้องการแก้ไขเพิ่มข้อกำหนดของกราฟที่สร้างไว้ก่อนหน้านี้โดยจะอ่านเพิ่มข้อกำหนดที่ต้องการแก้ไข จากนั้นรับข้อกำหนดต่างๆที่ต้องการแก้ไขจากผู้ใช้งานและบันทึกข้อกำหนดนั้นลงในเพิ่มข้อกำหนดเดิม แสดงรายละเอียดได้ดังตารางที่ 3.2

ตารางที่ 3.2 รายละเอียดยูสเคส Edit GraphConfiguration

Use Case:	Edit GraphConfiguration					
Actor:	User					
Purpose:	เพื่อปรับปรุงค่าข้อกำหนดของกราฟ					
Overview:	รับข้อมูลข้อกำหนดที่ต้องการแก้ไขเกี่ยวกับรูปแบบของกราฟจากผู้ใช้งาน ได้แก่ ตำแหน่งเริ่มต้นแกนตั้งและแกนนอน ขนาดพื้นที่แสดงกราฟ ชื่อกราฟ คำอธิบาย แกนนอน คำอธิบายแกนตั้ง ประเภทของกราฟ ชื่อฐานข้อมูล ข้อความสังเขป อัลเอล ตำแหน่งเพิ่มข้อมูลเอสวีจี ชื่อเพิ่มข้อมูลเอสวีจี และบันทึกข้อกำหนดนั้นลงเพิ่มข้อกำหนดเดิมซึ่งอยู่ในรูปแบบเอ็กซ์เอ็มแอล					
Typical Course of Event:	<table border="1"> <thead> <tr> <th>Actor Action</th> <th>System Response</th> </tr> </thead> <tbody> <tr> <td>1. ระบบเริ่มต้นเมื่อผู้ใช้งานต้องการแก้ไขการตั้งค่าข้อกำหนดของกราฟ โดยผู้ใช้งานระบุเพิ่มข้อกำหนดที่ต้องการแก้ไข เรียกใช้ฟังก์ชันการทำงาน</td> <td>2. ระบบทำการอ่านเพิ่มข้อกำหนดตามที่ผู้ใช้งานระบุชื่อเพิ่มข้อกำหนดขึ้นมาแสดง</td> </tr> </tbody> </table>	Actor Action	System Response	1. ระบบเริ่มต้นเมื่อผู้ใช้งานต้องการแก้ไขการตั้งค่าข้อกำหนดของกราฟ โดยผู้ใช้งานระบุเพิ่มข้อกำหนดที่ต้องการแก้ไข เรียกใช้ฟังก์ชันการทำงาน	2. ระบบทำการอ่านเพิ่มข้อกำหนดตามที่ผู้ใช้งานระบุชื่อเพิ่มข้อกำหนดขึ้นมาแสดง	
Actor Action	System Response					
1. ระบบเริ่มต้นเมื่อผู้ใช้งานต้องการแก้ไขการตั้งค่าข้อกำหนดของกราฟ โดยผู้ใช้งานระบุเพิ่มข้อกำหนดที่ต้องการแก้ไข เรียกใช้ฟังก์ชันการทำงาน	2. ระบบทำการอ่านเพิ่มข้อกำหนดตามที่ผู้ใช้งานระบุชื่อเพิ่มข้อกำหนดขึ้นมาแสดง					

ตารางที่ 3.2 รายละเอียดยูสเคส Edit GraphConfiguration (ต่อ)

Typical Course of Event:	Actor Action	System Response
	3. ผู้ใช้งานใส่ข้อมูลข้อกำหนดที่ต้องการแก้ไข	4. ระบบตรวจสอบความถูกต้องข้อมูลดังนี้ <ul style="list-style-type: none"> ▪ พื้นที่แสดงกราฟต้องมากกว่าศูนย์ ▪ ชนิดของกราฟต้องมีค่า ▪ รฐานข้อมูลต้องมีค่า ▪ ข้อความสังเขตคิวแอลถูกต้องตามหลักไวยากรณ์
		5. ระบบจะบันทึกข้อกำหนดลงเพิ่มข้อกำหนด ที่ผู้ใช้งานระบุชื่อไว้
	4a) ถ้าข้อมูลที่ผู้ใช้งานใส่ไม่ถูกต้อง : ระบบจะทำการแจ้งเป็นข้อผิดพลาด 4b) ถ้าไม่สามารถบันทึกเพิ่มข้อกำหนดได้ : ระบบจะทำการแจ้งเป็นข้อผิดพลาด	6. ระบบจะบันทึกข้อกำหนดลงเพิ่มข้อกำหนด ที่ผู้ใช้งานระบุชื่อไว้
Alternative Courses:	-	
Relate Use Case:	เรียกใช้เมทอด GetConfiguration	
Operations:		

3.1.3 ยูสเคส Create SVG

ยูสเคสนี้อธิบายถึงการที่ผู้ใช้งานกรณที่ผู้ใช้งานต้องการสร้างกราฟจากเพิ่มข้อกำหนดที่ตั้งไว้ โดยจะอ่านข้อกำหนดจากเพิ่มข้อกำหนดและดึงข้อมูลจากฐานข้อมูลเพื่อนำมาสร้างกราฟและบันทึกไว้ในรูปแบบเพิ่มเอสวีจี แสดงรายละเอียดได้ดังตารางที่ 3.3

ตารางที่ 3.3 รายละเอียดยูสเคส Create SVG

Use Case:	Create SVG	
Actor:	User	
Purpose:	เพื่อสร้างเพิ่มข้อมูลเอสวิจีของกราฟตามข้อกำหนดที่ตั้งไว้	
Overview:	รับคำสั่งสร้างกราฟจากผู้ใช้งานตามเพิ่มข้อกำหนด ระบบจะทำการดึงข้อมูลจากฐานข้อมูลเอสวิจีแอสเคิร์ฟเวอร์แล้วนำมาสร้างเพิ่มข้อมูลเอสวิจีของกราฟตามข้อกำหนด	
Typical Course of Event:	<p>Actor Action</p> <p>1. ระบบเริ่มต้นเมื่อผู้ใช้งานต้องการสร้างเพิ่มข้อมูลเอสวิจีของกราฟเรียกใช้การทำงาน</p>	<p>System Response</p> <p>2. รับชื่อเพิ่มข้อมูลข้อกำหนด แล้วระบบทำการอ่านข้อกำหนดในเพิ่มข้อมูลนั้นเพื่อทำงานดังต่อไปนี้</p> <ul style="list-style-type: none"> ▪ ดึงข้อมูลจากฐานข้อมูลเอสวิจีแอสเคิร์ฟเวอร์ ▪ นำมาสร้างกราฟตามประเภทที่กำหนด โดยคำนวณตามวิธีการที่กำหนดไว้ของกราฟแต่ละประเภท (Algorithm) ▪ ทำการบันทึกกราฟเป็นเพิ่มข้อมูลเอสวิจี ▪ แสดงรูปกราฟบนจอภาพ
Alternative Courses:	<p>2a) ถ้าไม่สามารถเปิดเพิ่มข้อมูลข้อกำหนดได้ : ระบบทำการแจ้งเป็นข้อผิดพลาด</p> <p>2b) ถ้าไม่สามารถดึงข้อมูลจากฐานข้อมูลเอสวิจีแอสเคิร์ฟเวอร์ : ระบบทำการแจ้งเป็นข้อผิดพลาด</p> <p>2c) ถ้าไม่สามารถบันทึกกราฟเป็นเพิ่มข้อมูลเอสวิจีได้ : ระบบทำการแจ้งเป็นข้อผิดพลาด</p>	
Relate Use Case:	-	
Operations:	เรียกใช้เมทอด CreateSVG	

3.2 การออกแบบคลาส

ในงานวิจัยนี้ผู้วิจัยได้ออกแบบคลาสโดยแบ่งเป็น 3 คลาสหลัก ได้แก่ คลาส GraphCtrl คลาส GraphConfiguration และคลาส Graph ซึ่งมีคลาสลูกอีก 3 คลาส ได้แก่ คลาส BarGraph คลาส LineGraph และคลาส PieGraph โดยคลาสต่างๆมีความสัมพันธ์และรายละเอียดคลาสดังต่อไปนี้

3.2.1 ความสัมพันธ์ระหว่างคลาส

ความสัมพันธ์ระหว่างคลาสต่างๆ สามารถแสดงได้ตามแผนภาพคลาส (Class Diagram) ดังแสดงดังรูปที่ 3.2

คลาส GraphCtrl เป็นคลาสควบคุมในการสร้างข้อกำหนด การแก้ไขข้อกำหนด และการสร้างกราฟในรูปแบบข้อมูลเอสวีจี โดยในการสร้างข้อกำหนดและการแก้ไขข้อกำหนดคลาส GrapCtrl จะเรียกใช้คลาส GraphConfiguration เพื่อสร้างเพิ่มข้อกำหนด และแก้ไขเพิ่มข้อกำหนดผ่านคลาส XMLTextWriter ซึ่งเป็นคลาสที่มีมาพร้อมกับ วิววลสตูดิโอเดสทอปเน็ต สำหรับในการสร้างกราฟคลาส GraphCtrl จะเรียกใช้คลาส GraphConfiguration เพื่ออ่านเพิ่มข้อกำหนดว่าต้องการสร้างกราฟประเภทใด จากนั้นเรียกใช้คลาส Graph ซึ่งจะนำข้อกำหนดต่างๆจากเพิ่มข้อกำหนดนั้นมาทำการคำนวณ และสร้างเพิ่มข้อมูลเอสวีจีของกราฟตามข้อกำหนดนั้นๆ ผ่านคลาส XMLTextWriter

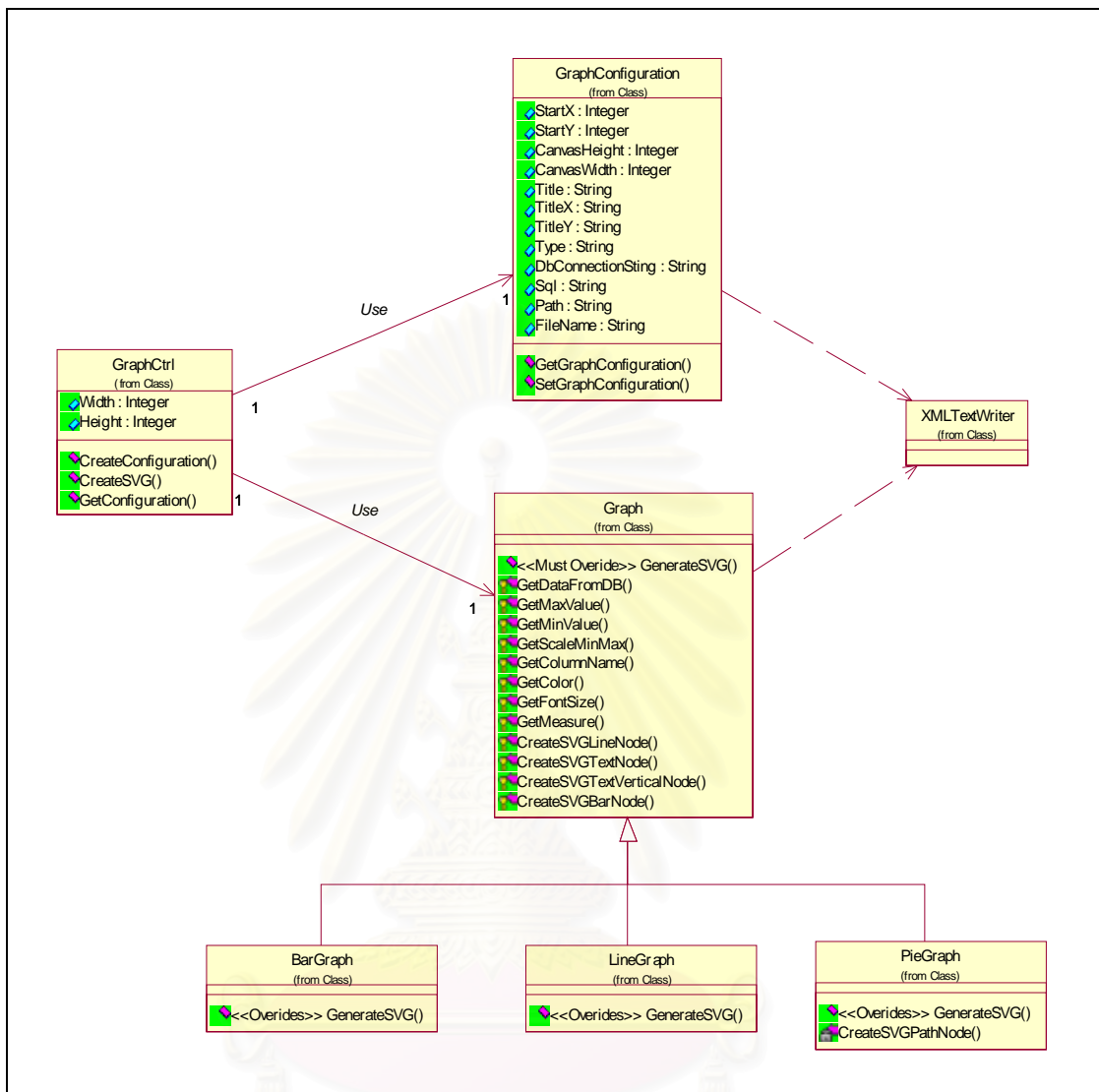
คลาส Graph จะการถ่ายทอดคุณสมบัติให้แก่คลาสลูกซึ่งได้แก่ คลาส BarGraph คลาส LineGraph และคลาส PieGraph ซึ่งมีหน้าที่ในการคำนวณและสร้างกราฟประเภทกราฟแท่ง กราฟเส้น และกราฟวงกลม ตามลำดับ

3.2.2 รายละเอียดคลาส

รายละเอียดของคลาสต่างๆ มีดังต่อไปนี้

3.2.2.1 คลาส GraphCtrl

คลาส GraphCtrl ประกอบด้วยคุณสมบัติและเมทอด โดยมีรายละเอียดดังแสดงในตารางที่ 3.4



รูปที่ 3.2 แผนภาพแสดงความสัมพันธ์ระหว่างคลาส

ตารางที่ 3.4 รายละเอียดของคลาส GraphCtrl

ชื่อคลาส	GraphCtrl
ประเภท	คลาสหลัก
คำอธิบาย	คลาสสำหรับควบคุมการเรียกใช้งานเมทอดของคลาสอื่น
ตัวแปร	<ul style="list-style-type: none"> ▪ Width ความกว้างของพื้นที่แสดงกราฟ ▪ Height ความสูงของพื้นที่แสดงกราฟ

ตารางที่ 3.4 รายละเอียดของคลาส GraphCtrl (ต่อ)

เม็ท็อด	<ul style="list-style-type: none"> ▪ CreateConfiguration() ทำหน้าที่ในการสร้างแฟ้มข้อกำหนดโดยจะไปเรียกเม็ท็อด SetGraphConfiguration() ของคลาส GraphConfiguration ▪ GetConfiguration() ทำหน้าที่ในการดึงรายละเอียดข้อกำหนดจากแฟ้มข้อกำหนดตามที่ระบุโดยจะเรียกเม็ท็อด GetGraphConfiguration() ของคลาส GraphConfiguration ▪ CreateSVG() ทำหน้าที่สร้างแฟ้มข้อมูลเอสวีจีตามประเภทกราฟที่กำหนดในแฟ้มข้อกำหนดโดยจะไปเรียกเม็ท็อด GenerateSVG() ของคลาส BarGraph คลาส LineGraph หรือคลาส PieGraph ตามประเภทของกราฟ
---------	--

3.2.2.2 คลาส GraphConfiguration

คลาส GraphConfiguration ประกอบด้วยคุณสมบัติและเม็ท็อด โดยมีรายละเอียดดังแสดงในตารางที่ 3.5

ตารางที่ 3.5 รายละเอียดของคลาส GraphConfiguration

ชื่อคลาส	GraphConfiguration
ประเภท	คลาสหลัก
คำอธิบาย	คลาสสำหรับรับข้อกำหนดที่จะนำมาใช้ในการสร้างแฟ้มข้อมูลเอสวีจี
ตัวแปร	<ul style="list-style-type: none"> ▪ StartX จุดเริ่มต้นของกราฟทางแกนตั้ง ▪ StartY จุดเริ่มต้นของกราฟทางแกนนอน ▪ CanvasHeight ความสูงของพื้นที่แสดงกราฟ ▪ CanvasWidth ความกว้างของพื้นที่แสดงกราฟ ▪ Title ชื่อกราฟ ▪ TitleX คำอธิบายทางแกนตั้ง ▪ TitleY คำอธิบายทางแกนนอน

ตารางที่ 3.5 รายละเอียดของคลาส GraphConfiguration (ต่อ)

ตัวแปร	<ul style="list-style-type: none"> ▪ Type ประเภทของกราฟ ▪ DbConnectionString ส่วนติดต่อฐานข้อมูล ▪ Sql ข้อความสั่งเอสคิวแอล ▪ Path ตำแหน่งแฟ้มข้อมูลเอสวีจี ▪ FileName ชื่อแฟ้มข้อมูลเอสวีจี
เมทอด	<ul style="list-style-type: none"> ▪ SetGraphConfiguration() ทำหน้าที่ในการรับข้อกำหนดจากผู้ใช้งานและบันทึกลงแฟ้มข้อกำหนด ▪ GetGraphConfiguration() ทำหน้าที่อ่านข้อกำหนดจากแฟ้มข้อกำหนดที่บันทึก

3.2.2.3 คลาส Graph

คลาส Graph ประกอบด้วยคุณสมบัติและเมทอด โดยมีรายละเอียดดังแสดงในตารางที่ 3.6

ตารางที่ 3.6 รายละเอียดของคลาส Graph

ชื่อคลาส	Graph
ประเภท	คลาสหลัก
คำอธิบาย	คลาสสำหรับใช้ในการสร้างแฟ้มข้อมูลเอสวีจี
ตัวแปร	-
เมทอด	<ul style="list-style-type: none"> ▪ GenerateSVG() ทำหน้าที่ในการสร้างแฟ้มข้อมูลเอสวีจีตามประเภทของกราฟต่างๆ ▪ GetDataFromDB() ทำหน้าที่ดึงข้อมูลจากฐานข้อมูลตามข้อความสั่งเอสคิวแอลที่อยู่ในแฟ้มข้อกำหนด ▪ GetMaxValue() ทำหน้าที่หาข้อมูลที่มีมากที่สุด ▪ GetMinValue() ทำหน้าที่หาข้อมูลที่มีน้อยที่สุด

ตารางที่ 3.6 รายละเอียดของคลาส Graph (ต่อ)

เมทอด	<ul style="list-style-type: none"> ▪ GetScaleMinMax() ทำหน้าที่หาสเกลที่มากที่สุดและน้อยที่สุดและระยะห่างระหว่างสเกล ▪ GetColumnName() ทำหน้าที่หาคอลัมน์ที่จะนำมาเป็นป้ายชื่อแกน ▪ GetColor() ทำหน้าที่หาสีที่จะนำมาวาดกราฟ ▪ GetFontSize() ทำหน้าที่หาขนาดตัวอักษรที่เหมาะสมกับความกว้างและความยาวของพื้นที่ที่มีอยู่ ▪ GetMeasure() ทำหน้าที่หาขนาดความกว้างและความยาวของตัวอักษร ▪ CreateSVGLineNode() ทำหน้าที่วาดเส้นตรง ▪ CreateSVGBarNode() ทำหน้าที่วาดสี่เหลี่ยม ▪ CreateSVGTextNode() ทำหน้าที่วาดข้อความทางแนวนอน ▪ CreateSVGTextVerticalNode() ทำหน้าที่วาดข้อความทางแนวตั้ง
-------	--

3.2.2.4 คลาส BarGraph

คลาส BarGraph ประกอบด้วยคุณสมบัติและเมทอดที่สืบทอดมาจากคลาส Graph โดยมีรายละเอียดดังแสดงในตารางที่ 3.7

ตารางที่ 3.7 รายละเอียดของคลาส BarGraph

ชื่อคลาส	BarGraph
ประเภท	คลาสลูกของคลาส Graph
คำอธิบาย	คลาสสำหรับสร้างแฟ้มข้อมูลเอสวีจีสำหรับกราฟแท่ง
ตัวแปร	-
เมทอด	<ul style="list-style-type: none"> ▪ GenerateSVG() ทำหน้าที่ในการสร้างแฟ้มข้อมูลเอสวีจีสำหรับกราฟแท่ง

3.2.2.5 คลาส LineGraph

คลาส LineGraph ประกอบด้วยคุณสมบัติและเมทอดที่สืบทอดมาจากคลาส Graph โดยมีรายละเอียดดังแสดงในตารางที่ 3.8

ตารางที่ 3.8 รายละเอียดของคลาส LineGraph

ชื่อคลาส	LineGraph
ประเภท	คลาสลูกของคลาส Graph
คำอธิบาย	คลาสสำหรับสร้างเพิ่มข้อมูลเอสวีจีสำหรับกราฟเส้น
ตัวแปร	-
เมทอด	<ul style="list-style-type: none"> ▪ GenerateSVG() ทำหน้าที่ในการสร้างเพิ่มข้อมูลเอสวีจีสำหรับกราฟเส้น

3.2.2.6 คลาส PieGraph

คลาส PieGraph ประกอบด้วยคุณสมบัติและเมทอดที่สืบทอดมาจากคลาส Graph และมีเมทอด CreateSVGPathNode() เพิ่มเติม โดยมีรายละเอียดดังแสดงในตารางที่ 3.9

ตารางที่ 3.9 รายละเอียดของคลาส PieGraph

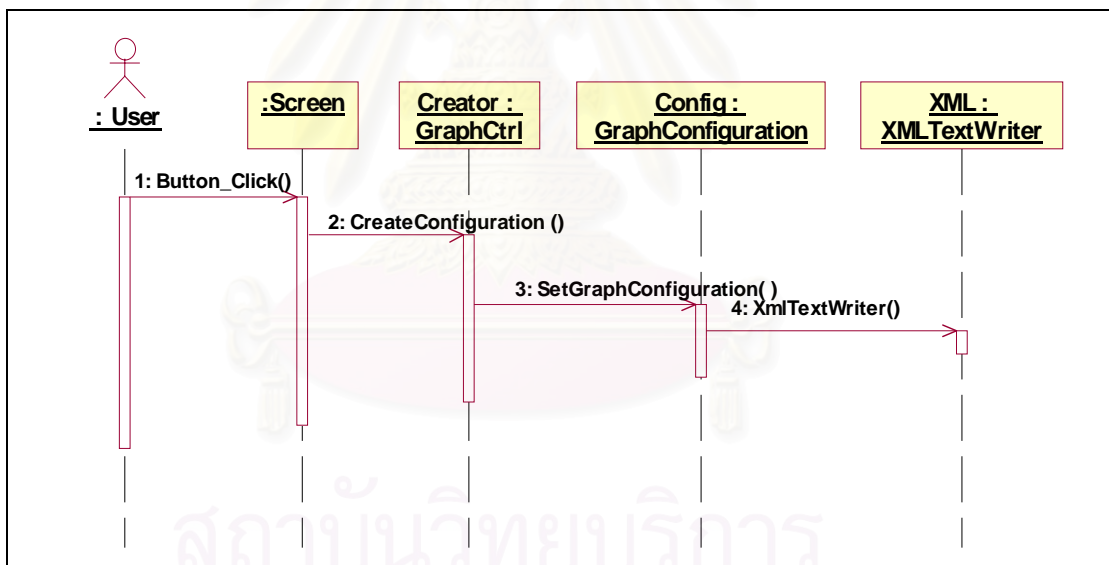
ชื่อคลาส	PieGraph
ประเภท	คลาสลูกของคลาส Graph
คำอธิบาย	คลาสสำหรับสร้างเพิ่มข้อมูลเอสวีจีสำหรับกราฟวงกลม
ตัวแปร	-
เมทอด	<ul style="list-style-type: none"> ▪ GenerateSVG() ทำหน้าที่ในการสร้างเพิ่มข้อมูลเอสวีจีสำหรับกราฟวงกลม ▪ CreateSVGPathNode() เพื่อทำหน้าที่วาดส่วนของวงกลม

3.3 แผนภาพซีเควนซ์ (Sequence Diagram)

ลำดับขั้นตอนการทำงานและการติดต่อกันระหว่างอ็อบเจกต์ของส่วนประกอบกราฟสำหรับสร้างเพิ่มข้อมูลเอสวีจีสามารถอธิบายได้ด้วยแผนภาพซีเควนซ์ โดยมีรายละเอียดดังต่อไปนี้

3.3.1 แผนภาพซีเควนซ์ของการตั้งค่าข้อกำหนด

แผนภาพซีเควนซ์ของการตั้งค่าข้อกำหนดแสดงได้ดังรูปที่ 3.3 โดยการทำงานเริ่มจากผู้ใช้งานจะป้อนค่าข้อกำหนดต่างๆ จากนั้นกดปุ่มบันทึก จะส่งข้อความไปยังคลาส GraphCtrl เพื่อเรียกใช้เมธอด CreateConfiguration() หลังจากนั้นจะส่งข้อความไปยังคลาส GraphConfiguration เพื่อเรียกใช้เมธอด SetGraphConfiguration() เพื่อบันทึกข้อกำหนดต่างๆลงเพิ่มข้อกำหนดผ่านเมธอด XMLTextWriter()

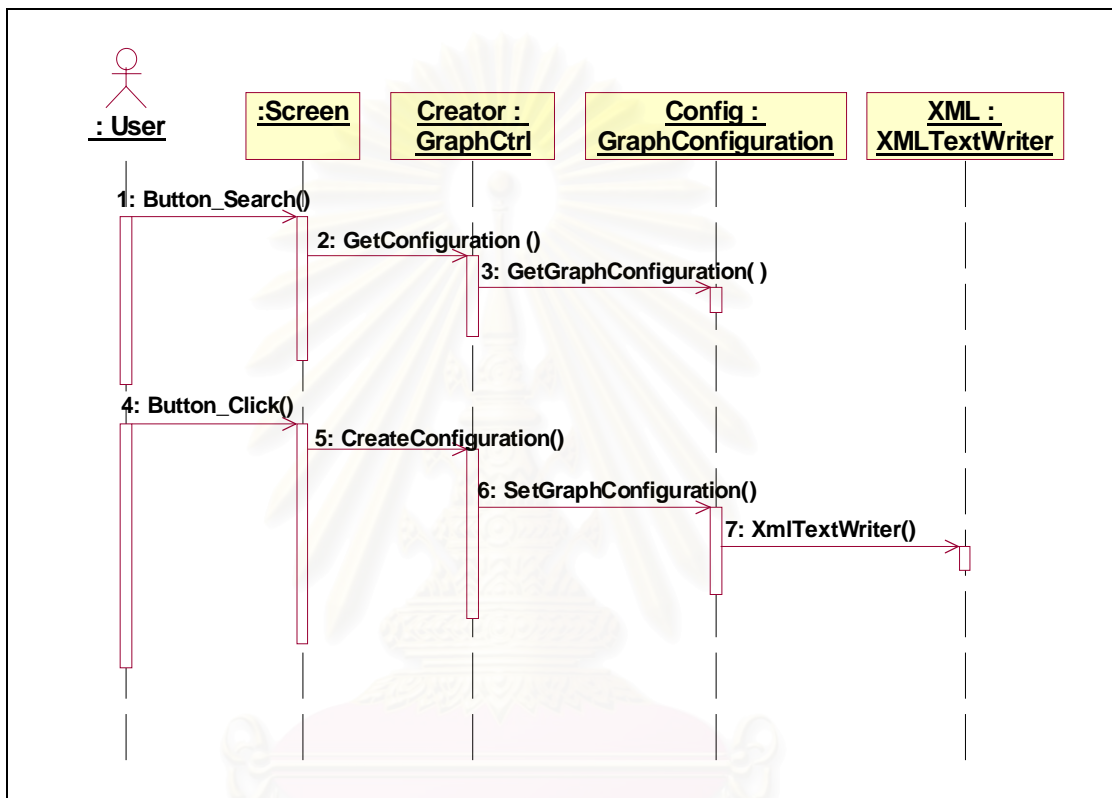


รูปที่ 3.3 แผนภาพซีเควนซ์ของการตั้งค่าข้อกำหนด

3.3.2 แผนภาพซีเควนซ์ของการแก้ไขข้อกำหนด

แผนภาพซีเควนซ์ของการแก้ไขเพิ่มข้อกำหนดแสดงได้ดังรูปที่ 3.4 โดยการทำงานเริ่มจากผู้ใช้งานใส่ชื่อเพิ่มข้อกำหนดที่ต้องการแก้ไข จากนั้นกดปุ่มค้นหาจะส่งข้อความไปยังคลาส GraphCtrl เพื่อเรียกใช้เมธอด GetConfiguration() หลังจากนั้น GraphCtrl จะส่งข้อความไปยังคลาส GraphConfiguration เพื่อเรียกใช้เมธอด GetGraphConfiguration() เพื่อดึงข้อกำหนดต่างๆจากเพิ่มข้อกำหนดนั้น เมื่อทำการแก้ไข

ไซ้ข้อกำหนดต่างๆเสร็จแล้วกดปุ่มบันทึก จะส่งข้อความไปยังคลาส GraphCtrl เพื่อเรียกใช้เมธอด CreateConfiguration() จากนั้นจะส่งข้อความไปยังคลาส GraphConfiguration เพื่อเรียกใช้เมธอด SetGraphConfiguration() เพื่อบันทึกข้อกำหนดต่างๆลงแฟ้มข้อกำหนดที่เรียกขึ้นมาแก้ไขผ่านเมธอด XMLTextWriter()



รูปที่ 3.4 แผนภาพซีควเอนซ์ของการแก้ไขข้อกำหนด

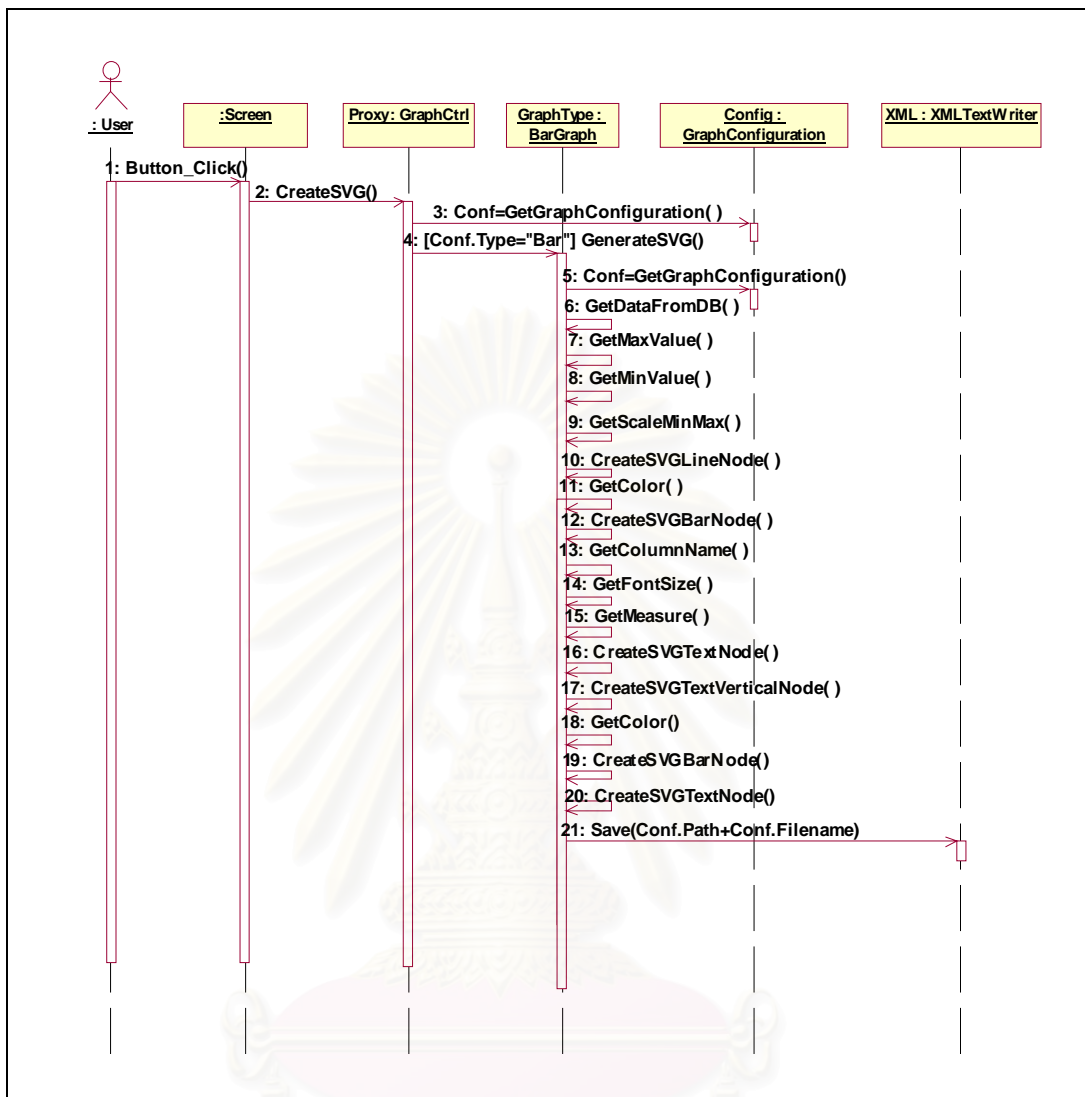
3.3.3 แผนภาพซีควเอนซ์ของการสร้างแฟ้มข้อมูลเอสวีจีของกราฟแท่ง

แผนภาพซีควเอนซ์ของการสร้างแฟ้มข้อมูลเอสวีจีของกราฟประเภทแท่งแสดงได้ดังรูปที่ 3.5 เริ่มจากผู้ใช้กดปุ่มสร้างกราฟจะส่งข้อความไปยัง GraphCtrl ซึ่งจะเรียกใช้เมธอด CreateSVG() แล้วส่งข้อความไปยังคลาส GraphConfiguration เรียกใช้เมธอด GetGraphConfiguration() เพื่ออ่านแฟ้มข้อกำหนดและตรวจสอบประเภทของกราฟที่ต้องการซึ่งเมื่อพบว่าเป็นกราฟแท่งจะเรียกใช้เมธอด GenerateSVG() ของคลาส BarGraph เพื่อทำการสร้างกราฟโดยมีรายละเอียดดังนี้

- ทำการอ่านข้อมูลข้อกำหนดของกราฟอื่นๆที่ต้องการโดยเรียกใช้เมธอด GetGraphConfiguration() ของคลาส GraphConfiguration

- นำประโยคคำสั่งเอสควิลที่ได้จากการอ่านข้อมูลข้อกำหนด ไปดึงข้อมูลจากฐานข้อมูลโดยเรียกใช้เมทอด GetDataFromDB()
- หาข้อมูลที่มีค่าสูงสุดโดยเรียกใช้เมทอด GetMaxValue()
- หาข้อมูลที่มีค่าต่ำสุดโดยเรียกใช้เมทอด GetMinValue()
- นำค่าสูงสุดและต่ำสุดไปหาสเกลของกราฟโดยเรียกใช้เมทอด GetScaleMinMax()
- วาดเส้นแกน X แกน Y และเส้นตารางหลักโดยเรียกใช้เมทอด CreateSVGLineNode()
- ทำการหาสีของกราฟแต่ละแท่งโดยเรียกใช้เมทอด GetColor()
- วาดกราฟแท่งโดยเรียกใช้เมทอด CreateSVGBarNode()
- หาประเภทของข้อมูลที่จะไปแสดงบนแกน X โดยเรียกใช้เมทอด GetColumnName()
- ทำการหาขนาดของตัวอักษรโดยเรียกใช้เมทอด GetFontSize()
- คำนวณความกว้างและความยาวของตัวอักษรโดยเรียกใช้เมทอด GetMeasure()
- วาดตัวอักษรประเภทข้อมูล ชื่อกราฟ ชื่อแกน X โดยเรียกใช้เมทอด CreateSVGTextNode()
- วาดตัวอักษรชื่อแกน Y โดยเรียกใช้เมทอด CreateSVGTextVerticalNode()
- ทำการวาดคำอธิบายข้อมูลซึ่งประกอบด้วยสัญลักษณ์และคำอธิบาย โดยเรียกใช้เมทอด GetColor() เพื่อหาสีที่ใช้แสดงรูปคำอธิบายกราฟ แล้วรูปสี่เหลี่ยมคำอธิบายโดยใช้เมทอด CreateSVGBarNode() แล้วเรียกเมทอด CreateSVGTextNode() เพื่อวาดตัวอักษรคำอธิบายกราฟ

เมื่อเสร็จสิ้นขั้นตอนการคำนวณเพื่อสร้างกราฟข้างต้น ก็จะทำการบันทึกเป็นรูปแบบแฟ้มข้อมูลเอสควิล



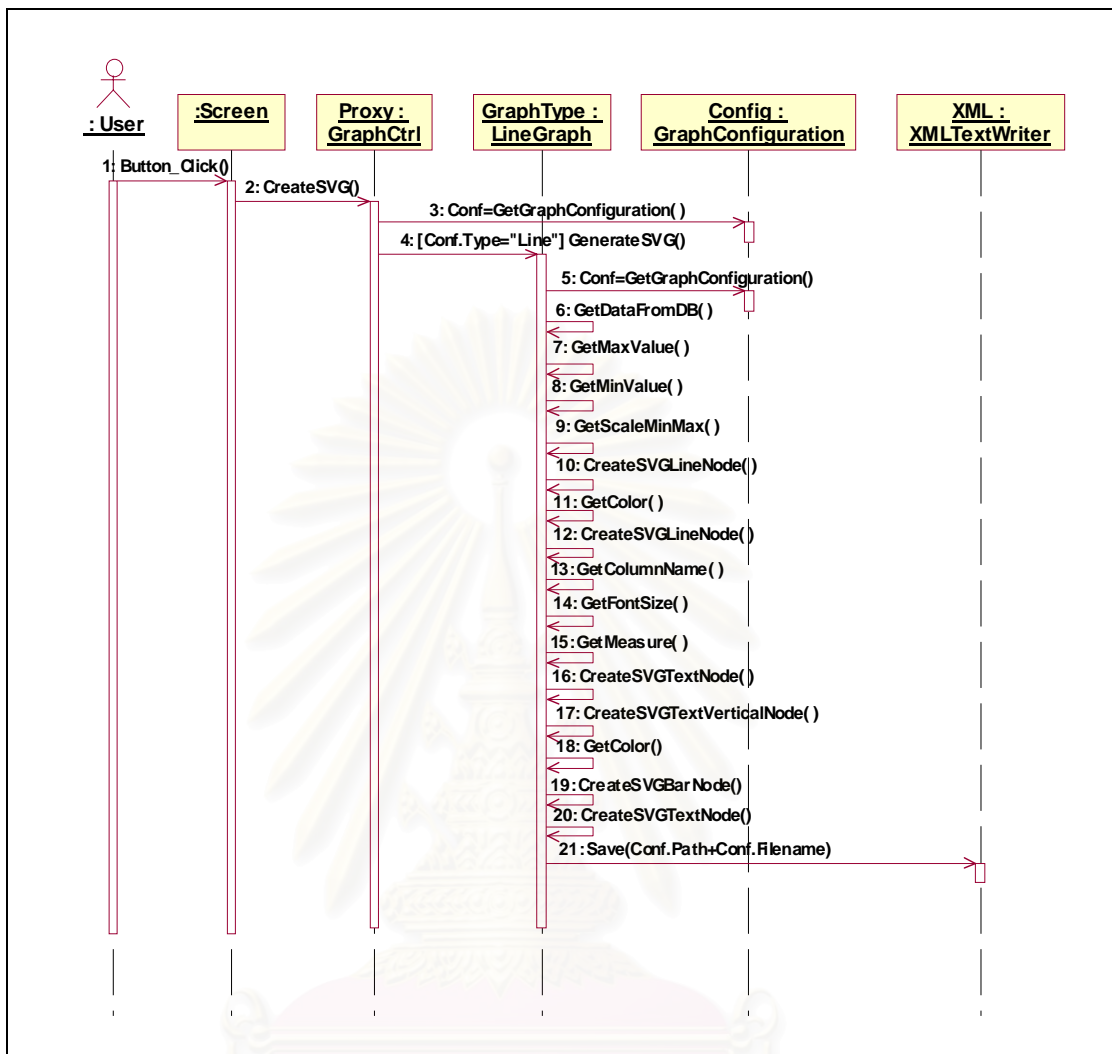
รูปที่ 3.5 แผนภาพซีควเอนซ์ของการสร้างแฟ้มข้อมูลเอสวิจีของกราฟแท่ง

3.3.4 แผนภาพซีควเอนซ์ของการสร้างแฟ้มข้อมูลเอสวิจีของกราฟเส้น

แผนภาพซีควเอนซ์ของการสร้างแฟ้มข้อมูลเอสวิจีของกราฟประเภทเส้นแสดงได้ดังรูปที่ 3.6 เริ่มจากผู้ใช้กดปุ่มสร้างกราฟจะส่งข้อความไปยัง GraphCtrl ซึ่งจะเรียกใช้เมทอด CreateSVG() แล้วส่งข้อความไปยังคลาส GraphConfiguration เรียกใช้เมทอด GetGraphConfiguration() เพื่อตรวจสอบประเภทของกราฟที่ต้องการซึ่งเมื่อพบว่าเป็นกราฟเส้นจะเรียกใช้เมทอด GenerateSVG() ของคลาส LineGraph เพื่อทำการสร้างกราฟโดยมีรายละเอียดดังนี้

- ทำการอ่านข้อมูลข้อกำหนดของกราฟอื่นๆที่ต้องการโดยเรียกใช้เมธอด `GetGraphConfiguration()` ของคลาส `GraphConfiguration`
- นำประโยคคำสั่งเอสคิวแอลที่ได้จากการอ่านข้อมูลข้อกำหนด ไปดึงข้อมูลจากฐานข้อมูลโดยเรียกใช้เมธอด `GetDataFromDB()`
- หาข้อมูลที่มีค่าสูงสุดโดยเรียกใช้เมธอด `GetMaxValue()`
- หาข้อมูลที่มีค่าต่ำสุดโดยเรียกใช้เมธอด `GetMinValue()`
- นำค่าสูงสุดและต่ำสุดไปหาสเกลของกราฟโดยเรียกใช้เมธอด `GetScaleMinMax()`
- วาดเส้นแกน X แกน Y และเส้นตารางหลักโดยเรียกใช้เมธอด `CreateSVGLineNode()`
- ทำการหาสีของกราฟแต่ละเส้นโดยเรียกใช้เมธอด `GetColor()`
- วาดกราฟเส้นโดยเรียกใช้เมธอด `CreateSVGLineNode()`
- หาประเภทของข้อมูลที่จะไปแสดงบนแกน X โดยเรียกใช้เมธอด `GetColumnName()`
- ทำการหาขนาดของตัวอักษรโดยเรียกใช้เมธอด `GetFontSize()`
- คำนวณความกว้างและความยาวของตัวอักษรโดยเรียกใช้เมธอด `GetMeasure()`
- วาดตัวอักษรประเภทข้อมูล ชื่อกราฟ ชื่อแกน X โดยเรียกใช้เมธอด `CreateSVGTextNode()`
- วาดตัวอักษรชื่อแกน Y โดยเรียกใช้เมธอด `CreateSVGTextVerticalNode()`
- ทำการวาดคำอธิบายข้อมูลซึ่งประกอบด้วยสัญลักษณ์และคำอธิบาย โดยเรียกใช้เมธอด `GetColor()` เพื่อหาสีที่ใช้แสดงรูปคำอธิบายกราฟ แล้วรูปสี่เหลี่ยมคำอธิบายโดยใช้เมธอด `CreateSVGBarNode()` แล้วเรียกเมธอด `CreateSVGTextNode()` เพื่อวาดตัวอักษรคำอธิบายกราฟ

เมื่อเสร็จสิ้นขั้นตอนการคำนวณเพื่อสร้างกราฟข้างต้น ก็จะทำการบันทึกเป็นรูปแบบแฟ้มข้อมูลเอสวีจี



รูปที่ 3.6 แผนภาพซีควเอนซ์ของการสร้างแฟ้มข้อมูลเอสวีจีของกราฟเส้น

3.3.5 แผนภาพซีควเอนซ์ของการสร้างแฟ้มข้อมูลเอสวีจีของกราฟวงกลม

แผนภาพซีควเอนซ์ของการสร้างแฟ้มข้อมูลเอสวีจีของกราฟประเภทวงกลมแสดงได้ดังรูปที่ 3.7 โดยการทำงานเริ่มจากผู้ใช้กดปุ่มสร้างจะส่งข้อความไปยัง GraphCtrl ซึ่งจะเรียกใช้เมทอด CreateSVG() แล้วส่งข้อความไปยังคลาส GraphConfiguration เรียกใช้เมทอด GetGraphConfiguration() เพื่อตรวจสอบประเภทของกราฟที่ต้องการซึ่งเมื่อพบว่าเป็นกราฟวงกลมจากนั้นจะเรียกใช้เมทอด GenerateSVG() ของคลาส PieGraph เพื่อทำการสร้างกราฟโดยมีรายละเอียดดังนี้

- ทำการอ่านข้อมูลข้อกำหนดของกราฟอื่น ๆ ที่ต้องการโดยเรียกใช้เมทอด GetGraphConfiguration() ของคลาส GraphConfiguration

- นำประโยคคำสั่งเอสคิวแอลที่ได้จากการอ่านข้อมูลข้อกำหนด ไปดึงข้อมูลจากฐานข้อมูลโดยเรียกใช้เมธอด GetDataFromDB()
- ทำการหาสีของกราฟแต่ละส่วนของวงกลมโดยเรียกใช้เมธอด GetColor()
- ทำการวาดส่วนของวงกลมโดยเรียกใช้เมธอด CreateSVGPathNode()
- ทำการหาขนาดของตัวอักษรโดยเรียกใช้เมธอด GetFontSize()
- คำนวณความกว้างและความยาวของตัวอักษรโดยเรียกใช้เมธอด GetMeasure()
- วาดตัวอักษรที่กราฟ โดยเรียกใช้เมธอด CreateSVGTextNode()
- ทำการวาดคำอธิบายข้อมูลซึ่งประกอบด้วยสัญลักษณ์และคำอธิบาย โดยเรียกใช้เมธอด GetColor() เพื่อหาสีที่ใช้แสดงรูปคำอธิบายกราฟ แล้วรูปสี่เหลี่ยมคำอธิบายโดยใช้เมธอด CreateSVGBarNode() แล้วเรียกเมธอด CreateSVGTextNode() เพื่อวาดตัวอักษรคำอธิบายกราฟ

เมื่อเสร็จสิ้นขั้นตอนการคำนวณเพื่อสร้างกราฟข้างต้น ก็จะทำการบันทึกเป็นรูปแบบแฟ้มข้อมูลเอสวีจี

3.4 แผนภาพกิจกรรม (Activity Diagram)

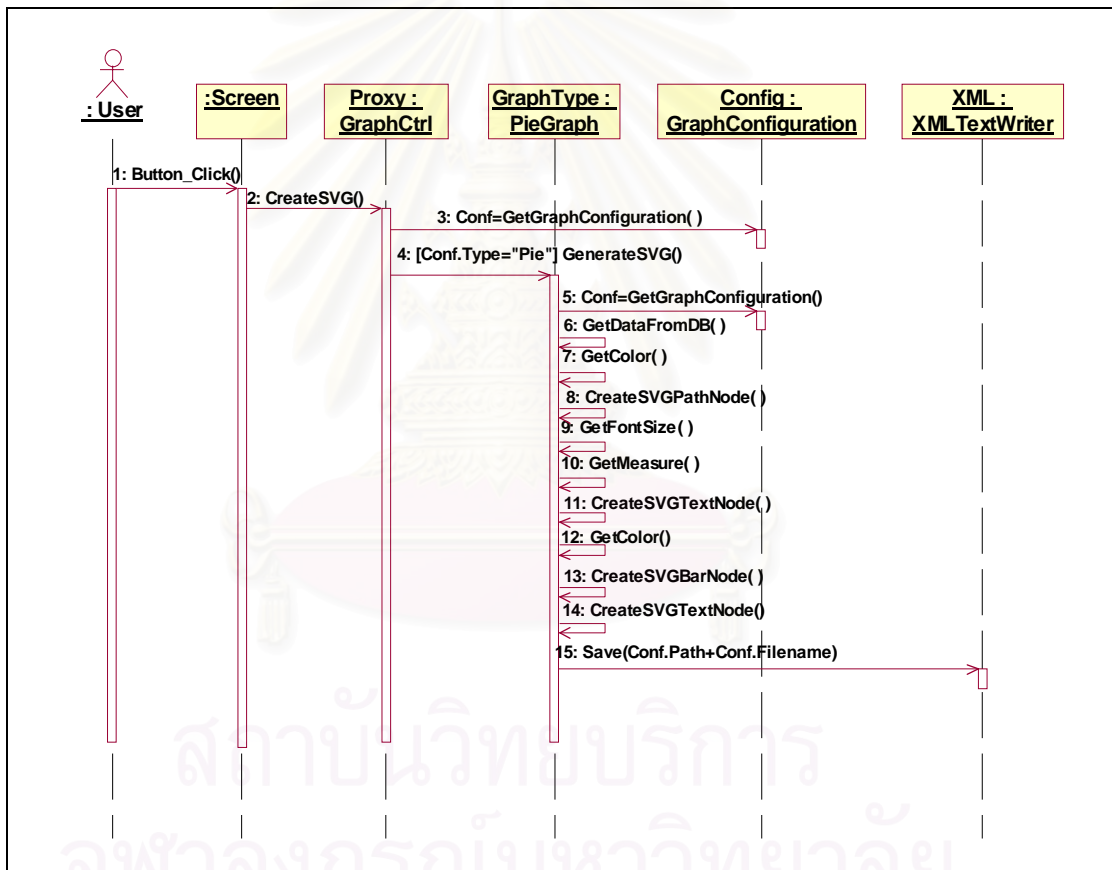
ลำดับขั้นตอนของการทำงานของส่วนประกอบกราฟสำหรับสร้างแฟ้มข้อมูลเอสวีจีสามารถแสดงได้จากแผนภาพกิจกรรมโดยสามารถแบ่งลำดับขั้นตอนออกเป็น 3 ส่วน คือ ลำดับขั้นตอนการกำหนดข้อกำหนดลงในแฟ้มข้อกำหนด ลำดับขั้นตอนการแก้ไขข้อกำหนดลงในแฟ้มข้อกำหนด และลำดับขั้นตอนการสร้างแฟ้มข้อมูลเอสวีจีดังรายละเอียดต่อไปนี้

3.4.1 แผนภาพกิจกรรมแสดงการตั้งค่าข้อกำหนดลงในแฟ้มข้อกำหนด

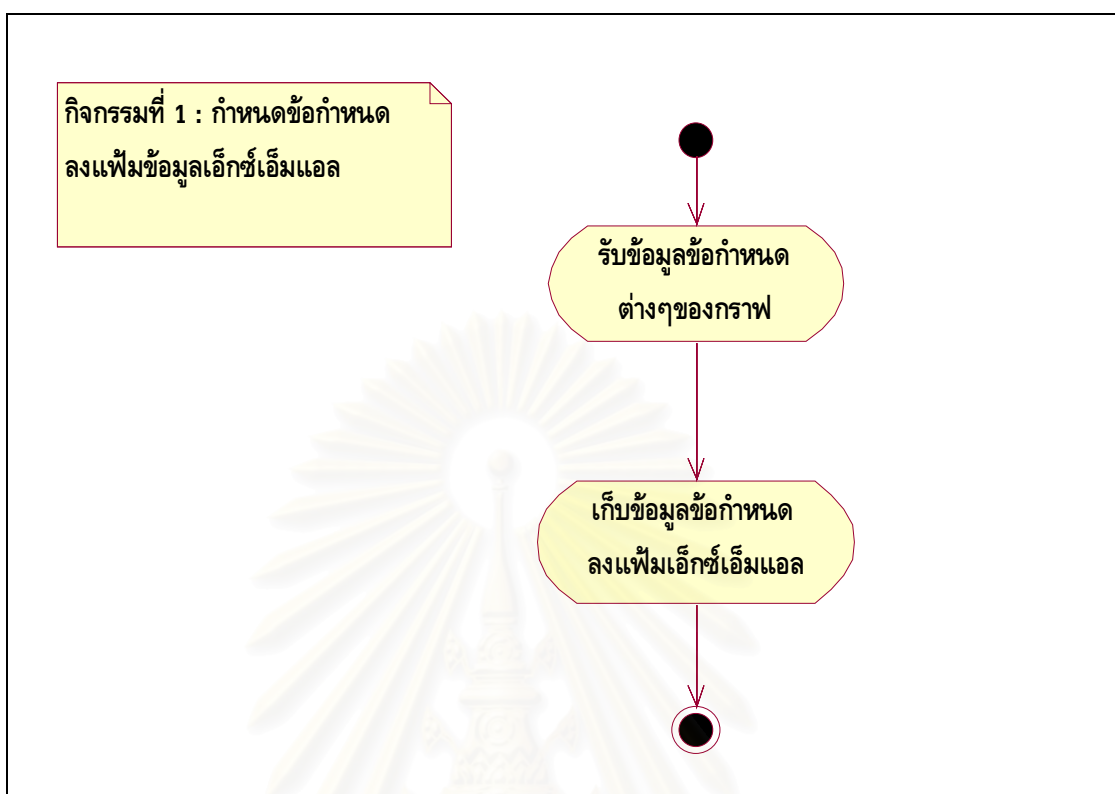
การตั้งค่าข้อกำหนดของกราฟลงในแฟ้มข้อกำหนด สามารถอธิบายได้ดังรูปที่ 3.8 เมื่อผู้ใช้งานป้อนข้อกำหนดต่างๆ ของกราฟที่ต้องการ ซึ่งได้แก่ ประเภทของกราฟ ความกว้าง ความยาวของพื้นที่แสดงกราฟ ข้อความสั่งเอสคิวแอล ชื่อกราฟ ชื่อแกนตั้ง ชื่อแกนนอน ชื่อและตำแหน่งที่เก็บแฟ้มข้อมูลเอสวีจี หลังจากนั้นระบบจะทำการบันทึกข้อมูลข้อกำหนดเหล่านั้นลงในแฟ้มข้อกำหนดซึ่งอยู่ในรูปแบบเอ็กซ์เอ็มแอล

3.4.2 แผนภาพกิจกรรมแสดงการแก้ไขข้อกำหนดลงในเพิ่มข้อกำหนด

การแก้ไขข้อกำหนดของกราฟลงในเพิ่มข้อกำหนด สามารถอธิบายได้ดังรูปที่ 3.9 เมื่อผู้ใช้งานใส่ชื่อเพิ่มข้อกำหนดที่ต้องการแก้ไข ระบบจะดึงข้อมูลจากเพิ่มข้อกำหนดนั้นขึ้นมาให้ผู้ใช้งานทำการแก้ไขข้อกำหนดต่างๆ ซึ่งได้แก่ ประเภทของกราฟ ความกว้าง ความยาวของพื้นที่แสดงกราฟ ข้อความสังเขปคิวแอล ชื่อกราฟ ชื่อแกนตั้ง ชื่อแกนนอน ชื่อและตำแหน่งที่เก็บเพิ่มข้อมูลเอสวิจี เมื่อผู้ใช้งานแก้ไขเรียบร้อยแล้วระบบจะทำการบันทึกข้อมูลข้อกำหนดเหล่านั้นลงในเพิ่มข้อกำหนดซึ่งอยู่ในรูปแบบเอ็กซ์เอ็มแอล



รูปที่ 3.7 แผนภาพซีควเอนซ์ของการสร้างเพิ่มข้อมูลเอสวิจีของกราฟวงกลม



รูปที่ 3.8 แผนภาพกิจกรรมการกำหนดข้อกำหนดของกราฟ

3.4.3 แผนภาพกิจกรรมแสดงการสร้างเพิ่มข้อมูลเอสวีจี

การสร้างเพิ่มข้อมูลเอสวีจี สามารถอธิบายได้ดังรูปที่ 3.10 เมื่อผู้ใช้งานเริ่มต้นให้ระบบสร้างเพิ่มข้อมูลเอสวีจี ระบบจะอ่านข้อกำหนดของกราฟจากเพิ่มข้อกำหนดที่บันทึกในรูปแบบเอ็กซ์เอ็มแอลไว้ และตรวจสอบว่าต้องการสร้างกราฟประเภทใด หลังจากนั้นจะเรียกเมธอดสำหรับสร้างกราฟประเภทนั้นๆ และบันทึกลงเป็นเพิ่มข้อมูลเอสวีจี

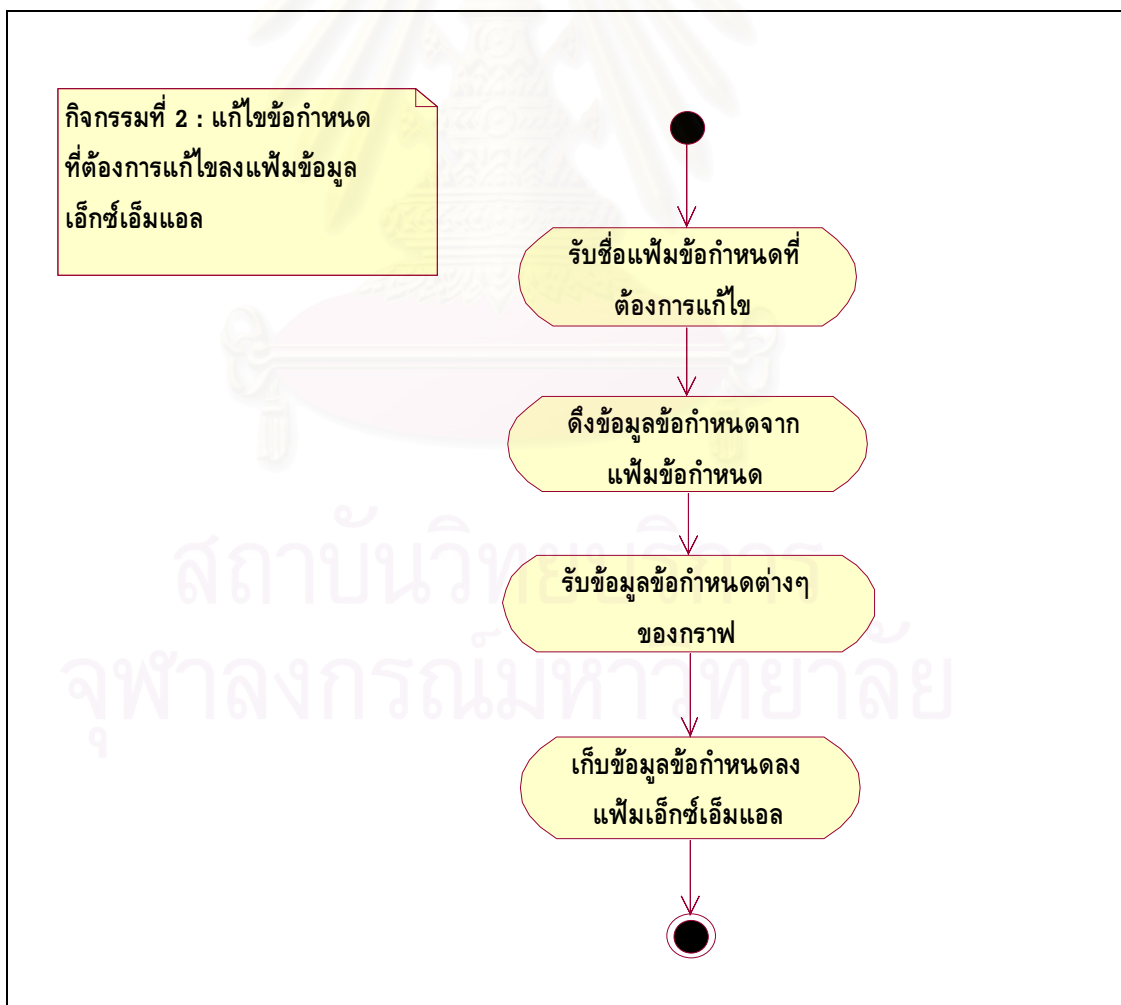
3.5 แผนภาพดีพลอยเมนต์ (Deployment Diagram)

การนำส่วนประกอบกราฟที่ได้ไปใช้งานสามารถอธิบายได้ดังรูปที่ 3.11 โดยจะเห็นว่าจะแบ่งเป็น 2 ฝั่งคือฝั่งไคลเอนต์และฝั่งเซิร์ฟเวอร์

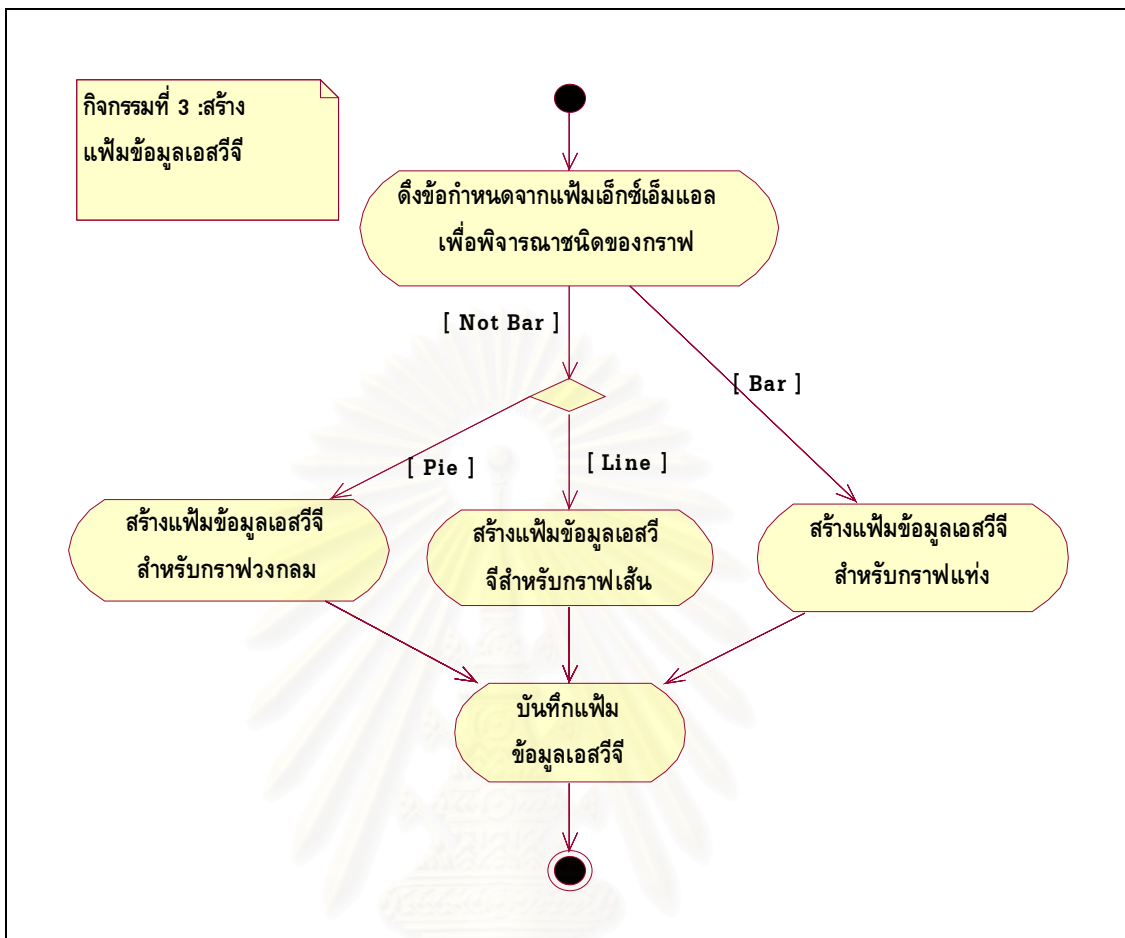
ฝั่งเซิร์ฟเวอร์ประกอบด้วยเว็บเซิร์ฟเวอร์ทำหน้าที่ให้บริการข้อมูลบนอินเทอร์เน็ต ส่วนประกอบกราฟสำหรับสร้างเพิ่มข้อมูลเอสวีจีทำหน้าที่สร้างกราฟ ฐานข้อมูลสำหรับดึงข้อมูลตามข้อกำหนด ส่วนประกอบเอ็กซ์เอ็มแอลทำหน้าที่สร้างและเก็บข้อมูลข้อกำหนด

ฝั่งไคลเอนต์ประกอบด้วยเว็บเบราว์เซอร์ทำหน้าที่แสดงเว็บเพจสำหรับผู้ใช้ ส่วนประกอบเอสวีจีวีวีเวอร์ทำหน้าที่แสดงเพิ่มข้อมูลเอสวีจี

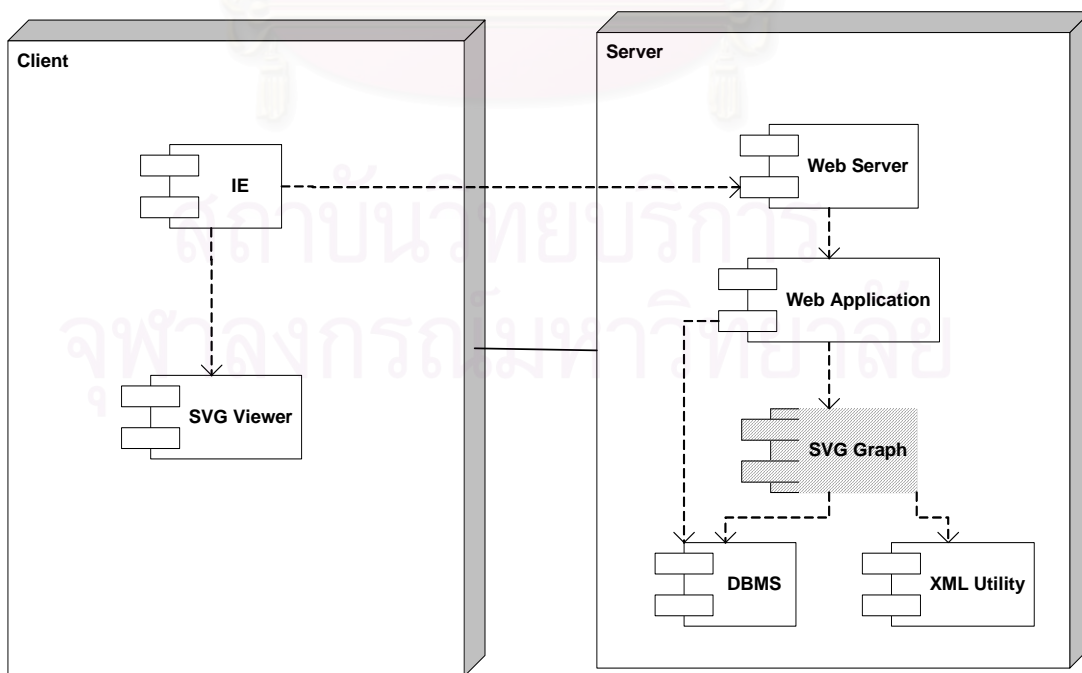
ในการใช้งานผู้ใช้งานซึ่งเป็นโปรแกรมเมอร์จะทำการพัฒนาซอฟต์แวร์ประเภทเว็บแอปพลิเคชันบนฝั่งเซิร์ฟเวอร์โดยมีการเรียกใช้ส่วนประกอบกราฟเพื่อสร้างรูปภาพในรูปแบบเอสวีจี เมื่อมีผู้ต้องการดูกราฟจากฝั่งไคลเอนต์จะทำการเรียกใช้เว็บแอปพลิเคชันผ่านเว็บเบราว์เซอร์ เว็บเบราว์เซอร์จะติดต่อไปยังเว็บเซิร์ฟเวอร์เพื่อทำการเรียกใช้งานส่วนประกอบกราฟ ซึ่งส่วนประกอบกราฟจะสร้างกราฟในรูปแบบเอสวีจีตามประเภทของกราฟที่อยู่ในเพิ่มข้อกำหนด ส่วนประกอบกราฟจะดึงข้อมูลจากฐานข้อมูลแล้วทำการสร้างเพิ่มข้อมูลเอสวีจีโดยเรียกส่วนประกอบเอ็กซ์เอ็มแอล จากนั้นเว็บเซิร์ฟเวอร์จะส่งเพิ่มข้อมูลที่ได้ไปแสดงบนเว็บเบราว์เซอร์ โดยเว็บเบราว์เซอร์จะเรียกใช้ส่วนประกอบเอสวีจีวีวีเวอร์ในการแสดงกราฟ



รูปที่ 3.9 แผนภาพกิจกรรมการแก้ไขเพิ่มข้อกำหนดของกราฟ



รูปที่ 3.10 แผนภาพกิจกรรมการสร้างเพิ่มข้อมูลเอสวิจีของกราฟ



รูปที่ 3.11 แผนภาพดีพลอยเมนต์

3.6 ลักษณะข้อมูลนำเข้า

ในส่วนของข้อมูลนำเข้าสำหรับนำมาสร้างกราฟ ผู้วิจัยได้ออกแบบให้มีการนำเข้าข้อมูลจากฐานข้อมูลเชิงสัมพันธ์ผ่านระบบจัดการฐานข้อมูลไมโครซอฟต์เอสคิวแอลเซิร์ฟเวอร์ โดยตารางข้อมูลสำหรับนำเข้านั้นจะต้องมีข้อมูลที่พร้อมสำหรับนำมาสร้างกราฟได้

```
SELECT ชื่อชุดข้อมูล, ชื่อประเภทข้อมูลที่ 1 [ชื่อประเภทข้อมูลที่ 2,...,ชื่อประเภทข้อมูลที่ n]
FROM ชื่อตารางฐานข้อมูล [ WHERE เงื่อนไขในการเลือกข้อมูล]
```

รูปที่ 3.12 รูปแบบของข้อความสั่งเอสคิวแอลที่กำหนดไว้

ผู้ใช้งานจะใช้ข้อความสั่งเอสคิวแอลในการเรียกข้อมูลที่ต้องการ รูปแบบของข้อความสั่งเอสคิวแอล ได้ถูกกำหนดไว้ดังรูปที่ 3.12 โดย

- ชื่อชุดข้อมูล (Data Series Name) : จะถูกนำไปแสดงบนคำอธิบายกราฟ (Legend) ในกรณีที่ข้อมูลในชุดข้อมูลมากกว่า 1 ชุด ชุดข้อมูลที่แตกต่างกันจะถูกแสดงในกราฟด้วยสีที่ต่างกัน
- ชื่อประเภทข้อมูล (Category Name) : สามารถระบุได้มากกว่า 1 ประเภท โดยในกรณีของกราฟแท่ง (Bar Chart) และกราฟเส้น (Line Chart) ที่มีประเภทข้อมูลมากกว่า 1 ประเภท ชื่อประเภทข้อมูลจะถูกนำไปแสดงบนแกนนอน ของกราฟ แต่ในกรณีของกราฟวงกลม (Pie Chart) จะมีประเภทข้อมูลอยู่เพียงหนึ่งประเภทเท่านั้น

```
SELECT SALESMAN,JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,
OCT,NOV,DEC FROM GENERAL_STATISTIC
```

รูปที่ 3.13 ตัวอย่างของข้อความสั่งเอสคิวแอล

ตัวอย่างของข้อความสั่งเอสคิวแอลสำหรับสร้างกราฟเปรียบเทียบยอดขายของพนักงานขายแต่ละคน จากเดือนมกราคมถึงเดือนธันวาคม สามารถแสดงได้รูปที่ 3.13 โดยจะเห็นว่าชื่อชุดข้อมูลคือ SALESMAN และ ชื่อประเภทข้อมูลคือ JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC

SALESMAN	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
A	20	55	30	40	50	55	45	35	54	89	52	76
B	30	45	56	39	30	40	42	46	50	53	59	100
C	40	35	42	28	27	34	37	42	49	58	67	87

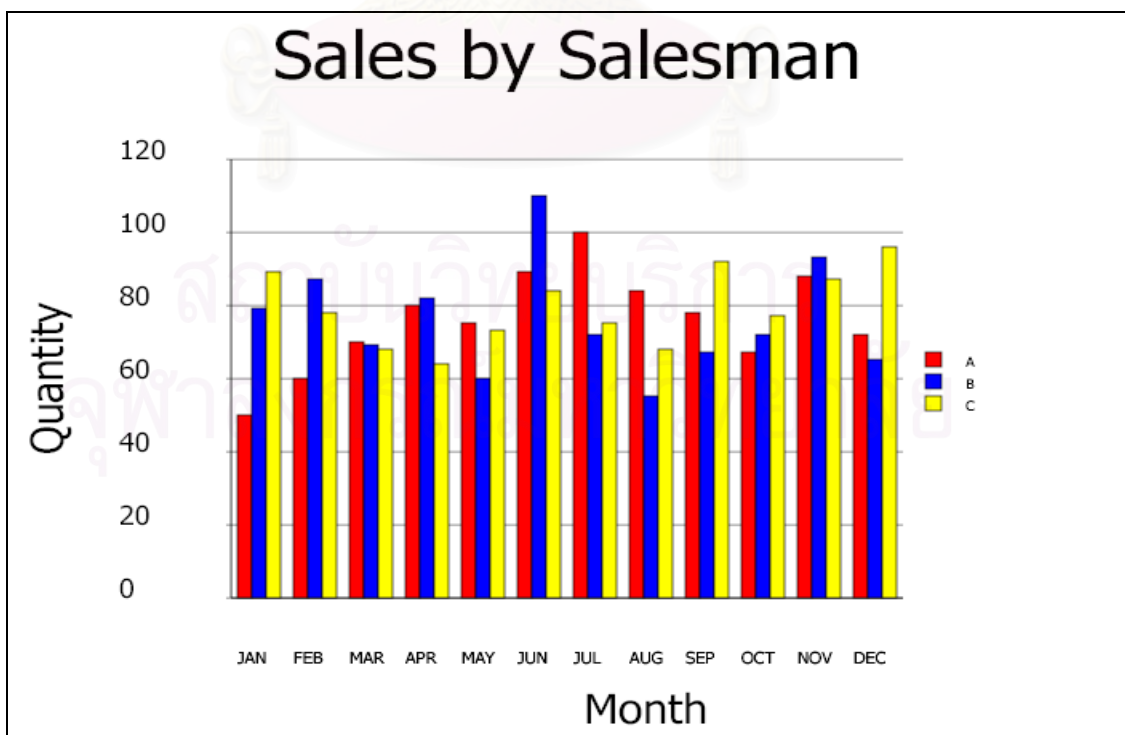
ส่วนที่ 1: เซลล์ A, B, C ในคอลัมน์แรก
ส่วนที่ 2: แถว A, B, C ในคอลัมน์ที่ 2-12
ส่วนที่ 3: เซลล์ A, B, C ในคอลัมน์ที่ 12

รูปที่ 3.14 แสดงตัวอย่างผลข้อมูลที่ได้จากข้อความสั่งเอสคิวแอล

ข้อมูลที่ได้จากข้อความสั่งเอสคิวแอลข้างต้น จะได้ผลดังแสดงในรูปที่ 3.14 ซึ่งประกอบด้วยส่วนต่างๆดังนี้

- ส่วนที่ 1 ในคอลัมน์ที่ 1 จะเป็นข้อมูลของ 'ชื่อชุดข้อมูล' ต่างๆ ซึ่งในตัวอย่างนี้คือ ชื่อของพนักงานขายแต่ละคน
- ส่วนที่ 2 ในแถวที่ 1 จะเป็นชื่อประเภทของข้อมูล ซึ่งจากตัวอย่างนี้คือ ชื่อเดือนต่างๆ
- ส่วนที่ 3 จะเป็นข้อมูลของชุดข้อมูลและประเภทข้อมูลต่างๆ สำหรับใช้ในการคำนวณและสร้างกราฟ ข้อมูลในส่วนนี้ต้องเป็นข้อมูลชนิดตัวเลขเท่านั้น

ตัวอย่างของกราฟที่ได้จากข้อมูลข้างต้น แสดงได้ดังรูปที่ 3.15



รูปที่ 3.15 ตัวอย่างรูปภาพที่ได้จากข้อความสั่งเอสคิวแอล

ในกรณีที่ตารางข้อมูลนำเข้ามีข้อมูลที่ซับซ้อน ผู้ใช้งานจะต้องแปลงข้อมูลนำเข้านั้นให้อยู่ในรูปแบบพิวอทเทเบิล (Pivot Table) ก่อน ซึ่งพิวอทเทเบิลหมายถึง ตารางข้อมูลโดยที่แปลงมาจากตารางข้อมูลเดิม โดยมีการจัดกลุ่มข้อมูล เรียงลำดับ และการแสดงค่ารวมของตัวเลขของข้อมูลที่มีความสัมพันธ์กัน ยกตัวอย่างเช่น การสร้างกราฟเปรียบเทียบยอดขายของพนักงานขายแต่ละคนในแต่ละเดือนจากตารางข้อมูลนำเข้าในตารางที่ 3.10 จะเห็นว่าไม่สามารถเรียกข้อมูลยอดขายรวมของพนักงานขายแต่ละคนในแต่ละเดือนด้วยข้อความสั่งเอสคิวแอลตามรูปแบบที่กำหนดไว้ได้ ดังนั้นในกรณีนี้ผู้ใช้งานต้องแปลงลักษณะข้อมูลดังกล่าวให้อยู่ในรูปแบบพิวอทเทเบิลก่อน ดังแสดงในตารางที่ 3.11

ตารางที่ 3.10 ตัวอย่างรูปแบบข้อมูลนำเข้าที่ต้องผ่านการแปลงก่อนนำไปสร้างกราฟ

SALESMAN	MONTH	QUANTITY
A	JAN	20
A	FEB	55
A	MAR	30
A	APR	40
A	MAY	50
A	JUN	55
B	JAN	30
B	FEB	45
B	MAR	56
B	APR	39
B	MAY	30
B	JUN	40
C	JAN	40
C	FEB	35
C	MAR	42
C	APR	28
C	MAY	27
C	JUN	34

ตารางที่ 3.11 ตัวอย่างรูปแบบข้อมูลนำเข้าที่แปลงให้อยู่ในรูปแบบpivotเทเบิล

SALESMAN	JAN	FEB	MAR	APR	MAY	JUN
A	20	55	30	40	50	55
B	30	45	56	39	30	40
C	40	35	42	28	27	34



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

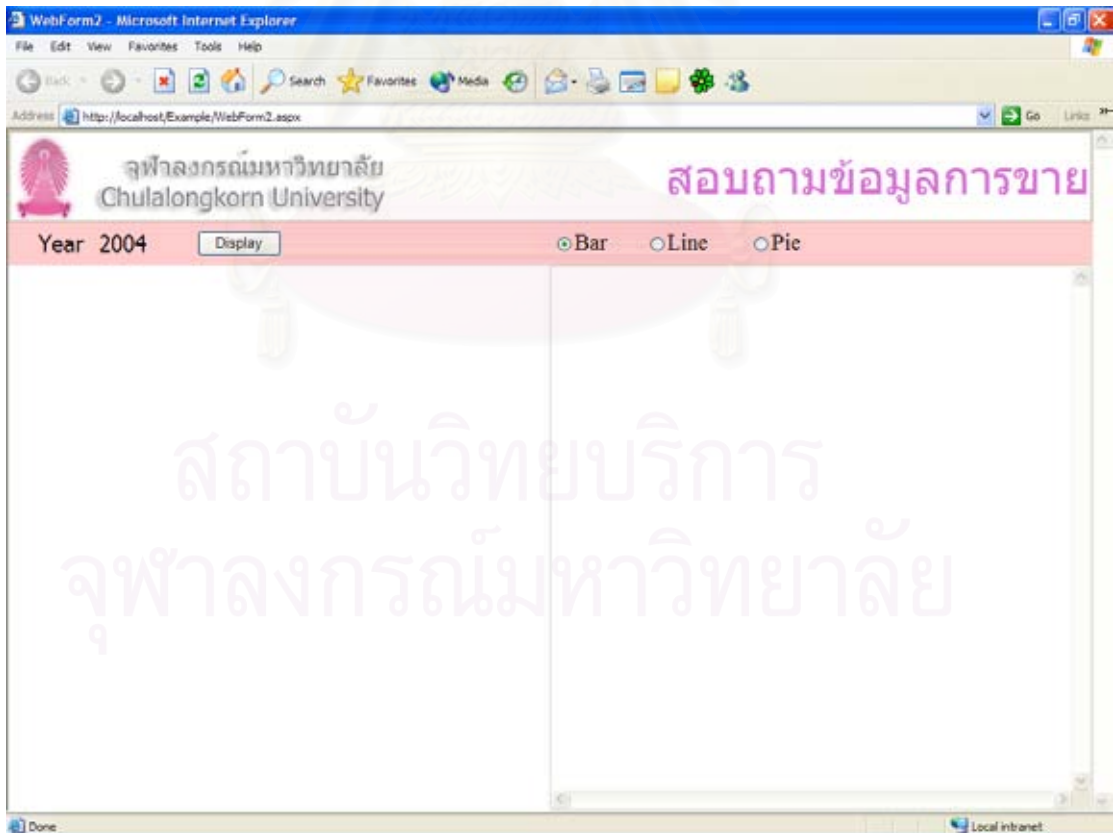
บทที่ 4

การทดสอบการใช้งานส่วนประกอบกราฟ

ผู้วิจัยได้ทำการทดสอบส่วนประกอบกราฟสำหรับสร้างแฟ้มข้อมูลเอสวีจีโดยมีวัตถุประสงค์คือสามารถเรียกใช้ส่วนประกอบกราฟได้มากกว่า 1 ภาษา ซึ่งในการทดสอบได้เรียกใช้ส่วนประกอบกราฟผ่าน 2 ภาษา ได้แก่ ภาษาวิซวลเบสิกคอตเน็ต และภาษาซีชาร์ป

4.1 การทดสอบส่วนประกอบกราฟด้วยภาษาวิซวลเบสิกคอตเน็ต

ผู้วิจัยทำการทดสอบการใช้ส่วนประกอบกราฟ โดยเป็นการแสดงข้อมูลยอดขายสินค้าไตรมาสแรกตามลูกค้าของปี 2004 โดยดึงข้อมูลจากฐานข้อมูลเอสคิวแอลเซิร์ฟเวอร์ ซึ่งใช้ฐานข้อมูลชื่อ "SVGBASE" เป็นฐานข้อมูลที่ผู้วิจัยสร้างขึ้นเป็นตัวทดสอบ และผู้วิจัยได้สร้างหน้าจอสำหรับการแสดงผลดังรูปที่ 4.1 ประกอบด้วย ส่วนที่ใช้เลือกชนิดของกราฟ ส่วนแสดงผลตารางข้อมูลและส่วนแสดงกราฟ



รูปที่ 4.1 หน้าจอเริ่มต้นการแสดงผลยอดขาย

จากหน้าจอนี้สามารถเลือกแสดงกราฟได้ 3 ประเภทได้แก่ กราฟแท่ง กราฟเส้น และกราฟวงกลม โดยแสดงรายละเอียดดังต่อไปนี้

4.1.1 การแสดงผลกราฟแท่ง

เมื่อผู้วิจัยเลือกกราฟประเภทแท่ง ส่วนประกอบกราฟจะใช้เพิ่มข้อกำหนดซึ่งแสดงไว้ดังรูปที่ 4.2 โดยจะกำหนดจุดเริ่มต้นด้าน X และ Y ขนาดความกว้าง ความยาวของพื้นที่แสดงกราฟ ชื่อกราฟ ชื่อทางด้านแกน X ชื่อทางด้านแกน Y ประเภทของกราฟ ชื่อฐานข้อมูล ชื่อความถี่เอสคิวแอล ตำแหน่งที่เก็บแฟ้มข้อมูล และชื่อแฟ้มข้อมูล

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <start startX="0" startY="0" />
  <canvas width="600" height="400" />
  <heading title="First Quarter Sales by Customer" titleX="Month" titleY="Quantity" />
  <graphType type="Bar" />
  <db connStr="data source=localhost;initial catalog=svgbase;user id=sa;password=">
  <sql type="sql">select cv_code as CUSTOMER,jan as JAN,feb as FEB,mar as MAR,apr as
APR from general_statistic where year_no=2004
  </sql>
  </db>
  <file path="Result" name="bar" />
</configuration>
```

รูปที่ 4.2 เพิ่มข้อกำหนดสำหรับแสดงกราฟแท่ง

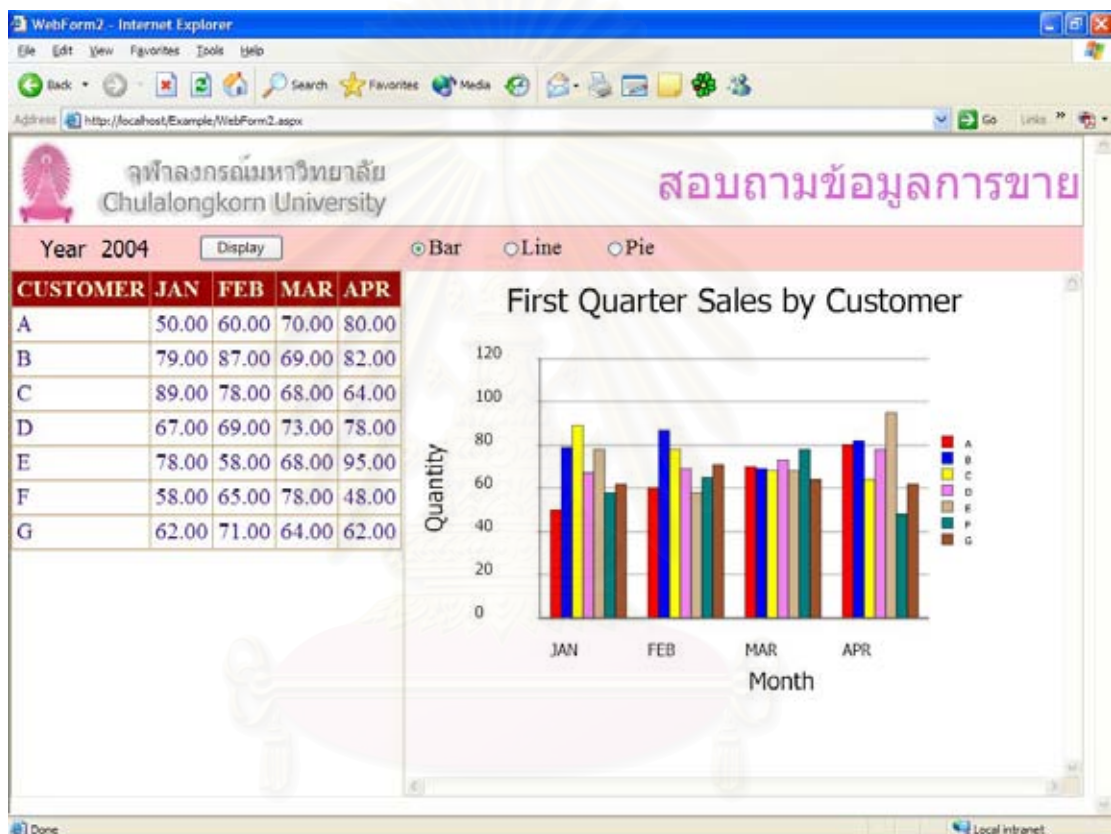
สำหรับการนำส่วนประกอบกราฟไปใช้ทำได้โดยการเพิ่มการอ้างอิง (Add Reference) SVGComponent.dll เข้าไปในโปรเจค (Project) ของวิซวลสตูดิโอเอกซทเน็ต จากนั้นทำการเรียกใช้ส่วนประกอบกราฟดังรูปที่ 4.3

```
Dim graph As New GraphCtrl
svg = graph.CreateSVG(configPath)
```

รูปที่ 4.3 การเรียกใช้ส่วนประกอบกราฟสำหรับกราฟแท่งโดยภาาษาวิซวลเบสิกเอกซทเน็ต

จากรูปที่ 4.3 จะทำการสร้างอ็อบเจกต์ GraphCtrl แล้วเรียกเมทอด CreateSVG โดยมีพารามิเตอร์ (Parameter) เป็นเพิ่มข้อกำหนดซึ่งแสดงไว้ดังรูป 4.2

ผลลัพธ์จากการเรียกใช้ส่วนประกอบกราฟจะแสดงได้ดังรูปที่ 4.4 และรูปภาพที่ได้จะถูกบันทึกไว้ที่ตำแหน่งชื่อ Result จากการตรวจสอบพบว่าเพิ่มข้อมูลเอสวีจีที่ได้มีขนาด 8 กิโลไบต์ ซึ่งเมื่อนำรูปภาพนั้นจากหน้าจอมาแปลงให้อยู่ในรูปแบบเจเพ็ก (JPEG) มีขนาด 73 กิโลไบต์ เมื่อเปรียบเทียบกันแล้วขนาดจะลดลงไปประมาณ 89 เปอร์เซ็นต์



รูปที่ 4.4 ข้อมูลและรูปภาพแท่ง

4.1.2 การแสดงผลกราฟเส้น

เมื่อผู้วิจัยเลือกกราฟประเภทเส้น ส่วนประกอบกราฟจะใช้เพิ่มข้อกำหนดซึ่งแสดงไว้ดังรูปที่ 4.5 ซึ่งกำหนดจุดเริ่มต้นด้าน X และ Y ขนาดความกว้าง ความยาวของพื้นที่แสดงกราฟ ชื่อกราฟ ชื่อทางด้านแกน X ชื่อทางด้านแกน Y ประเภทของกราฟ ชื่อฐานข้อมูล ชื่อความถี่เอสคิวแอล ตำแหน่งที่เก็บเพิ่มข้อมูล และชื่อเพิ่มข้อมูล

สำหรับการนำส่วนประกอบกราฟไปใช้ทำได้โดยการเพิ่มการอ้างอิง SVGComponent.dll เข้าไปในโพเจคของวิซวลสตูดิโอเดสทอปเน็ต จากนั้นทำการเรียกใช้ ส่วนประกอบกราฟดังรูปที่ 4.6

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <start startX="0" startY="0" />
  <canvas width="600" height="400" />
  <heading title="First Quarter Sales by Customer" titleX="Month" titleY="Quantity" />
  <graphType type="Line" />
  <db connStr="data source=localhost;initial catalog=svgbase;user id=sa;password=";
  <sql type="sql">select cv_code as CUSTOMER,jan as JAN,feb as FEB,mar as
MAR,apr as APR from general_statistic where year_no=2004
</sql>
</db>
<file path="Result" name="line" />
</configuration>
```

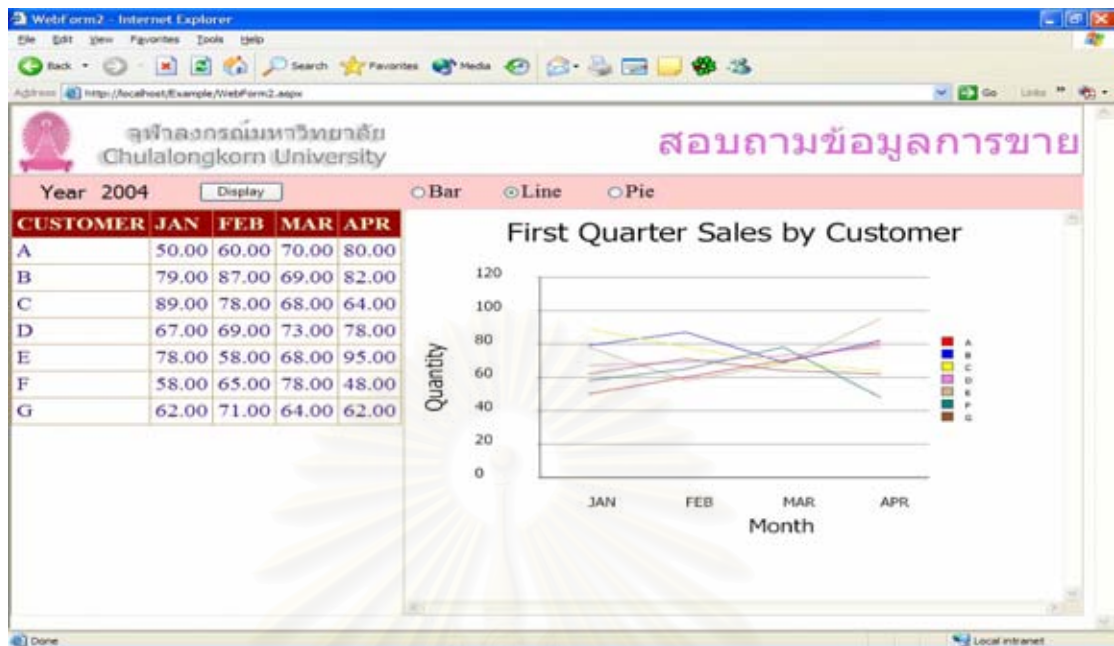
รูปที่ 4.5 เพิ่มข้อกำหนดสำหรับแสดงกราฟเส้น

```
Dim graph As New GraphCtrl
svg = graph.CreateSVG(configPath)
```

รูปที่ 4.6 การเรียกใช้ส่วนประกอบกราฟสำหรับกราฟเส้นโดยภาษาวิซวลเบสิกเดสทอปเน็ต

จากรูปที่ 4.6 จะทำการสร้างอ็อบเจกต์ GraphCtrl แล้วเรียกเมทอด CreateSVG โดยมีพารามิเตอร์เป็นเพิ่มข้อกำหนดซึ่งแสดงไว้ดังรูป 4.5

ผลลัพธ์จากการเรียกใช้ส่วนประกอบกราฟจะแสดงได้ดังรูปที่ 4.7 และรูปภาพที่ได้จะถูกบันทึกไว้ที่ตำแหน่งชื่อ Result จากการตรวจสอบพบว่าเพิ่มข้อมูลเอสวีจีที่ได้มีขนาด 8 กิโลไบต์ ซึ่งเมื่อนำรูปภาพนั้นจากหน้าจอมาแปลงให้อยู่ในรูปแบบเจเพ็ก มีขนาด 63 กิโลไบต์ เมื่อเปรียบเทียบกันแล้วขนาดจะลดลงไปประมาณ 87 เปอร์เซ็นต์



รูปที่ 4.7 ข้อมูลและรูปภาพเส้น

4.1.3 การแสดงผลกราฟวงกลม

เมื่อผู้วิจัยเลือกกราฟประเภทวงกลม ส่วนประกอบกราฟจะใช้แฟ้มข้อกำหนดซึ่งแสดงไว้ดังรูปที่ 4.8 ซึ่งกำหนดจุดเริ่มต้นด้าน X และ Y ขนาดความกว้าง ความยาวของพื้นที่แสดงกราฟ ชื่อกราฟ ชื่อทางด้านแกน X ชื่อทางด้านแกน Y ประเภทของกราฟ ชื่อฐานข้อมูล ชื่อความสังเขยคิวแอล ตำแหน่งที่เก็บแฟ้มข้อมูล และชื่อแฟ้มข้อมูล

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <start startX="0" startY="0" />
  <canvas width="600" height="400" />
  <heading title="First Quarter Sales by Customer" titlex="Month" titley="Quantity" />
  <graphType type="Pie" />
  <db connStr="data source=localhost;initial catalog=svgbase;user id=sa;password=" />
  <sql type="sql">select cv_code as CUSTOMER,jan as JAN,feb as FEB,mar as MAR,apr as
APR from general_statistic where year_no=2004
  </sql>
</db>
  <file path="Result" name="pie" />
</configuration>
```

รูปที่ 4.8 แฟ้มข้อกำหนดสำหรับแสดงกราฟวงกลม

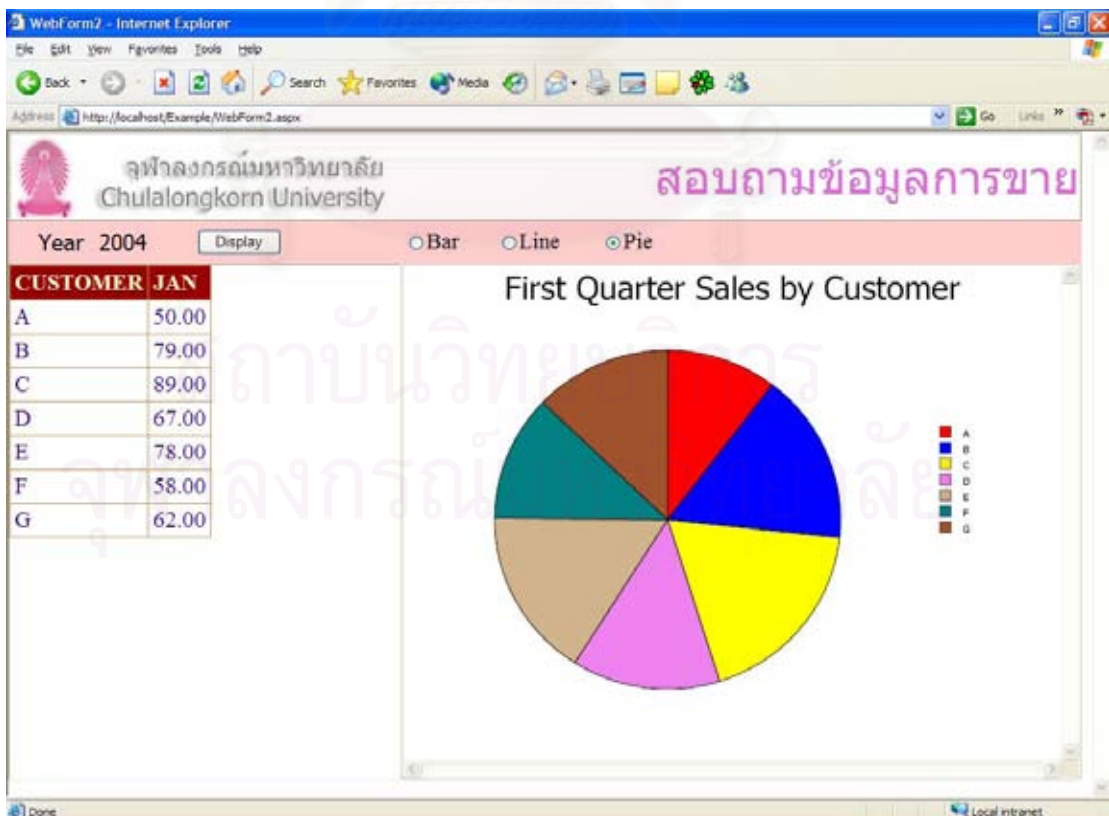
สำหรับการนำส่วนประกอบกราฟไปใช้ทำได้โดยการเพิ่มการอ้างอิง SVGComponent.dll เข้าไปในโพรเจกต์ของวิซวลสตูดิโอเดอทอเน็ต จากนั้นทำการเรียกใช้ ส่วนประกอบกราฟดังรูปที่ 4.9

```
Dim graph As New GraphCtrl
svg = graph.CreateSVG(configPath)
```

รูปที่ 4.9 การเรียกใช้ส่วนประกอบกราฟสำหรับกราฟวงกลมโดยภาษาวิซวลเบสิกเดอทอเน็ต

จากรูปที่ 4.9 จะทำการสร้างอ็อบเจกต์ GraphCtrl แล้วเรียกเมธอด CreateSVG โดยมีพารามิเตอร์เป็นแฟ้มข้อกำหนดซึ่งแสดงไว้ดังรูป 4.8

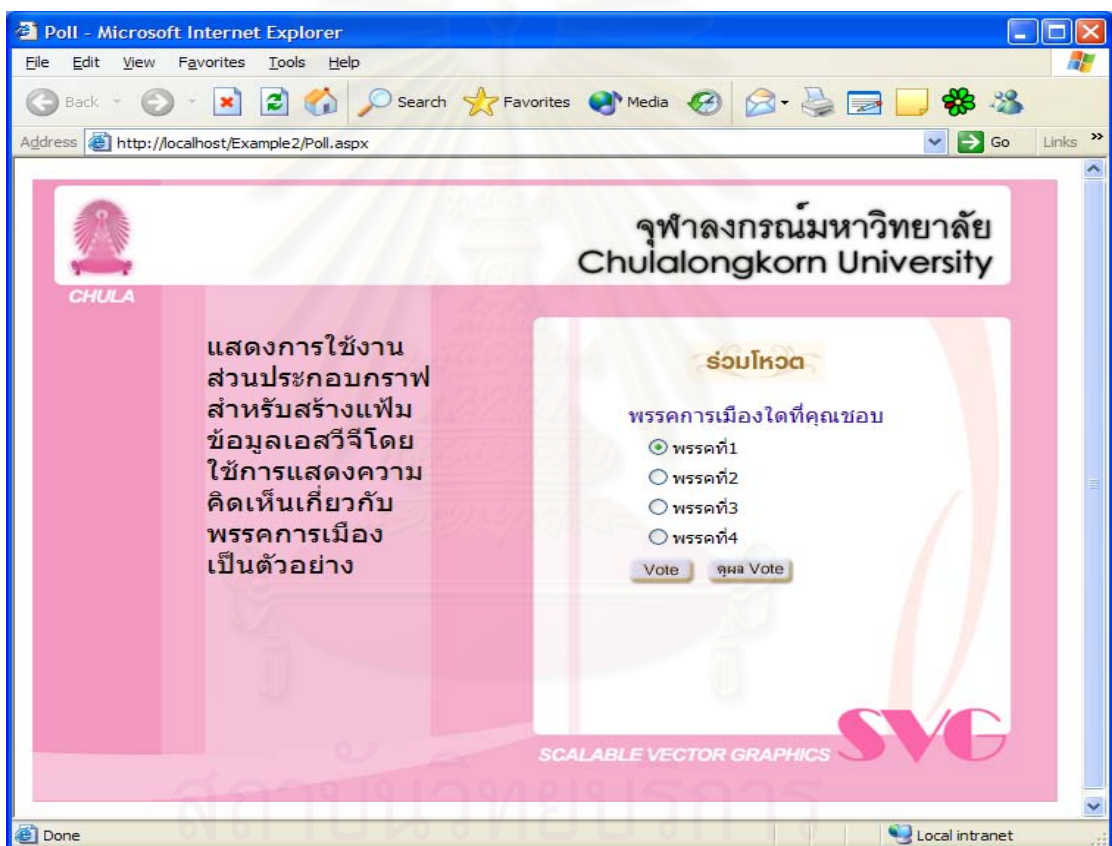
ผลลัพธ์จากการเรียกใช้ส่วนประกอบกราฟจะแสดงได้ดังรูปที่ 4.10 และรูปภาพที่ได้จะถูกบันทึกไว้ที่ตำแหน่งชื่อ Result จากการตรวจสอบพบว่าแฟ้มข้อมูลเอสตีจีที่ได้มีขนาด 4 กิโลไบต์ ซึ่งเมื่อนำรูปภาพนั้นจากหน้าจอมาแปลงให้อยู่ในรูปแบบเจเพ็ก มีขนาด 54 กิโลไบต์ เมื่อเปรียบเทียบกันแล้วขนาดจะลดลงไปประมาณ 92 เปอร์เซ็นต์



รูปที่ 4.10 ข้อมูลและรูปภาพวงกลม

4.2 การทดสอบส่วนประกอบกราฟด้วยภาษาซีชาร์ป

ผู้วิจัยทดสอบการใช้งานส่วนประกอบกราฟสำหรับแสดงผลการสำรวจความนิยมของพรรคการเมือง ได้แก่ พรรคที่1 พรรคที่2 พรรคที่3 และพรรคที่4 โดยดึงข้อมูลจากฐานข้อมูลเอสคิวแอลเซิร์ฟเวอร์ ซึ่งใช้ฐานข้อมูลชื่อ “SVGBASE” เป็นฐานข้อมูลที่ผู้วิจัยสร้างขึ้นเป็นตัวทดสอบ และผู้วิจัยได้สร้างหน้าจอแสดงผลดังรูปที่ 4.11 ซึ่งประกอบด้วย ส่วนของตัวเลือกพรรคการเมืองที่ทำการสำรวจความนิยม



รูปที่ 4.11 หน้าจอสำหรับแสดงความคิดเห็น

ส่วนประกอบกราฟจะใช้แฟ้มข้อกำหนดซึ่งแสดงไว้ดังรูปที่ 4.12 ซึ่งกำหนดจุดเริ่มต้นด้าน X และ Y ขนาดความกว้าง ความยาวของพื้นที่แสดงกราฟ ชื่อกราฟ ชื่อทางด้านแกน X ชื่อทางด้านแกน Y ประเภทของกราฟ ชื่อฐานข้อมูล ชื่อความถี่เอสคิวแอล ตำแหน่งที่เก็บแฟ้มข้อมูล และชื่อแฟ้มข้อมูล

```

<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <start startX="0" startY="0" />
  <canvas width="450" height="400" />
  <heading title="อันดับความนิยมของพรรคการเมือง" titlex="พรรคการเมือง" titley="จำนวน" />
  <graphType type="Bar" />
  <db connStr="data source=localhost;initial catalog=svgbase;user id=sa;password=";>
    <sql type="sql"> select b.party_name,a.score as คะแนน from poll a,party b
                      where a.party_code=b.party_code
  </sql>
</db>
<file path="Result" name="Poll" />
</configuration>

```

รูปที่ 4.12 เพิ่มข้อกำหนดสำหรับแสดงกราฟแท่ง

สำหรับการนำส่วนประกอบกราฟไปใช้ทำได้โดยการเพิ่มการอ้างอิง SVGComponent.dll เข้าไปในโปรเจกต์ของ Visual Studio IDE จากนั้นทำการเรียกใช้ส่วนประกอบกราฟดังรูปที่ 4.13

```

GraphCtrl graph = new GraphCtrl();
svg = graph.CreateSVG(configPath);

```

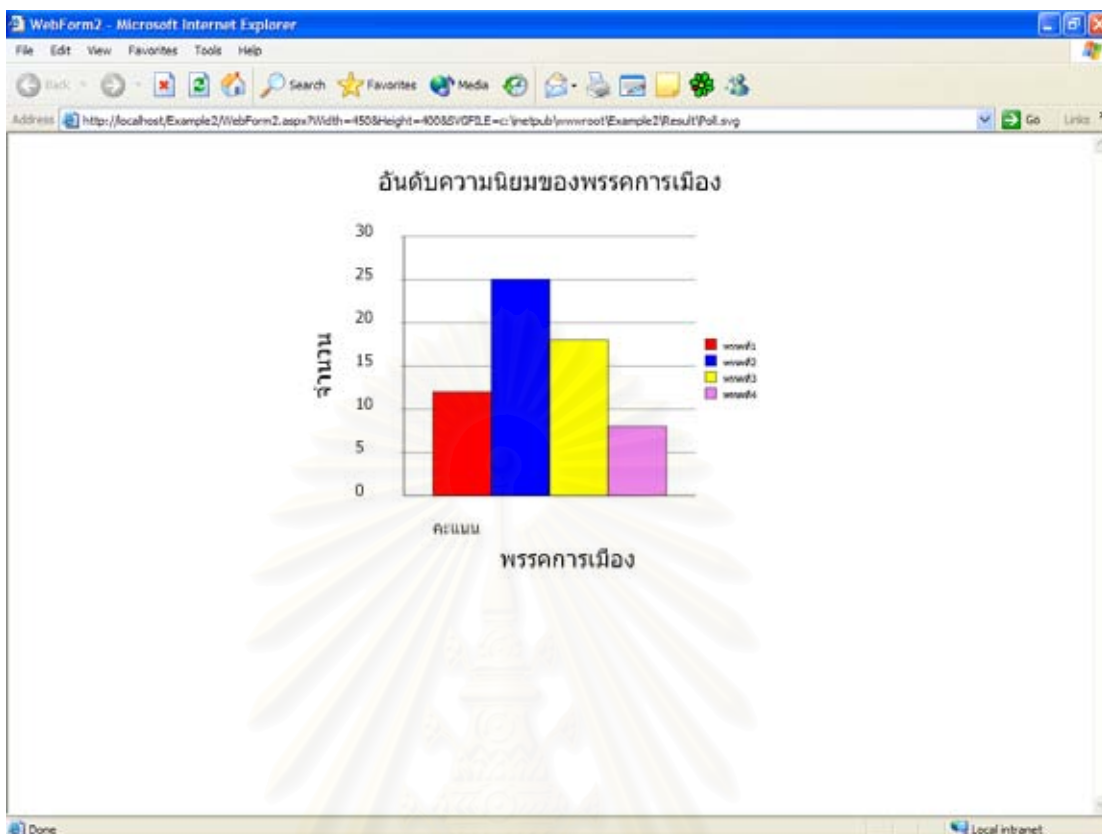
รูปที่ 4.13 การเรียกใช้ส่วนประกอบกราฟสำหรับการสำรวจความนิยมโดยภาษาซีชาร์ป

จากรูปที่ 4.13 จะทำการสร้างอ็อบเจกต์ GraphCtrl แล้วเรียกเมธอด CreateSVG โดยมีพารามิเตอร์เป็นเพิ่มข้อกำหนดซึ่งแสดงไว้ดังรูป 4.12

ผลลัพธ์จากการเรียกใช้ส่วนประกอบกราฟจะแสดงได้ดังรูปที่ 4.14 และรูปภาพที่ได้จะถูกบันทึกไว้ที่ตำแหน่งชื่อ Result จากการตรวจสอบพบว่าเพิ่มข้อมูลเอสวีจีที่มีขนาด 5 กิโลไบต์ ซึ่งเมื่อนำรูปภาพนั้นจากหน้าจอมาแปลงให้อยู่ในรูปแบบเจเพ็ก มีขนาด 42 กิโลไบต์ เมื่อเปรียบเทียบกันแล้วขนาดจะลดลงไปประมาณ 88 เปอร์เซ็นต์

4.3 การวิเคราะห์ผลการทดสอบ

จากการทดสอบการใช้งานส่วนประกอบกราฟด้วยภาษา Visual Basic .NET และภาษาซีชาร์ปพบว่าส่วนประกอบกราฟสามารถนำมาใช้งานและแสดงผลได้อย่างถูกต้อง โดยจะได้เพิ่มข้อมูลเอสวีจีของรูปภาพที่มีขนาดต่างกัน ซึ่งเมื่อเปรียบเทียบขนาดเพิ่มข้อมูลเอสวีจีของรูปภาพในการทดสอบกรณีต่างๆกับเพิ่มรูปแบบเจเพ็กจะได้ผลดังแสดงในตารางที่ 4.1



รูปที่ 4.14 กราฟคะแนนความนิยมของพรรคการเมืองต่างๆ

ตารางที่ 4.1 สรุปขนาดแฟ้มรูปภาพระหว่างรูปแบบเอสวีจีกับรูปแบบเจพีค

กรณี	กราฟแท่ง		กราฟเส้น		กราฟวงกลม	
	รูปแบบเอสวีจี	รูปแบบเจพีค	รูปแบบเอสวีจี	รูปแบบเจพีค	รูปแบบเอสวีจี	รูปแบบเจพีค
1) แสดงข้อมูลยอดขายสินค้าไตรมาสแรกตามลูกค้าของปี 2004 โดยใช้ภาษาวีซวลเบสิกดอทเน็ต	8 KB	73 KB	8 KB	63KB	4 KB	54 KB
2) แสดงผลการสำรวจความนิยมของพรรคการเมือง โดยใช้ภาษาซีชาร์ป	5 KB	42 KB	-	-	-	-

จากตารางที่ 4.1 พบว่าแฟ้มในรูปแบบเอสวีจีมีขนาดเล็กกว่าแฟ้มในรูปแบบเจพีคประมาณ 80-90 เปอร์เซ็นต์

ผู้วิจัยได้ทำการเปรียบเทียบคุณสมบัติของส่วนประกอบกราฟที่สร้างขึ้นโดยเลือกเปรียบเทียบ กับ ChartDirector [12] ซึ่งเป็นส่วนประกอบซอฟต์แวร์สำหรับใช้ในการสร้างกราฟที่มีขายในเชิงพาณิชย์เนื่องจากเป็นส่วนประกอบที่ใช้ได้กับภาษาในมาตรฐานอินเทอร์เน็ตเหมือนกัน การเปรียบเทียบคุณสมบัติของส่วนประกอบกราฟทั้งสองแสดงไว้ในตารางที่ 4.2

ตารางที่ 4.2 เปรียบเทียบคุณสมบัติของส่วนประกอบกราฟ

คุณสมบัติ	ส่วนประกอบกราฟของงานวิจัย	ส่วนประกอบกราฟ ChartDirector
รูปแบบและขนาดของแฟ้มรูปภาพ (สำหรับกราฟวงกลมขนาด 360*300 พิกเซล)	SVG(4KB)	GIF(6KB),PNG(9KB) JPEG(14KB),BMP(317KB)
วิธีการกำหนดรายละเอียดของกราฟ	กำหนดในแฟ้มเอ็กซ์เอ็มแอล	กำหนดในโปรแกรม
ข้อมูลนำเข้า	จากฐานข้อมูลโดยใช้คำสั่งเอสคิวแอล	นำเข้าในรูปแบบอาร์เรย์
ประเภทของกราฟ	กราฟแท่ง,กราฟเส้น,กราฟวงกลม	กราฟแท่ง,กราฟเส้น,กราฟวงกลม,กราฟพื้นที่,กราฟกระจาย,กราฟฟอง และอื่นๆ

จากการเปรียบเทียบคุณสมบัติของส่วนประกอบซอฟต์แวร์ทั้งสองจะเห็นได้ว่าส่วนประกอบกราฟที่พัฒนาขึ้นมีข้อดีและข้อเสียดังนี้

ข้อดี

- 1) รูปแบบแฟ้มกราฟที่ได้อยู่ในรูปแบบเอสวีจีซึ่งมีขนาดเล็กทำให้ใช้เวลาในการส่งข้อมูลไปแสดงบนอินเทอร์เน็ตได้รวดเร็วกว่า นอกจากนี้แฟ้มเอสวีจีสามารถขยายได้โดยไม่เสียความคมชัด
- 2) ในการกำหนดรายละเอียดของกราฟจะกำหนดอยู่ในแฟ้มข้อกำหนดซึ่งอยู่ในรูปแบบเอ็กซ์เอ็มแอลทำให้สามารถแก้ไขรายละเอียดของกราฟที่ต้องการได้ง่าย ไม่จำเป็นต้องแก้ไขในโปรแกรม

3) สามารถนำเข้าข้อมูลเพื่อสร้างกราฟจากฐานข้อมูลได้โดยตรง

ข้อเสีย

1) นำเข้าข้อมูลได้จากฐานข้อมูล ไม่สามารถนำเข้าข้อมูลในรูปแบบอาร์เรย์ได้

2) สามารถแสดงกราฟได้เพียง 3 ประเภท

3) การแสดงรูปภาพต้องมีการติดตั้งเอสวีจีวีวเวอร์บนบราวเซอร์



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 5

บทสรุปปัญหาและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงผลสรุปของงานวิจัย ปัญหาและข้อจำกัดต่าง ๆ รวมทั้งข้อเสนอแนะที่สามารถนำไปปรับปรุงและพัฒนาได้ต่อไปในอนาคต โดยมีรายละเอียดดังต่อไปนี้

5.1 สรุปผลการวิจัย

ในการวิจัยครั้งนี้ผู้วิจัยได้สร้างส่วนประกอบกราฟสำหรับสร้างแฟ้มข้อมูลเอสวีจีด้วยภาษาซีชาร์ป โดยผู้ใช้งานสามารถเลือกแสดงกราฟได้สามประเภท ได้แก่ กราฟแท่ง กราฟเส้น และกราฟวงกลม และสามารถตั้งข้อกำหนดอื่น ๆ สำหรับกราฟที่ต้องการได้โดยบันทึกไว้ในแฟ้มข้อกำหนดในรูปแบบเอ็กซ์เอ็มแอล เมื่อผู้ใช้งานต้องการสร้างกราฟ ส่วนประกอบกราฟจะสร้างกราฟโดยนำข้อมูลมาจากรฐานข้อมูลไมโครซอฟท์เอสคิวแอลเซิร์ฟเวอร์ ตามข้อความสั่งเอสคิวแอลที่ผู้ใช้งานกำหนดไว้ในแฟ้มข้อกำหนด และบันทึกกราฟนั้นไว้ในรูปของแฟ้มเอสวีจี เพื่อให้ผู้ใช้งานนำไปใช้ต่อไป

ส่วนประกอบกราฟนี้สร้างขึ้นตามมาตรฐานดอทเน็ต ซึ่งสามารถเรียกใช้ได้จากภาษาต่างๆ ในมาตรฐานดอทเน็ตเช่นเดียวกัน เช่น ภาษาซีชาร์ป ภาษาวิซวลเบสิกดอทเน็ต เป็นต้น จากการทดสอบการใช้งานด้วยภาษาซีชาร์ปและภาษาวิซวลเบสิกดอทเน็ต พบว่าสามารถเรียกใช้งานได้อย่างสะดวก โดยสามารถสร้างกราฟจากรฐานข้อมูลที่ต้องการได้อย่างถูกต้อง และแฟ้มข้อมูลเอสวีจีของกราฟที่ได้ก็มีขนาดเล็กเมื่อเทียบกับแฟ้มประเภทรูปภาพชนิดอื่นๆ เช่น JPEG เป็นต้น ทำให้ส่วนประกอบกราฟนี้มีความเหมาะสมในการนำไปใช้ในการพัฒนาซอฟต์แวร์บนอินเทอร์เน็ตได้ เช่น การนำส่วนประกอบกราฟไปใช้ในการแสดงข้อมูลทางธุรกิจต่างๆ ในรูปแบบกราฟเพื่อให้ผู้บริหารดูได้ผ่านทางอินเทอร์เน็ต หรือการนำไปใช้ในการแสดงผลลัพธ์ของแบบสอบถามความคิดเห็นในเรื่องต่างๆ เป็นต้น

5.2 ประโยชน์ที่คาดว่าจะได้รับ

- 1) ได้ส่วนประกอบกราฟสำหรับนำข้อมูลจากรฐานข้อมูลเชิงสัมพันธ์มาสร้างเป็นแฟ้มเอสวีจีซึ่งเป็นกราฟในรูปแบบต่างๆ ทั้งกราฟแท่ง กราฟเส้น กราฟวงกลม
- 2) ได้ส่วนประกอบกราฟที่สามารถนำไปเป็นส่วนประกอบของซอฟต์แวร์ได้หลายภาษา
- 3) ได้กราฟที่อยู่บนมาตรฐานเอสวีจี

5.3 ข้อจำกัดของส่วนประกอบกราฟสำหรับสร้างเพิ่มข้อมูลเอสวีจี

- 1) ข้อมูลนำเข้าสำหรับใช้ในส่วนประกอบกราฟจะต้องอยู่ในรูปของตารางซึ่งมีข้อมูลพร้อมสำหรับสร้างกราฟ ในกรณีที่ข้อมูลมีความซับซ้อน ผู้ใช้ต้องแปลงข้อมูลนำเข้านั้นให้อยู่ในรูปพีวอทเทเบิลก่อน
- 2) ส่วนประกอบกราฟสามารถใช้ได้กับภาษาที่สนับสนุนมาตรฐานคอทเน็ตเท่านั้น
- 3) ในการเลือกสีสำหรับแสดงแทนชื่อชุดของข้อมูลต่างๆ ผู้วิจัยได้ออกแบบให้มีการเลือกสีมาจากจำนวนสี 20 สี ดังนั้นผู้ใช้งานจะไม่สามารถเลือกสีได้ หากชื่อของข้อมูลมีมากกว่า 20 ชื่อ ข้อมูลบางข้อมูลต้องใช้สีซ้ำกัน
- 4) ข้อมูลที่ได้จะจำกัดการดึงข้อมูลจากฐานข้อมูลเอสคิวแอลเซิร์ฟเวอร์เท่านั้น
- 5) เมื่อต้องการเพิ่มหรือลดประเภทของกราฟจะต้องมีการแก้ไขส่วนประกอบกราฟนี้ในส่วน of คลาส GraphCtrl และคลาสของกราฟที่ต้องการเพิ่มหรือลดนั้น

5.4 ข้อเสนอแนะ

- 1) เนื่องจากมีระบบฐานข้อมูลอื่นๆ นอกเหนือจากเอสคิวแอลเซิร์ฟเวอร์ เช่น ออราเคิล (Oracle) ไมโครซอฟท์ แอคเซส (Microsoft Access) จึงควรพัฒนาส่วนประกอบกราฟให้สามารถทำงานร่วมกับระบบฐานข้อมูลอื่นๆ ได้
- 2) พัฒนาให้สามารถดึงข้อมูลจากตารางอื่นๆ ที่ไม่ได้อยู่ในรูปฐานข้อมูลได้
- 3) พัฒนาให้มีประเภทกราฟแบบอื่นๆ เพิ่มเติม
- 4) พัฒนาส่วนประกอบกราฟให้มีแบบ 3 มิติ ซึ่งทำให้มองเห็นภาพชัดเจน สวยงามขึ้น และกำลังเป็นที่นิยมในปัจจุบัน

รายการอ้างอิง

- [1] A.W. Brown. Large-Scale,Component –Based Development. NJ: Prentice-Hall,2000.
- [2] I. Jacobson, G. Booch, J. Rumbaugh. The Unified Software Development Process. (n.p.):Pearson Education, 2003.
- [3] Microsoft Corporation. Available from : <http://www.microsoft.com/com> [March 2002]
- [4] Microsoft Corporation. Available from : <http://www.microsoft.com/net> [March 2002]
- [5] Object Management Group. Available from : <http://www.omg.org/corba/whatiscorba.html>[March 2002]
- [6] Sun Microsystem,Inc. Available from : <http://java.sun.com/products/javabeans/> and <http://java.sun.com/products/ejb/> [March 2002]
- [7] M. Laaker. Sams Teach Yourself SVG in 24 Hours.(n.p.): Sams Publishing, 2002.
- [8] Adobe System Incorporated. Adobe SVG Viewer. Available from : <http://www.adobe.com/svg/viewer>[November,2001]
- [9] J. Gabriel, D.Gosnell, Jeffrey Hasan, K.Hoffman, C.Holm, E.Musters, J.Narkiewicz, J.Ortiz, J.Schenken, T.Thangarathinam, S.wylie.Professional .NET Framework. (n.p.): Wrox Press, 2001.
- [10] Sybase Inc. Using SVG to generate Graphs and graphics in a server component. Available from : <http://my.sybase.com/detail?id=1017125>[January, 2002]
- [11] Apache XML Project. Batik SVG Toolkit. Available from : <http://xml.apache.org/batik/index.html>[August ,2002]
- [12] Advanced Software Engineering. Available from <http://www.advsofteng.com/product.html> [March 2005]



ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ตัวอย่างเพิ่มข้อมูลเอสวีจีของกราฟในรูปแบบต่างๆ

เพิ่มข้อมูลเอสวีจีของกราฟแท่งที่ได้จากการส่วประกอบกราฟ

```
<?xml version="1.0" encoding="utf-8"?>
<svg width="600px" height="400px" viewBox="0 0 600 400"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <script type="text/javascript"><![CDATA[
    function showTooltip(evt)
    {
      data = evt.target.getAttribute("data");
      document.getElementById("comment").firstChild.data = data;
      document.getElementById("message").setAttribute("transform","translate(" +
      (evt.clientX+10) + "," + (evt.clientY+10) + ")");
      document.getElementById("message").setAttribute("visibility","visible");
    }

    function hideTooltip(evt)
    {
      document.getElementById("message").setAttribute("visibility","hidden");
    }
  ]]></script>
  <line x1="120" y1="320" x2="480" y2="320" data="" stroke="black" stroke-width="1" />
  <line x1="120" y1="80" x2="120" y2="320" data="" stroke="black" stroke-width="1" />
  <text x="60" y="320" font-family="Tahoma" font-size="15" fill="black" text-
  anchor="start">0</text>
  <text x="60" y="280" font-family="Tahoma" font-size="15" fill="black" text-
  anchor="start">20</text>
  <line x1="117" y1="280" x2="480" y2="280" data="" stroke="black" stroke-width="0.5" />
  <text x="60" y="240" font-family="Tahoma" font-size="15" fill="black" text-
  anchor="start">40</text>
  <line x1="117" y1="240" x2="480" y2="240" data="" stroke="black" stroke-width="0.5" />
  <text x="60" y="200" font-family="Tahoma" font-size="15" fill="black" text-
  anchor="start">60</text>
  <line x1="117" y1="200" x2="480" y2="200" data="" stroke="black" stroke-width="0.5" />
  <text x="60" y="160" font-family="Tahoma" font-size="15" fill="black" text-
  anchor="start">80</text>
  <line x1="117" y1="160" x2="480" y2="160" data="" stroke="black" stroke-widt_="0.5" />
  <text x="60" y="120" font-family="Tahoma" font-size="15" fill="black" text-
  anchor="start">100</text>
  <line x1="117" y1="120" x2="480" y2="120" data="" stroke="black" stroke-width="0.5" />
  <text x="60" y="80" font-family="Tahoma" font-size="15" fill="black" text-
  anchor="start">120</text>
  <line x1="117" y1="80" x2="480" y2="80" data="" stroke="black" stroke-width="0.5" />
  <g id="g1" onmouseover="showTooltip(evt)" onmouseout="hideTooltip(evt)">
    <rect x="130" y="220" width="10" height="100" fill="Red" data="50" stroke="black" stroke-
    width="1" />
    <rect x="220" y="200" width="10" height="120" fill="Red" data="60" stroke="black" stroke-
    width="1" />
    <rect x="310" y="180" width="10" height="140" fill="Red" data="70" stroke="black" stroke-
    width="1" />
  </g>
</svg>
```

```

    <rect x="400" y="160" width="10" height="160" fill="Red" data="80" stroke="black" stroke-
width="1" />
    <rect x="140" y="162" width="10" height="158" fill="Blue" data="79" stroke="black" stroke-
width="1" />
    <rect x="230" y="146" width="10" height="174" fill="Blue" data="87" stroke="black" stroke-
width="1" />
    <rect x="320" y="182" width="10" height="138" fill="Blue" data="69" stroke="black" stroke-
width="1" />
    <rect x="410" y="156" width="10" height="164" fill="Blue" data="82" stroke="black" stroke-
width="1" />
    <rect x="150" y="142" width="10" height="178" fill="Yellow" data="89" stroke="black"
stroke-width="1" />
    <rect x="240" y="164" width="10" height="156" fill="Yellow" data="78" stroke="black"
stroke-width="1" />
    <rect x="330" y="184" width="10" height="136" fill="Yellow" data="68" stroke="black"
stroke-width="1" />
    <rect x="420" y="192" width="10" height="128" fill="Yellow" data="64" stroke="black"
stroke-width="1" />
    <rect x="160" y="186" width="10" height="134" fill="Violet" data="67" stroke="black" stroke-
width="1" />
    <rect x="250" y="182" width="10" height="138" fill="Violet" data="69" stroke="black" stroke-
width="1" />
    <rect x="340" y="174" width="10" height="146" fill="Violet" data="73" stroke="black" stroke-
width="1" />
    <rect x="430" y="164" width="10" height="156" fill="Violet" data="78" stroke="black" stroke-
width="1" />
    <rect x="170" y="164" width="10" height="156" fill="Tan" data="78" stroke="black" stroke-
width="1" />
    <rect x="260" y="204" width="10" height="116" fill="Tan" data="58" stroke="black" stroke-
width="1" />
    <rect x="350" y="184" width="10" height="136" fill="Tan" data="68" stroke="black" stroke-
width="1" />
    <rect x="440" y="130" width="10" height="190" fill="Tan" data="95" stroke="black" stroke-
width="1" />
    <rect x="180" y="204" width="10" height="116" fill="Teal" data="58" stroke="black" stroke-
width="1" />
    <rect x="270" y="190" width="10" height="130" fill="Teal" data="65" stroke="black" stroke-
width="1" />
    <rect x="360" y="164" width="10" height="156" fill="Teal" data="78" stroke="black" stroke-
width="1" />
    <rect x="450" y="224" width="10" height="96" fill="Teal" data="48" stroke="black" stroke-
width="1" />
    <rect x="190" y="196" width="10" height="124" fill="Sienna" data="62" stroke="black"
stroke-width="1" />
    <rect x="280" y="178" width="10" height="142" fill="Sienna" data="71" stroke="black"
stroke-width="1" />
    <rect x="370" y="192" width="10" height="128" fill="Sienna" data="64" stroke="black"
stroke-width="1" />
    <rect x="460" y="196" width="10" height="124" fill="Sienna" data="62" stroke="black"
stroke-width="1" />
</g>
<text x="130" y="355" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">JAN</text>
<text x="220" y="355" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">FEB</text>
<text x="310" y="355" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">MAR</text>
<text x="400" y="355" font-family="Tahoma" font-size="15" fill="black" text-

```

```

anchor="start">APR</text>
  <text x="300" y="37" font-family="Tahoma" font-size="30" fill="black" text-
anchor="middle">First Quarter Sales by Customer</text>
  <text x="313.374137878418" y="386" font-family="Tahoma" font-size="22" fill="black" text-
anchor="start">Month</text>
  <text x="30" y="200" transform="rotate(270,30,200)" font-family="Tahoma" font-size="22"
fill="black" text-anchor="middle">Quantity</text>
  <g id="g2" onmouseover="showTooltip(evt)" onmouseout="hideTooltip(evt)">
    <rect x="492" y="152" width="10" height="10" fill="Red" data="A" stroke="black" stroke-
width="1" />
    <text x="513.6" y="162" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">A</text>
    <rect x="492" y="167" width="10" height="10" fill="Blue" data="B" stroke="black" stroke-
width="1" />
    <text x="513.6" y="177" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">B</text>
    <rect x="492" y="182" width="10" height="10" fill="Yellow" data="C" stroke="black" stroke-
width="1" />
    <text x="513.6" y="192" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">C</text>
    <rect x="492" y="197" width="10" height="10" fill="Violet" data="D" stroke="black" stroke-
width="1" />
    <text x="513.6" y="207" font-family="Tahoma" font-size="9" fill="black" te_t-
anchor="start">D</text>
    <rect x="492" y="212" width="10" height="10" fill="Tan" data="E" stroke="black" stroke-
width="1" />
    <text x="513.6" y="222" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">E</text>
    <rect x="492" y="227" width="10" height="10" fill="Teal" data="F" stroke="black" stroke-
width="1" />
    <text x="513.6" y="237" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">F</text>
    <rect x="492" y="242" width="10" height="10" fill="Sienna" data="G" stroke="black" stroke-
width="1" />
    <text x="513.6" y="252" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">G</text>
  </g>
  <g id="message">
    <text id="comment" x="20" y="13" font-family="Tahoma" text-anchor="end">
</text>
  </g>
</svg>

```

เพิ่มข้อมูลเอสวีจีของกราฟเส้นที่ได้จากการส่วนประกอบกราฟ

```

<?xml version="1.0" encoding="utf-8"?>
<svg width="600px" height="400px" viewBox="0 0 600 400"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <script type="text/javascript"><![CDATA[ function showTooltip(evt) { data =
evt.target.getAttribute("data"); document.getElementById("comment").firstChild.data = data;
document.getElementById("message").setAttribute("transform", "translate(" +
(evt.clientX+10) + "," + (evt.clientY+10) + ")");
document.getElementById("message").setAttribute("visibility", "visible"); } function
hideTooltip(evt) {Xdocument.getElementById("message").setAttribute("visibility", "hidden");

```



```

]]></script>
<line x1="120" y1="320" x2="480" y2="320" data="" stroke="black" stroke-width="1" />
<line x1="120" y1="80" x2="120" y2="320" data="" stroke="black" stroke-width="1" />
<text x="60" y="320" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">0</text>
<text x="60" y="280" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">20</text>
<line x1="117" y1="280" x2="480" y2="280" data="" stroke="black" stroke-width="0.5" />
<text x="60" y="240" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">40</text>
<line x1="117" y1="240" x2="480" y2="240" data="" stroke="black" stroke-width="0.5" />
<text x="60" y="200" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">60</text>
<line x1="117" y1="200" x2="480" y2="200" data="" stroke="black" stroke-width="0.5" />
<text x="60" y="160" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">80</text>
<line x1="117" y1="160" x2="480" y2="160" data="" stroke="black" stroke-width="0.5" />
<text x="60" y="120" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">100</text>
<line x1="117" y1="120" x2="480" y2="120" data="" stroke="black" stroke-width="0.5" />
<text x="60" y="80" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">120</text>
<line x1="117" y1="80" x2="480" y2="80" data="" stroke="black" stroke-width="0.5" />
<g id="g1" onmouseover="showTooltip(evt)" onmouseout="hideTooltip(evt)">
<line x1="165" y1="220" x2="255" y2="200" data="50" stroke="Red" stroke-width="1" />
<line x1="255" y1="200" x2="345" y2="180" data="60" stroke="Red" stroke-width="1" />
<line x1="345" y1="180" x2="435" y2="160" data="70" stroke="Red" stroke-width="1" />
<line x1="435" y1="160" x2="435" y2="160" data="80" stroke="Red" stroke-width="1" />
<line x1="165" y1="162" x2="255" y2="146" data="79" stroke="Blue" stroke-width="1" />
<line x1="255" y1="146" x2="345" y2="182" data="87" stroke="Blue" stroke-width="1" />
<line x1="345" y1="182" x2="435" y2="156" data="69" stroke="Blue" stroke-width="1" />
<line x1="435" y1="156" x2="435" y2="156" data="82" stroke="Blue" stroke-width="1" />
<line x1="165" y1="142" x2="255" y2="164" data="89" stroke="Yellow" stroke-width="1" />
<line x1="255" y1="164" x2="345" y2="184" data="78" stroke="Yellow" stroke-width="1" />
<line x1="345" y1="184" x2="435" y2="192" data="68" stroke="Yellow" stroke-width="1" />
<line x1="435" y1="192" x2="435" y2="192" data="64" stroke="Yellow" stroke-width="1" />
<line x1="165" y1="186" x2="255" y2="182" data="67" stroke="Violet" stroke-width="1" />
<line x1="255" y1="182" x2="345" y2="174" data="69" stroke="Violet" stroke-width="1" />
<line x1="345" y1="174" x2="435" y2="164" data="73" stroke="Violet" stroke-width="1" />
<line x1="435" y1="164" x2="435" y2="164" data="78" stroke="Violet" stroke-width="1" />
<line x1="165" y1="164" x2="255" y2="204" data="78" stroke="Tan" stroke-width="1" />
<line x1="255" y1="204" x2="345" y2="184" data="58" stroke="Tan" stroke-width="1" />
<line x1="345" y1="184" x2="435" y2="130" data="68" stroke="Tan" stroke-width="1" />
<line x1="435" y1="130" x2="435" y2="130" data="95" stroke="Tan" stroke-width="1" />
<line x1="165" y1="204" x2="255" y2="190" data="58" stroke="Teal" stroke-width="1" />
<line x1="255" y1="190" x2="345" y2="164" data="65" stroke="Teal" stroke-width="1" />
<line x1="345" y1="164" x2="435" y2="224" data="78" stroke="Teal" stroke-width="1" />
<line x1="435" y1="224" x2="435" y2="224" data="48" stroke="Teal" stroke-width="1" />
<line x1="165" y1="196" x2="255" y2="178" data="62" stroke="Sienna" stroke-width="1" />
<line x1="255" y1="178" x2="345" y2="192" data="71" stroke="Sienna" stroke-width="1" />
<line x1="345" y1="192" x2="435" y2="196" data="64" stroke="Sienna" stroke-width="1" />
<line x1="435" y1="196" x2="435" y2="196" data="62" stroke="Sienna" stroke-width="1" />
</g>
<text x="165" y="355" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">JAN</text>
<text x="255" y="355" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">FEB</text>
<text x="345" y="355" font-family="Tahoma" font-size="15" fill="black" text-

```

```

anchor="start">MAR</text>
<text x="435" y="355" font-family="Tahoma" font-size="15" fill="black" text-
anchor="start">APR</text>
<text x="300" y="37" font-family="Tahoma" font-size="30" fill="black" text-
anchor="middle">First Quarter Sales by Customer</text>
<text x="313.374137878418" y="386" font-family="Tahoma" font-size="22" fill="black" text-
anchor="start">Month</text>
<text x="30" y="200" transform="rotate(270,30,200)" font-family="Tahoma" font-size="22"
fill="black" text-anchor="middle">Quantity</text>
<g id="g2" onmouseover="showTooltip(evt)" onmouseout="hideTooltip(evt)">
<rect x="492" y="152" width="10" height="10" fill="Red" data="A" stroke="black" stroke-
width="1" />
<text x="513.6" y="162" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">A</text>
<rect x="492" y="167" width="10" height="10" fill="Blue" data="B" stroke="black" stroke-
width="1" />
<text x="513.6" y="177" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">B</text>
<rect x="492" y="182" width="10" height="10" fill="Yellow" data="C" stroke="black" stroke-
width="1" />
<text x="513.6" y="192" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">C</text>
<rect x="492" y="197" width="10" height="10" fill="Violet" data="D" stroke="black" stroke-
width="1" />
<text x="513.6" y="207" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">D</text>
<rect x="492" y="212" width="10" height="10" fill="Tan" data="E" stroke="black" stroke-
width="1" />
<text x="513.6" y="222" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">E</text>
<rect x="492" y="227" width="10" height="10" fill="Teal" data="F" stroke="black" stroke-
width="1" />
<text x="513.6" y="237" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">F</text>
<rect x="492" y="242" width="10" height="10" fill="Sienna" data="G" stroke="black" stroke-
width="1" />
<text x="513.6" y="252" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">G</text>
</g>
<g id="message">
<text id="comment" x="20" y="13" font-family="Tahoma" text-anchor="middle">
</text>
</g>
</svg>

```

เพิ่มข้อมูลเอชวีซีของกราฟวงกลมที่ได้จากการส่วนประกอบกราฟ

```

<?xml version="1.0" encoding="utf-8"?>
<svg width="600px" height="400px" viewBox="0 0 600 400"
xmlns:xlink="http://www.w3.org/1999/xlink">
<script type="text/javascript"><![CDATA[ function showTooltip(evt) { data =
evt.target.getAttribute("data"); document.getElementById("comment").firstChild.data = data;
document.getElementById("message").setAttribute("transform","translate(" +
(evt.clientX+10) + "," + (evt.clientY+10) + ")");

```

```

        document.getElementById("message").setAttribute("visibility","visible"); } function
hideTooltip(evt) {pdocument._etElementById("message").setAttribute("visibility","hidden");
}}></script>
<g id="g1" onmouseover="showTooltip(evt)" onmouseout="hideTooltip(evt)">
  <path d="M240,240L240,80A160,160 0 0,1 336.885001835881,112.668556831938Z"
fill="Red" stroke="black" data="50" />
  <path d="M240,240L336.885001835881,112.668556831938A160,160 0 0,1
399.079451928491,257.138493928316Z" fill="Blue" stroke="black" data="79" />
  <path d="M240,240L399.079451928491,257.138493928316A160,160 0 0,1
288.154281435061,392.581667245685Z" fill="Yellow" stroke="black" data="89" />
  <path d="M240,240L288.154281435061,392.581667245685A160,160 0 0,1
154.214998152554,375.058999914977Z" fill="Violet" stroke="black" data="67" />
  <path d="M240,240L154.214998152554,375.058999914977A160,160 0 0,1
80.0076150812942,238.438984937719Z" fill="Tan" stroke="black" data="78" />
  <path d="M240,240L80.0076150812942,238.438984937719A160,160 0 0,1
124.496756441702,129.279628218144Z" fill="Teal" stroke="black" data="58" />
  <path d="M240,240L124.496756441702,129.279628218144A160,160 0 0,1 240,80Z"
fill="Sienna" stroke="black" data="62" />
</g>
<text x="300" y="32" font-family="Tahoma" font-size="30" fill="black" text-
anchor="middle">First Quarter Sales by Customer</text>
<g id="g2" onmouseover="showTooltip(evt)" onmouseout="hideTooltip(evt)">
  <rect x="492" y="152" width="10" height="10" fill="Red" data="A" stroke="black" stroke-
width="1" />
  <text x="513.6" y="162" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">A</text>
  <rect x="492" y="167" width="10" height="10" fill="Blue" data="B" stroke="black" stroke-
width="1" />
  <text x="513.6" y="177" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">B</text>
  <rect x="492" y="182" width="10" height="10" fill="Yellow" data="C" stroke="black" stroke-
width="1" />
  <text x="513.6" y="192" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">C</text>
  <rect x="492" y="197" width="10" height="10" fill="Violet" data="D" stroke="black" stroke-
width="1" />
  <text x="513.6" y="207" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">D</text>
  <rect x="492" y="212" width="10" height="10" fill="Tan" data="E" stroke="black" stroke-
width="1" />
  <text x="513.6" y="222" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">E</text>
  <rect x="492" y="227" width="10" height="10" fill="Teal" data="F" stroke="black" stroke-
width="1" />
  <text x="513.6" y="237" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">F</text>
  <rect x="492" y="242" width="10" height="10" fill="Sienna" data="G" stroke="black" stroke-
width="1" />
  <text x="513.6" y="252" font-family="Tahoma" font-size="9" fill="black" text-
anchor="start">G</text>
</g>
<g id="messa_e">
  <text id="comment" x="20" y="13" font-family="Tahoma" text-anchor="end">
  </text>
</g>
</svg>

```

ประวัติผู้เขียนวิทยานิพนธ์

นางสาววิมลศรี หอมหวล เกิดเมื่อวันที่ 31 กรกฎาคม พ.ศ. 2517 สำเร็จการศึกษาหลักสูตรสถิติศาสตร์บัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศทางธุรกิจ ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2539 เข้าทำงานที่บริษัทเครือเจริญโภคภัณฑ์ ตำแหน่งโปรแกรมเมอร์ในปี พ.ศ. 2540 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2544



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย