

ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล.

การพัฒนาโปรแกรมสำหรับการค้นหาข้อมูลภายในเอกสารเอ็กซ์เอ็มแอล
ด้วยเอ็กซ์คิวรี่บนเฟรมเวิร์กคือตเน็ต

THE DEVELOPMENT OF XML DATA QUERYING PROGRAM
USING XQUERY ON .NET FRAMEWORK



วัน เดือน ปี..... 21 พ.ค. 2550
เลขทะเบียน..... 03340
เลขเรียกหนังสือ..... 2พ. ๖ 4๖๗๓ 254๙
"ห้องสมุดคณะเทคโนโลยีสารสนเทศ สจล."

6117๕7472
11 ๒๙ ๒๕๕๐

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ภาคเรียนที่ 1 ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**THE DEVELOPMENT OF XML DATA QUERYING PROGRAM
USING XQUERY ON .NET FRAMEWORK**



**A SYSTEM DEVELOPMENT PROJECT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY
FACULTY OF INFORMATION TECNOLOGY
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

1/ 2006

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2006

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การพัฒนาโปรแกรมสำหรับการค้นหาข้อมูลภายในเอกสาร เอ็กซ์เอ็มแอลด้วยเอ็กซ์คิวรีบนเฟรมเวิร์กคือดเน็ต
นักศึกษา	นายปรีชาพล เหมะสุคนธ์
รหัสนักศึกษา	47066121
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2549
อาจารย์ที่ปรึกษา	ผศ.ดร.ภัทรชัย ลลิต โรจนวงศ์

บทคัดย่อ

โปรแกรมสำหรับการค้นหาข้อมูลภายในเอกสารเอ็กซ์เอ็มแอลด้วยเอ็กซ์คิวรีบนเฟรมเวิร์กคือดเน็ต เป็นโปรแกรมประยุกต์ประเภทเอดิเตอร์ที่มีส่วนต่อประสานผู้ใช้เป็นภาพกราฟิกที่พัฒนาขึ้นด้วยเครื่องมือพัฒนาแอปพลิเคชันวิซวลสตูดิโอบนเฟรมเวิร์กคือดเน็ต โดยโปรแกรมนี้พัฒนาขึ้นเพื่อความสะดวกในการทำงานกับเอ็กซ์คิวรี ซึ่งโปรแกรมสามารถให้ผู้ใช้สร้าง แก้ไข ตั้งประมวลผล บันทึก ไวยากรณ์เอ็กซ์คิวรี แสดงโครงสร้างเอกสารเอ็กซ์เอ็มแอล รวมถึงสามารถนำออกผลลัพธ์เป็นเอกสารเอ็กซ์เอ็มแอลเพื่อนำไปใช้ประโยชน์อย่างอื่นต่อไปได้ โดยในการออกแบบและพัฒนาโดยเฉพาะในส่วนการประมวลผลจะมีการเรียกใช้คลาสไลบรารีภายนอกที่มีการนำเข้ามาใช้ตอนเขียน โปรแกรม ซึ่งเป็นคลาสไลบรารีที่พัฒนาโดยบริษัทอัลโทวาและไมโครซอฟต์ เพื่อรับไวยากรณ์เอ็กซ์คิวรีไปประมวลผล และคืนค่ากลับมาให้โปรแกรมนำไปแสดงผล นอกจากนี้ยังออกแบบและพัฒนาให้โปรแกรมสามารถสร้างไวยากรณ์เอ็กซ์คิวรีโดยผู้ใช้ป้อนเองกับการใช้ตัวช่วยสร้างไวยากรณ์เอ็กซ์คิวรี ซึ่งทำให้โปรแกรมเหมาะกับทั้งผู้ใช้ที่มีความรู้ไวยากรณ์เอ็กซ์คิวรี และผู้ใช้ที่ไม่มีความรู้ไวยากรณ์เอ็กซ์คิวรี

Title	The Development of XML Data Querying Program using XQuery on .NET Framework
Student	Mr. Preechapol Hemasukhon
Student ID	47066121
Degree	Master of Science
Programme	Information Science
Academic Year	2006
Advisor	Asst.Prof.Dr. Pattarachai Lalitrojwong

ABSTRACT

The Development of XML Data Querying Program using XQuery on .NET Framework is an editor program which has graphic user interface and all interfaces developed by using Visual Studio .NET Framework Application tool. This program was developed to help user working with XQuery more comfortable. Users will be able to create, modify, processing order, save XQuery, represent structure of XML data and also can produce result to XML document which is very useful result to use for further more. In designing and developing phase especially in processing part, there will be calling procedure for class library from outside. This class library was develop by Altova company and Microsoft. Class library is imported in programming phase to receive XQuery for processing and also return values to the program for executing outcomes. Moreover, this program will be able to create XQuery by letting users input by their own or using XQuery wizard. This purpose make this program comfortable to both users who do not know very well about XQuery and users who know well about XQuery.

กิตติกรรมประกาศ

โครงการพัฒนาระบบงานฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำ และคำปรึกษาจาก ผศ.ดร. ภัทรชัย ลลิตโรจน์วงศ์ ซึ่งเป็นอาจารย์ที่ปรึกษาควบคุมโครงการพัฒนาระบบ ข้าพเจ้ารู้สึกทราบบซึ่งในความอนุเคราะห์จากท่านอาจารย์ และขอขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้า

ขอขอบคุณอาจารย์สุรารักษ์ เรืองวานิช อาจารย์ประจำโปรแกรมวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏจันทรเกษม และคุณกรวิภา เกตุเรืองโรจน์ ที่ช่วยให้คำแนะนำในการเขียนบทคัดย่อภาษาอังกฤษ

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

ขอขอบคุณบัณฑิตศึกษาและบัณฑิตวิทยาลัย คณะเทคโนโลยีสารสนเทศ ที่ให้ความช่วยเหลือในเรื่องต่างๆ

สุดท้ายนี้ข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจ และให้การสนับสนุนในทุกเรื่องๆ ทำให้ข้าพเจ้าสามารถทำโครงการพัฒนาระบบงานฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากโครงการพัฒนาระบบงานฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

ปรีชาพล เหมะสุคนธ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของการพัฒนาระบบงาน.....	2
1.4 ขั้นตอนการดำเนินงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.6 เครื่องมือที่ใช้ในการดำเนินงาน.....	3
บทที่ 2 เทคโนโลยี XMLและภาษา XQuery.....	4
2.1 เทคโนโลยี XML.....	4
2.2 XQuery.....	7
2.3 แบบจำลองกระบวนการทำงานของ XQuery.....	20
บทที่ 3 เครื่องมือที่ใช้ในการพัฒนาโครงการ.....	22
3.1 เทคโนโลยี.NET.....	22
3.2 ชุดคลาสไลบรารี.....	24
3.3 ยูเอ็มแอล.....	28
บทที่ 4 การวิเคราะห์และออกแบบโปรแกรม.....	30
4.1 ความต้องการและข้อจำกัดของโปรแกรม.....	30
4.2 แผนภาพยูสเคส.....	31

สารบัญ (ต่อ)

	หน้า
4.4 แผนภาพคลาส	39
4.5 แผนภาพซีเควนซ์.....	53
4.6 แผนภาพดีพลอยเมนต์	61
บทที่ 5 การออกแบบส่วนต่อประสานกับผู้ใช้	62
5.1 หน้าจอต้อนรับเข้าสู่โปรแกรม.....	62
5.2 หน้าจอหลัก	63
5.3 หน้าจอช่วยสร้างนิพจน์ XQuery.....	67
บทที่ 6 การพัฒนาและทดสอบ โปรแกรม	75
6.1 สถาปัตยกรรมของโปรแกรม.....	75
6.2 เครื่องมือที่ใช้พัฒนาโปรแกรม.....	75
6.3 การทดสอบ โปรแกรม.....	76
บทที่ 7 บทสรุป.....	79
7.1 สรุปโครงการ.....	79
7.2 ปัญหาที่พบและข้อเสนอแนะ.....	80
บรรณานุกรม.....	81
ภาคผนวก.....	83
ภาคผนวก ก. ตัวอย่างเอกสาร XML	83
ภาคผนวก ข. การติดตั้งและการถอดถอนโปรแกรม KMITL Stepz XQuery Editor 1.0	87
ประวัติผู้เขียน	92

สารบัญตาราง

ตารางที่	หน้า
2.1 แกนพื้นฐาน 4 แกน และตัวย่อ	10
2.2 คำมูลที่เป็นไปได้ของ and และ or.....	14
3.1 คลาสต่างๆ ในเนมสเปซ Microsoft.Xml.XQuery	25
3.2 คุณสมบัติและเมธอดในคลาส XQuery ในเนมสเปซ Altova.AltovaXML	27
6.1 ผลการทดสอบประสิทธิภาพของเวลาการทำงานของโปรแกรมค้นหาข้อมูล ภายในเอกสาร XML ด้วย XQuery.....	77
6.2 ค่าคะแนนเฉลี่ยของความพึงพอใจของผู้ใช้งานโปรแกรมค้นหาข้อมูลภายใน เอกสาร XML ด้วย XQuery	78



สารบัญภาพ

ภาพที่	หน้า
2.1 คุณลักษณะของเอกสาร XML.....	5
2.2 หลักการทำงานของ DOM	7
2.3 ความสัมพันธ์ระหว่างข้อกำหนดต่างๆ ของ XML	7
2.4 รูปแบบของนิพจน์	8
2.5 โครงสร้างของนิพจน์เส้นทาง	9
2.6 แบบจำลองการทำงานของ XQuery	21
3.1 สถาปัตยกรรมเฟรมเวิร์ก .NET	22
3.2 รูปแบบการทำงานของชุดคลาสิกภายในเนมสเปซ Microsoft.Xml.XQuery.....	26
4.1 แผนภาพยูสเคส	32
4.2 แผนภาพแอกทิวิตีของการสร้างไวยากรณ์ XQuery	33
4.3 แผนภาพแอกทิวิตีของการสร้างไวยากรณ์ XQuery ด้วยตัวช่วยสร้าง	35
4.4 แผนภาพแอกทิวิตีของการแสดงโครงสร้างของเอกสาร XML	36
4.5 แผนภาพแอกทิวิตีของการประมวลผลไวยากรณ์ XQuery.....	37
4.6 แผนภาพแอกทิวิตีของการอัปเดตไวยากรณ์ XQuery.....	38
4.7 แผนภาพแอกทิวิตีของการสร้างเอกสาร XML จากผลลัพธ์	39
4.8 แผนภาพคลาสของโปรแกรม.....	40
4.9 แผนภาพคลาสของแพ็คเกจ KMITL.....	40
4.10 แผนภาพคลาสของแพ็คเกจ Stepz.....	41
4.11 แผนภาพคลาสของแพ็คเกจ Base.....	42
4.12 แผนภาพคลาสของแพ็คเกจ Editor.....	45
4.13 แผนภาพคลาสของแพ็คเกจ Wizard.....	48
4.14 แผนภาพซีเควนซ์การสร้างหน้าจอดีเตอร์ใหม่.....	53
4.15 แผนภาพซีเควนซ์การเปิดไฟล์	54
4.16 แผนภาพซีเควนซ์การบันทึกไวยากรณ์ XQuery	55
4.17 แผนภาพซีเควนซ์การค้นหาข้อมูล	56
4.18 แผนภาพซีเควนซ์การแสดงผลโครงสร้างของเอกสาร XML.....	57
4.19 แผนภาพซีเควนซ์การกำหนดโหนดเริ่มต้นในการค้นหาในหน้าจอช่วยสร้าง	58

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.20 แผนภาพซีเควนซ์การสร้างเงื่อนไขในหน้าจอช่วยสร้าง	59
4.21 แผนภาพซีเควนซ์การนำออกผลลัพธ์	60
4.22 แผนภาพดีพลอยเมนต์	61
5.1 หน้าจอหลักและแท็บต้อนรับเข้าสู่โปรแกรม	62
5.2 ส่วนประกอบหน้าจอหลักของโปรแกรม	63
5.3 แท็บสำหรับเลือกเอกสาร XML	64
5.4 แท็บแสดงโครงสร้างเอกสาร XML	64
5.5 แท็บจัดการเลย์เอาต์ของโปรแกรม	64
5.6 แท็บจัดการลักษณะต่างๆ และพฤติกรรมของโปรแกรม	65
5.7 แท็บกำหนดส่วนประมวลผลไวยากรณ์ XQuery	65
5.8 แท็บแสดงผลลัพธ์แบบโครงสร้างต้นไม้	66
5.9 แท็บแสดงผลลัพธ์แบบตัวอักษรธรรมดา	66
5.10 แท็บแสดงผลลัพธ์แบบเปิดผ่านอินเทอร์เน็ตเอ็กซ์พลอเรอร์	66
5.11 แท็บแสดงข้อผิดพลาด	67
5.12 แท็บแสดงลำดับเหตุการณ์ขณะทำงานกับโปรแกรม	67
5.13 แถบสถานะ	67
5.14 หน้าจอช่วยสร้างนิพจน์ XQuery	68
5.15 ไดอะล็อกสำหรับกำหนดโหนดที่จะใช้เริ่มต้นในการค้นหา	69
5.16 หน้าจอสำหรับเลือกเอกสาร XML หลังเลือกเอกสาร XML แล้ว	69
5.17 หน้าจอสำหรับเลือกโหนดที่ต้องการค้นหาพร้อมแสดงลิสต์โหนดที่ผู้ใช้เลือก	70
5.18 ไดอะล็อกสำหรับการกำหนดเงื่อนไข	71
5.19 ไดอะล็อกถามว่าผู้ใช้ไม่มีเงื่อนไขอื่นๆ อีกใช่หรือไม่	71
5.20 ไดอะล็อกให้ผู้ใช้กำหนดนิพจน์ Quantified	71
5.21 หน้าจอสำหรับกำหนดเงื่อนไขพร้อมแสดงเงื่อนไขที่ผู้ใช้กำหนด	72
5.22 หน้าจอสำหรับกำหนดการเรียงลำดับและรีเทิร์นรูตเอลิเมนต์	73
5.23 หน้าจอจบการใช้ตัวช่วยสร้างนิพจน์ XQuery	73
5.24 แท็บเอกสารที่ถูกสร้างขึ้นจากการใช้ตัวช่วยสร้างนิพจน์ XQuery	74

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
5.25 หน้าจอช่วยสร้างนิพจน์ XQuery พร้อมกับหน้าจอแสดงข้อมูลที่ถูกเลือกก่อนหน้า	74
ข.1 ไดอะล็อกเริ่มต้นการติดตั้งโปรแกรม	87
ข.2 ไดอะล็อกระบุตำแหน่งการติดตั้งโปรแกรม	88
ข.3 ไดอะล็อกยืนยันขั้นการติดตั้งโปรแกรม	88
ข.4 ไดอะล็อกสถานะการดำเนินการติดตั้งโปรแกรม	89
ข.5 ไดอะล็อกการติดตั้งโปรแกรมสมบูรณ์	89
ข.6 ไดอะล็อกการเพิ่มและการถอดถอนโปรแกรมบนวินโดวส์	90
ข.7 ไดอะล็อกรายละเอียดของโปรแกรม KMITL Stepz Editor 1.0	90
ข.8 ไดอะล็อกยืนยันขั้นการถอดถอนโปรแกรม	91
ข.9 ไดอะล็อกสถานะการดำเนินการถอดถอนโปรแกรม	91

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

แวดวงคอมพิวเตอร์ปัจจุบันนี้ ภาษามาร์กอัปอย่าง XML เป็นภาษาที่มีบทบาทเพิ่มขึ้น เนื่องจากด้วยศักยภาพของ XML ที่มีมาตรฐาน มีความยืดหยุ่นสูง อีกทั้งยังสามารถอธิบายข้อมูลหรือสารสนเทศได้ทุกชนิด รวมถึงการนำ XML มาอธิบายข้อมูลคล้ายฐานข้อมูล ทำให้สะดวกในการจัดการข้อมูลมากยิ่งขึ้น และปัจจุบันนี้ภาษาที่ใช้ในการพัฒนาแอปพลิเคชันหลายภาษาที่รองรับการทำงานกับ XML ยิ่งทำให้ภาษาอย่าง XML ได้รับความสนใจมากขึ้นไปอีก ดังนั้น จึงไม่เป็นการยากที่จะเห็น XML ถูกใช้ในหลายๆ รูปแบบและกว้างขวาง

เมื่อมาตรฐานอย่าง XML ถูกหยิบยกรมาใช้มากขึ้น ข้อมูลหรือสารสนเทศที่ถูกอธิบายด้วย XML ก็มีขนาดหรือจำนวนบรรทัดที่มากขึ้นเช่นกัน ซึ่งไม่ต่างอะไรกับการมีฐานข้อมูลที่มีข้อมูลในตารางจำนวนมากแล้วมีการเกิดขึ้นของภาษา SQL ที่ใช้สำหรับสืบค้นข้อมูลในฐานข้อมูล ดังนั้นจึงเกิดเป็นความต้องการใหม่ที่จะสืบค้นข้อมูลภายใน XML ซึ่งองค์กรที่พัฒนาและออกข้อกำหนดต่างๆ ของ XML คือ World Wide Web Consortium (W3C) จึงได้คิดค้นภาษาที่ใช้สำหรับสืบค้นข้อมูลจาก XML ที่ชื่อ XQuery ขึ้น

ในปัจจุบันได้เริ่มมีบุคลากรที่สนใจ องค์กรและบริษัทต่างๆ ได้พัฒนาซอฟต์แวร์ที่เป็นเครื่องมือสำหรับ XQuery ขึ้นมากมายพอสมควร แต่ก็ยังใช้งานไม่สะดวกมากนัก เนื่องจากบางซอฟต์แวร์ยังต้องใช้คำสั่งค่อนข้างยุ่งยาก ซึ่งอาจเป็นเรื่องยากสำหรับผู้ไม่ถนัดคำสั่งบนคอส แต่ก็ยังมีการพัฒนาซอฟต์แวร์ดังกล่าวออกมาในอีกรูปแบบหนึ่งคือ เป็นคลาสไลบรารีให้สำหรับผู้มีความรู้ด้านโปรแกรมมิ่งสามารถนำไปพัฒนาต่อยอดได้ ดังนั้น ผู้จัดทำจึงมีแนวความคิดที่จะพัฒนาโปรแกรมสำหรับการค้นหาข้อมูลภายใน XML ด้วยไวยากรณ์ XQuery โดยอาศัยคลาสไลบรารีในการพัฒนาเป็นโปรแกรมแบบเอดิเตอร์เพื่อรับไวยากรณ์ XQuery จากผู้ใช้ ซึ่งสามารถพิมพ์ผ่านหน้าจอหรืออาศัยตัวช่วยสร้างในการสร้างไวยากรณ์ XQuery

1.2 วัตถุประสงค์ของโครงการ

วัตถุประสงค์ในการพัฒนาระบบงาน มีดังนี้

1. เพื่อศึกษาเทคโนโลยี XML ที่มีใช้งานในปัจจุบัน
2. เพื่อศึกษาไวยากรณ์ XQuery สำหรับเป็นแนวทางในการพัฒนาโปรแกรมในส่วนตัว

ช่วยสร้างไวยากรณ์ XQuery

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เพื่อศึกษาการวิเคราะห์และออกแบบโดยใช้ UML (Unified Modeling Language)
4. เพื่อศึกษาการพัฒนาโปรแกรมบนเฟรมเวิร์ก .NET
5. เพื่อศึกษาวิธีการเขียน โปรแกรมเชิงวัตถุโดยใช้ภาษา VB.NET

1.3 ขอบเขตของการพัฒนาระบบงาน

ขอบเขตของการพัฒนาระบบงาน มีดังนี้

1. สามารถพัฒนาเป็น โปรแกรมที่ติดตั้งบนเครื่องผู้ใช้ได้
2. สามารถสร้างไวยากรณ์ XQuery บนเอดิเตอร์ที่กำหนด หรือจากตัวช่วยสร้างได้
3. สามารถส่งประมวลผลไวยากรณ์ XQuery ด้วยคลาสไลบรารีที่กำหนดได้
4. สามารถค้นหาข้อมูลจาก XML ที่อยู่ในเครื่องผู้ใช้ หรือผ่านทาง URL ได้
5. สามารถบันทึกเป็นไฟล์นามสกุล XQuery ไว้เปิดใช้งานในภายหลังได้
6. สามารถบันทึกผลลัพธ์เป็นไฟล์นามสกุล XML เพื่อนำไปใช้ประโยชน์อย่างอื่นได้

1.4 ขั้นตอนการดำเนินงาน

ขั้นตอนการดำเนินงาน มีดังนี้

1. ศึกษาเทคโนโลยี XML และเทคโนโลยีที่เกี่ยวข้อง
2. ศึกษาไวยากรณ์ XQuery
3. ศึกษาการพัฒนาโปรแกรมด้วย VB บนเฟรมเวิร์ก .NET
4. วิเคราะห์และออกแบบโดยใช้ UML
5. พัฒนาโปรแกรมด้วย VB บนเฟรมเวิร์ก .NET
6. ทดลองติดตั้งโปรแกรมบนเครื่องผู้ใช้จริง
7. ทดสอบประสิทธิภาพและประสิทธิผลของโปรแกรม

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่คาดว่าจะได้รับเมื่อพัฒนาระบบงานนี้ มีดังนี้

1. ผู้ใช้สามารถสร้าง ประมวลผล และจัดการไวยากรณ์ XQuery ได้ง่ายและสะดวกมากขึ้นในโปรแกรมเดียว
2. ผู้ใช้ไม่ต้องยุ่งยากกับการเขียนคำสั่งบนคอส
3. ไม่ต้องยุ่งยากเกี่ยวกับการจัดการติดตั้งส่วนประมวลผล XQuery เพราะโปรแกรมได้มี

เอกสารนี้เป็นเอกสารผู้รวบรวมเข้ากับโปรแกรมเรียบร้อยแล้ว เมื่อติดตั้งสมบูรณ์ กรุณาให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ประหยัดเวลาในการทำงาน ไม่ต้องทำงานกับโปรแกรมหลายๆ โปรแกรมพร้อมกัน
5. ผู้ใช้สามารถสร้างนิพจน์ FLWOR อย่างง่ายได้ผ่านหน้าจอช่วยสร้าง ซึ่งเหมาะกับผู้ใช้ที่ไม่มีความรู้ไวยากรณ์ XQuery

1.6 เครื่องมือที่ใช้ในการดำเนินงาน

เครื่องมือที่ใช้ในการดำเนินงานพัฒนาระบบงาน มีดังนี้

1. Microsoft Visual Studio 2005 Professional Edition เป็นชุดเครื่องมือสำหรับพัฒนาแอปพลิเคชันภายใต้ .NET โดยเป็นโปรแกรมประเภท IDE (Integrated Development Environment) ซึ่งจะประกอบไปด้วยเครื่องมือต่างๆ ที่ช่วยอำนวยความสะดวกในเขียนโปรแกรม ตั้งแต่การออกแบบหน้าจอ เขียนโปรแกรม คอมไพล์ ตรวจสอบความถูกต้องของโปรแกรม และเครื่องมืออื่นๆ ที่จำเป็น ซึ่งโปรแกรมที่จะพัฒนาขึ้นจะมีการพัฒนาบนเครื่องมือนี้
2. ชุดคลาสไลบรารี Altova XML 2006 เป็นชุดคลาสไลบรารีที่จัดการเกี่ยวกับ XML คือ XML Validator, XQuery และ XSLT ของบริษัท Altova GmbH โดยในส่วนของ XQuery ได้มีการพัฒนาตามข้อกำหนดในเอกสารฉบับวันที่ 3 พฤศจิกายน พ.ศ.2548 ขององค์กร W3C ซึ่งโปรแกรมที่จะพัฒนาขึ้น จะมีการเขียนโปรแกรมเพื่อเรียกใช้คลาสที่อยู่ภายในชุดคลาสไลบรารีนี้
3. ชุดคลาสไลบรารี Microsoft Xml Xquery เป็นชุดคลาสไลบรารีสำหรับจัดการกับ XQuery ที่ไมโครซอฟต์พัฒนาขึ้น โดยได้มีการพัฒนาตามข้อกำหนดในเอกสารฉบับร่างวันที่ 7 มิถุนายน พ.ศ.2544 ขององค์กร W3C ซึ่งโปรแกรมที่จะพัฒนาขึ้น จะมีการเขียนโปรแกรมเพื่อเรียกใช้คลาสที่อยู่ภายในชุดคลาสไลบรารีนี้
4. Poseidon for UML Professional Edition 4.1 เป็นเคสทูลไว้สำหรับเขียนแผนภาพ UML ทั้งแผนภาพยูสเคส แผนภาพแอกทิวิตี แผนภาพคลาส แผนภาพซีเควนซ์ และอื่นๆ

บทที่ 2

เทคโนโลยี XML และภาษา XQuery

การพัฒนากระบวนการนี้เป็นการทำงานกับข้อมูลหรือสารสนเทศที่อยู่ในรูปแบบของ XML ดังนั้น ในบทนี้จะเป็นการศึกษาเกี่ยวกับเทคโนโลยี XML ทฤษฎีอื่นที่เกี่ยวข้อง รวมทั้งศึกษาไวยากรณ์ XQuery เพื่อนำไปใช้ในการพัฒนากระบวนการต่อไป

2.1 เทคโนโลยี XML

XML ย่อมาจาก Extensible Markup Language ถูกพัฒนาโดย W3C (The World Wide Web Consortium) เป็นภาษามาร์กอัป (Markup) เช่นเดียวกับ HTML ประกอบด้วยโค้ดหรือที่เรียกว่าแท็ก (Tag) เป็นพื้นฐานสำคัญที่ใช้เพิ่มเข้าไปในข้อความ ในการปรับเปลี่ยนรูปแบบมุมมองหรือความหมาย

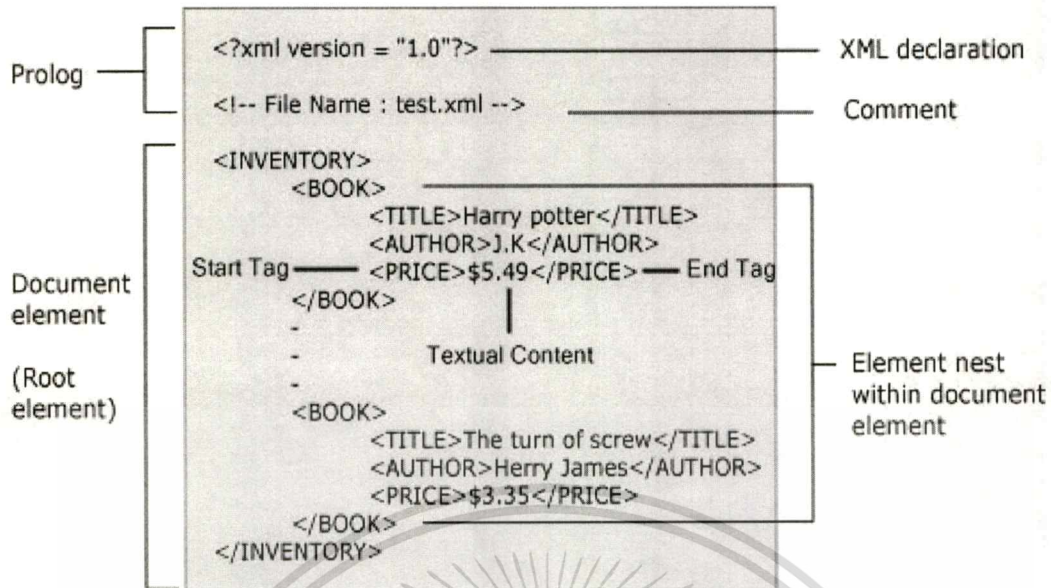
XML เป็นอภิภาษา (Meta-Language) นั่นคือ XML เป็นภาษาที่ใช้อธิบายภาษาอื่น การกำหนดเอลิเมนต์ (Element) สามารถกำหนดได้ตามความต้องการของผู้พัฒนา เพื่อให้สอดคล้องกับโปรแกรมการทำงานของพัฒนาเอง (รหัสสากลสภาอุตสาหกรรมแห่งประเทศไทย. 2546)

2.1.1 คุณลักษณะต่างๆ ของ XML

XML เป็นเอกสารที่เขียนด้วยข้อความปกติธรรมดา จึงสามารถสร้างเอกสารหรือแก้ไขไฟล์ XML ได้ง่ายดาย โดยการใช้บรรณาธิกรณข้อความหรือเอดิเตอร์ ซึ่งเป็นมากกว่าเครื่องมือการเขียนโปรแกรมที่ไม่ซับซ้อน หรือถ้าหากต้องการใช้โปรแกรมที่มีความสามารถพิเศษมากกว่านี้ ก็ต้องใช้บรรณาธิกรณข้อความ เช่น Stylus Studio หรือ XML Altova เป็นต้น

ส่วนประกอบในเอกสาร XML มีอยู่ 2 ส่วนหลักด้วยกัน คือ โปรล็อกเอลิเมนต์ (Prolog Element) และ ค็อกคิวเมนต์เอลิเมนต์ (Document Element) หรือรูตเอลิเมนต์ ในส่วนของเอกสาร XML คือ เอลิเมนต์เดี่ยว ซึ่งสามารถบรรจุเอลิเมนต์เพิ่มเติมในเอกสาร XML ได้ โดยในเอกสาร XML นั้น เอลิเมนต์จะแสดงลักษณะโครงสร้างของเอกสาร และจะแสดงส่วนประกอบเนื้อหาของเอกสารอยู่ในสัญลักษณ์เอลิเมนต์ ซึ่งประกอบด้วย แท็กเริ่มต้น เนื้อหาภายในเอลิเมนต์ และแท็กสิ้นสุด ส่วนเนื้อหาภายในเอลิเมนต์สามารถเป็นได้ทั้งข้อมูลหรือเอลิเมนต์อื่นๆ ที่ซ่อนอยู่ภายในหรือทั้งสองแบบ (เกียรติศักดิ์ อนุธรรม. 2548) ดังภาพที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 คุณลักษณะของเอกสาร XML (รหัสสากลสภาอุตสาหกรรมแห่งประเทศไทย. 2546)

2.1.2 รูปแบบของเอกสาร XML

เอกสาร XML แบ่งได้เป็น 2 รูปแบบ ดังนี้ (ประณต บุญไชยอภิสิทธิ์. 2546 : 37)

2.1.2.1 เอกสารรูปแบบถูกต้อง (Well-Formed)

เอกสาร XML ที่มีความถูกต้องตามกับกฎเกณฑ์พื้นฐานของภาษา XML และมีความสอดคล้องกับข้อบังคับของการเป็นเอกสารรูปแบบถูกต้องที่เรียกว่า Well-Formedness Constraint

ต่อไปนี้เป็นกฎของเอกสาร XML ที่มีรูปแบบถูกต้อง (กิตติชัย พินิจคำ. 2544 : 61)

- เอลิเมนต์ต้องประกอบด้วยแท็กเปิดและแท็กปิด
- เอกสารจะต้องมีรูตเอลิเมนต์เพียงหนึ่งเดียวเท่านั้น
- ชื่อแอตทริบิวต์หรือเอลิเมนต์ไวต่ออักษรใหญ่เล็ก (Case-Sensitive)
- เอลิเมนต์จะต้องเข้าคู่กันพอดี
- ไม่ควรใช้อักขระบางตัว หากจำเป็นให้ใช้ลักษณะการผสมอักขระแทน ที่เรียกว่า Entity

References ซึ่งมีจำนวน 5 ตัวคือ < (<), > (>), & (&), ' ('), " (")

- ค่าของแอตทริบิวต์ต้องอยู่ในเครื่องหมายคำพูด
- เอลิเมนต์ว่าง (Empty Element) จะมีรูปแบบเฉพาะให้ใช้ คือ <element-name/>

2.1.2.2 เอกสารสมเหตุสมผล (Valid)

เอกสารสมเหตุสมผลนั้น นอกจากจะต้องเป็นเอกสารรูปแบบถูกต้องแล้ว ยังต้องมีความถูกต้องและมีความสอดคล้องกันของโครงสร้างเอกสาร XML กับการประกาศโครงสร้างทางตรรกะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของเอกสาร XML และมีความสอดคล้องกับข้อบังคับของการเป็นเอกสารสมเหตุสมผลที่เรียกว่า Validity Constraint

การทำให้เอกสาร XML มีลักษณะสมเหตุสมผลสามารถทำได้ 2 วิธีคือ DTD (Document Type Definition) และ XML Data Schema โดยทั้ง 2 วิธีเป็นข้อตกลงที่กำหนดขึ้น เพื่อใช้ตรวจสอบและเป็นข้อบังคับให้ข้อมูลต่างๆ ที่จัดเก็บในแต่ละแท็ก แต่ละเอลิเมนต์ให้ข้อมูลมีความถูกต้อง และเป็นไปตามที่ผู้ออกแบบเอกสาร XML ต้องการ (ศุภชัย สมพานิช. 2544 : 100)

2.1.3 ตัววิเศษหรือประมวลผล XML (XML Parser/Processor)

ตัววิเศษหรือประมวลผล XML ใช้สำหรับอ่านเอกสาร XML และเข้าถึงข้อมูลและโครงสร้างต่างๆ ของเอกสาร โดยที่ตัววิเศษหรือประมวลผล XML ทำงานเป็นตัวแทนให้กับอีกมอดูลหนึ่งที่เรียกว่า แอปพลิเคชัน ซึ่งการทำงานจะกระจายโครงสร้างของเอกสาร XML ออกมาแล้วส่งข้อมูลต่างๆ จากเอกสารไปยังแอปพลิเคชัน (สุวรรณ สุขสมจินต์. 2545 : 85) ซึ่งสามารถแบ่งตัววิเศษหรือประมวลผล XML ออกเป็น 2 ประเภทใหญ่ๆ คือ

2.1.3.1 Non-Validating Parser และ Validating Parser

Non-Validating Parser เป็นตัววิเศษที่มีความสามารถในการตรวจสอบความถูกต้องของการเป็นเอกสารรูปแบบถูกต้องเท่านั้น ส่วน Validating Parser เป็นตัววิเศษที่มีความสามารถในการตรวจสอบความถูกต้องของการเป็นเอกสารรูปแบบถูกต้องและ เอกสารสมเหตุสมผล

2.1.3.2 Tree-Based Parser และ Event-Based Parser

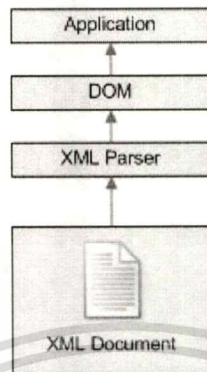
ตัววิเศษที่ทำงานแบบ Tree-Based Parser คือ DOM (Document Object Model) ส่วนตัววิเศษที่ทำงานแบบ Event-Based Parser คือ SAX (Simple API for XML) (ประณต บุญไชยอภิสิทธิ์. 2546 : 39)

2.1.4 Document Object Model (DOM)

DOM เป็นวิธีที่ประมวลโครงสร้างของเอกสาร XML ให้เป็นโครงสร้างแบบต้นไม้ เพื่อให้แอปพลิเคชันสามารถเข้าหาจุดต่างๆ ของโครงสร้างต้นไม้ได้ โดยที่ DOM จะโหลดไฟล์ XML ทั้งไฟล์ในหน่วยความจำซึ่งก็มีทั้งข้อดีและข้อเสีย ข้อดี คือครั้งแรกสุดที่มีการเรียกใช้งานจะช้า แต่หลังจากนั้นการเข้าถึงจุดต่างๆ ของไฟล์จะเร็วเพราะถูกเก็บในหน่วยความจำแล้ว ข้อเสีย คือ ไม่เหมาะกับ XML ที่มีขนาดใหญ่ โดยที่ขอบเขตจะถูกจำกัดด้วยหน่วยความจำที่เรียกใช้ในขณะนั้น

DOM มักจะถูกใส่เข้าไปเป็นชั้นที่คั่นระหว่างตัววิเศษหรือประมวลผล XML กับแอปพลิเคชันที่ต้องการข้อมูลในเอกสาร นั้นหมายความว่า ตัววิเศษจะอ่านข้อมูลจากเอกสาร XML แล้วป้อนข้อมูลนั้นให้กับ DOM จากนั้น DOM ก็จะถูกใช้โดยแอปพลิเคชันระดับที่สูงกว่า ดังภาพที่ 2.2 แอปเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พลีเคชันสามารถทำอะไรที่มันต้องการกับข้อมูลดังกล่าวก็ได้ รวมถึงการใส่มันเข้าไปในโมเดลเชิงวัตถุอื่นๆ ถ้ามันต้องการเช่นนั้น (สุวรรณา สุขสมจินตน์. 2545 : 263)

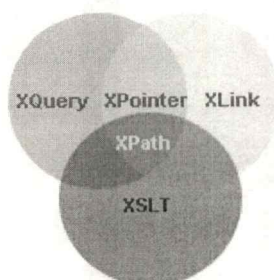


ภาพที่ 2.2 หลักการทำงานของ DOM (สุวรรณา สุขสมจินตน์. 2545 : 263)

2.2 XQuery

XQuery คือภาษาที่ใช้ในการค้นหาข้อมูลภายในเอกสาร XML ถูกคิดค้นและพัฒนาโดยองค์กร W3C เพื่อให้เป็นมาตรฐานสำหรับการค้นหาข้อมูลภายในเอกสาร XML การออกแบบ XQuery นั้น W3C ได้ออกแบบให้เป็นภาษาสำหรับคิวรีที่รัดกุม ง่ายต่อการทำความเข้าใจ มีความยืดหยุ่นพอสำหรับการคิวรีเอกสาร XML มากกว่า 1 เอกสาร โดยที่เอกสาร XML ไม่จำเป็นที่จะต้องจัดเก็บอยู่ที่เดียวกัน และได้ออกแบบให้มีรูปแบบภาษาทั้งที่อยู่ในรูปของ XML (XML-Based Query Syntax) ที่เรียกว่า XQueryX และไม่อยู่ในรูปของ XML (Non-XML Query Syntax) หรือมีรูปแบบภาษาที่มนุษย์สามารถอ่านเข้าใจได้ ซึ่งก็คือ XQuery ที่จะได้อธิบายในรายงานฉบับนี้ การพัฒนา XQuery นั้นได้รับการพัฒนามาจากภาษาคิวรี XML ที่เรียกว่า Quilt ที่ได้ยืมคุณลักษณะต่างๆ เช่น โครงสร้างภาษา รูปแบบการเข้าถึง เป็นต้น มาจากภาษาอื่นๆ อีกหลายภาษา ไม่ว่าจะเป็น XQL, XML-QL, SQL, OQL รวมถึง XPath 1.0 ด้วย (Boag. 2005)

XQuery 1.0 ถือได้ว่าเป็นส่วนขยายของ XPath 2.0 หรือมองอีกมุมคือ XPath เป็นสับเซตของ XQuery นั่นเอง (Siegers. 2004 : 4) ดังภาพที่ 2.3



เอกสารนี้เป็นเอกสารที่สงวนไว้เฉพาะใช้เพื่อการเรียนการสอนเท่านั้น ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้านการค้า
 ภาพที่ 2.3 ความสัมพันธ์ระหว่างข้อกำหนดต่างๆ ของ XML (W3Schools. 2005)
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 นิพจน์ที่ใช้ใน XQuery

นิพจน์ที่ใช้ใน XQuery มีอยู่หลายนิพจน์ด้วยกัน นิพจน์หลักๆ ทั้งหมดมี 14 นิพจน์ รายงานฉบับนี้จะเลือกอธิบายเพียง 5 นิพจน์ที่สำคัญ คือ นิพจน์เส้นทาง (Path Expression) นิพจน์ FLWOR (FLWOR Expression) นิพจน์เงื่อนไข (Conditional Expression) นิพจน์ตรรกะ (Logical Expression) และนิพจน์เปรียบเทียบ (Comparison Expression) (Boag, 2005)

นิพจน์โดยทั่วไป (Expr) สามารถประกอบไปด้วยหลายๆ ExprSingle ซึ่งถูกแยกด้วยเครื่องหมายจุลภาค (,) ดังภาพที่ 2.4

Expr	::=	ExprSingle ("," ExprSingle)*
ExprSingle	::=	FLWORExpr QuantifiedExpr TypeswitchExpr IfExpr

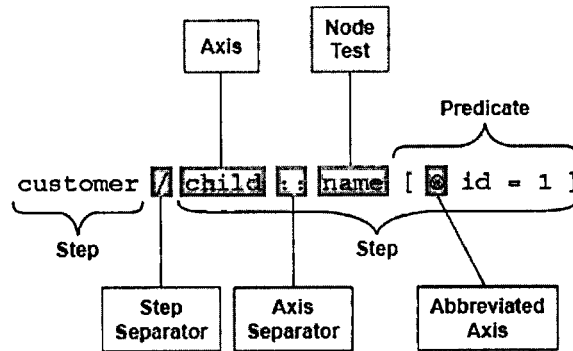
ภาพที่ 2.4 รูปแบบของนิพจน์ (Boag, 2005)

XQuery ให้ความสำคัญกับตัวอักษรพิมพ์เล็ก-ใหญ่ เหมือนกับการกำหนดแท็กใน XML ดังนั้น จึงต้องมีการกำหนดรูปแบบของการใช้นิพจน์ใน XQuery ให้ใช้ตัวอักษรพิมพ์เล็กในทุกๆ นิพจน์ของ XQuery (Boag, 2005)

หมายเหตุ: ตัวอย่างประกอบความเข้าใจทั้งหมดจากส่วนนี้เป็นต้นไป จะเป็นการค้นหาข้อมูลจากเอกสาร XML ชื่อ *videos.xml* โดยอ้างอิงมาจากภาคผนวก และเนื่องด้วยอาจมีบางการค้นหาที่ได้ผลลัพธ์ค่อนข้างมาก จึงขอละไว้ด้วยตัวอักษรจุด 3 จุด (...) แทนส่วนที่ไม่ได้นำเสนอ

2.2.1.1 นิพจน์เส้นทาง

นิพจน์เส้นทาง (Path Expression) เป็นนิพจน์ที่ถูกนำมาใช้ในการชี้ที่ตั้งของโหนดภายในโครงสร้างต้นไม้ โดยนิพจน์นี้ประกอบไปด้วยชุดของ Step อาจจะหนึ่งหรือหลายๆ Step ซึ่งแบ่งแยกระหว่าง Step ด้วย / หรือ // ดังภาพที่ 2.5



ภาพที่ 2.5 โครงสร้างของนิพจน์เส้นทาง (Brundage. 2004 : 2)

/ เป็นการบอกถึงคอนเท็กซ์โหนด (ส่วนของเอกสารที่จะเริ่มต้นเดิน) ปัจจุบัน ดังตัวอย่างต่อไปนี่

ตัวอย่างที่ 2.1 การแบ่งแยะระหว่าง Step ด้วย /

```
doc("videos.xml")/result/actors
```

คำอธิบาย: คอนเท็กซ์โหนด คือ เอลิเมนต์ `<result>` ของ `videos.xml`

ผลลัพธ์ที่ได้คือ

```
<actors>
  <actor id="00000003">Jones, Tommy Lee</actor>
  <actor id="00000006">Ford, Harrison</actor>
  <actor id="00000009">Smith, Will</actor>
  <actor id="0000000c">Hackman, Gene</actor>
  <actor id="0000000f">Bonet, Lisa</actor>
  ...
</actors>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน // เป็นการเข้าถึงโหนดลูกหลานใดๆ ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.2 การแบ่งแยกระหว่าง Step ด้วย //

```
doc("videos.xml")//title
คำอธิบาย: เป็นการบอกว่าจะเข้าถึงโหนด <title>
ไม่ว่ามันจะเป็นลูกหลานของใครใน videos.xml

ผลลัพธ์ที่ได้คือ
<title>The Fugitive</title>
<title>Enemy of the State</title>
<title>Star Wars</title>
<title>Six Days Seven Nights</title>
<title>Seven</title>
```

ภายในแต่ละ Step ประกอบไปด้วย 3 ส่วน (Brundage. 2004 : 3) คือ

- แกน (Axis)

ใน XQuery จะพบแกนที่ใช้บ่อย 4 แกน คือ แกนลูก แกนแอตทริบิวต์ แกนแม่ และแกนตัวเอง โดยจะใช้เครื่องหมายมหัพภาคคู่ (:) 2 เครื่องหมายติดกัน (::) ในการแบ่งชื่อแกน กับ Node Tests และการไม่ใช่ชื่อแกน XQuery จะถือว่าเป็นแกนลูกทันที (แกนลูกถือว่าเป็นค่าโดยปริยาย)

ตารางที่ 2.1 แกนพื้นฐาน 4 แกน และตัวย่อ

ชื่อแกน	ตัวย่อ	ตัวอย่าง
แม่	..	x/parent::node() = x/..
ลูก		x/child::y = x/y
ตัวเอง	.	x/self::node() = x/.
แอตทริบิวต์	@	x/attribute::y = x/@y

- Node Test

Node Test มีทั้งหมด 3 ชนิด คือ Name Test, Node Kind Test และ Wildcard อธิบายแต่ละชนิด ดังนี้

Name Test เป็น Node Test ที่ถูกนำไปใช้บ่อยมากที่สุด โดย Name Test จะเป็นคำเดียวกับชื่อของโหนดในเอกสาร XML ซึ่งจะมีลักษณะไวต่ออักษรใหญ่-เล็กด้วย ดังตัวอย่างต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2.3 การใช้ Name Test

```
doc("videos.xml")/result/actors
```

คำอธิบาย: เข้าถึงเอลิเมนต์ <actors> ที่เป็นลูกของเอลิเมนต์ result จะสังเกตได้ว่าคำว่า actors เป็นคำเดียวกับชื่อของเอลิเมนต์ <actors> ใน videos.xml

ผลลัพธ์ที่ได้คือ เช่นเดียวกับตัวอย่างที่ 2.1

Node Kind Test เป็น Node Test ที่ใช้สำหรับโหนดของ XML ที่ไม่มีชื่อ เช่น โหนดแท็ก (text()) โหนดคอมเมนต์ (comment()) โหนดเอลิเมนต์ (element()) เป็นต้น ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.4 การใช้ Node Kind Test

```
doc("videos.xml")/result/actors/actor/text()
```

คำอธิบาย: เลือกโหนดแท็กทั้งหมดที่เป็นลูกของเอลิเมนต์ <actor>

ผลลัพธ์ที่ได้คือ

```
Jones, Tommy LeeFord, HarrisonSmith,
WillHackman, GeneBonet, LisaVoight,
JohnHamill, MarkFisher, CarrieHeche,
AnneSchwimmer, DavidPitt, BradFreeman,
Morgan
```

Wildcard เป็น Node Test ที่ใช้สำหรับเลือกโหนดทั้งหมด โดยใช้เครื่องหมายดอกจัน (*) ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.5 การใช้ Wildcards

```
doc("videos.xml")/result/actors/*
```

คำอธิบาย: เลือกโหนดทั้งหมดที่เป็นลูกของเอลิเมนต์ <actors>

ผลลัพธ์ที่ได้คือ

```
<actor id="00000003">Jones, Tommy Lee</actor>
<actor id="00000006">Ford, Harrison</actor>
<actor id="00000009">Smith, Will</actor>
<actor id="0000000c">Hackman, Gene</actor>
<actor id="0000000f">Bonet, Lisa</actor>
...
```

- Predicate

Predicate เป็นการกรองข้อมูลจากโหนดที่ถูกเลือก โดยจะถูกกำหนดอยู่ภายใน [และ] ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.6 การใช้ Predicate

```
doc("videos.xml")//video[genre='action']/title
คำอธิบาย: เลือกเอลิเมนต์ <title> ซึ่งเป็นลูกของเอลิเมนต์ <video> แล้วกรองเอาเฉพาะค่าของเอลิเมนต์ <genre> ที่มีค่าเป็น action
```

ผลลัพธ์ที่ได้คือ

```
<title>The Fugitive</title>
<title>Enemy of the State</title>
<title>Star Wars</title>
```

2.2.1.2 นิพจน์ FLWOR

นิพจน์ FLWOR (FLWOR Expression) FLWOR อ่านออกเสียงว่า ฟลาวา-เออะ (flower) ย่อมาจากคีย์เวิร์ด for, let, where, order by และ return ซึ่งเป็นนิพจน์ที่มีลักษณะการทำงานคล้าย SQL โดยสามารถที่จะคิวรีเอกสาร XML ได้มากกว่า 1 เอกสาร ใน 1 ครั้ง จึงเกิดเป็นเอกสาร XML ที่มีโครงสร้างใหม่ตามที่กำหนดได้ (Boag, 2005) ซึ่งจะอธิบายรายละเอียดแต่ละคีย์เวิร์ด ดังนี้

2.2.1.2.1 For และ Let

For และ Let ใช้สำหรับกำหนดตัวแปร และ/หรือชุดของตัวแปร ผูกเข้ากับเอกสารที่จะค้นหา ซึ่งอาจจะกำหนดจุดตั้งต้นในการหาด้วยนิพจน์เส้นทาง เมื่อทำงาน for และ let จะคืนค่าเป็นกลุ่มของโหนดและค่าผลลัพธ์ในแต่ละโหนด (รวมถึงโหนดลูกหลาน) ที่ได้มาเก็บไว้ในตัวแปรที่กำหนด โดยที่ for จะมีการวนซ้ำในการคืนค่าของผลลัพธ์ ในขณะที่ let ไม่มีการวนซ้ำในการคืนค่าของผลลัพธ์

2.2.1.2.2 Where

Where เป็นค่าทางเลือก (Optional) คือจะกำหนดหรือไม่กำหนดก็ได้ ใช้ในการกรองข้อมูลที่ต้องการจากค่าที่ได้จาก for และ/หรือ let ซึ่งจะคืนค่าในรูปของบูล โดยการกำหนด where อาจจะเกิดจากการใช้นิพจน์เปรียบเทียบ หรือนิพจน์ตรรกะ (ที่จะได้อธิบายรายละเอียดในหัวข้อถัดไป) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.2.3 Order by

Order by เป็นค่าทางเลือกเช่นเดียวกับ where ใช้ในการกำหนดการเรียงลำดับข้อมูล และ/หรือชุดของข้อมูล โดยจะต้องมีการกำหนดไว้ก่อน return

2.2.1.2.4 Return

Return ใช้กำหนดรูปแบบในการแสดงผลลัพธ์ โดยอาจกำหนดเป็นเอลิเมนต์ใหม่ ซึ่งผลลัพธ์ที่ได้จะมีการเรียงลำดับตามในเอกสาร XML ที่ใช้ในการค้นหา หากไม่มีการกำหนดลำดับโดย order by

ตัวอย่างที่ 2.7 ค้นหาชื่อภาพยนตร์และความยาวของภาพยนตร์ที่เป็นภาพยนตร์ประเภทแอคชัน โดยแสดงผลเรียงลำดับตามความยาวของภาพยนตร์จากมากไปหาน้อย

```
for      $video in doc("videos.xml")//video
where    $video/genre='action'
order by number($video/runtime) descending
return
<result>
  {
    $video/title, $video/runtime
  }
</result>
```

ผลลัพธ์ที่ได้คือ

```
<result>
  <title>Star Wars</title>
  <runtime>121</runtime>
</result>
<result>
  <title>Enemy of the State</title>
  <runtime>113</runtime>
</result>
<result>
  <title>The Fugitive</title>
  <runtime>110</runtime>
</result>
```

2.2.1.3 นิพจน์เงื่อนไข

นิพจน์เงื่อนไข (Conditional Expression) เป็นนิพจน์ที่ใช้ในการเลือกเส้นทางการทำงาน โดยประกอบไปด้วยคีย์เวิร์ด 3 ตัว คือ if, then และ else ซึ่งนิพจน์ที่ตามหลัง if จะเรียกว่า Test Expression และเรียกนิพจน์ที่ตามหลัง then และ else ว่า Then Expression และ Else Expression ตามลำดับ (Boag. 2005) ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.8 ตรวจสอบว่าภายในรายชื่อนักแสดงมีนักแสดงชื่อขึ้นต้นด้วยคำว่า Lisa หรือไม่
ถ้ามีให้ตอบว่า <Yes/> ถ้าไม่มีให้ตอบว่า <No/>

```
if (doc("videos.xml")//actor[ends-with(. , 'Lisa')])
then <Yes/>
else <No/>
```

ผลลัพธ์ที่ได้คือ
<Yes/>

2.2.1.4 นิพจน์ตรรกะ

นิพจน์ตรรกะ (Logical Expression) เป็นนิพจน์ที่ใช้ในการตรวจสอบ 2 นิพจน์ (ตัวถูกดำเนินการ) ซึ่งจะคืนค่าเป็นบูล คือ จริง หรือ เท็จ โดยนิพจน์นี้ประกอบไปด้วย 2 คีย์เวิร์ด คือ and กับ or ต่อไปนี้จะป็นค่าที่เป็นไปได้ของ and และ or ตามลำดับ โดย EBV คือ Effective Boolean Value (Boag. 2005)

ตารางที่ 2.2 ค่าบูลที่เป็นไปได้ของ and และ or (Boag. 2005)

AND:	EBV ₂ = จริง	EBV ₂ = เท็จ	Error in EBV ₂
EBV ₁ = จริง	จริง	เท็จ	error
EBV ₁ = เท็จ	เท็จ	เท็จ	เท็จ/error
Error in EBV ₁	error	เท็จ/error	Error
OR:	EBV ₂ = จริง	EBV ₂ = เท็จ	Error in EBV ₂
EBV ₁ = จริง	จริง	จริง	จริง/error
EBV ₁ = เท็จ	จริง	เท็จ	error
Error in EBV ₁	จริง/error	error	error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2.9 นิพจน์ตรรกะในแบบต่างๆ

- นิพจน์เหล่านี้จะคืนค่าเป็น จริง
 - 1 = 1 **and** 2 = 2
 - 1 = 1 **or** 2 = 3
- นิพจน์นี้อาจคืนค่าเป็นเท็จหรือไม่ก็เป็น error
 - 1 = 2 **and** 3 div 0 = 1
- นิพจน์นี้อาจคืนค่าเป็นจริงหรือไม่ก็เป็น error
 - 1 = 1 **or** 3 div 0 = 1
- นิพจน์นี้คืนค่าเป็น error
 - 1 = 1 **and** 3 div 0 = 1

2.2.1.5 นิพจน์เปรียบเทียบ

นิพจน์เปรียบเทียบ (Comparison Expression) เป็นนิพจน์ที่ใช้ในการเปรียบเทียบค่า 2 ค่า ใน XQuery มีนิพจน์เปรียบเทียบทั้งหมด 3 ชนิด คือ เปรียบเทียบค่า (Value Comparison) เปรียบเทียบทั่วไป (General Comparison) และ เปรียบเทียบโหนด (Node Comparison) (Boag, 2005) ในรายงานฉบับนี้ขออธิบายเพียง เปรียบเทียบทั่วไปเท่านั้น เพราะเป็นการเปรียบเทียบที่เกิดขึ้นบ่อย

ตัวดำเนินการที่ใช้ในการเปรียบเทียบทั่วไป คือ =, !=, <, <=, >, และ >= โดยผลลัพธ์การเปรียบเทียบทั่วไปคือ จริง หรือ เท็จ ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.10 เปรียบเทียบค่า 2 ค่าระหว่าง ปีที่ภาพยนตร์ออกฉาย กับปี 1999

```
for $video in
doc("videos.xml")//video
where number($video/year) <= 1999
return $video/title
```

ผลลัพธ์ที่ได้คือ

```
<title>The Fugitive</title>
<title>Enemy of the State</title>
<title>Star Wars</title>
<title>Six Days Seven Nights</title>
<title>Seven</title>
```

2.2.2 ฟังก์ชัน

XQuery ได้จัดเตรียมฟังก์ชันไว้ให้อยู่หลายฟังก์ชันที่เรียกว่า ฟังก์ชันในตัว อย่างที่เห็นบ่อยแล้วในหลายๆ ตัวอย่าง คือ ฟังก์ชัน doc() ที่ใช้ในการอ้างถึงเอกสาร XML ที่จะใช้ในการค้นหา นอกจากนี้ยังมีอีกหลายๆ ฟังก์ชันที่มีให้ใช้ เช่น

- ฟังก์ชันการนับจำนวน (count)
- ฟังก์ชันการหาค่าเฉลี่ย (avg)
- ฟังก์ชันการหาค่าสูงสุด (max)
- ฟังก์ชันการหาค่าต่ำสุด (min)
- ฟังก์ชันการหาผลรวม (sum)
- ฟังก์ชันการตรวจสอบค่าว่าง(empty)
- ฟังก์ชันการตรวจสอบค่าที่มีอยู่ (exists)
- ฟังก์ชันแปลงค่าใดๆ ไปเป็นสตริง (string)
- ฟังก์ชันการเชื่อมสตริงต่างๆ เหล่านั้นเข้าไว้ด้วยกัน (concat)
- ฟังก์ชันการรับสตริงย่อย ณ ตำแหน่งที่ต้องการมาจากสตริงที่กำหนด (substring)
- ฟังก์ชันการบอกว่าส่วนท้ายของสตริงจะตรงกับสตริงที่กำหนดหรือไม่ (ends-with)
- ฟังก์ชันการบอกว่าส่วนเริ่มต้นของสตริงจะตรงกับสตริงที่กำหนดหรือไม่ (starts-with)

ตัวอย่างที่ 2.11 นับจำนวนภาพยนตร์ที่เป็นภาพยนตร์ประเภท PG-13

```
count
(
  for $video in doc("videos.xml")//video
  where $video/rating = 'PG-13'
  return $video
)
```

ผลลัพธ์ที่ได้คือ

2

นอกจากนี้ยังสามารถให้ผู้สร้างฟังก์ชันขึ้นใช้งานเอง (User-Defined Function) ได้อีกด้วย สำหรับการสร้างฟังก์ชันขึ้นใช้งานนั้น ต้องกำหนดชื่อฟังก์ชัน ประเภทข้อมูลของพารามิเตอร์ และ ประเภทข้อมูลของผลลัพธ์ (Boag, 2005) ดังตัวอย่างต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2.12 การสร้างและใช้ฟังก์ชันคำนวณส่วนลดราคา DVD ตามระดับความนิยมของผู้ชม โดยถ้าระดับนิยมของผู้ชมมากกว่าหรือเท่ากับ 4 ลดราคาให้ 3 หน่วย (เงิน) แต่ถ้าน้อยกว่านั้นลดราคาให้ 1 หน่วย (เงิน)

(:การสร้างฟังก์ชัน:)

```
declare function local:discountDVDPrice(
  $price as xs:decimal?,
  $userrating as xs:decimal?) as xs:decimal? {
  let $actualDiscount := ( if($userrating >= 4)
    then 3
    else 1)

  return
    ($price - $actualDiscount)
};
```

(:การใช้ฟังก์ชัน:)

```
for $video in doc("videos.xml")//video
where exists($video/dvd)
return
<result>
{
  $video/title,
  <oldprice>{
    $video/dvd/text()
  }
  </oldprice>,
  <dvdprice>{
    local:discountDVDPrice($video/dvd,
    $video/user_rating)
  }
  </dvdprice>
}
</result>
```

ผลลัพธ์ที่ได้คือ

```
<result>
  <title>The Fugitive</title>
  <oldprice>14.99</oldprice>
  <dvdprice>13.99</dvdprice>
</result>
<result>
  <title>Enemy of the State</title>
  <oldprice>29.99</oldprice>
  <dvdprice>26.99</dvdprice>
</result>
<result>
  <title>Seven</title>
  <oldprice>22</oldprice>
  <dvdprice>21</dvdprice>
</result>
```

2.2.3 การทำงานกับแอตทริบิวต์

ในการค้นหาข้อมูลภายในเอกสาร XML ที่มีแอตทริบิวต์เป็นส่วนประกอบ แอตทริบิวต์สามารถทำงานได้เหมือนเอลิเมนต์ทั่วไป ซึ่งใช้เป็นส่วนหนึ่งในการกำหนดเงื่อนไข และการแสดงผล

2.2.3.1 การกำหนดเงื่อนไขด้วยแอตทริบิวต์

นิพจน์ FLWOR ในส่วนของ where สามารถกำหนดเงื่อนไขโดยใช้แอตทริบิวต์กรองข้อมูลได้ ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.13 ค้นหาชื่อภาพยนตร์ที่มีค่าของแอตทริบิวต์ id ไม่ขึ้นต้นด้วยคำว่า id

```
for $video in doc("videos.xml")//video
where not(starts-with($video/@id , 'id'))
return $video/title
```

ผลลัพธ์ที่ได้คือ

```
<title>Seven</title>
```

2.2.3.2 การแสดงแอตทริบิวต์ในผลลัพธ์

ผลลัพธ์ที่ได้จากการค้นหา ในส่วนของแอตทริบิวต์สามารถนำมาแสดงในเอลิเมนต์ที่สร้างใหม่ หรือนำมาแสดงเป็นเนื้อหาในเอลิเมนต์ก็ได้ ซึ่งอาจต้องใช้ฟังก์ชัน string() ทำงานร่วมกับ ดังตัวอย่างต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2.14 ค้นหาชื่อภาพยนตร์ที่มีค่าของแอตทริบิวต์ id ไม่ขึ้นต้นด้วยคำว่า id แล้วแสดง
แอตทริบิวต์ id ภายในเอลิเมนต์ใหม่ คือ <video> และเพิ่มคำว่า id ลงในค่า
ของแอตทริบิวต์ id เป็นเนื้อหาในเอลิเมนต์ของเอลิเมนต์ <newid>

```
for $video in doc("videos.xml")//video
where not(starts-with($video/@id, 'id'))
return
<video>
{
  $video/@id,
  <newid>{concat('id', string($video/@id))}</newid>,
  $video/title
}
</video>
```

ผลลัพธ์ที่ได้คือ

```
<video id="647599249">
  <newid>id647599249</newid>
  <title>Seven</title>
</video>
```

2.2.4 การค้นหาข้อมูลภายในเอกสาร XML แบบอ้างอิงข้อมูลเดียวกัน (Join)

ในการค้นหาข้อมูลจากเอกสาร XML ที่มีมากกว่า 2 เอกสารขึ้นไปนั้น หรือการค้นหาข้อมูลจากเอกสาร XML ที่มีการจัดเก็บข้อมูลเหมือนฐานข้อมูลเชิงสัมพันธ์ คือ รูดเอลิเมนต์เปรียบเทียบกับฐานข้อมูล เอลิเมนต์ย่อยๆ เปรียบเหมือนตาราง แล้วเอลิเมนต์ภายในเอลิเมนต์ย่อยๆ ก็คือ เรคคอร์ด ซึ่งประกอบไปด้วยฟิลด์ข้างใน ซึ่งอาจเป็นเอลิเมนต์หรือแอตทริบิวต์ โดยในการค้นหาข้อมูลจากเอกสารลักษณะนี้ จะต้องมีการกำหนดเงื่อนไขเพื่อให้อ้างอิงข้อมูลเดียวกัน ซึ่งเปรียบได้กับการค้นหาข้อมูลจาก 2 ตารางขึ้นไป (Join) ใน SQL ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.15 ค้นหารายชื่อนักแสดงที่แสดงภาพยนตร์แอคชันแล้วตัดชื่อที่ซ้ำออก เพราะนักแสดง
1 คนอาจแสดงในภาพยนตร์หลายเรื่อง แล้วแสดงผลโดย
นำเอาชื่อขึ้นต้น แล้วตามด้วยนามสกุล

```

distinct-values(
let $doc := doc("videos.xml")
for $video in $doc//video,
    $actor in $doc//actor
where      $video/actorRef = $actor/@id
           and $video/genre = 'action'
return
concat(substring-after($actor, ',', ' '), ' ',
        substring-before($actor, ',', ' ')
)

```

ผลลัพธ์ที่ได้คือ

Tommy Lee Jones	Harrison Ford	Will Smith
Gene Hackman	Lisa Bonet	John Voight
Hamill	Carrie Fisher	Mark

2.3 แบบจำลองกระบวนการทำงานของ XQuery

แบบจำลองกระบวนการทำงาน (XQuery Process Model) ประกอบด้วย 4 ขั้นตอน ซึ่งในแต่ละขั้นตอนจะนำผลลัพธ์ที่ได้จากขั้นตอนก่อนหน้ามาเป็นอินพุต เพื่อนำมาเข้าสู่กระบวนการทำงานเพื่อให้ได้เอาต์พุตไปทำงานในขั้นต่อไป ดังภาพที่ 2.6 ซึ่งรายละเอียดในแต่ละขั้นตอนมีดังนี้ (ขนิษฐสรณ์ เวียรศิลป์, 2546 : 19)

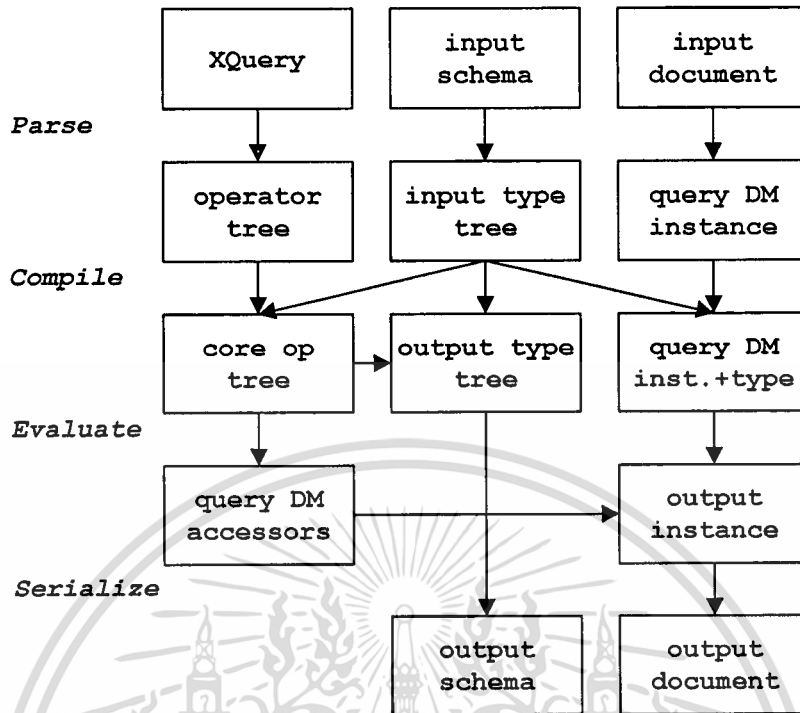
กระบวนการวิภาษหรือแจงข้อมูล (Parse) มีหน้าที่ในการตรวจสอบความถูกต้องของนิพจน์ชุดคำสั่ง XQuery ความถูกต้องของเค้าร่างเอกสาร และความถูกต้องของรูปแบบเอกสาร XML หลังจากตรวจสอบความถูกต้องเรียบร้อยแล้ว จะสร้างโครงสร้างตัวดำเนินการในรูปแบบต้นไม้สร้างโครงสร้างเอกสารในรูปแบบต้นไม้และโครงสร้างข้อมูลเอกสารในรูปแบบต้นไม้ตามลำดับ

กระบวนการแปลโปรแกรม (Compile) มีหน้าที่แปลงนิพจน์ชุดคำสั่ง XQuery ให้อยู่ในรูปแบบของ XQuery Core และนำโครงสร้างเอกสารกับเอกสาร XML มาตรวจสอบความถูกต้องร่วมกัน

กระบวนการประเมินหาคำตอบ (Evaluate) มีหน้าที่นำ XQuery Core มาประมวลผลเพื่อค้นหาข้อมูลผลลัพธ์ และสร้างโครงสร้างผลลัพธ์

กระบวนการเรียงลำดับสร้างผลลัพธ์ (Serialize) มีหน้าที่นำข้อมูลผลลัพธ์ และโครงสร้างผลลัพธ์มาสร้างเป็นเอกสาร XML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.6 แบบจำลองการทำงานของ XQuery

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

เครื่องมือที่ใช้ในการพัฒนาโครงการ

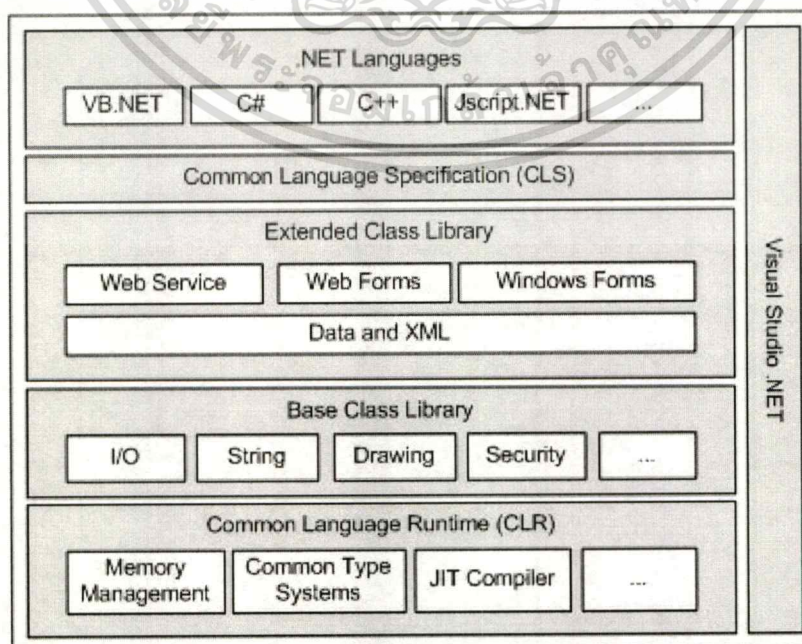
หลังจากได้ศึกษาเทคโนโลยี XML และ XQuery เพื่อใช้ประกอบเป็นพื้นฐานความรู้ในการพัฒนาโครงการแล้ว ขั้นตอนต่อไปคือการศึกษาเกี่ยวกับเครื่องมือต่างๆ ที่จะใช้ในการพัฒนาโครงการนี้ซึ่งประกอบไปด้วยเทคโนโลยี .NET ชุดคลาสไลบรารีและ UML

3.1 เทคโนโลยี .NET

ไมโครซอฟต์ .NET หรือเรียกสั้นว่า .NET เป็นเทคโนโลยีล่าสุดที่พัฒนาขึ้นโดยบริษัทไมโครซอฟต์ .NET คือ แพลตฟอร์มในการพัฒนาซอฟต์แวร์สำหรับระบบปฏิบัติการวินโดวส์ โดยมีแนวคิดที่ เขียนโปรแกรมด้วยภาษาใดๆก็ได้ (Language Neutral) แต่ทำงานได้บนระบบปฏิบัติการวินโดวส์เท่านั้น (Windows Centric) หรือที่เรียกว่า Write any languages, run on Windows (สุรสิทธิ์ ทิวประสพศักดิ์ และนันท์นิ แขวงโสภา. 2546 : 14)

3.1.1 สถาปัตยกรรมเฟรมเวิร์กค็อดเน็ต

สิ่งที่เป็นหัวใจสำคัญของ .NET คือ เฟรมเวิร์กค็อดเน็ต ซึ่งมีส่วนประกอบต่างๆ ดังภาพที่ 3.1 (สุรสิทธิ์ ทิวประสพศักดิ์ และนันท์นิ แขวงโสภา. 2546 : 14) ซึ่งสามารถอธิบายจากบนลงล่างได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ประกอบการเรียนการสอนเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า

ภาพที่ 3.1 สถาปัตยกรรมเฟรมเวิร์กค็อดเน็ต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.NET มีภาษาให้เลือกใช้มากมายมากกว่า 20 ภาษาในการพัฒนาซอฟต์แวร์ เช่น VB.NET, C#.NET, C++.NET, Pascal .NET, Perl.NET หรือ COBOL.NET เป็นต้น ซึ่งทุกภาษาเหล่านี้อยู่ภายใต้กฎเกณฑ์มาตรฐานเดียวกันที่เรียกว่า Common Language Specifications (CLS) และโครงสร้างพื้นฐานตั้งแต่ชนิดข้อมูล ชุดคำสั่งพื้นฐาน เช่น การจัดการอินพุต/เอาต์พุต ฐานข้อมูล เป็นต้น

การสร้างแอปพลิเคชันสามารถทำได้หลากหลายรูปแบบ ได้แก่ Win Form, Web Form และ Web Service ซึ่งโปรแกรมที่พัฒนาขึ้นด้วย .NET จะมีการเรียกใช้ข้อมูลประเภทเดียวกันทั้งหมดไม่ว่าจะถูกเขียนด้วยภาษา C#.NET, VB.NET หรือภาษาใดๆ ก็ตาม ประเภทข้อมูลเหล่านี้จะอยู่ในกลุ่มของคลาส Data และ XML

คลาสพื้นฐาน (Base Class) เป็นที่รวมของคลาสพื้นฐานต่างๆ ซึ่งไมโครซอฟต์พัฒนาขึ้นมาให้สามารถเรียกใช้งานและพัฒนาต่อออกเพิ่มเติมได้ ซึ่งคลาสพื้นฐานนี้ครอบคลุมถึงสิ่งที่จำเป็นในการพัฒนาโปรแกรม เช่น การจัดการอินพุต/เอาต์พุต การจัดการข้อมูลชนิดสตริง การจัดการกราฟิก การจัดการเกี่ยวกับความปลอดภัยของระบบ เป็นต้น

และชั้นสุดท้ายซึ่งเป็นส่วนที่สำคัญที่สุดของเฟรมเวิร์กคือดอตเน็ต คือ Common Language Runtime (CLR) มีหน้าที่คือ เป็น Execution Engine ในการประมวลผลและจัดการโปรแกรมที่คอมไพล์แล้วให้ทำงานได้บนระบบปฏิบัติการวินโดวส์ โดย CLR จะแปลงโค้ดในรูปภาษากลาง (Intermediate Language) (ทางไมโครซอฟต์มีชื่อเรียกว่า Microsoft Intermediate Language: MSIL) ไปเป็นคำสั่งภาษาเครื่อง ซึ่งใช้เทคโนโลยีในการแปลงแบบ Just In Time (JIT) คือ แปลงเฉพาะส่วนที่จะนำมาใช้งานเท่านั้น หลังจากนั้น ถ้าต้องการนำส่วนอื่นๆ มาใช้งานอีกก็จะแปลงเพิ่มเฉพาะในส่วนนั้น นอกจากนี้ CLR ยังทำหน้าที่ติดต่อกับระบบปฏิบัติการ จัดสรรหน่วยความจำให้กับโปรแกรมต่างๆ และคืนหน่วยความจำที่ไม่ถูกใช้งานให้กับระบบที่เรียกว่า การเก็บขยะ จัดการข้อผิดพลาด รวมถึงดูแลเรื่องความปลอดภัยด้วย (สุรสิทธิ์ ธิวประสพศักดิ์ และนนทนี แขวงโสภา. 2546 : 15)

3.1.2 Visual Basic.NET

Visual Basic .NET หรือ VB.NET เป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมแบบ Visual Programming ซึ่งได้รับการพัฒนามาจากภาษาเบสิก โดย VB.NET เป็นเวอร์ชันล่าสุดของ Visual Basic โดยเวอร์ชันนี้ได้เพิ่มขีดความสามารถมากขึ้น สิ่งที่โดดเด่นคือการปรับเปลี่ยนภาษาเป็นลักษณะ OOP (Object Oriented Programming) เต็มตัวเหมือนภาษาโปรแกรมสมัยใหม่ เช่น C++, C#, Delphi และ Java เป็นต้น จึงทำให้ VB.NET มีความสามารถในการสืบทอด (Inheritance) Overloading, Overriding, Polymorphism และอื่นๆ (สุรสิทธิ์ ธิวประสพศักดิ์ และนนทนี แขวงโสภา. 2546 : 16)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ชุดคลาสไลบรารี

.NET ได้จัดเก็บกลุ่มคลาสที่ทำงานคล้ายกันไว้เป็นกลุ่มเดียวกันเป็นชุดคลาสไลบรารี โดยกลุ่มของคลาสเหล่านี้ เรียกว่า เนมสเปซ

3.2.1 เนมสเปซที่เกี่ยวข้องกับ XML

การจัดการกับ XML .NET ได้จัดเตรียมคลาสไว้มากมายหลายคลาสจากเนมสเปซ ทั้ง 8 เนมสเปซในคลาสไลบรารีของเฟรมเวิร์ก .NET เวอร์ชัน 1.0, 1.1 และ 2.0 (Chand. 2003; Microsoft. 2005)

เนมสเปซในเวอร์ชัน 1.0 และ 1.1 ได้แก่

- System.Xml
- System.Xml.Schema
- System.Xml.Serialization
- System.Xml.XPath
- System.Xml.Xsl

เนมสเปซที่เพิ่มเข้ามาในเวอร์ชัน 2.0 ได้แก่

- System.Xml.Serialization.Advanced
- System.Xml.Serialization.Configuration
- System.Xml.Xsl.Runtime

การนำคลาสภายในเนมสเปซเหล่านี้มาใช้ได้ จำเป็นต้องอ้างอิงไปยังไฟล์แอสเซมบลีที่ชื่อ System.Xml.dll เฉพาะ System.Xml.Xsl.Runtime เท่านั้นที่ต้องอ้างอิงไปยังไฟล์แอสเซมบลีที่ชื่อ System.Data.Sqlxml.dll และใช้เนมสเปซในโปรแกรมด้วยการใช้คีย์เวิร์ด Imports สำหรับ VB.NET ตัวอย่างคลาสที่นำมาใช้ คลาส XmlDocument คลาส XmlElement คลาส XmlNode คลาส XmlAttribute เป็นต้น

3.2.2 Microsoft.Xml.XQuery

Microsoft.Xml.XQuery เป็นเนมสเปซที่ถูกออกแบบให้ทำงานร่วมกับคลาสที่มีอยู่แล้วในคลาสไลบรารีของเฟรมเวิร์ก .NET โดยเฉพาะกับคลาสที่อยู่ในเนมสเปซ System.Xml ได้เป็นอย่างดี ภายในเนมสเปซนี้ประกอบไปด้วยคลาสที่จำเป็นในการเขียนโปรแกรมทั้งหมด 5 คลาส (Siegers. 2004 : 6) คือ

- XQueryDocument
- XQueryNavigator
- XQueryNavigatorCollection

- XQueryExpression
- XQueryException

รายละเอียดของคลาสต่างๆ ของเนมสเปซ Microsoft.Xml.XQuery อธิบายตามตารางที่ 3.1

ตารางที่ 3.1 คลาสต่างๆ ในเนมสเปซ Microsoft.Xml.XQuery

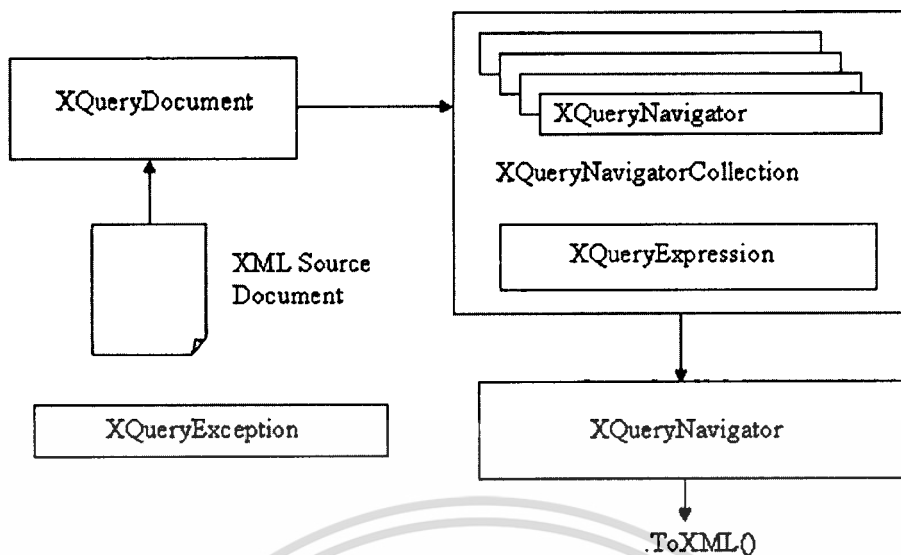
คลาส	รายละเอียด
XQueryDocument	<ul style="list-style-type: none"> - จัดเตรียมอ็อบเจกต์ XML เพื่อใช้ในการคิวรี - มีเมธอดที่สำคัญคือ CreateNavigator() คืนค่าเป็น XQueryNavigator
XQueryNavigator	<ul style="list-style-type: none"> - สร้างผู้นำทาง (Navigator) โดยจะคืนค่าเมื่อมีการคิวรี - มีเมธอดที่สำคัญคือ ToXml() คืนค่าเป็นสตริง
XQueryNavigatorCollection	<ul style="list-style-type: none"> - กลุ่มของ XQueryNavigator โดยการคิวรีของ XQuery จะทำผ่านคลาสนี้ - มีเมธอดที่สำคัญคือ AddNavigator(XQueryNavigator)
XQueryExpression	<ul style="list-style-type: none"> - ห่อหุ้มนิพจน์ XQuery - แจง (Parsed) - ตรวจสอบข้อผิดพลาดจากรูปแบบภาษาผิดพลาด (Syntax Error) - คอมไพล์ - มีเมธอดที่สำคัญคือ Execute(XQueryNavigatorCollection) คืนค่าเป็น XQueryNavigator
XQueryException	<ul style="list-style-type: none"> - จัดการรองรับเกี่ยวกับข้อผิดพลาดที่เกิดขึ้นภายในกลุ่มคลาส XQuery

การนำคลาสเหล่านี้มาใช้จำเป็นต้องอ้างถึงไฟล์แอสเซมบลีที่ชื่อ

Microsoft.Xml.XQuery.dll และนำอังกเนมสเปซในโปรแกรมด้วยการใช้คีย์เวิร์ด Imports สำหรับ VB.NET

โดยทั่วไป คลาสที่ใช้เป็นพื้นฐานในการทำงาน XQuery ประกอบไปด้วย XQueryNavigator, XQueryNavigatorCollection และ XQueryExpression โดยสามารถแสดงความสัมพันธ์ระหว่างคลาสได้ ดังภาพที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.2 รูปแบบการทำงานของชุดคลาสนภายในเนมสเปซ Microsoft.Xml.XQuery

จากภาพที่ 3.2 XQueryDocument จัดการจัดเตรียมเอกสาร XML ที่จะสอบถาม โดยการสร้าง XQueryNavigator ด้วยเมธอด CreateNavigator ซึ่ง XQueryNavigator ที่สร้างขึ้นจะเป็นตัวแทนของเซตของข้อมูล เช่น ไฟล์ XML เป็นต้น โดย XQueryNavigator จะต้องถูกเก็บรวบรวมอยู่ใน XQueryNavigatorCollection ซึ่งสามารถเพิ่มเข้าไปได้ด้วยเมธอด AddNavigator

นิพจน์จะถูกห่อหุ้มด้วย XQueryExpression แล้วจะถูกส่งให้คิวรีด้วยเมธอด Execute โดยการ Execute จะทำผ่าน XQueryNavigatorCollection ซึ่งจะถูกคืนค่าออกมาเป็น XQueryNavigator ต่อจากนั้นจึงใช้เมธอด ToXML เพื่อคืนค่าเป็นสตริงเพื่อแสดงผลต่อไป (Siegers, 2004 : 7)

3.2.3 Altova.AltovaXML

Altova.AltovaXML เป็นเนมสเปซที่ถูกแพคเกจรวมอยู่ใน AltovaXML 2006 ที่ทางบริษัท Altova GmbH พัฒนาขึ้น โดย AltovaXML 2006 เป็นแพคเกจแอปพลิเคชันที่จัดการเกี่ยวกับ XML ที่สามารถนำไปใช้โดยไม่เสียค่าใช้จ่าย (Altova GmbH, 2006)

เนมสเปซ Altova.AltovaXML ประกอบด้วยคลาสนทั้งหมด 5 คลาส ได้แก่

- คลาส Application
- คลาส XMLValidator
- คลาส XQuery
- คลาส XSLT1
- คลาส XSLT2

การนำคลาสนี้มาใช้จำเป็นต้องอ้างถึงไฟล์แอสเซมบลีที่ชื่อ Altova.AltovaXML.dll

และนำอ้างเนมสเปซในโปรแกรมด้วยการใช้คีย์เวิร์ด Imports สำหรับ VB.NET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส XQuery มีคุณสมบัติ (Properties) และเมธอด ดังอธิบายตามตารางที่ 3.2

ตารางที่ 3.2 คุณสมบัติและเมธอดในคลาส XQuery ในเนมสเปซ Altova.AltovaXML

คุณสมบัติ	รายละเอียด
XQueryFileName	กำหนด URL ที่เป็นที่ตั้งของชื่อไฟล์ XQuery ชนิดข้อมูลเป็นสตริง ตัวอย่างเช่น <code>objcXQuery.XQueryFileName = "C:\abc.xquery"</code>
InputXMLFileName	กำหนด URL ที่เป็นที่ตั้งของชื่อไฟล์ XML ชนิดข้อมูลเป็นสตริง ตัวอย่างเช่น <code>objcXQuery.InputXMLFileName = "C:\abc.xml"</code>
XQueryFromText	กำหนดนิพจน์ XQuery ชนิดข้อมูลเป็นสตริง ตัวอย่างเช่น <code>objcXQuery.XQueryFromText = "doc("C:\abc.xquery")"</code>
InputXMLFromText	กำหนดข้อมูล XML ชนิดข้อมูลเป็นสตริง ตัวอย่างเช่น <code>objcXQuery.InputXMLFromText =</code> <code>"<root><a>content<a/></root>"</code>
OutputMethod	กำหนดรูปแบบของผลลัพธ์ มีค่าเป็นไปได้ 4 ค่า คือ xml, xhtml, html และ text โดยมี xml เป็นค่าโดยปริยาย ตัวอย่างเช่น <code>objcXQuery.OutputMethod = "xml"</code>
OutputOmitXMLDeclaration	กำหนดว่าจะให้ละทิ้งในส่วนของการประกาศให้เป็น XML หรือไม่ มีค่าเป็นไปได้ 2 ค่า คือ จริง หรือ เท็จ โดยมีค่าจริงเป็นค่าโดยปริยาย ตัวอย่างเช่น <code>objcXQuery.OutputOmitXMLDeclaration = True</code>
OutputIndent	กำหนดว่าจะให้มีจัดเว้นวรรคหรือไม่ มีค่าเป็นไปได้ 2 ค่า คือ จริง หรือ เท็จ โดยมีค่าเท็จ เป็นค่าโดยปริยาย ตัวอย่างเช่น <code>objcXQuery.OutputIndent = True</code>
OutputEncoding	กำหนดรูปแบบการเข้ารหัสข้อมูล มี UTF-8 เป็นค่าโดยปริยาย ตัวอย่างเช่น <code>objcXQuery.OutputEncoding = "UTF-8"</code>

ตารางที่ 3.2 (ต่อ)

เมธอด	รายละเอียด
Excute(String)	รับค่า URL ที่เป็นที่ตั้งของชื่อ ไฟล์ที่จะถูกสร้างและจัดเก็บ ผลลัพธ์ ตัวอย่างเช่น <code>objcXQuery.Excute("C:\abc.xml")</code>
ExcuteAndgetResultAsString () as String	ประมวลผล XQuery แล้วคืนค่าเป็นสตริง ตัวอย่างเช่น <code>Result = objcXQuery.ExcuteAndgetResultAsString</code>

3.3 ยูเอ็มแอล

ยูเอ็มแอล (UML หรือ Unified Modeling Language) คือ สัญลักษณ์มาตรฐานที่ใช้ในวิธีการออกแบบการพัฒนาเชิงวัตถุ ซึ่งสัญลักษณ์ที่ใช้กันนั้นได้มาจากการรวมเอาวิธีการในการออกแบบและวิเคราะห์แบบเชิงวัตถุ 3 แบบ คือ (Hamilton and Miles, 2006)

1. Booch Method ของ Grady Booch
2. Object-Modeling Technique (OMT) ของ James Rumbaugh
3. Use Case Methodology ของ Ivar Jacobson

ใน UML ประกอบด้วยแผนภาพต่างๆ ดังต่อไปนี้ (Hamilton and Miles, 2006)

- แผนภาพยูสเคส (Use Case Diagram) แสดงปฏิสัมพันธ์ระหว่างระบบกับผู้ใช้ระบบหรือระบบภายนอก
- แผนภาพแอคทิวิตี (Activity Diagram) แสดงลำดับการทำงานของระบบในรูปแบบกิจกรรม
- แผนภาพคลาส (Class Diagram) แสดงคลาส ชนิดของคลาส อินเทอร์เฟซ และความสัมพันธ์ระหว่างคลาส
- แผนภาพอ็อบเจกต์ (Object Diagram) แสดงถึงอินสแตนซ์ที่สืบทอดมาจากคลาสต่างๆ โดยจะประกอบด้วยอ็อบเจกต์และค่าข้อมูลของอ็อบเจกต์นั้นๆ
- แผนภาพซีเควนซ์ (Sequence Diagram) แสดงถึงคลาสที่มีส่วนร่วมในแต่ละยูสเคสและข้อความ (Message) ที่ส่งผ่านระหว่างคลาสบนเวลาต่างๆ
- แผนภาพคอมมิวนิเคชัน (Communication Diagram)
- แผนภาพไทม์มิง (Timing Diagram) แสดงปฏิสัมพันธ์ระหว่างอ็อบเจกต์ที่เน้นไปในเรื่องของเวลาเป็นสำคัญ
- แผนภาพอินเทอร์แอคชัน โอเวอร์วิว (Interaction Overview Diagram)

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี หากท่านใดนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แผนภาพคอมโพเนนต์ (Component Diagram) แสดงคอมโพเนนต์ที่สำคัญต่างๆ ภายในระบบ และอินเตอร์เฟซที่มันใช้ติดต่อกับคอมโพเนนต์อื่นๆ
- แผนภาพแพคเกจ (Package Diagram) แสดงการจัดการ โครงสร้างลำดับชั้นของกลุ่มของคลาสและคอมโพเนนต์
- แผนภาพสเตตแมชีน (State Machine Diagram) แสดงสถานะของอ็อบเจกต์ที่เป็นไปได้ ในระหว่างช่วงชีวิตในการตอบสนองต่อเหตุการณ์ที่เกิดขึ้น
- แผนภาพดีพลอยเมนต์ (Deployment Diagram) ใช้สำหรับแสดงสถาปัตยกรรมของระบบในลักษณะเป็นสถาปัตยกรรมเชิงกายภาพ คือ แสดงว่ามีคอมพิวเตอร์และอุปกรณ์อะไรบ้างที่ต้องการใช้ในระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การวิเคราะห์และออกแบบโปรแกรม

ในบทนี้จะอธิบายถึงการวิเคราะห์และออกแบบโปรแกรม ซึ่งจะเกี่ยวข้องกับการกำหนดความต้องการทั้งแบบเชิงฟังก์ชันและไม่เชิงฟังก์ชัน ข้อจำกัดของโปรแกรม รวมถึงอธิบายระบบงานด้วยแผนภาพแบบต่างๆ ได้แก่ แผนภาพยูสเคส แผนภาพแอกทิวิตี แผนภาพคลาส แผนภาพซีเควนซ์ และแผนภาพดีพลอยเมนต์

4.1 ความต้องการและข้อจำกัดของโปรแกรม

ระบบงานนี้สามารถกำหนดความต้องการทั้งแบบเชิงฟังก์ชันและไม่เชิงฟังก์ชัน รวมถึงข้อจำกัดของโปรแกรม เพื่อใช้เป็นขอบเขตในการพัฒนาระบบงานได้ดังต่อไปนี้

4.1.1 ความต้องการเชิงฟังก์ชัน

การทำงานของโปรแกรมจะต้องรองรับการทำงานต่างๆ ดังต่อไปนี้

- โปรแกรมจะต้องมีหน้าจอสำหรับให้ผู้ใช้ที่มีความรู้ไวยากรณ์ XQuery สร้างไวยากรณ์ XQuery ได้
- โปรแกรมจะต้องสนับสนุนการทำงานในกรณีที่ผู้ใช้ไม่มีความรู้ไวยากรณ์ XQuery โดยมีหน้าจอช่วยเหลือเป็นลำดับขั้นตอนในการสร้างไวยากรณ์ XQuery
- โปรแกรมต้องสนับสนุนการเปิดไฟล์นามสกุล xquery มากกว่า 1 ไฟล์ต่อการทำงานหนึ่งหน้าจอ
- โปรแกรมสามารถแสดงโครงสร้างเอกสาร XML ในรูปแบบโครงสร้างต้นไม้ได้ ทั้งหน้าจอสำหรับให้ผู้ใช้ที่มีความรู้ไวยากรณ์ XQuery และหน้าจอช่วยเหลือ
- โปรแกรมจะต้องสามารถค้นหาข้อมูลภายในเอกสาร XML ได้ผ่านทางงานบันทึกเฉพาะที่ และ URL
- โปรแกรมสามารถตรวจสอบคุณสมบัติรูปแบบถูกต้องของเอกสาร XML ที่ต้องการค้นหา พร้อมแสดงข้อความแจ้งผู้ใช้ในกรณีที่ตรวจพบความไม่ถูกต้องของเอกสาร XML
- โปรแกรมต้องสามารถแสดงผลลัพธ์ได้ 3 มุมมองคือ มุมมองแบบโครงสร้างต้นไม้ มุมมองแบบตัวอักษรธรรมดา และมุมมองแบบเปิดผ่านอินเทอร์เน็ตเอ็กซ์พลอเรอร์
- โปรแกรมสามารถบันทึกเป็นไฟล์ XQuery นามสกุล xquery เพื่อนำมาใช้งานในภายหลังได้

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ผ่านการอนุญาตจากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ถือเป็นการละเมิดลิขสิทธิ์และอาจมีความผิดตามกฎหมายที่เกี่ยวข้อง

4.1.2 ความต้องการที่ไม่ใช่เชิงฟังก์ชัน

- ผู้ใช้มีความรู้ และความเข้าใจในโครงสร้าง XML เบื้องต้น
- เครื่องคอมพิวเตอร์ที่นำมาติดตั้งโปรแกรม และทำงานกับโปรแกรม ควรมีความต้องการ ดังนี้
 - หน่วยประมวลผล Pentium III ขึ้นไป
 - หน่วยความจำตั้งแต่ 256 MB ขึ้นไป
 - เฟรมเวิร์ก .NET เวอร์ชัน 2.0
 - ไมโครซอฟต์อินเทอร์เน็ตเอ็กซ์พลอเรอร์

4.1.3 ข้อจำกัดของโปรแกรม

- หน้าจอช่วยเหลือสำหรับผู้ที่ไม่มีความรู้ไวยากรณ์ XQuery โปรแกรมสามารถสร้างเฉพาะนิพจน์ FLWOR
- หน้าจอช่วยเหลือสำหรับผู้ที่ไม่มีความรู้ไวยากรณ์ XQuery โปรแกรมไม่สามารถสร้างฟังก์ชันที่ผู้ใช้กำหนดเอง (User-Defined Function)
- ในส่วนการกำหนดเงื่อนไขในการเปรียบเทียบ (Where Clause) จากหน้าจอวิซาร์ด สามารถเปรียบเทียบชนิดข้อมูลที่เป็นสตริง (String) วันที่ (Date) ตัวเลขจำนวนเต็ม (Integer) ตัวเลขทศนิยม (Decimal) และตัวเลข Double
- โปรแกรมสามารถค้นหาข้อมูลจากเอกสาร XML ที่มีขนาดข้อมูลไม่จำกัด ขึ้นอยู่กับขนาดหน่วยความจำและประสิทธิภาพของเครื่องคอมพิวเตอร์ที่ติดตั้งโปรแกรม
- โปรแกรมสามารถประมวลผลไวยากรณ์ XQuery กับส่วนประมวลผล XQuery (XQuery Engine) 2 ตัว คือ Altova และ Microsoft
- โปรแกรมสามารถประมวลผลไวยากรณ์ XQuery ได้มีประสิทธิภาพและประสิทธิผลสูงสุดขึ้นอยู่กับส่วนประมวลผล XQuery

4.2 แผนภาพยูสเคส

จากความต้องการในหัวข้อที่ 4.1 สามารถนำมาวิเคราะห์ในเชิงแนวคิด ซึ่งนำเสนอด้วยแผนภาพยูสเคส

4.2.1 การออกแบบแผนภาพยูสเคส

แอกเตอร์และบทบาทหน้าที่ของแอกเตอร์ที่ติดต่อกับโปรแกรม มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

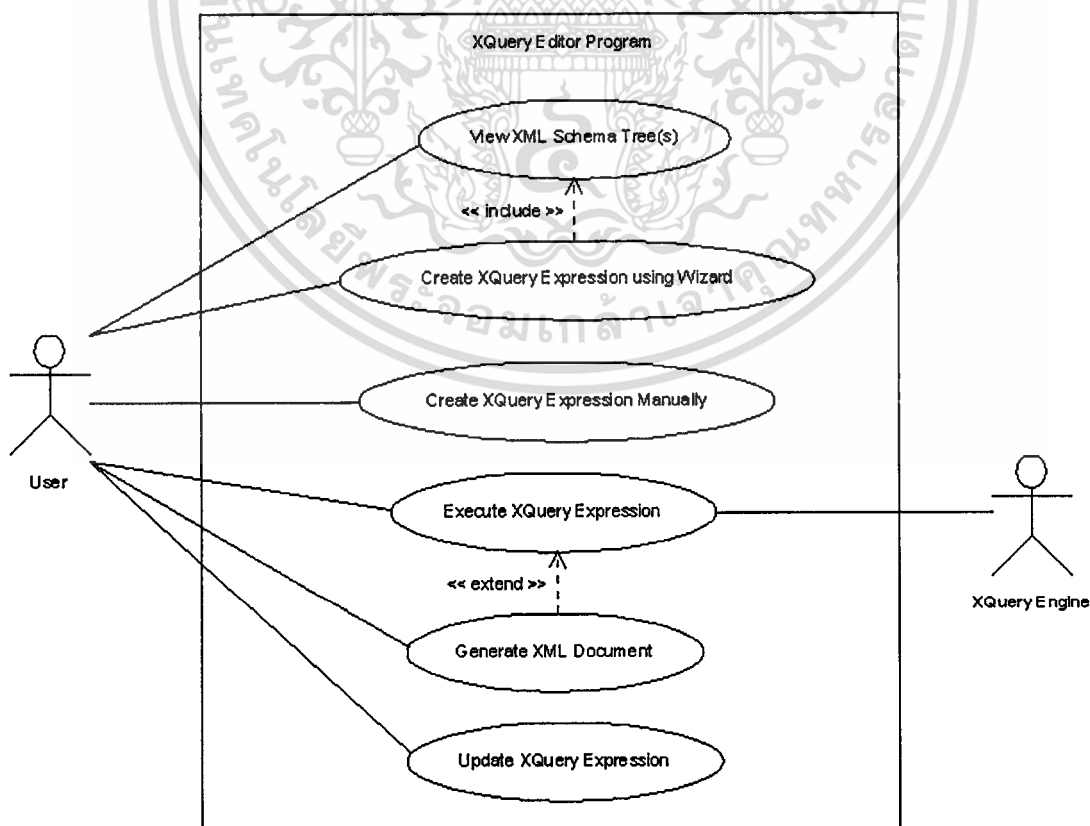
1. User เป็นบุคคลที่เข้าใช้โปรแกรม เพื่อค้นหาข้อมูลภายในเอกสาร XML ซึ่งมีบทบาทคือ ป้อนไวยากรณ์ XQuery เข้าโปรแกรม ซึ่งผู้ใช้อาจเป็นผู้ใช้ที่มีความรู้ในการสร้างไวยากรณ์ XQuery หรือผู้ใช้ที่ไม่มีความรู้ในการสร้างไวยากรณ์ XQuery

2. XQuery Engine เป็นชุดคลาสไลบรารี ซึ่งมีบทบาทในการรับไวยากรณ์ XQuery ไปประมวลผลและคืนค่าผลลัพธ์ให้กับโปรแกรม ซึ่งในโปรแกรมมี XQuery Engine 2 ตัว คือ Microsoft Engine และ Altova Engine

กิจกรรมหลักที่ทำงานภายในโปรแกรม (รายละเอียดของกิจกรรม จะได้อธิบายในหัวข้อคำอธิบายยูสเคส) มีดังนี้

1. Create XQuery Expression Manually
2. Create XQuery Expression using Wizard
3. View XML Schema Tree(s)
4. Execute XQuery Expression
5. Update XQuery Expression
6. Generate XML Document

สามารถนำมาวาดเป็นแผนภาพยูสเคสได้ดังภาพที่ 4.1



ภาพที่ 4.1 แผนภาพยูสเคส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 คำอธิบายยูสเคส

จากแผนภาพยูสเคสสามารถนำมาอธิบายละเอียดของแต่ละยูสเคสได้ด้วยการใช้คำอธิบายยูสเคส (Use Case Description) ได้ดังต่อไปนี้

4.2.2.1 คำอธิบายยูสเคส Create XQuery Expression Manually

คำอธิบาย

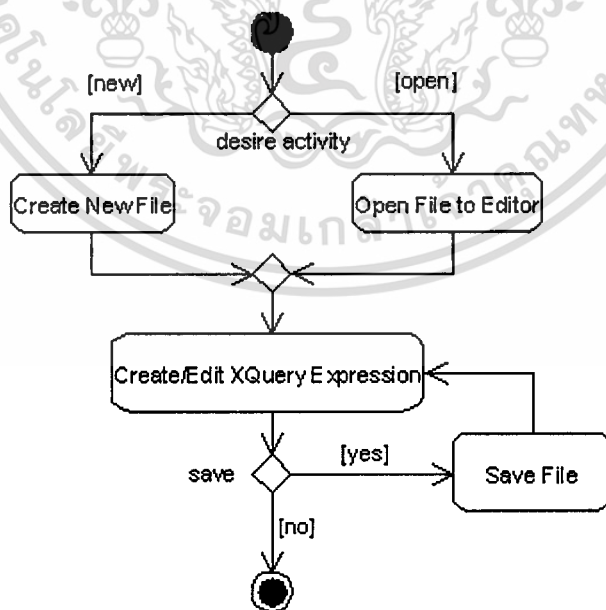
ยูสเคสนี้ใช้ในการอธิบายว่า ผู้ใช้เข้ามาสร้างไวยากรณ์ XQuery ได้อย่างไร โดยเป็นการสร้างไฟล์ไวยากรณ์ XQuery ขึ้นใหม่ หรือเปิดไฟล์ไวยากรณ์ XQuery ที่เคยบันทึกไว้แล้วขึ้นมาทำงาน จากนั้นสามารถบันทึกไวยากรณ์ XQuery ไว้ตามที่ที่ต้องการ

แอกเตอร์ User

ขั้นตอนการทำงาน

- 1 ผู้ใช้คลิกปุ่ม New: โปรแกรมสร้างไฟล์ใหม่ หรือผู้ใช้คลิกปุ่ม Open: ผู้ใช้เลือกเปิดไฟล์ชุดไวยากรณ์ XQuery ที่เคยบันทึกเอาไว้ (ไฟล์นามสกุล xquery) จากงานบันทึกเฉพาะที่
- 2 ผู้ใช้ป้อนและ/หรือแก้ไขไวยากรณ์ XQuery
- 3 ผู้ใช้เลือกบันทึกชุดไวยากรณ์ XQuery ลงบนงานบันทึกเฉพาะที่ และยังคงทำงานยูสเคสนี้ต่อไปตามปกติ

แผนภาพแอกทิวิตี



ภาพที่ 4.2 แผนภาพแอกทิวิตีของการสร้างไวยากรณ์ XQuery

4.2.2.2 คำอธิบายยูสเคส Create XQuery Expression Using Wizard

คำอธิบาย

ยูสเคสสำหรับผู้ใช้ที่ไม่มีความรู้ไวยากรณ์ XQuery โดยผู้ใช้ต้องสร้างไวยากรณ์ XQuery ด้วยหน้าจอช่วยสร้างไวยากรณ์เป็นลำดับขั้นตอน

แอกเตอร์ User

ขั้นตอนการทำงาน

- 1 ผู้ใช้เลือกเอกสาร XML จากบันทึกเฉพาะที่
- 2 โปรแกรมทำยูสเคส View XML Schema Tree(s)
- 3 ผู้ใช้กำหนดโหนดเริ่มต้นของการค้นหาข้อมูล ถ้าต้องการเลือกเอกสาร XML

อีก กลับไปทำข้อที่ 1

- 4 ผู้ใช้เลือกโหนดที่ต้องการค้นหาจากโครงสร้างเอกสาร XML ที่โปรแกรมแสดง
- 5 ผู้ใช้กำหนดเงื่อนไขของการค้นหาข้อมูล
- 6 ผู้ใช้เลือกโหนดที่ใช้ในการเรียงลำดับ
- 7 ผู้ใช้กำหนดรูปแบบของรูตเอลิเมนต์
- 8 โปรแกรมแปลงสิ่งที่ผู้ใช้กำหนดให้อยู่ในรูปของไวยากรณ์ XQuery และสร้าง

ไฟล์

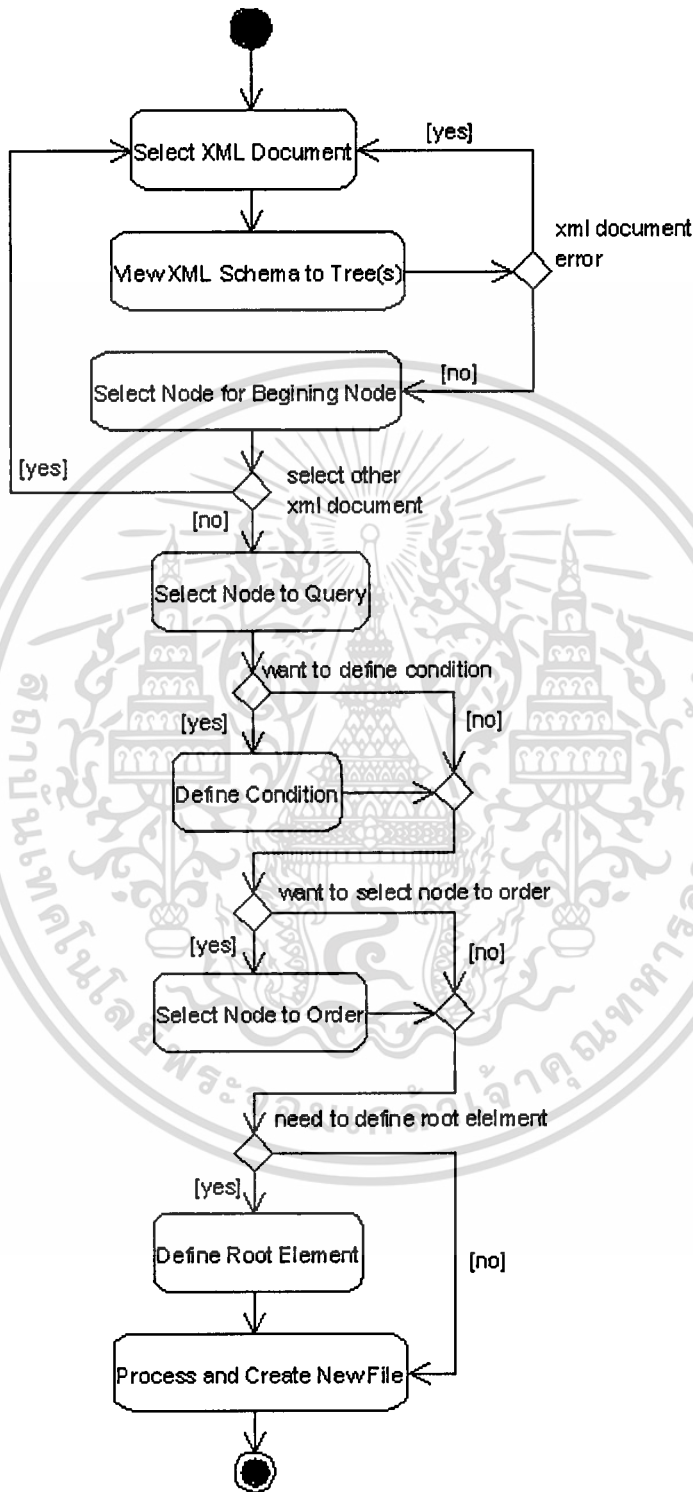
ขั้นตอนทางเลือก

- 2.1 เอกสาร XML เกิดข้อผิดพลาด: โปรแกรมแสดงข้อความเตือน กลับไปทำข้อที่

1

- 5.1 ผู้ใช้ไม่กำหนดเงื่อนไขของการค้นหา ไปทำข้อที่ 6
- 6.1 ผู้ใช้กำหนดการเรียงลำดับตามลำดับข้อมูลใน XML ไปทำข้อที่ 7
- 7.1 โปรแกรมไม่แสดงให้กำหนดรูปแบบของรูตเอลิเมนต์ เนื่องจากมีการเลือกค้นหาจากเอลิเมนต์เพียงเอลิเมนต์เดียว ไปทำข้อที่ 8

แผนภาพแอคทีวิตี



ภาพที่ 4.3 แผนภาพแอคทีวิตีของการสร้างไวยากรณ์ XQuery ด้วยตัวช่วยสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.3 คำอธิบายยูสเคส View XML Schema Tree(s)

คำอธิบาย

ยูสเคสนี้มีหน้าที่นำเอกสาร XML ที่ต้องการค้นหาข้อมูลมาแสดงเป็นรูปแบบกราฟิก โครงสร้างลำดับชั้นแบบต้นไม้ ซึ่งจะช่วยให้ผู้ใช้เห็นภาพรวมของเอกสาร XML ก่อนลงมือสร้าง และ/หรือแก้ไขไวยากรณ์ XQuery

แอกเตอร์ User

เงื่อนไขก่อน เอกสาร XML อยู่ในรูปแบบถูกต้อง

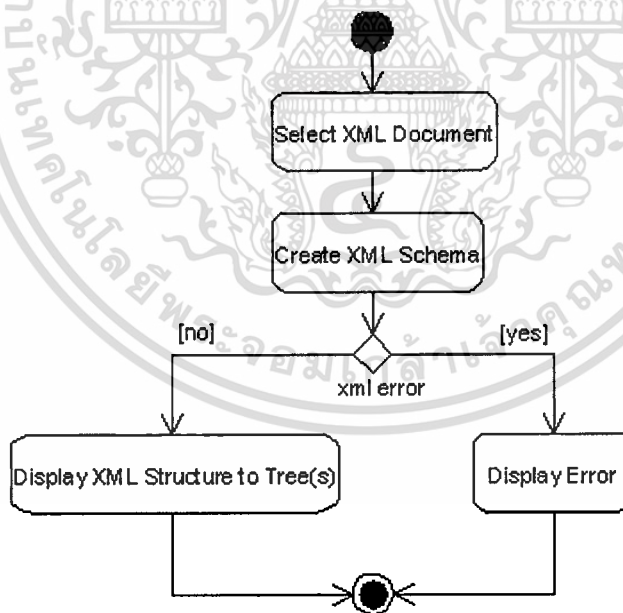
ขั้นตอนการทำงาน

- 1 ผู้ใช้เลือกเอกสาร XML ที่ต้องการแสดงโครงสร้าง
- 2 โปรแกรมสร้าง XML Schema ของเอกสาร XML ที่กำหนด
- 3 โปรแกรมแสดงโครงสร้างของเอกสาร XML

ขั้นตอนทางเลือก

- 2.1 เอกสาร XML เกิดข้อผิดพลาด: โปรแกรมแสดงข้อความเตือน จบการทำงาน

แผนภาพแอกทิวิตี



ภาพที่ 4.4 แผนภาพแอกทิวิตีของการแสดงโครงสร้างของเอกสาร XML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.4 คำอธิบายยูสเคส Execute XQuery Expression

คำอธิบาย

ยูสเคสนี้มีหน้าที่ตรวจสอบความถูกต้องและประมวลผลไวยากรณ์ XQuery โดยโปรแกรม นำชุดไวยากรณ์ XQuery ไปให้ XQuery Engine ประมวลผล

แอกเตอร์ User และ XQuery Engine

เงื่อนไขก่อน ผู้ใช้ต้องสร้างไวยากรณ์ XQuery แล้ว

ขั้นตอนการทำงาน

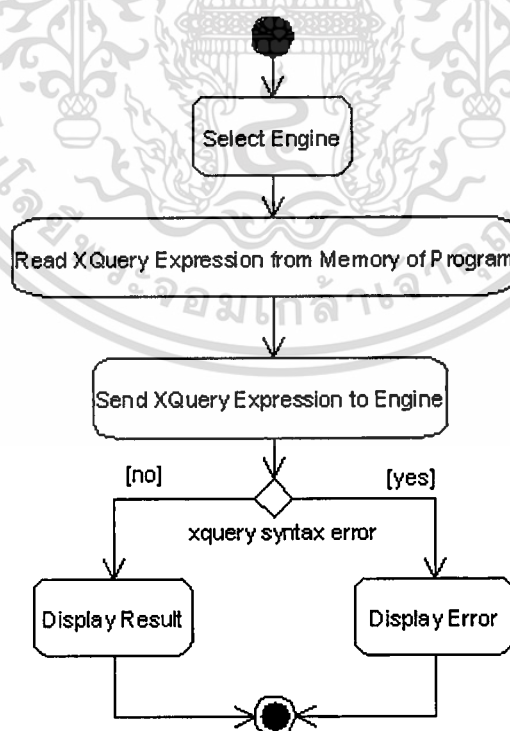
- 1 ผู้ใช้เลือก Engine ที่ใช้การประมวลผล
- 2 ผู้ใช้คลิกปุ่ม Execute: โปรแกรมอ่านไวยากรณ์ XQuery จากหน่วยความจำ
- 3 โปรแกรมสั่งให้ Engine ที่ผู้ใช้กำหนด ประมวลผลไวยากรณ์ XQuery
- 4 โปรแกรมแสดงผลลัพธ์

ขั้นตอนทางเลือก

- 3.1 ไวยากรณ์ XQuery เกิดข้อผิดพลาด: โปรแกรมแสดงข้อความเตือน จบการ

ทำงาน

แผนภาพแอกทิวิตี



ภาพที่ 4.5 แผนภาพแอกทิวิตีของการประมวลผลไวยากรณ์ XQuery

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.5 คำอธิบายยูสเคส Update XQuery Expression

คำอธิบาย

ยูสเคสนี้ทำการอัปเดตชุดไวยากรณ์ XQuery โดยผู้ใช้เปิดไฟล์นามสกุล xquery ที่เคยบันทึกไว้แล้วขึ้นมาแก้ไข แล้วบันทึกลงบนงานบันทึกเฉพาะที่

แอกเตอร์ User

เงื่อนไขก่อน ผู้ใช้ต้องมีไฟล์นามสกุล xquery

ขั้นตอนการทำงาน

- 1 ผู้ใช้เปิดไฟล์นามสกุล xquery จากงานบันทึกเฉพาะที่
- 2 ผู้ใช้แก้ไขไวยากรณ์ XQuery
- 3 ผู้ใช้ร้องขอโปรแกรมบันทึกไฟล์
- 4 โปรแกรมบันทึกไฟล์

แผนภาพแอกทिवิตี



ภาพที่ 4.6 แผนภาพแอกทिवิตีของการอัปเดตไวยากรณ์ XQuery

4.2.2.6 คำอธิบายยูสเคส Generate XML Document

คำอธิบาย

ยูสเคสนี้ทำการสร้างเอกสาร XML จากผลลัพธ์ที่ได้จากการประมวลชุดไวยากรณ์ XQuery แล้วบันทึกลงบนงานบันทึกเฉพาะที่

แอกเตอร์ User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เงื่อนไขก่อน ผู้ใช้ต้องมีผลลัพธ์จากการประมวลผลไวยากรณ์ XQuery

ขั้นตอนการทำงาน

- 1 ผู้ใช้คลิกปุ่ม Export: โปรแกรมอ่านค่าผลลัพธ์จากการประมวลผลไวยากรณ์ XQuery จากหน่วยความจำ
- 2 โปรแกรมบันทึกผลลัพธ์ออกเป็นเอกสาร XML (ไฟล์นามสกุล xml) ลงบนงานบันทึกเฉพาะที่

แผนภาพแอกทิวิตี



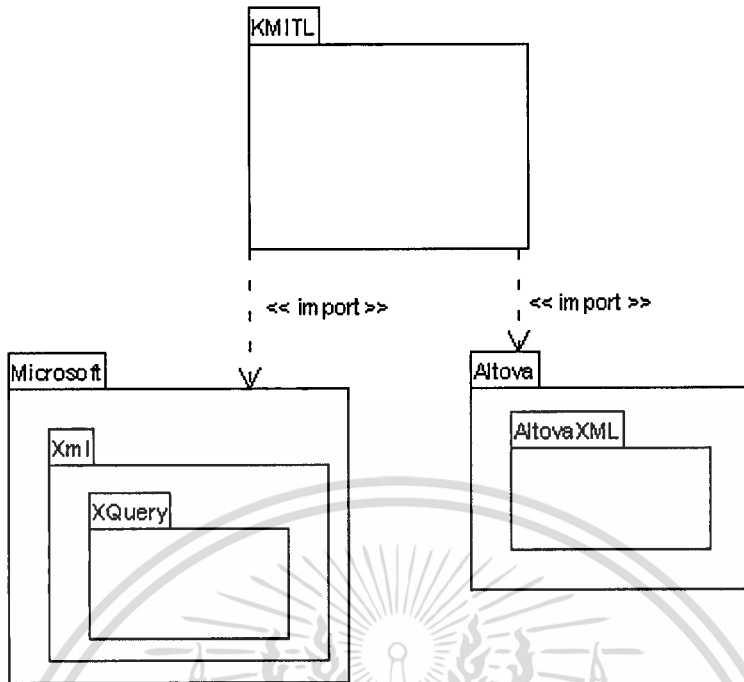
ภาพที่ 4.7 แผนภาพแอกทิวิตีของการสร้างเอกสาร XML จากผลลัพธ์

4.3 แผนภาพคลาส

ในส่วนนี้จะเป็นการออกแบบแผนภาพคลาสต่างๆ ของโปรแกรมว่าประกอบด้วยคลาสใด และคลาสนั้นมีคุณสมบัติและเมธอดใดบ้าง รวมถึงแสดงความสัมพันธ์ระหว่างคลาส ซึ่งแผนภาพคลาสทั้งหมด มีดังนี้

4.4.1 แผนภาพคลาสของโปรแกรม

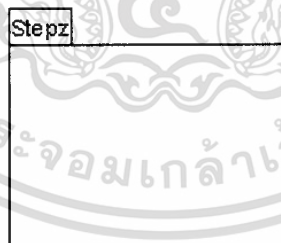
แผนภาพคลาสของโปรแกรม เป็นการแสดงภาพโดยรวมของโปรแกรม ซึ่งแสดงเป็นแพ็คเกจที่รวบรวมคลาสต่างๆ โดยแบ่งเป็น 3 แพ็คเกจ คือ แพ็คเกจ KMITL รวบรวมคลาสของโปรแกรม แพ็คเกจ Microsoft และแพ็คเกจ Altova รวบรวมคลาสที่เกี่ยวกับส่วนประมวลผลไวยากรณ์ XQuery ที่โปรแกรมเรียกใช้ ดังภาพที่ 4.8



ภาพที่ 4.8 แผนภาพคลาสของโปรแกรม

4.4.2 แผนภาพคลาสของแพ็คเกจ KMITL

แผนภาพคลาสของแพ็คเกจ KMITL มีแพ็คเกจย่อย คือ Stepz ซึ่งเป็นแพ็คเกจที่รวบรวมคลาสของโปรแกรม ดังภาพที่ 4.9

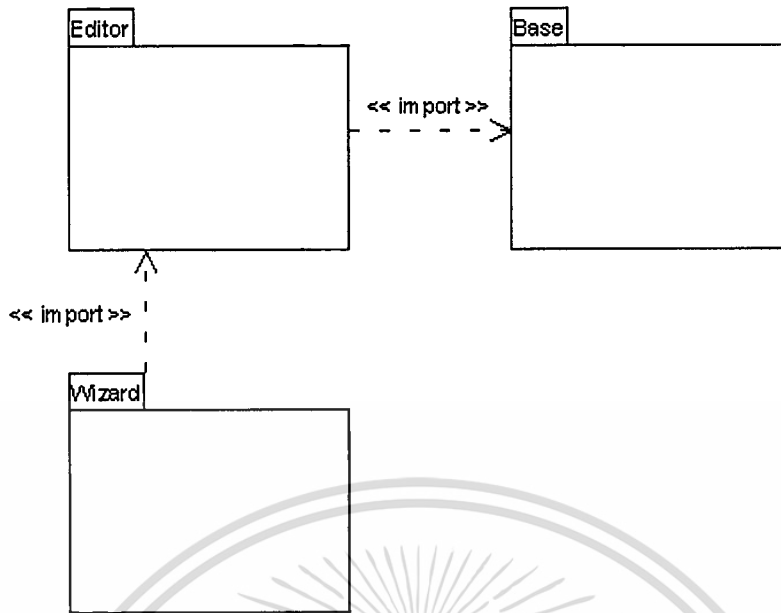


ภาพที่ 4.9 แผนภาพคลาสของแพ็คเกจ KMITL

4.4.3 แผนภาพคลาสของแพ็คเกจ Stepz

แผนภาพคลาสของแพ็คเกจ Stepz มีแพ็คเกจย่อย 3 แพ็คเกจ คือ แพ็คเกจ Base รวบรวมคลาสที่เกี่ยวกับการทำงานพื้นฐานของเอดิเตอร์ แพ็คเกจ Editor รวบรวมคลาสที่เกี่ยวกับการทำงานกับเอดิเตอร์ และแพ็คเกจ Wizard ที่รวบรวมคลาสที่เกี่ยวกับตัวช่วยสร้างไวยากรณ์ XQuery ดังภาพที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.10 แผนภาพคลาสของแพ็คเกจ Stepz

4.4.4 แผนภาพคลาสของแพ็คเกจ Base

แผนภาพคลาสของแพ็คเกจ Base ประกอบด้วย 2 คลาส คือ คลาส Editor และคลาส XQueryEditor ซึ่งคลาส XQueryEditor จะมีการสืบทอดมาจากคลาส Editor ด้วย ดังภาพที่ 4.11 สำหรับรายละเอียดของแต่ละคลาสมีดังนี้

- คลาส KMITL.Stepz.Base.Editor ประกอบด้วยคุณสมบัติและเมธอด ดังต่อไปนี้

คุณสมบัติ มีดังนี้

Private DocumentName:String

เก็บชื่อไฟล์

เมธอด มีดังนี้

Protected New(FilePath:String)

กำหนดคอนสตรัคเตอร์ในการสร้างอินสแตนซ์ใหม่ของคลาส

Protected New(FilePath:String,Text:String)

กำหนดคอนสตรัคเตอร์ในการสร้างอินสแตนซ์ใหม่ของคลาส

Private ReadFile(FilePath:String):String

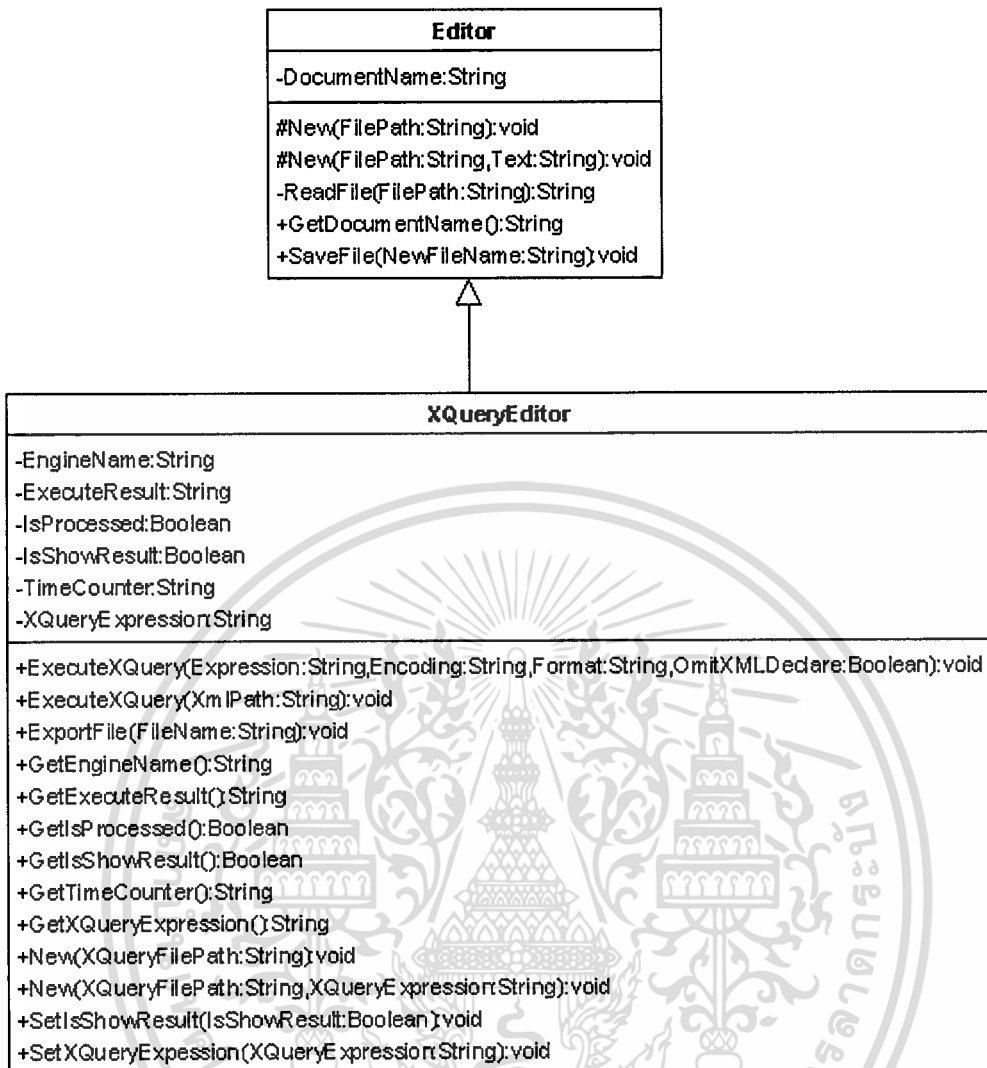
อ่านข้อมูลจากไฟล์ที่กำหนด

คืนค่าเป็น สตริง

Public GetDocumentName():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ DocumentName

คืนค่าเป็น สตริง



ภาพที่ 4.11 แผนภาพคลาสของแพคเกจ Base

Public SaveFile(NewFileName:String)

บันทึกข้อมูลลงไฟล์ตามชื่อที่กำหนด

▪ คลาส KMITL.Stepz.Base.XQueryEditor ประกอบด้วยคุณสมบัติและเมธอด
ดังต่อไปนี้

คุณสมบัติ มีดังนี้

Private EngineName:String

เก็บชื่อ Engine ที่ถูกใช้ในการประมวลผล

Private ExecuteResult:String

เก็บผลลัพธ์จากการประมวลผล

Private IsProcessed:Boolean

เก็บค่าบูลว่า มีการประมวลผลหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private IsShowResult:Boolean

เก็บค่าบูลว่า มีการแสดงผลลัพธ์แล้วหรือไม่

Private TimeCounter:String

เก็บเวลาที่ใช้ในการประมวลผล

Private XQueryExpression:String

เก็บนิพจน์ XQuery

เมธอด มีดังนี้

Public ExecuteXQuery(Expression:String, Encoding:String, Format:String,
OmitXMLDeclare:Boolean)

ประมวลผลไวยากรณ์ XQuery ด้วย Altova Engine

Public ExecuteXQuery(XmlPath:String)

ประมวลผลไวยากรณ์ XQuery ด้วย Microsoft Engine

Public ExportFile(FileName:String)

นำออกผลลัพธ์ บันทึกเป็นไฟล์ XML

Public GetEngineName():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ EngineName

คืนค่าเป็น สตริง

Public GetExecuteResult():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ ExecuteResult

คืนค่าเป็น สตริง

Public GetIsProcessed():Boolean

เข้าถึงค่าที่ถูกเก็บอยู่ที่ IsProcessed

คืนค่าเป็น บูล

Public GetIsShowResult():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ IsShowResult

คืนค่าเป็น บูล

Public GetTimeCounter():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ TimeCounter

คืนค่าเป็น สตริง

Public GetXQueryExpression():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ XQueryExpression

เอกสารนี้เป็นเอกสารที่สงวนไว้ **คืนค่าเป็น สตริง** เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Public New(XQueryFilePath:String)

กำหนดคอนสตรัคเตอร์ในการสร้างอินสแตนซ์ใหม่ของคลาส

Public New(XQueryFilePath:String, XQueryExpression:String)

กำหนดคอนสตรัคเตอร์ในการสร้างอินสแตนซ์ใหม่ของคลาส

Public SetIsShowResult(IsShowResult:Boolean)

กำหนดค่าให้กับ IsShowResult

Public SetXQueryExpression(XQueryExpression:String)

กำหนดค่าให้กับ XQueryExpression

4.4.5 แผนภาพคลาสของแพ็คเกจ Editor

แผนภาพคลาสของแพ็คเกจ Editor ประกอบด้วย 2 คลาส คือ คลาส FormXQueryEditor และคลาส DialogSaveAll ดังภาพที่ 4.12

■ คลาส KMITL.Stepz.Editor.FormXQueryEditor ประกอบด้วยคุณสมบัติและเมธอดดังต่อไปนี้

คุณสมบัติ มีดังนี้

Private Documents:XQueryEditor

เก็บอ็อบเจกต์ XQueryEditor ที่สร้างขึ้นทั้งหมดขณะทำงานกับโปรแกรม

เมธอด มีดังนี้

Private CloseDocument()

รองรับเหตุการณ์เมื่อผู้ใช้ปิดแท็บเอกสารใดๆ

Private CloseFormXQueryEditor()

รองรับเหตุการณ์เมื่อผู้ใช้ออกจากโปรแกรม มีการเรียก DialogSaveAll ทำงาน

Private DisplayError(ErrorText:String)

แสดงข้อความเตือนเมื่อเกิดข้อผิดพลาด

Private ExecuteDocument(Xe:XQueryEditor)

ประมวลผลไวยากรณ์ XQuery

Private ExportResult(FileName:String)

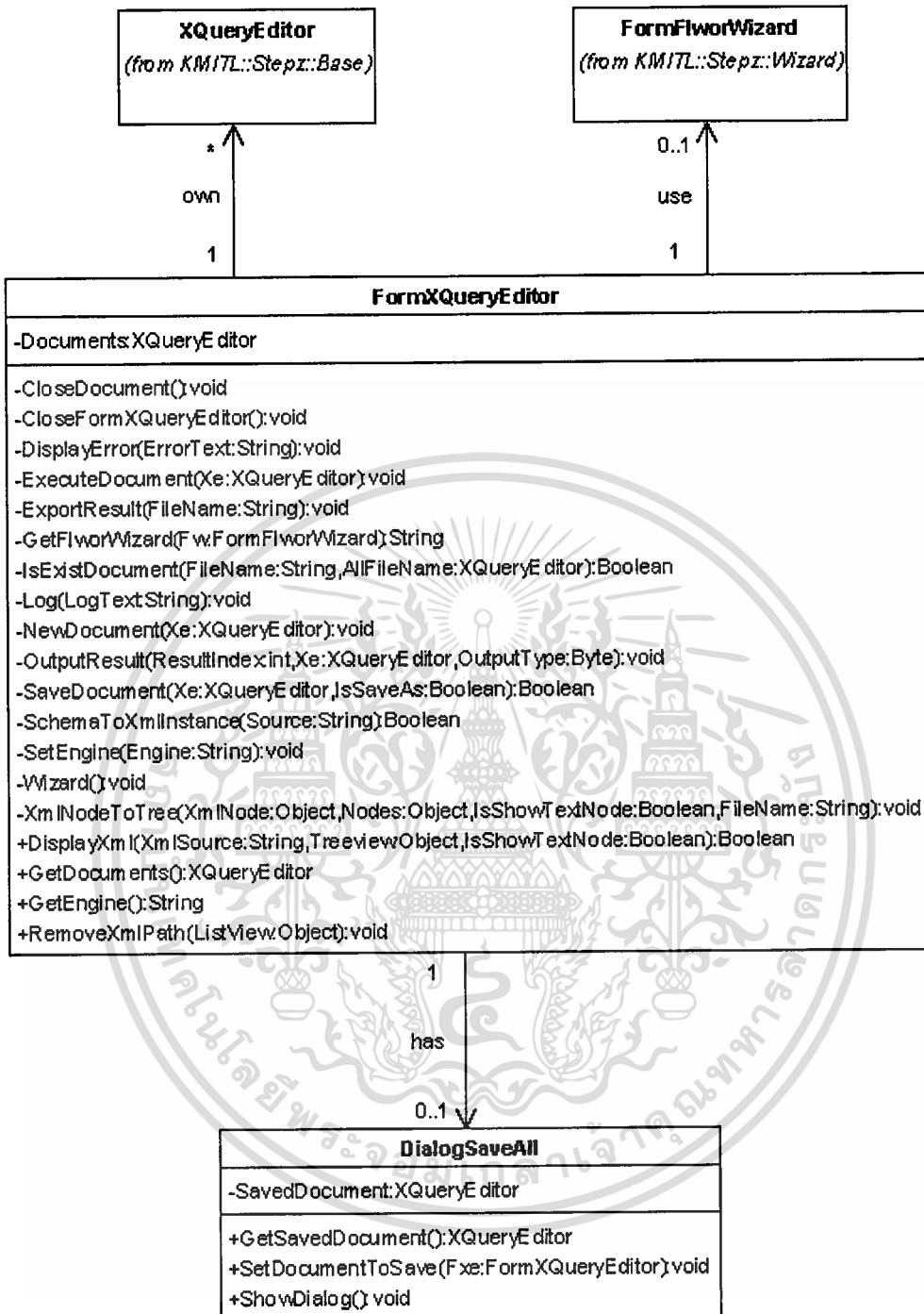
นำออกผลลัพธ์ บันทึกเป็นไฟล์ XML

Private GetFlworWizard(Fw:FormFlworWizard):String

รับนิพจน์ FLWOR จากตัวช่วยสร้าง

คืนค่าเป็น สตริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.12 แผนภาพคลาสของแพ็คเกจ Editor

Private IsExistDocument(FileName:String, AllFileName:XQueryEditor)

:Boolean

ตรวจสอบไฟล์ที่จะถูกเปิดว่า มีการทำงานอยู่กับโปรแกรมแล้วหรือไม่

คืนค่าเป็นจริง ถ้ามีการเปิดไฟล์นั้นแล้ว

คืนค่าเป็นเท็จ ถ้ายังไม่มีเปิดไฟล์นั้น

Private Log(LogText:String)

บันทึกการทำงานต่างๆ ที่เกิดกับโปรแกรม

Private NewDocument(Xe:XQueryEditor)

สร้างเอดิเตอร์ขึ้นมาใหม่

Private OutputResult(ResultIndex:int, Xe:XQueryEditor, OutputType:Byte)

กำหนดรูปแบบผลลัพธ์ว่าจะให้แสดงรูปแบบใด มี 3 รูปแบบ คือ
แบบเท็กซ์ แบบโครงสร้างต้นไม้ และแบบอินเทอร์เน็ทเอ็กซ์พลอเรอร์

Private SaveDocument(Xe:XQueryEditor, IsSaveAs:Boolean):Boolean

บันทึกข้อมูลลงไฟล์

คืนค่าเป็นจริง ถ้าสามารถบันทึกได้สำเร็จ

คืนค่าเป็นเท็จ ถ้าไม่สามารถบันทึกได้

Private SchemaToXmlInstance(Source:String):Boolean

สร้าง XML Schema จากเอกสาร XML ที่กำหนด

คืนค่าเป็นจริง ถ้าสามารถสร้าง XML Schema สำเร็จ

คืนค่าเป็นเท็จ ถ้าเกิดข้อผิดพลาด สร้าง XML Schema ไม่สำเร็จ

Private SetEngine(Engine:String)

กำหนด Engine ที่จะใช้ในการประมวลไวยากรณ์ XQuery

Private Wizard()

แสดงฟอร์มตัวช่วยสร้างไวยากรณ์ XQuery

Private XmlNodeToTree(XmlNode:Object, Nodes:Object, IsShowTextNode
:Boolean, FileName:String)

อ่านโหนดจากเอกสาร XML มาแสดงอยู่ในรูปแบบโครงสร้างต้นไม้

Public DisplayXml(XmlSource:String, Treeview:Object, IsShowTextNode
:Boolean):Boolean

แสดงโครงสร้างเอกสาร XML มีการทำงานร่วมกับ XmlNodeToTree
และ SchemaToXmlInstance ด้วย

คืนค่าเป็นจริง ถ้าสามารถแสดงโครงสร้างเอกสาร XML ได้

คืนค่าเป็นเท็จ ถ้าเกิดข้อผิดพลาด ไม่สามารถแสดงโครงสร้างเอกสาร
XML ได้

Public GetDocuments():XQueryEditor

เข้าถึงค่าที่ถูกเก็บอยู่ที่ Documents

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Public GetEngine():String

คืนค่าเป็น สตริง เป็นค่าของ Engine คือ Altova Engine หรือ Microsoft Engine

Public RemoveXmlPath(ListView:Object)

ลบพารของเอกสาร XML

▪ คลาส KMITL.Stepz.Editor.DialogSaveAll ประกอบด้วยคุณสมบัติและเมธอด ดังต่อไปนี้

คุณสมบัติ มีดังนี้

Private SavedDocument:XQueryEditor

เก็บอ็อบเจกต์ XQueryEditor ที่ทำงานอยู่ ที่มีการเปลี่ยนแปลง ซึ่งเมื่อปิด โปรแกรม แล้วต้องการการบันทึก

เมธอด มีดังนี้

Public GetSavedDocument():XQueryEditor

เข้าถึงค่าที่ถูกเก็บอยู่ที่ SavedDocument
คืนค่าเป็น XQueryEditor

Public SetDocumentToSave(Fxe:FormXQueryEditor)

รับค่าจาก FormXQueryEditor แล้วเข้าถึงค่า Document ว่ามีเอกสาร ใดบ้างที่ต้องการบันทึก แล้วเก็บลงใน SavedDocument

Public ShowDialog()

แสดงไดอะล็อกรายชื่อไฟล์ที่ต้องการการบันทึก

4.4.6 แผนภาพคลาสของแพคเกจ Wizard

แผนภาพคลาสของแพคเกจ Wizard ประกอบด้วย 5 คลาส คือ คลาส FormFlworWizard คลาส FormContextNode คลาส FormCondition คลาส FormQuantified และคลาส FormFlworLog ดังภาพที่ 4.13

สำหรับรายละเอียดของแต่ละคลาสมีดังนี้

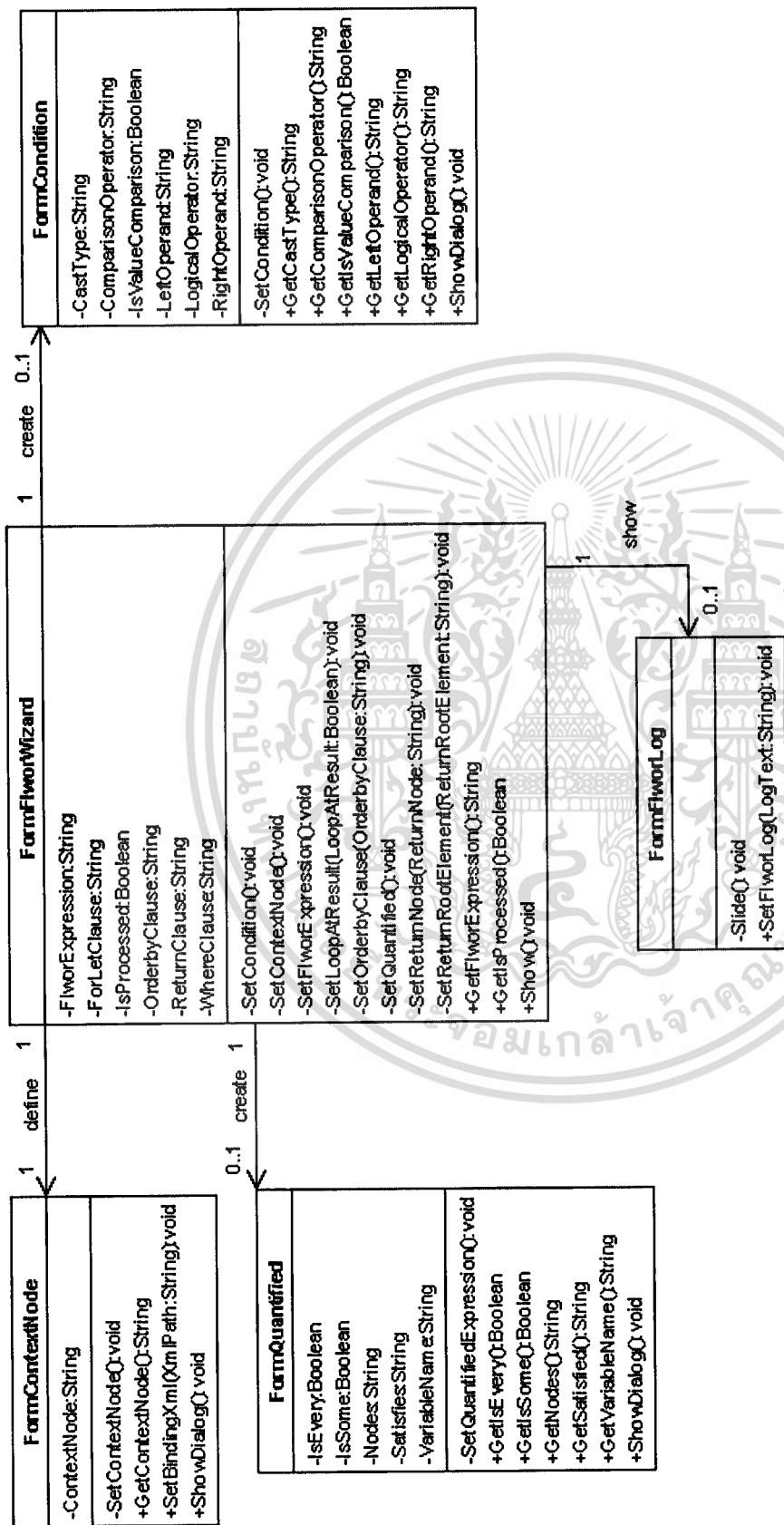
▪ คลาส KMITL.Stepz.Wizard.FormFlworWizard ประกอบด้วยคุณสมบัติและเมธอด ดังต่อไปนี้

คุณสมบัติ มีดังนี้

Private FlworExpression:String

เก็บผลลัพธ์นิพจน์ FLWOR ที่สร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.13 แผนภาพคลาสของแพคเกจ Wizard

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private ForLetClause:String

เก็บนิพจน์ FLWOR ในส่วนประโยค For และ Let

Private IsProcessed:Boolean

เก็บค่าบูลว่า มีการสร้างนิพจน์ FLWOR สำเร็จหรือไม่

Private OrderbyClause:String

เก็บนิพจน์ FLWOR ในส่วนประโยค Orderby

Private ReturnClause:String

เก็บนิพจน์ FLWOR ในส่วนประโยค Return

Private WhereClause:String

เก็บนิพจน์ FLWOR ในส่วนประโยค Where

เมธอด มีดังนี้

Private SetCondition()

กำหนดเงื่อนไขด้วย FormCondition

Private SetContextNode()

กำหนดโหนดที่ใช้ในการเริ่มต้นการค้นหา ด้วย FormContextNode

Private SetFlworExpression()

รวบรวมคำสั่งทั้งหมดแล้วสร้างนิพจน์ FLWOR แล้วเก็บที่ FlworExpression

Private SetLoopAtResult(LoopAtResult:Boolean)

กำหนดค่าว่าให้มีการวนลูปที่ผลลัพธ์หรือไม่

Private SetOrberbyClause(OrderbyClause:String)

กำหนดประโยคสำหรับการเรียงลำดับ

Private SetQuantified()

กำหนดนิพจน์ Quantified ด้วย FormQuantified

Private SetReturnNode(Node:String)

กำหนดโหนดที่ต้องการค้นหาและแสดงผล

Private SetReturnRootElement(ReturnRootElement:String)

กำหนดรีเทิร์นรูตเอลิเมนต์

Public GetFlworExpression():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ FlworExpression

คืนค่าเป็น สตริง เป็นค่าของผลลัพธ์นิพจน์ FLWOR ที่สร้างขึ้น

เอกสารนี้เป็นเอกสาร **Public GetIsProcessed():Boolean** ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าถึงค่าที่ถูกเก็บอยู่ที่ IsProcessed

คืนค่าเป็นจริง ถ้าสามารถสร้างนิพจน์ FLWOR สำเร็จ

คืนค่าเป็นเท็จ ถ้ามีการยกเลิกการใช้ตัวช่วยสร้าง

Public Show()

แสดงฟอร์มตัวช่วยสร้างนิพจน์ FLWOR

▪ คลาส KMITL.Stepz.Wizard.FormContextNode ประกอบด้วยคุณสมบัติและเมธอดดังต่อไปนี้

คุณสมบัติ มีดังนี้

Private ContextNode:String

เก็บโหนดที่ใช้ในการเริ่มต้นการค้นหา

เมธอด มีดังนี้

Private SetContextNode()

กำหนดค่าให้กับ ContextNode

Public GetContextNode():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ ContextNode

คืนค่าเป็น สตริง เป็นค่าของโหนดที่ใช้ในการเริ่มต้นการค้นหาที่ผู้ใช้

กำหนด

Public SetBindingXml(XmlPath:String)

กำหนดเอกสาร XML เพื่อนำไปใช้แสดงโครงสร้างต้นไม้ เพื่อให้ผู้ใช้

คลิกเลือกโหนดที่ต้องการ

Public ShowDialog()

แสดงไดอะล็อกให้ผู้ใช้กำหนด โหนดที่ใช้ในการเริ่มต้นการค้นหา

▪ คลาส KMITL.Stepz.Wizard.FormCondition ประกอบด้วยคุณสมบัติและเมธอดดังต่อไปนี้

คุณสมบัติ มีดังนี้

Private CastType:String

เก็บชนิดข้อมูลสำหรับการเปรียบเทียบ

Private ComparisonOperator:String

เก็บ โอเปอเรเตอร์สำหรับการเปรียบเทียบ

Private IsValueComparison:Boolean

เก็บค่าบูลว่า โอเปอเรเตอร์สำหรับการเปรียบเทียบเป็นแบบ Value

เอกสารนี้เป็นเอกสารที่สงวนไว้หรือไม่การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private LeftOperand:String

เก็บโอเปอเรนด์ที่อยู่ข้างซ้าย

Private LogicalOperator:String

เก็บโอเปอเรเตอร์เชิงตรรกะ

Private RightOperand:String

เก็บโอเปอเรนด์ที่อยู่ข้างขวา

เมธอด มีดังนี้

Private SetCondition()

กำหนดค่าให้กับ CastType, ComparisonOperator, IsValueComparison, LeftOperand, LogicalOperator และ RightOperand

Public GetCastType():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ CastType

คืนค่าเป็น สตริง เป็นค่าของชนิดข้อมูลสำหรับการเปรียบเทียบ คือ

xs:string, xs:date, xs:integer, xs:decimal และ xs:double

Public GetComparisonOperator():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ ComparisonOperator

คืนค่าเป็น สตริง เป็นค่าของโอเปอเรเตอร์ eq, ne, lt, le, gt, ge, =, !=, <, <=, > และ >=

Public GetIsValueComparison():Boolean

เข้าถึงค่าที่ถูกเก็บอยู่ที่ IsValueComparison

คืนค่าเป็นจริง ถ้ามีการใช้โอเปอเรเตอร์ eq, ne, lt, le, gt และ ge

คืนค่าเป็นเท็จ ถ้ามีการใช้โอเปอเรเตอร์ =, !=, <, <=, > และ >=

Public GetLeftOperand():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ LeftOperand

คืนค่าเป็น สตริง เป็นค่าของโอเปอเรนด์ที่อยู่ข้างซ้ายที่ผู้ใช้กำหนด

Public GetLogicalOperator():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ LogicalOperator

คืนค่าเป็น สตริง เป็นค่าของโอเปอเรเตอร์ and และ or

Public GetRightOperand():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ RightOperand

คืนค่าเป็น สตริง เป็นค่าของโอเปอเรนด์ที่อยู่ข้างขวาที่ผู้ใช้กำหนด

เอกสารนี้เป็นเอกสาร **Public ShowDialog()** งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงไคอะล็อกให้ผู้ใช้กำหนดเงื่อนไขในการค้นหา

■ คลาส KMITL.Stepz.Wizard.FormQuantified ประกอบด้วยคุณสมบัติและเมธอดดังต่อไปนี้

คุณสมบัติ มีดังนี้

Private IsEvery:Boolean

เก็บค่าบูลว่า มีคีย์เวิร์ดเป็น Every หรือไม่

Private IsSome:Boolean

เก็บค่าบูลว่า มีคีย์เวิร์ดเป็น Some หรือไม่

Private Nodes:String

เก็บ โหนดที่จะถูก Satisfies

Private Satisfies:String

เก็บนิพจน์ Quantified ในส่วนประโยค Satisfies

Private VariableName:String

เก็บชื่อตัวแปร

เมธอด มีดังนี้

Private SetQuantifiedExpression()

กำหนดค่าให้กับ IsEvery, IsSome, Nodes, Satisfied และ VariableName

Public GetIsEvery():Boolean

เข้าถึงค่าที่ถูกเก็บอยู่ที่ IsEvery

คืนค่าเป็นจริง ถ้ามีการเลือกใช้คีย์เวิร์ด every

Public GetIsSome():Boolean

เข้าถึงค่าที่ถูกเก็บอยู่ที่ IsSome

คืนค่าเป็นจริง ถ้ามีการเลือกใช้คีย์เวิร์ด some

Public GetNodes():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ Nodes

คืนค่าเป็น สตริง เป็นค่าของโหนดที่จะถูก Satisfies ที่ผู้ใช้กำหนด

Public GetSatisfied():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ Satisfied

คืนค่าเป็น สตริง เป็นค่าของประโยค Satisfies ที่ผู้ใช้กำหนด

Public GetVariableName():String

เข้าถึงค่าที่ถูกเก็บอยู่ที่ VariableName

เอกสารนี้เป็นเอกสารที่สงวนไว้คืนค่าเป็น สตริง เป็นค่าของตัวแปรที่ผู้ใช้กำหนด ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Public ShowDialog()

แสดงไดอะล็อกให้ผู้ใช้งานกำหนดเงื่อนไขในการค้นหาแบบนิพจน์

Quantified

- คลาส KMITL.Stepz.Wizard.FormFlworLog ประกอบด้วยเมธอด ดังต่อไปนี้

Private Slide()

แสดงฟอร์มรายการการทำงานที่เกิดขึ้นขณะสร้างนิพจน์ XQuery ด้วยตัวช่วยสร้าง

Public SetFlworLog(LogText:String)

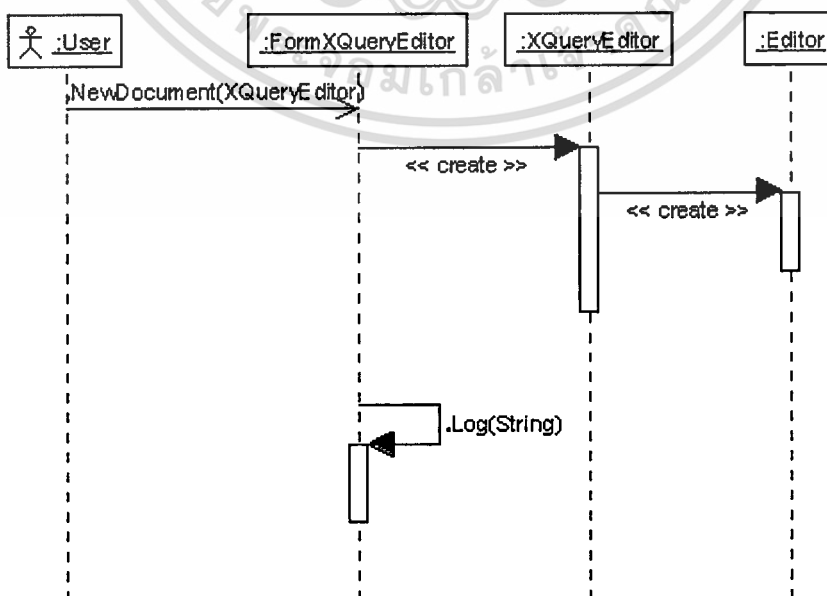
กำหนดรายการการทำงานที่เกิดขึ้นขณะสร้างนิพจน์ XQuery ด้วยตัวช่วยสร้าง

4.4 แผนภาพซีเควนซ์

ในส่วนนี้จะเป็นการลงลึกในเชิงฟังก์ชันการทำงานของโปรแกรม ซึ่งจะแสดงให้เห็นถึงการมีส่วนร่วมในแต่ละคลาสของโปรแกรมว่าคลาสใดทำงานกับคลาสใด เพื่อนำไปเป็นพื้นฐานในการเขียนโปรแกรม ซึ่งลำดับการทำงานในเชิงฟังก์ชันของโปรแกรมที่สำคัญๆ มีดังนี้

4.5.1 ลำดับการทำงานของการสร้างหน้าจอเอดิเตอร์ใหม่

ลำดับการทำงานของการสร้างหน้าจอเอดิเตอร์ใหม่เริ่มต้นจากที่ผู้ใช้คลิกปุ่ม New เพื่อสร้างเอดิเตอร์ใหม่ ดังภาพที่ 4.14



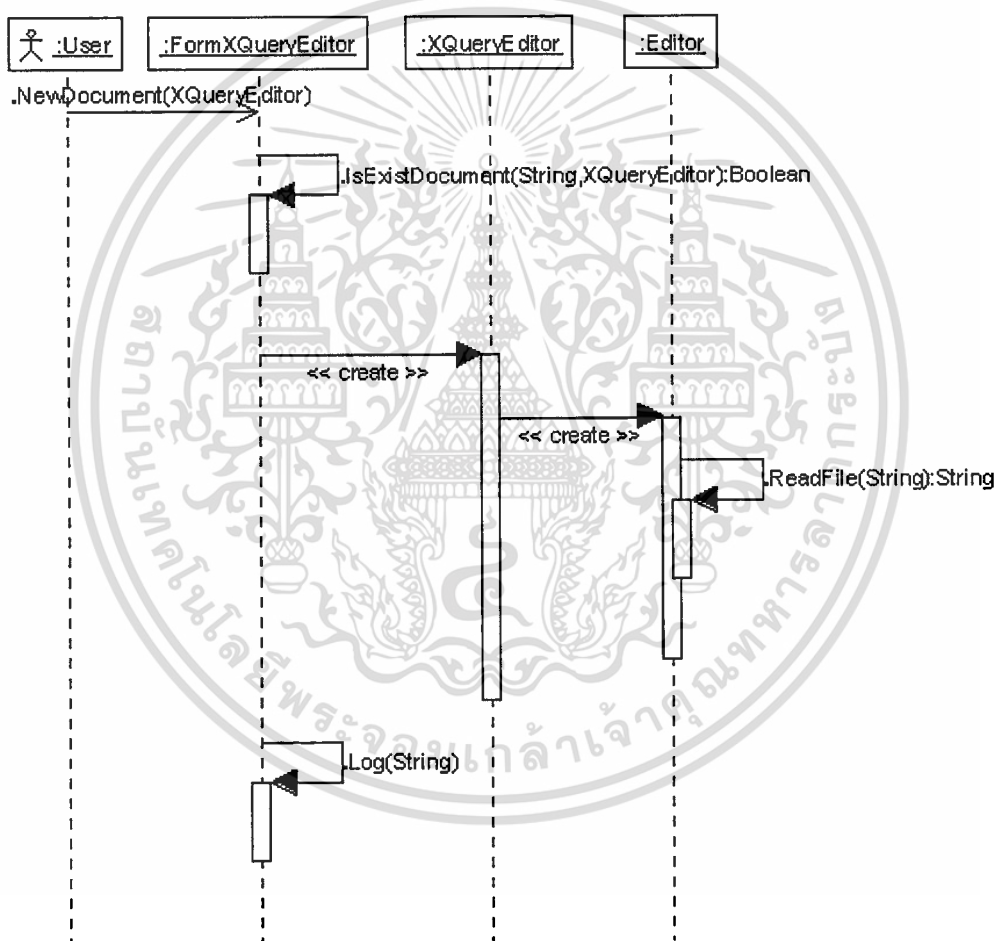
ภาพที่ 4.14 แผนภาพซีเควนซ์การสร้างหน้าจอเอดิเตอร์ใหม่

ขั้นตอนลำดับการสร้างหน้าจอดีเตอร์ใหม่ มีดังนี้

- ผู้ใช้คลิกปุ่ม New
- สร้างหน้าจอดีเตอร์ใหม่
- บันทึกรายการการทำงานกับโปรแกรม

4.5.2 ลำดับการทำงานของการเปิดไฟล์

ลำดับการทำงานของการเปิดไฟล์เริ่มต้นจากที่ผู้ใช้คลิกปุ่ม Open แล้วเลือกไฟล์นามสกุล xquery จากนั้น โปรแกรมจะอ่านข้อมูลในไฟล์แล้วแสดงที่หน้าจอดีเตอร์ ดังภาพที่ 4.15



ภาพที่ 4.15 แผนภาพซีควเอนซ์การเปิดไฟล์

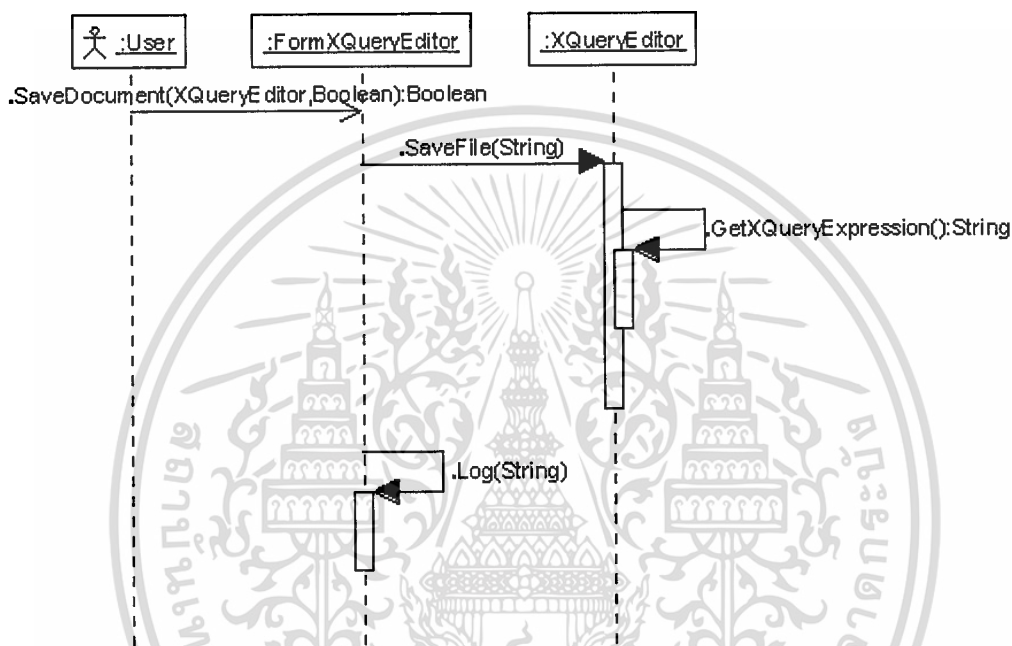
ขั้นตอนลำดับการเปิดไฟล์ มีดังนี้

- ผู้ใช้คลิกปุ่ม Open แล้วเลือกไฟล์นามสกุล xquery
- ตรวจสอบว่าไฟล์ที่กำลังจะเปิด ทำงานอยู่ในโปรแกรมแล้วหรือไม่
- สร้างเอดิเตอร์ใหม่
- อ่านข้อมูลจากไฟล์เข้าเอดิเตอร์

- บันทึกรายการการทำงานกับโปรแกรม

4.5.3 ลำดับการทำงานของการบินทักไวยากรณ์ XQuery

ลำดับการทำงานของการบินทักไวยากรณ์ XQuery เริ่มต้นจากที่ผู้ใช้คลิกปุ่ม Save แล้วกำหนดชื่อไฟล์ พร้อมตำแหน่งจัดเก็บ จากนั้นโปรแกรมจะบันทึกข้อมูลลงในไฟล์เก็บยังตำแหน่งที่กำหนด ดังภาพที่ 4.16



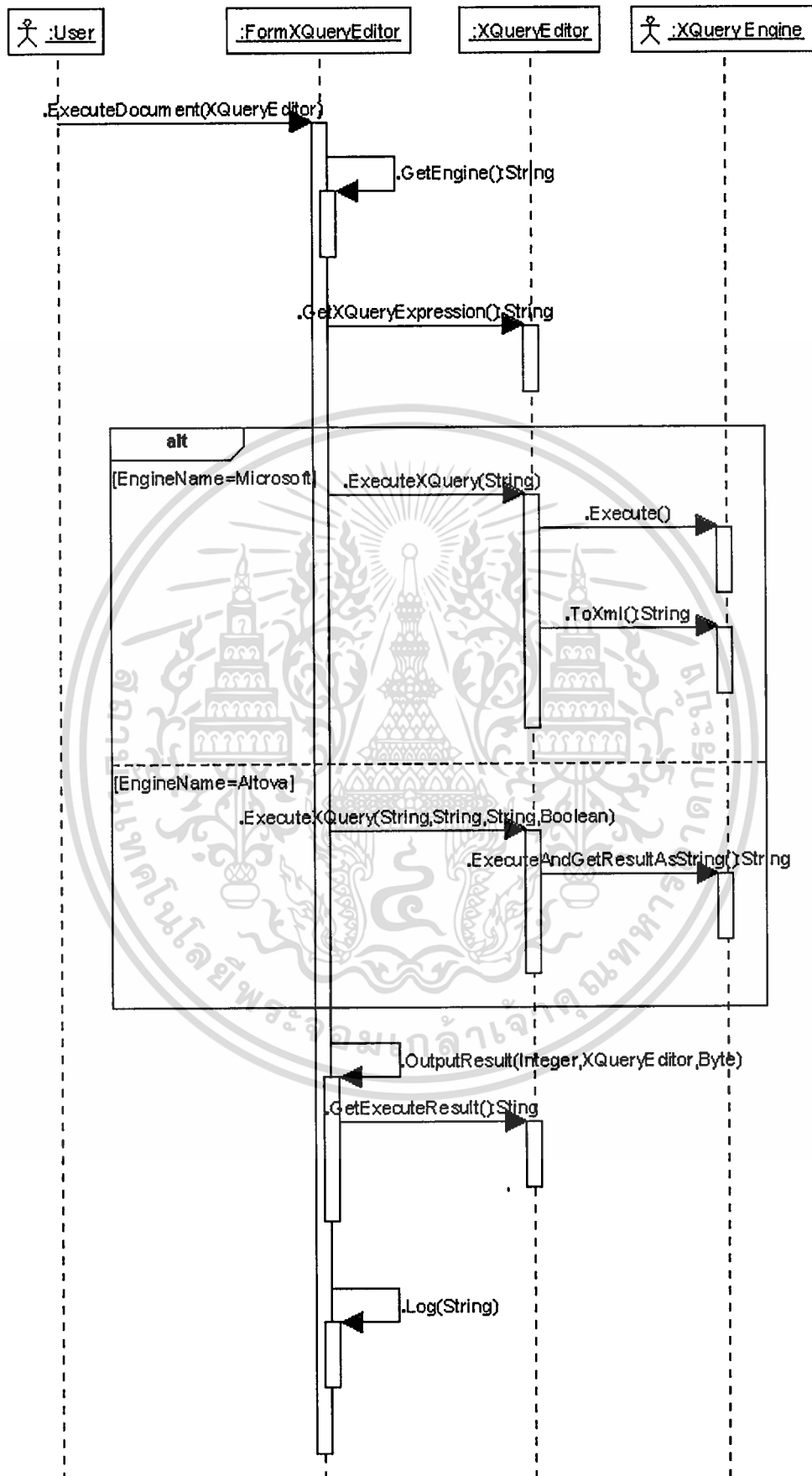
ภาพที่ 4.16 แผนภาพซีเคว้นซ์การบันทึกไวยากรณ์ XQuery

ขั้นตอนลำดับการบันทึกไวยากรณ์ XQuery มีดังนี้

- ผู้ใช้คลิกปุ่ม Save แล้วกำหนดชื่อไฟล์ พร้อมตำแหน่งจัดเก็บ
- อ่านไวยากรณ์ XQuery จากเอดิเตอร์
- บันทึกข้อมูลลงไฟล์ เก็บยังตำแหน่งที่กำหนด
- บันทึกรายการการทำงานกับโปรแกรม

4.5.4 ลำดับการทำงานของการค้นหาข้อมูล

ลำดับการทำงานของการค้นหาข้อมูลเริ่มต้นจากที่ผู้ใช้คลิกปุ่ม Execute โปรแกรมจะตรวจสอบว่าผู้ใช้ได้เลือกประมวลไวยากรณ์ XQuery ด้วย Engine ไต จากนั้นจะส่งไวยากรณ์ไปให้ Engine ดังกล่าวประมวลผล และแสดงผลลัพธ์ ดังภาพที่ 4.17



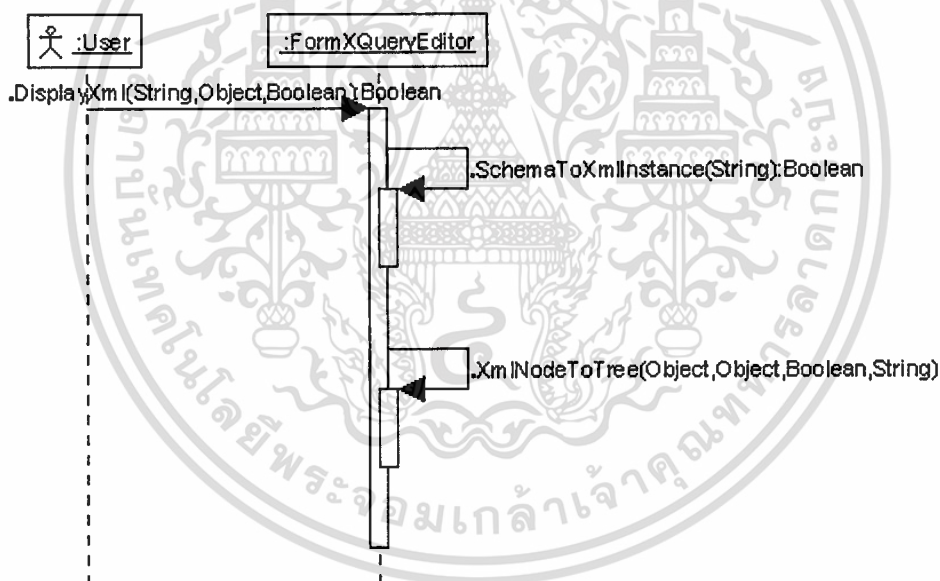
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ภาพที่ 4.17 แผนภาพชีวิตการทำงานการค้นหาข้อมูล
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนลำดับการค้นหาข้อมูล มีดังนี้

- ผู้ใช้คลิกปุ่ม Execute
- ตรวจสอบว่าเลือก Engine ไດ
- อ่านไวยากรณ์ XQuery จากเอคิเตอร์
- ประมวลผลไวยากรณ์ XQuery ด้วย Engine ที่กำหนด
- นำผลลัพธ์มาแสดงผล
- บันทึกรายการการทำงานกับโปรแกรม

4.5.5 ลำดับการทำงานของ การแสดงโครงสร้างของเอกสาร XML

ลำดับการทำงานของ การแสดงโครงสร้างของเอกสาร XML เริ่มต้นจากผู้ใช้เลือกเอกสาร XML ที่ต้องการ โปรแกรมจะประมวลผลเอกสาร XML แล้วแสดงผลที่หน้าจอแสดงโครงสร้าง ดังภาพที่ 4.18



ภาพที่ 4.18 แผนภาพซีควเอนซ์การแสดงผลโครงสร้างของเอกสาร XML

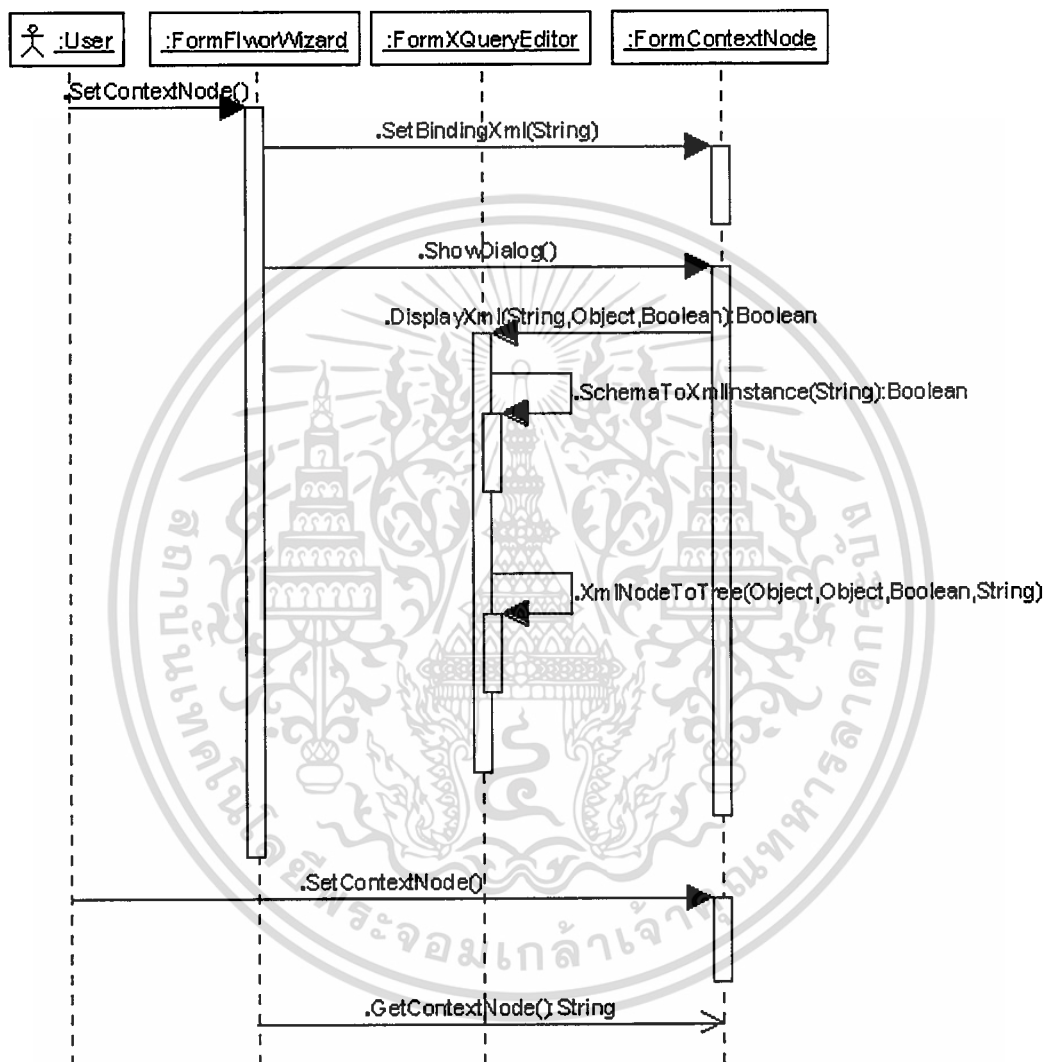
ขั้นตอนลำดับการแสดงผลโครงสร้างของเอกสาร XML มีดังนี้

- ผู้ใช้เลือกเอกสาร XML ที่ต้องการ
- ประมวลผลเอกสาร XML ว่าอยู่ในรูปแบบ Well-Formed
- สร้าง XML Schema จากเอกสาร XML ที่กำหนด
- อ่านโหนดของ XML สร้างเป็นโครงสร้างต้นไม้
- แสดงผลบนหน้าจอแสดงโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.6 ลำดับการทำงานของกาหนดโหนดเริ่มต้นในการค้นหาในหน้าจอช่วยสร้าง

ลำดับการทำงานของกาหนดโหนดเริ่มต้นในการค้นหาในหน้าจอช่วยสร้าง เริ่มต้นจากผู้ใช้เลือกเอกสาร XML ที่ต้องการค้นหา หน้าจอช่วยเหลือจะแสดงฟอร์มสำหรับการกาหนดโหนดเริ่มต้น จากนั้นผู้ใช้กาหนดโหนด และยืนยันการกาหนด ดังภาพที่ 4.19



ภาพที่ 4.19 แผนภาพซีควเอนซ์การกาหนดโหนดเริ่มต้นในการค้นหาในหน้าจอช่วยสร้าง

ขั้นตอนลำดับการกาหนดโหนดเริ่มต้นในการค้นหาในหน้าจอช่วยสร้าง มีดังนี้

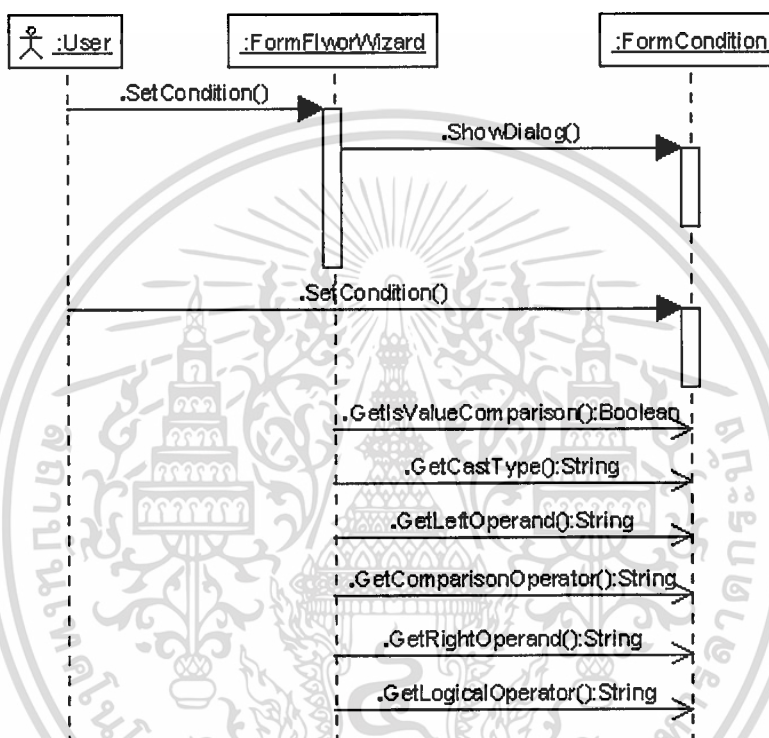
- ผู้ใช้เลือกเอกสาร XML ที่ต้องการค้นหา
- แสดงฟอร์ม
- แสดงโครงสร้างเอกสาร XML ข้างต้น
- ผู้ใช้กาหนดโหนดเริ่มต้นในการค้นหา โดยการคลิกบนโหนด

เอกสารนี้เป็นลิขสิทธิ์ของสถาบันพระจอมเกล้าเจ้าพระยาเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แสดงโหนดที่ถูกกำหนดบนหน้าจอช่วยสร้าง

4.5.7 ลำดับการทำงานของการสร้างเงื่อนไขในหน้าจอช่วยสร้าง

ลำดับการทำงานของการสร้างเงื่อนไขในหน้าจอช่วยสร้าง เริ่มต้นจากผู้ใช้เลือกที่จะสร้างเงื่อนไขในการค้นหา หน้าจอช่วยเหลือจะแสดงฟอร์มสำหรับการสร้างเงื่อนไข จากนั้นผู้ใช้สร้างเงื่อนไข และยืนยันการสร้าง ดังภาพที่ 4.20



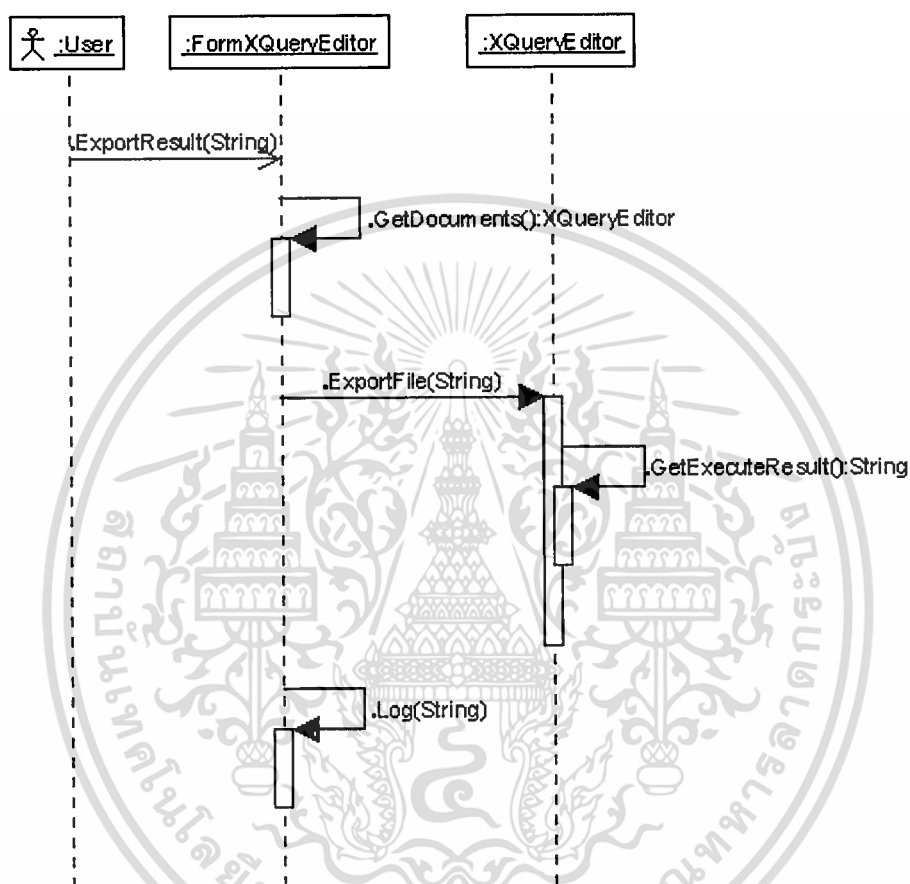
ภาพที่ 4.20 แผนภาพซีควเอนซ์การสร้างเงื่อนไขในหน้าจอช่วยสร้าง

ขั้นตอนลำดับการสร้างเงื่อนไขในหน้าจอช่วยสร้าง มีดังนี้

- ผู้ใช้คลิกปุ่ม Create Condition
- แสดงฟอร์มสำหรับการสร้างเงื่อนไข
- ผู้ใช้สร้างเงื่อนไข
- ยืนยันการสร้าง
- แสดงเงื่อนไขที่ผู้ใช้สร้างบนหน้าจอช่วยสร้าง

4.5.8 ลำดับการทำงานของการนำออกผลลัพธ์

ลำดับการทำงานของการนำออกผลลัพธ์เริ่มต้นจากผู้ใช้คลิกปุ่ม Export ในส่วนแสดงผลลัพธ์ ผู้ใช้กำหนดชื่อให้กับผลลัพธ์ พร้อมตำแหน่งจัดเก็บ จากนั้นโปรแกรมจะนำออกผลลัพธ์เก็บยังตำแหน่งที่กำหนด ดังภาพที่ 4.21



ภาพที่ 4.21 แผนภาพซีเควนซ์การนำออกผลลัพธ์

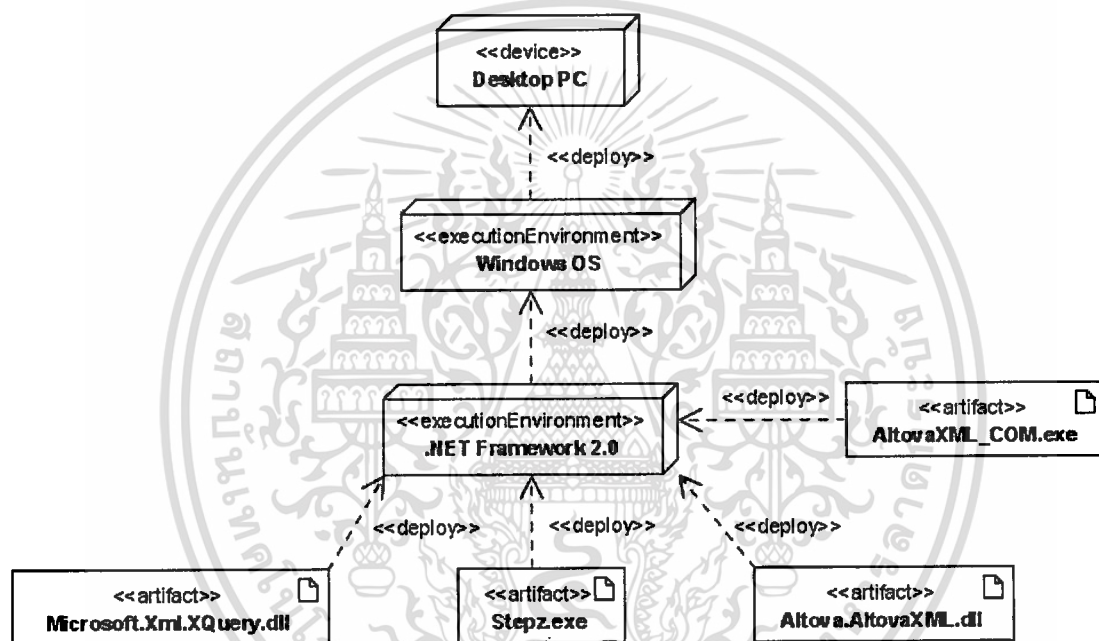
ขั้นตอนลำดับการนำออกผลลัพธ์ มีดังนี้

- ผู้ใช้คลิกปุ่ม Export แล้วกำหนดชื่อ พร้อมตำแหน่งจัดเก็บ
- เข้าถึงเอกสารที่เกี่ยวข้องกับผลลัพธ์
- เข้าถึงผลลัพธ์จากเอกสารข้างต้น
- นำออกผลลัพธ์
- บันทึกรายการการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 แผนภาพดีพลอยเมนต์

หลังจากที่ได้วิเคราะห์และออกแบบโปรแกรมเรียบร้อยแล้ว ต่อไปจะนำเสนอแผนภาพดีพลอยเมนต์ที่จะแสดงให้เห็นถึงการนำไปใช้งานจริง นั่นคือ จะนำโปรแกรมที่พัฒนาภายใต้ชื่อ Stepz.exe (ชื่อโปรแกรมคือ KMITL Stepz XQuery Editor) พร้อมไฟล์ไลบรารีของ Engine (ไฟล์นามสกุล dll) ที่ใช้ในการประมวลผลไวยากรณ์ XQuery ไปติดตั้งที่เครื่องคอมพิวเตอร์ส่วนตัวซึ่งต้องติดตั้งระบบปฏิบัติการวินโดวส์ และลงโปรแกรมเฟรมเวิร์ก .NET เวอร์ชัน 2.0 เรียบร้อยแล้ว ดังภาพที่ 4.22



ภาพที่ 4.22 แผนภาพดีพลอยเมนต์

จากภาพที่ 4.22 ไฟล์ AltovaXML_COM.exe, Altova.AltovaXML.dll, Microsoft.Xml.XQuery.dll และ Stepz.exe โดยแต่ละไฟล์เป็นซอฟต์แวร์ โดยใน UML เรียกว่า Artifact ซึ่งถูกนำไปติดตั้งบนโหนด โดยโหนดในที่นี้มี 2 ลักษณะคือ โหนด Execution Environment และ โหนดฮาร์ดแวร์ ในที่นี้ซอฟต์แวร์จะถูกติดตั้งบนโหนด Execution Environment คือ เฟรมเวิร์กคือ .NET 2.0 ซึ่งเฟรมเวิร์กคือ .NET 2.0 จะถูกติดตั้งบนโหนด Execution Environment คือ ระบบปฏิบัติการวินโดวส์ ซึ่งระบบปฏิบัติการวินโดวส์ถูกติดตั้งบนโหนดฮาร์ดแวร์ คือ คอมพิวเตอร์ส่วนตัวของผู้ใช้

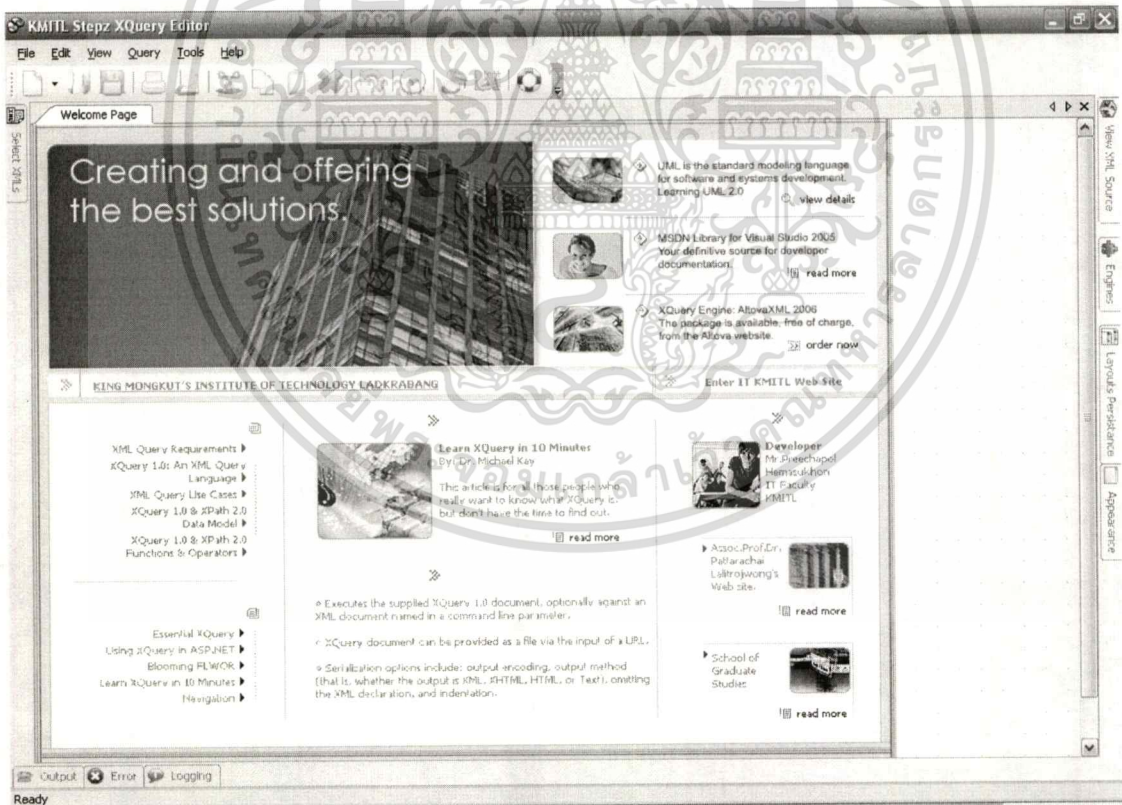
บทที่ 5

การออกแบบส่วนต่อประสานกับผู้ใช้

ในบทนี้จะอธิบายการออกแบบหน้าจอเพื่อให้ผู้ใช้สามารถใช้งานโปรแกรมได้ง่าย และสะดวก ซึ่งเริ่มต้นตั้งแต่หน้าจอต้อนรับเข้าสู่โปรแกรม หน้าจอหลัก แท็บ หน้าจอช่วยสร้างนิพจน์ XQuery และไดอะล็อกต่างๆ ที่จะเกิดขึ้น รวมไปถึงการทำงานของหน้าจอต่างๆ เหล่านี้

5.1 หน้าจอต้อนรับเข้าสู่โปรแกรม

หน้าจอต้อนรับเข้าสู่โปรแกรมจะเป็นหน้าจอแรกที่ผู้ใช้จะได้เห็นทุกครั้งเมื่อเปิดโปรแกรม ซึ่งหน้าจอนี้จะประกอบไปด้วยบทความต่างๆ ที่เกี่ยวกับไวยากรณ์ XQuery รวมถึงแสดงรายละเอียดเกี่ยวกับผู้พัฒนา ดังภาพที่ 5.1

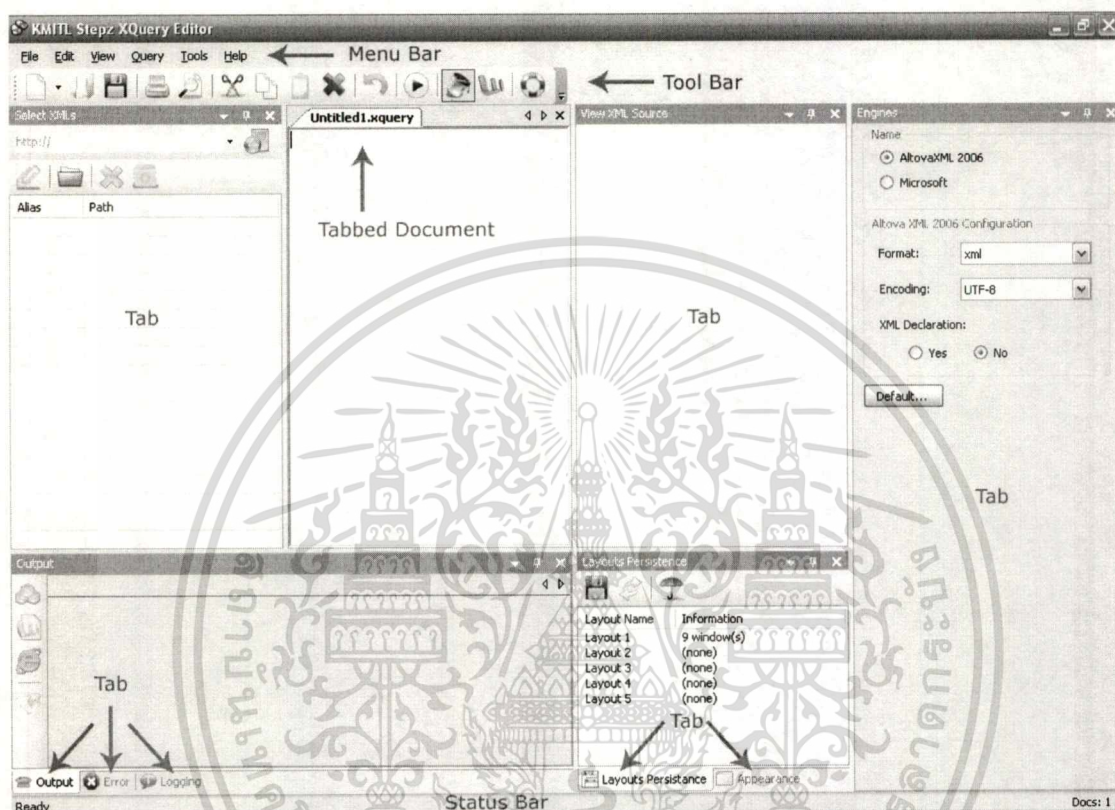


ภาพที่ 5.1 หน้าจอหลักและแท็บต้อนรับเข้าสู่โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 หน้าจอหลัก

หน้าจอหลักของโปรแกรมประกอบด้วยส่วนต่างๆ 5 ส่วน ดังภาพที่ 5.2 อธิบายแต่ละส่วนได้ดังต่อไปนี้



ภาพที่ 5.2 ส่วนประกอบหน้าจอหลักของโปรแกรม

5.2.1 แถบเมนู

แถบเมนู (Menu Bar) เป็นเมนูรวบรวมคำสั่งทั้งหมดที่สามารถใช้งานได้ รวมถึงการเรียกใช้เครื่องมืออื่นๆ โดยบางคำสั่งที่ใช้งานบ่อยๆ จะได้รับการกำหนดคีย์ลัด เพื่อช่วยให้เรียกใช้งานได้สะดวกขึ้น เช่น F5 จะเท่ากับการคลิกเมนู Query จากนั้นคลิก Execute เป็นต้น

5.2.2 แถบเครื่องมือ

แถบเครื่องมือ (Tool Bar) ประกอบด้วยปุ่มคำสั่งที่ช่วยให้เรียกใช้งานคำสั่งได้รวดเร็วยิ่งขึ้น ซึ่งประกอบด้วยปุ่ม New, Wizard, Open, Save, Print, Print Preview, Cut, Copy, Paste, Delete, Undo, Execute, Highlight, Word Wrap และ Help

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 แท็บ

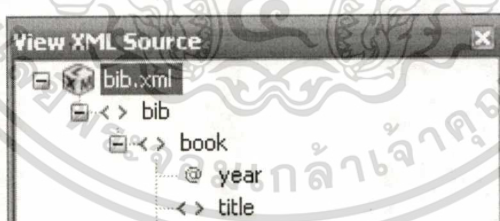
แท็บ (Tab) ประกอบด้วยทั้งหมด 8 แท็บ ดังนี้

- แท็บ Select XMLs ใช้สำหรับกำหนดเลือกเอกสาร XML ซึ่งสามารถเลือกจากงานบันทึกเฉพาะที่บนเครื่องผู้ใช้หรือกำหนด URL ของเอกสาร XML บนเครือข่าย โดยโปรแกรมจะตั้งชื่ออ้างอิง (Alias) ให้กับเอกสาร XML ซึ่งชื่อนี้จะนำไปใช้กับการเขียนนิพจน์ XQuery ที่จะต้องถูกประมวลผลด้วย Engine ของไมโครซอฟต์ ดังแสดงตัวอย่างในภาพที่ 5.3



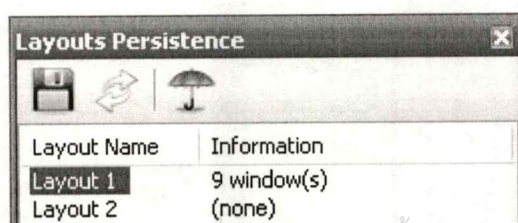
ภาพที่ 5.3 แท็บสำหรับเลือกเอกสาร XML

- แท็บ View XML Source ใช้สำหรับแสดงโครงสร้างเอกสาร XML โดยคลิกที่ชื่ออ้างอิงในแท็บ Select XMLs ซึ่งจะแสดงเป็นโครงสร้างต้นไม้ และมีไอคอนกำกับด้านหน้าว่าแต่ละส่วนคืออะไร เช่น < > หมายถึงเป็นเอลิเมนต์ @ หมายถึงเป็นแอตทริบิวต์ เป็นต้น ดังแสดงตัวอย่างในภาพที่ 5.4



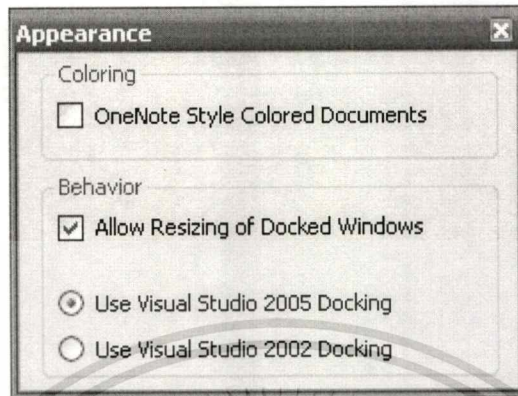
ภาพที่ 5.4 แท็บแสดงโครงสร้างเอกสาร XML

- แท็บ Layouts Persistence ใช้สำหรับจัดการเลย์เอาต์ของโปรแกรม ดังภาพที่ 5.5 โดยภายในจะมีปุ่มควบคุม 3 ปุ่ม คือ บันทึกเลย์เอาต์ โหลดเลย์เอาต์ และกระจายแท็บออกทั้งหมด



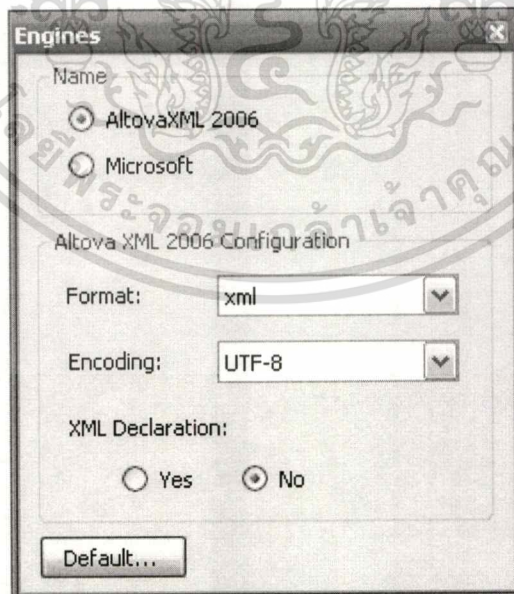
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับนักวิจัยภายในของจุฬาลงกรณ์มหาวิทยาลัย ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ภาพที่ 5.5 แท็บจัดการเลย์เอาต์ของโปรแกรม
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แท็บ Appearance ใช้สำหรับจัดการลักษณะต่างๆ และพฤติกรรมของโปรแกรม เช่น สีของแท็บเอกสาร การอนุญาตให้ลดหรือขยายขนาดของแท็บ เป็นต้น ดังภาพที่ 5.6



ภาพที่ 5.6 แท็บจัดการลักษณะต่างๆ และพฤติกรรมของโปรแกรม

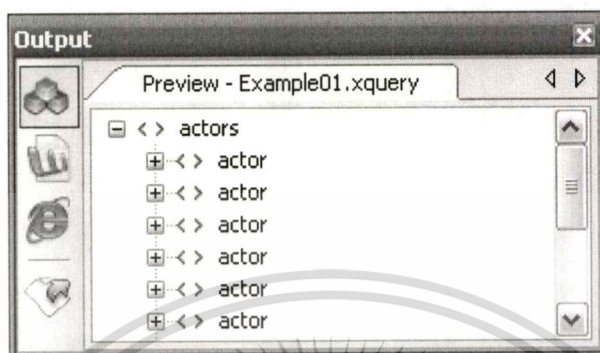
- แท็บ Engines ใช้สำหรับกำหนด Engine ที่ใช้ในการประมวลผลนิพจน์ XQuery โดยมีให้เลือก 2 ตัวคือ AltovaXML2006 และ Microsoft ซึ่งถ้าเลือก AltovaXML2006 จะสามารถกำหนดรูปแบบแสดงผล การเข้ารหัส และการประกาศว่าเป็น XML ดังภาพที่ 5.7 และสามารถกำหนดค่าโดยปริยาย (Default) ของ Engine ได้โดยคลิกปุ่ม Default... จากนั้นโปรแกรมจะบันทึกเก็บไว้เพื่อใช้เป็นค่าโดยปริยายต่อไป



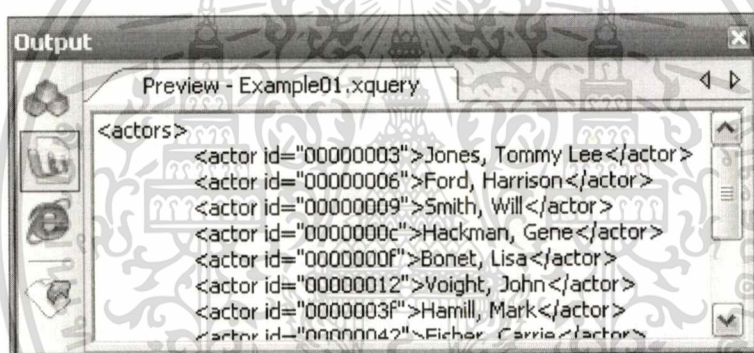
ภาพที่ 5.7 แท็บกำหนดส่วนประมวลผลไวยากรณ์ XQuery

- แท็บ Output ใช้สำหรับกำหนดรูปแบบผลลัพธ์ที่แสดงผล ซึ่งสามารถกำหนดได้ทั้งหมด 4 รูป คือ แบบโครงสร้างต้นไม้ (Tree View) ดังแสดงตัวอย่างในภาพที่ 5.8 แบบตัวอักษรธรรมดาไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

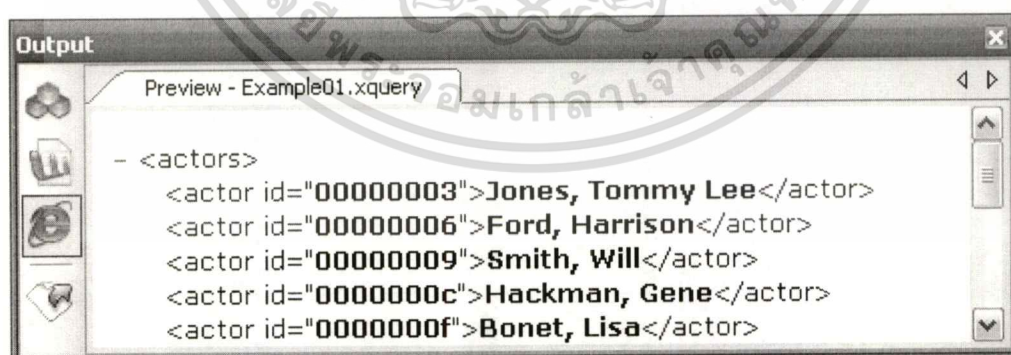
(Text View) ดังแสดงตัวอย่างในภาพที่ 5.9 แบบเปิดผ่านอินเทอร์เน็ตเอ็กซ์พลอเรอร์ ดังแสดงตัวอย่างในภาพที่ 5.10 และแบบนำเสนอเป็นไฟล์จัดเก็บลงบนเครื่องคอมพิวเตอร์ ซึ่งสามารถกำหนดรูปแบบผลลัพธ์โดยการเลือกคลิกปุ่มจาก 4 ปุ่ม ที่อยู่ทางด้านซ้ายมือของแท็บ



ภาพที่ 5.8 แท็บแสดงผลลัพธ์แบบโครงสร้างต้นไม้



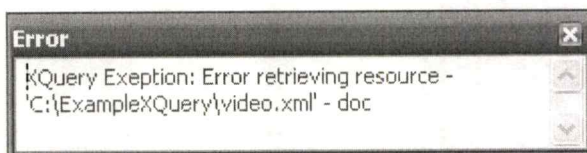
ภาพที่ 5.9 แท็บแสดงผลลัพธ์แบบตัวอักษรธรรมดา



ภาพที่ 5.10 แท็บแสดงผลลัพธ์แบบเปิดผ่านอินเทอร์เน็ตเอ็กซ์พลอเรอร์

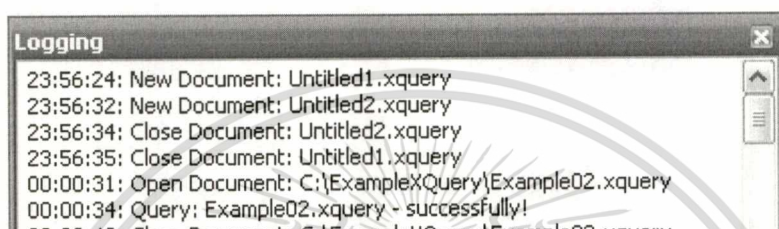
- แท็บ Error ใช้สำหรับแสดงข้อผิดพลาดที่เกิดขึ้นขณะทำงานกับโปรแกรม ดังแสดงตัวอย่างในภาพที่ 5.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.11 แท็บแสดงข้อผิดพลาด

- แท็บ Logging ใช้สำหรับแสดงลำดับเหตุการณ์ที่เกิดขึ้นขณะทำงานกับโปรแกรม ดังแสดงตัวอย่างในภาพที่ 5.12



ภาพที่ 5.12 แท็บแสดงลำดับเหตุการณ์ขณะทำงานกับ โปรแกรม

5.2.4 แท็บเอกสาร

แท็บเอกสาร (Tabbed Document) เป็นแท็บที่แสดงขึ้นเมื่อมีการเรียกใช้คำสั่ง New, Open และ Wizard เป็นแท็บที่โปรแกรมจะจัดเตรียมพื้นที่ให้ผู้ใช้เพิ่ม/แก้ไข นิพจน์ XQuery ซึ่งสามารถสลับการทำงานกับแท็บเอกสารที่เปิดอยู่

5.2.5 แถบสถานะ

แถบสถานะ (Status Bar) แสดงสถานะการทำงาน จำนวนเอกสารที่เปิดอยู่ ณ ขณะนั้น ซึ่งถ้ามีการคิวรีจะแสดงแถบความคืบหน้า (Progress Bar) ชื่อของ Engine ที่ใช้และเวลาที่ใช้ในการประมวลผลนิพจน์ XQuery เพิ่มเข้ามา ดังแสดงตัวอย่างในภาพที่ 5.12

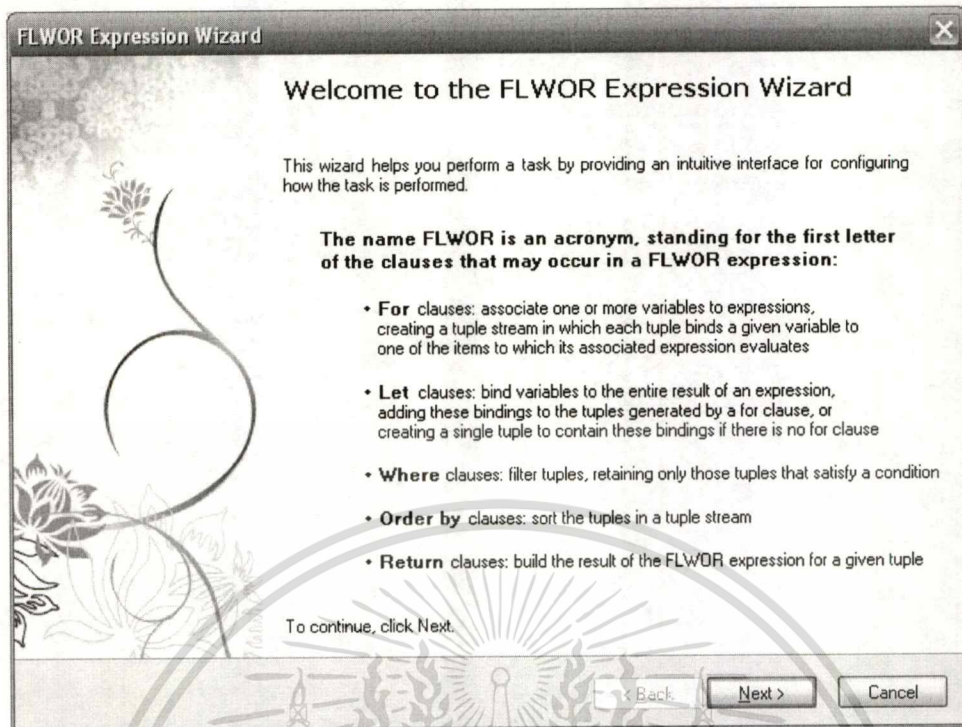


ภาพที่ 5.13 แถบสถานะ

5.3 หน้าจอช่วยสร้างนิพจน์ XQuery

หน้าจอช่วยสร้างนิพจน์ XQuery เป็นตัวช่วยสำหรับผู้ใช้ที่ไม่มีความรู้ไวยากรณ์ XQuery ซึ่งผู้ใช้สามารถเริ่มใช้งานได้โดยการเลือกไปที่แถบเครื่องมือที่เดียวกับ New แล้วเลือก Wizard หรือกดปุ่ม Ctrl+W หน้าจอช่วยสร้างนิพจน์ XQuery จะปรากฏขึ้น ดังภาพที่ 5.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



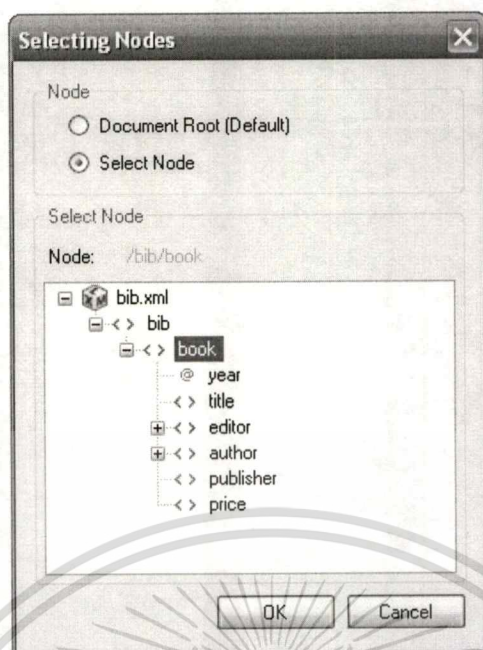
ภาพที่ 5.14 หน้าจอช่วยสร้างนิพจน์ XQuery

จากภาพที่ 5.14 เป็นหน้าจอแรกของตัวช่วยสร้างนิพจน์ XQuery แสดงการต้อนรับเข้าสู่ตัวช่วยสร้าง โดยเป็นตัวช่วยสร้างนิพจน์ FLWOR ซึ่งภายในหน้านี้จะมีการอธิบายถึงคำย่อจากตัวอักษร FLWOR เพื่อเพิ่มความเข้าใจให้กับผู้ใช้ จากนั้นเมื่อคลิกปุ่ม Next > จะเข้าสู่การสร้างนิพจน์ FLWOR ซึ่งจะมีขั้นตอนทั้งหมด 4 ขั้นตอน โดยจะอธิบายเป็นลำดับการทำงานพร้อมหน้าจอและไดอะล็อกต่างๆ ที่จะปรากฏขึ้นได้ดังต่อไปนี้

1. เลือกเอกสาร XML

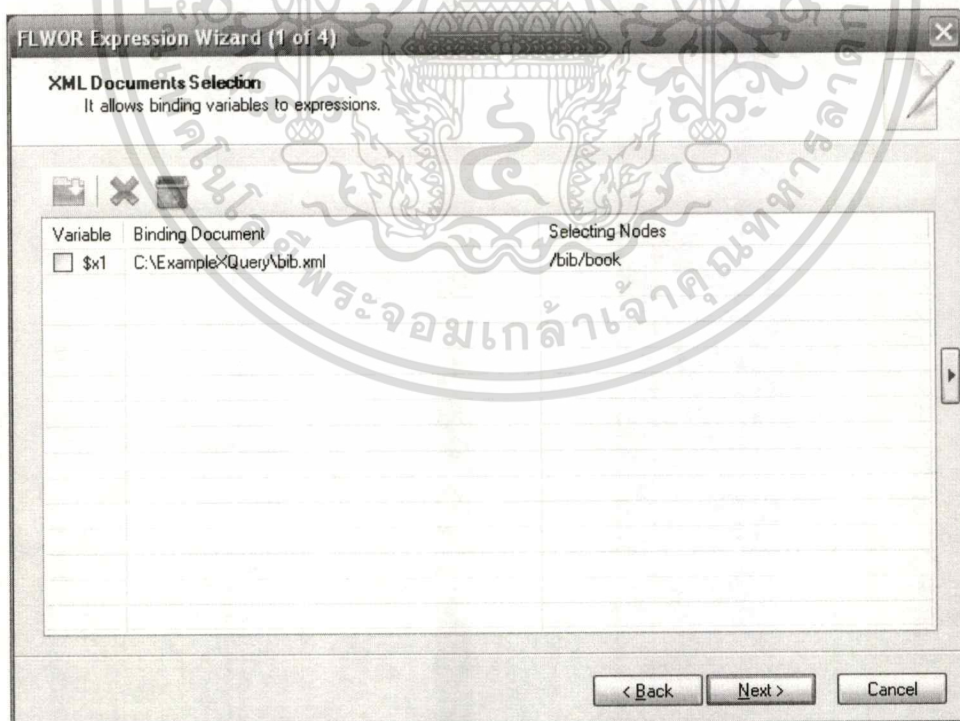
- เลือกเอกสาร XML ที่ต้องการค้นหา โดยการคลิกปุ่ม Browse XML โปรแกรมจะแสดงไดอะล็อก Browse XML Document ให้เลือกไฟล์ XML ที่ต้องการจากนั้นคลิกปุ่ม Open
- หลังจากที่ผู้ใช้คลิกปุ่ม Open โปรแกรมจะแสดงไดอะล็อก Selecting Nodes ดังแสดงตัวอย่างในภาพที่ 5.15 ซึ่งเป็นการกำหนดโหนดที่จะใช้เริ่มต้นในการค้นหา
- เริ่มต้นโปรแกรมจะกำหนดโหนดที่จะใช้เริ่มต้นในการค้นหาเป็น Document Root แต่สามารถกำหนดเป็นโหนดอื่นได้ด้วยการเลือก Select Node จากนั้นจึงสามารถดับเบิลคลิกที่โหนดที่ต้องการ ซึ่งโปรแกรมจะแสดงลำดับโครงสร้างการเข้าถึงโหนดนั้นด้วยตัวอักษรสีส้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.15 ไดอะล็อกสำหรับกำหนดโหนดที่จะใช้เริ่มต้นในการค้นหา

- จากนั้นคลิกปุ่ม OK ไดอะล็อกจะถูกปิด แล้วกลับมายังหน้าเลือกเอกสาร XML เพื่อเลือก XML เพิ่มเติมต่อไปได้ ดังแสดงตัวอย่างในภาพที่ 5.16



ภาพที่ 5.16 หน้าจอสำหรับเลือกเอกสาร XML หลังเลือกเอกสาร XML แล้ว

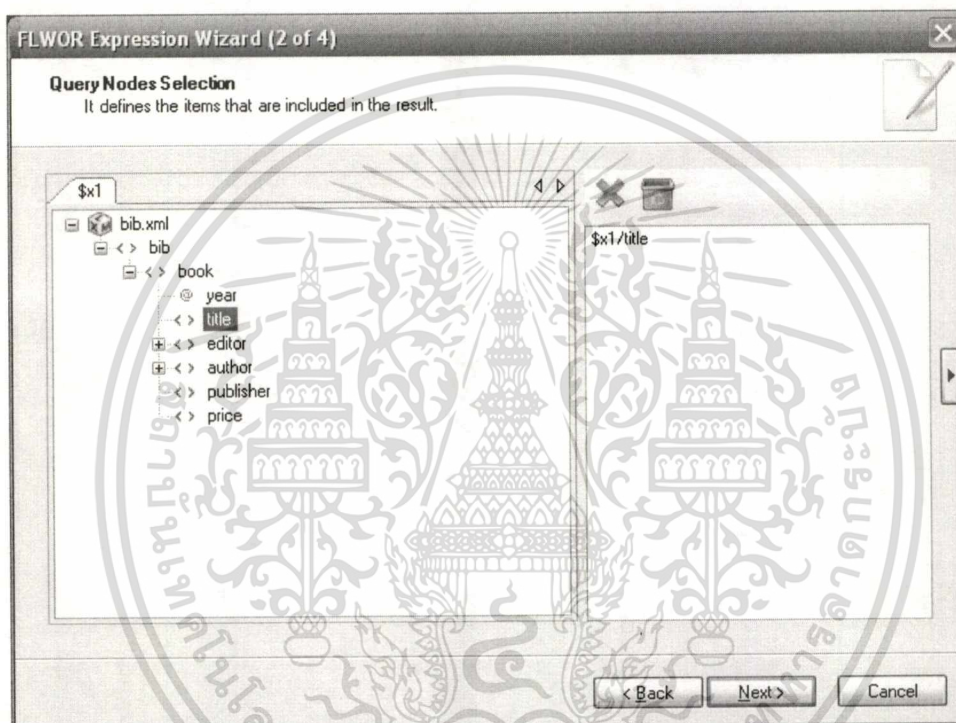
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คลิก Next> เพื่อไปยังหน้าถัดไป เพื่อเลือกโหนดที่ต้องการค้นหา หากต้องการยกเลิกคลิกปุ่ม Cancel หากต้องการย้อนกลับไปดูหน้าก่อนหน้า คลิกปุ่ม <Back

2. กำหนดโหนดที่ต้องการค้นหา

- กำหนดโหนดที่ต้องการค้นหา โดยการดับเบิลคลิกที่โหนดที่ต้องการ
- โปรแกรมจะเพิ่มโหนดที่เลือกเข้าลิสต์ที่อยู่ทางด้านขวามือ ดังแสดงตัวอย่างในภาพที่

5.17



ภาพที่ 5.17 หน้าจอสำหรับเลือกโหนดที่ต้องการค้นหาพร้อมแสดงลิสต์โหนดที่ผู้ใช้เลือก

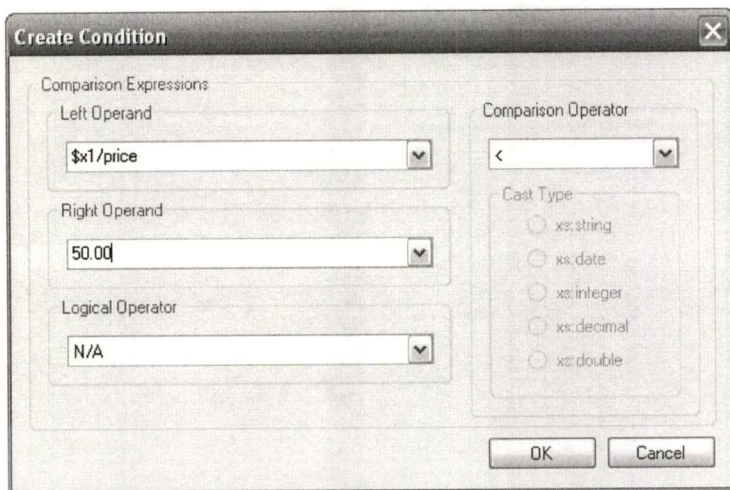
- ในกรณีที่ต้องลบโหนดใดโหนดหนึ่งออกจากลิสต์ ให้คลิกเลือกที่โหนดนั้น จากนั้นคลิกปุ่ม Remove Node หากต้องการเคลียร์ลิสต์ให้คลิกปุ่ม Clear Nodes
- คลิก Next> เพื่อไปยังหน้าถัดไป เพื่อกำหนดเงื่อนไขในการค้นหา หากต้องการยกเลิกคลิกปุ่ม Cancel หากต้องการย้อนกลับไปดูหน้าก่อนหน้า คลิกปุ่ม <Back

3. กำหนดเงื่อนไข (เป็นทางเลือก กำหนดหรือไม่กำหนดก็ได้)

- ในกรณีต้องการกำหนดเงื่อนไข สามารถพิมพ์เงื่อนไขลงในช่องที่กำหนดไว้ให้ได้ หรือคลิกปุ่ม Create Condition หรือคลิกปุ่ม Create Condition: Quantified Expressions
- ถ้าคลิกปุ่ม Create Condition โปรแกรมจะแสดงไดอะล็อก Create Condition ขึ้น ดัง

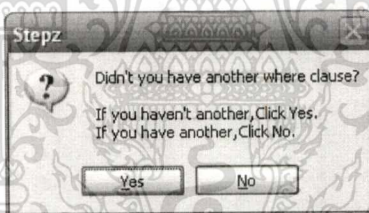
แสดงตัวอย่างในภาพที่ 5.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



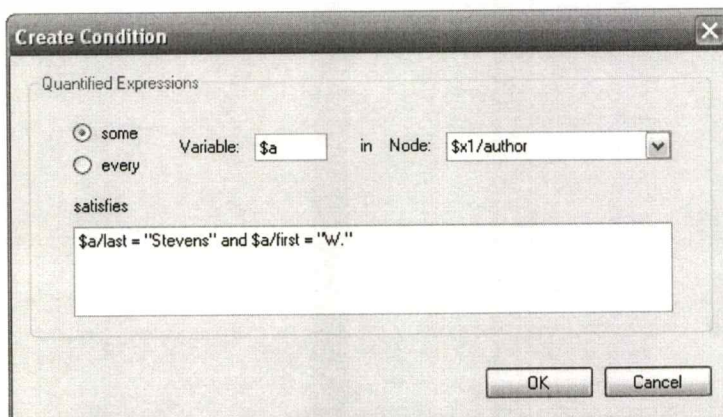
ภาพที่ 5.18 โค้ดเลือกสำหรับการกำหนดเงื่อนไข

- ในกรณีที่ไม่มีกำหนด Logical Operator โปรแกรมจะแสดงโค้ดเลือกถามว่า “ผู้ใช้ไม่มีเงื่อนไขอื่นๆ อีกใช่หรือไม่” ดังแสดงตัวอย่างในภาพที่ 5.19 ถ้าเลือกใช่ เมื่อกลับสู่หน้าจอเดิมปุ่ม Create Condition จะไม่สามารถคลิกได้ ถ้าเลือกไม่ เมื่อกลับสู่หน้าจอเดิมปุ่ม Create Condition จะทำงานได้ตามปกติ



ภาพที่ 5.19 โค้ดเลือกถามว่าผู้ใช้ไม่มีเงื่อนไขอื่นๆ อีกใช่หรือไม่

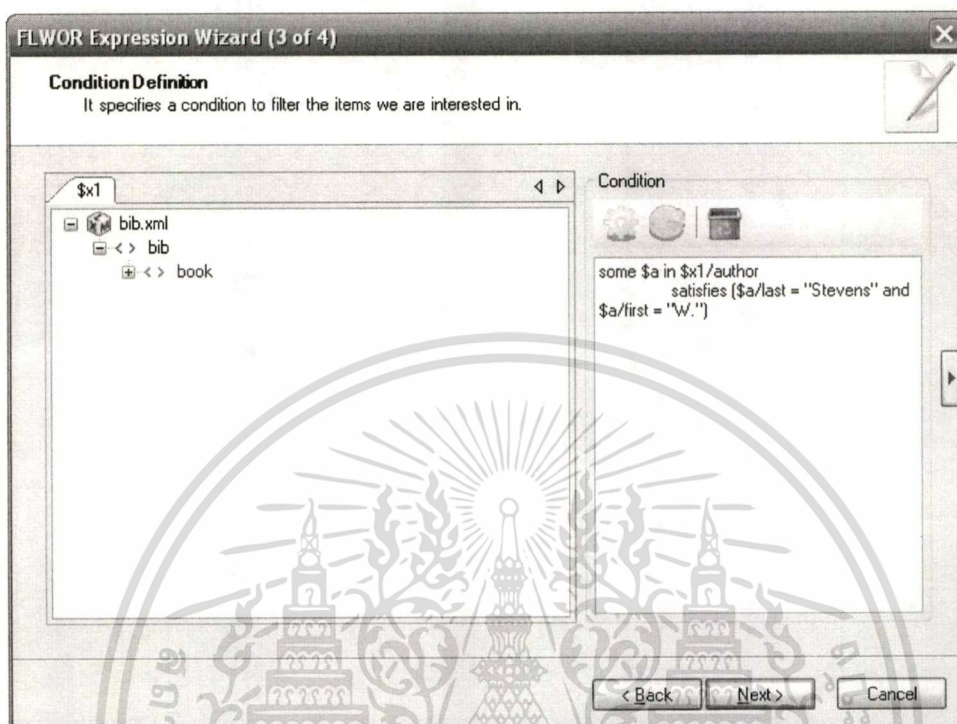
- ในกรณีที่ต้องการกำหนดเงื่อนไขแบบนิพจน์ Quantifier ให้คลิกปุ่ม Create Condition: Quantified Expressions โปรแกรมจะแสดงโค้ดเลือกให้กำหนดค่า ดังแสดงตัวอย่างในภาพที่ 5.20



ภาพที่ 5.20 โค้ดเลือกให้กำหนดนิพจน์ Quantified

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสำนักงานส่งเสริมการค้าในต่างประเทศ ณ นครเชียงใหม่ โดยขอให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จากนั้นคลิกปุ่ม OK ใดจะล็อกจะถูกปิด แล้วกลับมายังหน้ากำหนดเงื่อนไข ดังแสดงตัวอย่างในภาพที่ 5.21 ในกรณีที่ต้องการเคลียร์เงื่อนไขทั้งหมดให้คลิกปุ่ม Clear Conditions



ภาพที่ 5.21 หน้าจอสำหรับกำหนดเงื่อนไขพร้อมแสดงเงื่อนไขที่ผู้ใช้กำหนด

- คลิก Next> เพื่อไปยังหน้าถัดไป เพื่อกำหนดการเรียงลำดับและรีเทิร์นรูตเอลิเมนต์ หากต้องการยกเลิกคลิกปุ่ม Cancel หากต้องการย้อนกลับไปดูหน้าก่อนหน้า คลิกปุ่ม <Back

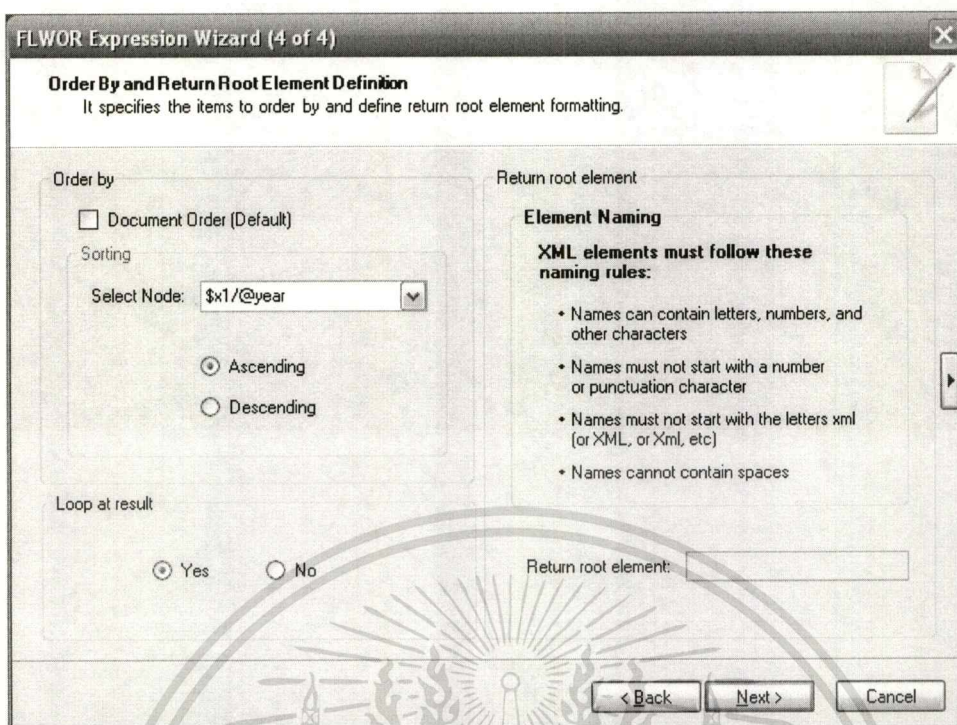
4. กำหนดการเรียงลำดับและรีเทิร์นรูตเอลิเมนต์

- กำหนดการเรียงลำดับ โดยโปรแกรมจะกำหนดให้เรียงลำดับตามลำดับข้อมูลในเอกสาร XML ที่ใช้ค้นหา (Document Order) เป็นค่าโดยปริยาย แต่สามารถให้ผู้ใช้กำหนดการเรียงลำดับตามที่ผู้ใช้ต้องการได้ โดยให้เลือกเอาเครื่องหมายถูกหน้า Document Order (Default) ออก แล้วเลือกโหมดที่ต้องการใช้สำหรับการเรียงลำดับผลลัพธ์ โดยสามารถกำหนดการเรียงลำดับได้ 2 แบบ คือ น้อยไปมาก (Ascending) และมากไปน้อย (Descending)

- กำหนดการวนลูปที่ผลลัพธ์ โดยโปรแกรมจะกำหนดให้มีการวนลูปที่ผลลัพธ์เป็นค่าโดยปริยาย

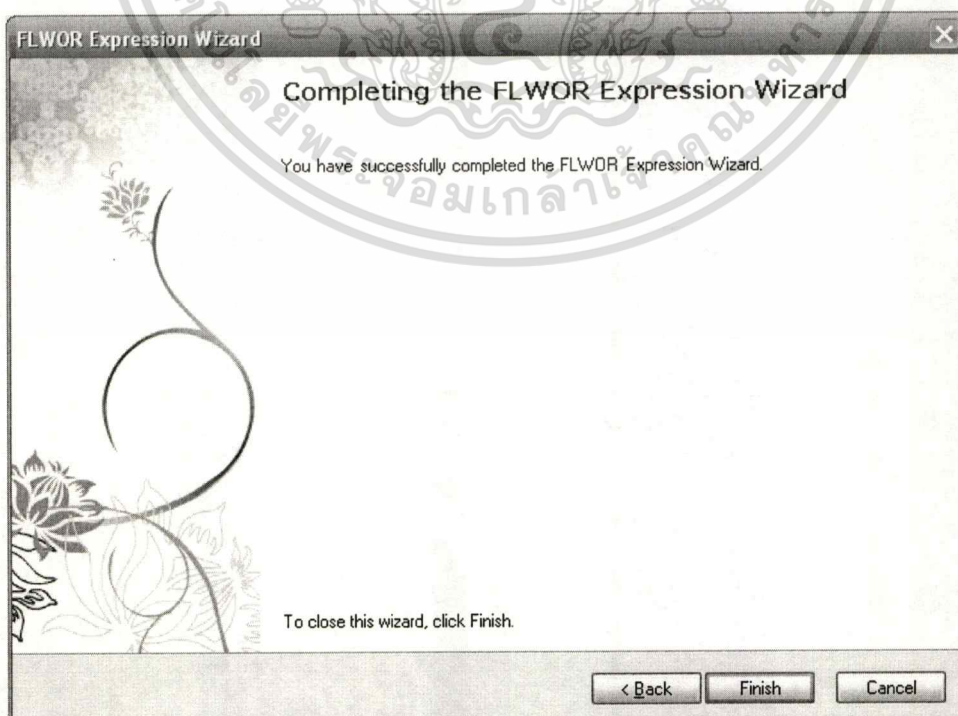
- กำหนดรีเทิร์นรูตเอลิเมนต์ โดยพิมพ์ลงในช่องว่างตรง Return root element: ในกรณีที่ผู้ใช้กำหนดชื่อไม่ถูกต้อง โปรแกรมจะแสดงข้อความเตือนข้างช่องว่างที่ผู้ใช้พิมพ์เป็นตัวอักษรสีแดง

เอกสารนี้เป็นกรณีกำหนดค่าต่างๆ ทั้งหมด แสดงตัวอย่างในภาพที่ 5.22 มอนูญาต์ให้เข้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.22 หน้าจอสำหรับกำหนดการเรียงลำดับและรีเทิร์นรูตเอลิเมนต์

- คลิกปุ่ม Next> เพื่อไปยังหน้าถัดไป เพื่อจบการใช้ตัวช่วยสร้างนิพจน์ XQuery ดังแสดงตัวอย่างในภาพที่ 5.23 หากต้องการยกเลิกคลิกปุ่ม Cancel หากต้องการย้อนกลับไปดูหน้าก่อนหน้าคลิกปุ่ม <Back



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์โดยมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ภาพที่ 5.23 หน้าจอจบการใช้ตัวช่วยสร้างนิพจน์ XQuery นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การพัฒนาและทดสอบโปรแกรม

หลังจากได้ออกแบบส่วนต่อประสานกับผู้ใช้เรียบร้อยแล้ว ถัดมาในบทนี้จะได้อธิบายถึงสถาปัตยกรรมของโปรแกรม เครื่องมือที่ใช้พัฒนาโปรแกรม และทดสอบโปรแกรม โดยมีรายละเอียดดังต่อไปนี้

6.1 สถาปัตยกรรมของโปรแกรม

สถาปัตยกรรมของโปรแกรม จะแสดงความต้องการทางด้านฮาร์ดแวร์ ซอฟต์แวร์ของโปรแกรมบนเครื่องผู้ใช้ เพื่อเตรียมสภาพแวดล้อมที่เหมาะสมเมื่อนำโปรแกรมไปติดตั้ง ซึ่งรายละเอียดมีดังต่อไปนี้

6.1.1 คุณสมบัติของฮาร์ดแวร์

คุณสมบัติของฮาร์ดแวร์ที่โปรแกรมต้องการ มีดังต่อไปนี้

- หน่วยประมวลผลระดับเพนเทียม โพรหรือเทียบเท่า
- ขนาดพื้นที่บนฮาร์ดดิสก์ประมาณ 20 MB สำหรับติดตั้ง โปรแกรม
- หน้าจอความละเอียดที่ 1024 x 768 จุดภาพ
- หน่วยความจำ 256 MB ขึ้นไป
- เม้าส์และคีย์บอร์ด

6.1.2 ซอฟต์แวร์ของโปรแกรม

ซอฟต์แวร์ที่จำเป็นต้องติดตั้ง เพื่อสร้างสภาวะแวดล้อมในการทำให้โปรแกรมสามารถติดตั้งและทำงานได้อย่างราบรื่น มีดังต่อไปนี้

- ระบบปฏิบัติการวินโดวส์เอ็กซ์พี Service Pack 2
- เฟรมเวิร์ก .NET เวอร์ชัน 2.0
- อินเทอร์เน็ตเอ็กซ์พลอเรอร์เวอร์ชัน 6.0

6.2 เครื่องมือที่ใช้พัฒนาโปรแกรม

เครื่องมือที่ใช้ในการพัฒนาโปรแกรมนี้นี้ คือ Visual Studio .NET 2005 ซึ่งเลือกใช้ภาษา VB ในการพัฒนาตลอดทั้งโปรแกรม โดยในการพัฒนาโปรแกรมจะใช้ Visual Studio .NET 2005 ตั้งแต่การคำนวณว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบหน้าจอ ปุ่มต่างๆ รวมถึงเขียนโปรแกรมควบคุมการทำงาน ทั้งนี้จะมีการเรียกใช้คลาสที่มีมากับเฟรมเวิร์ก .NET รวมถึงมีการเรียกใช้คลาสภายนอกซึ่งอยู่ในรูปของเครื่องมือที่จะมีการนำเข้ามาเพิ่มเติม เช่น ในการออกแบบหน้าจอให้สวยงามและน่าใช้งานนั้น จำเป็นต้องมีการใช้เครื่องมืออื่นนอกเหนือจากที่มีมาให้ใน Visual Studio .NET 2005 คือ SandDock และ SandBar สำหรับ Visual Studio 2005 ของบริษัท Divelements โดยนำมาใช้ในส่วนการพัฒนาออกแบบหน้าจอหลัก ตั้งแต่ส่วนแถบเมนู แถบเครื่องมือ แท็บ แท็บเอกสาร และแถบสถานะ หรือในส่วนการพัฒนาออกแบบหน้าจอช่วยสร้างไวยากรณ์ XQuery ได้มีการใช้เครื่องมือชื่อ WizardFramework มาช่วยในการวางลำดับเหตุการณ์การทำงานของตัวช่วยสร้าง และเครื่องมือชื่อ SlideForm มาใช้สำหรับการสร้างฟอร์มที่สามารถเลื่อนออกมาจากด้านใดด้านหนึ่งของฟอร์มอื่นได้ ซึ่งทำให้สามารถพัฒนาโปรแกรมได้สะดวก รวดเร็ว และน่าสนใจมากยิ่งขึ้น

ลำดับถัดมาสำหรับคุณสมบัติของคอมพิวเตอร์ที่จะนำมาใช้ในการพัฒนาโปรแกรมนี้ ต้องมีสมรรถนะค่อนข้างสูง เนื่องจากเลือกที่จะพัฒนาโปรแกรมบน Visual Studio .NET 2005 ซึ่งมีการใช้ทรัพยากรของคอมพิวเตอร์ค่อนข้างมาก ดังนั้น คุณสมบัติของคอมพิวเตอร์ที่ใช้พัฒนาโปรแกรมครั้งนี้ (คอมพิวเตอร์ของผู้พัฒนา) มีดังต่อไปนี้

- พื้นที่ว่างของฮาร์ดดิสก์ 4 GB
- การ์ดวิดีโอ 1024x768
- หน่วยความจำขนาด 768 MB
- ไดรฟ์ซีดีรอมหรือดีวีดีรอม
- ระบบปฏิบัติการวินโดวส์เอ็กซ์พี Service Pack 2
- เฟรมเวิร์ก .NET เวอร์ชัน 2.0
- อินเทอร์เน็ตเอ็กซ์พลอเรอร์เวอร์ชัน 6.0

6.3 การทดสอบโปรแกรม

ในหัวข้อนี้จะได้แสดงถึงการทดสอบโปรแกรม ซึ่งเป็นการทดสอบประสิทธิภาพของเวลาการทำงานของโปรแกรม และทดสอบการใช้งานโปรแกรมโดยแบบสอบถาม ซึ่งมีรายละเอียดดังต่อไปนี้

6.3.1 การทดสอบประสิทธิภาพของเวลาการทำงานของโปรแกรม

การทดสอบประสิทธิภาพของเวลาการทำงานของโปรแกรม เป็นการทดสอบโปรแกรมเพื่อพิจารณาถึงช่วงเวลาที่ใช้ในการประมวลผลไวยากรณ์ XQuery ว่าใช้ระยะเวลาสั้นเพียงใด ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดสอบ กำหนดกลุ่มตัวอย่างของไวยากรณ์ XQuery จำนวน 15 ตัวอย่าง (อ้างอิงตัวอย่างที่ 2.1 – 2.15 ในบทที่ 2) แล้ววิเคราะห์หาค่าเฉลี่ย

ตารางที่ 6.1 ผลการทดสอบประสิทธิภาพของเวลาการทำงานของโปรแกรมค้นหาข้อมูลภายในเอกสาร XML ด้วย XQuery

กลุ่มตัวอย่าง	เวลา (วินาที)					ค่าเฉลี่ย
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
1. ตัวอย่างที่ 2.1	0.0183	0.0184	0.0183	0.0181	0.0186	0.01834
2. ตัวอย่างที่ 2.2	0.0172	0.0161	0.0172	0.0155	0.016	0.0164
3. ตัวอย่างที่ 2.3	0.0158	0.0154	0.0156	0.015	0.0162	0.0156
4. ตัวอย่างที่ 2.4	0.0155	0.0177	0.0159	0.0154	0.0155	0.016
5. ตัวอย่างที่ 2.5	0.018	0.0163	0.0156	0.0154	0.0158	0.01622
6. ตัวอย่างที่ 2.6	0.0504	0.0164	0.0162	0.0166	0.0169	0.0233
7. ตัวอย่างที่ 2.7	0.0834	0.0169	0.0169	0.0163	0.0165	0.03
8. ตัวอย่างที่ 2.8	0.0174	0.0166	0.0167	0.0169	0.0175	0.01702
9. ตัวอย่างที่ 2.9	0.0351	0.009	0.0083	0.0088	0.009	0.01404
10. ตัวอย่างที่ 2.10	0.0164	0.0168	0.0159	0.0166	0.0161	0.01636
11. ตัวอย่างที่ 2.11	0.0245	0.016	0.0186	0.0171	0.0165	0.01854
12. ตัวอย่างที่ 2.12	0.0368	0.0185	0.0212	0.0167	0.0182	0.02228
13. ตัวอย่างที่ 2.13	0.0165	0.0165	0.0189	0.0152	0.0227	0.01796
14. ตัวอย่างที่ 2.14	0.0462	0.0169	0.0196	0.0152	0.0165	0.02288
15. ตัวอย่างที่ 2.15	0.0444	0.0208	0.0207	0.0199	0.0209	0.02534
ค่าเฉลี่ยรวม:						0.019352

จากตารางที่ 6.1 สามารถสรุปประสิทธิภาพของเวลาการทำงานของโปรแกรมเฉลี่ยอยู่ที่ 0.019352 วินาที ซึ่งเวลาเฉลี่ยที่ได้ถือว่าโปรแกรมสามารถทำได้ดีและรวดเร็วเป็นที่น่าพอใจ

6.3.2 การทดสอบการใช้งานโปรแกรมโดยแบบสอบถาม

การทดสอบการใช้งานโปรแกรมใช้การตอบแบบสอบถาม โดยกำหนดกลุ่มประชากรจำนวน 20 คน จากนั้นใช้การวิเคราะห์ค่าคะแนนเฉลี่ย และกำหนดเกณฑ์ในการแปลความหมายค่า

คะแนนเฉลี่ยของความพึงพอใจของผู้ใช้งานโปรแกรม ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าคะแนนเฉลี่ยระหว่าง 3.67-5.00 หมายถึง ผู้ใช้มีความพึงพอใจอยู่ในระดับมาก
 ค่าคะแนนเฉลี่ยระหว่าง 2.34-3.66 หมายถึง ผู้ใช้มีความพึงพอใจอยู่ในระดับปานกลาง
 ค่าคะแนนเฉลี่ยระหว่าง 1.00-2.33 หมายถึง ผู้ใช้มีความพึงพอใจอยู่ในระดับน้อย
 และจากการทดสอบการใช้งานของผู้ใช้สามารถสรุปผลการทดสอบ ได้ดังนี้

ตารางที่ 6.2 ค่าคะแนนเฉลี่ยของความพึงพอใจของผู้ใช้งาน โปรแกรมค้นหาข้อมูลภายในเอกสาร XML ด้วย XQuery

หัวข้อ	ค่าคะแนนเฉลี่ย
การออกแบบส่วนต่อประสานกับผู้ใช้	
1. หน้าจอหลักง่ายต่อการเรียนรู้และใช้งาน	2.7
2. รูปแบบของหน้าจอมีความน่าสนใจ	2.9
3. แสดงข้อความอยู่ในตำแหน่งที่เหมาะสมบนจอภาพ	3.5
4. การแบ่งสัดส่วนของหน้าจอชัดเจน	3.25
5. ในส่วนของเอดิเตอร์ขนาดตัว สี่อักษรชัดเจน	3
6. ในหน้าจอช่วยสร้างนิพจน์ XQuery ง่ายต่อการเรียนรู้และใช้งาน	2.7
7. แสดงข้อผิดพลาด อธิบายเกี่ยวกับข้อผิดพลาดชัดเจน	3.15
การค้นหาข้อมูล	
8. ความรวดเร็ว	3.15
9. ความถูกต้อง	3.1
ผลลัพธ์	
10. หน้าจอแสดงผลการค้นหาชัดเจน สวยงาม	2.85
11. นิพจน์ FLWOR ที่สร้างด้วยตัวช่วยสร้างถูกต้องตามที่ต้องการ	2.4
12. แสดงโครงสร้างเอกสาร XML ถูกต้อง	2.4
การใช้งานโปรแกรม	
13. ความเสถียรของโปรแกรม	2.6
14. ความน่าใช้งาน	2.85
15. ประโยชน์ที่ได้จากการใช้งานโปรแกรม	3.15
เฉลี่ยรวม:	2.91

จากตารางสามารถสรุปความพึงพอใจของผู้ใช้ต่อการออกแบบส่วนต่อประสานกับผู้ใช้
 การค้นหาข้อมูล ผลลัพธ์ และการใช้งานโปรแกรมอยู่ในระดับปานกลาง
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

บทสรุป

หลังจากได้ศึกษาโครงการ วิเคราะห์ ออกแบบ และพัฒนาโครงการสำเร็จเสร็จสิ้นแล้ว สามารถนำมาสรุปผลการดำเนินงานในบทนี้ อีกทั้งกล่าวถึงปัญหาที่พบในการทำงาน และมีการเพิ่มเติมข้อเสนอแนะสำหรับผู้สนใจโครงการนี้สามารถนำไปพัฒนาต่อ ทำให้โครงการนี้มีคุณค่าและประสิทธิภาพที่ดียิ่งขึ้นต่อไป

7.1 สรุปโครงการ

การพัฒนาโปรแกรมสำหรับการค้นหาข้อมูลภายในเอกสาร XML ด้วย XQuery บนเฟรมเวิร์ก .NET สามารถวิเคราะห์ ออกแบบ และพัฒนาโปรแกรมไปในทางเดียวกันคือ ในลักษณะเชิงวัตถุ เพื่อให้ง่ายต่อการพัฒนาและใช้ประโยชน์จากความสามารถในการนำกลับมาใช้ใหม่ โดยมีความจำเป็นที่จะต้องศึกษา ค้นคว้า เรียนรู้ทางทฤษฎี และวิธีการใช้เครื่องมือต่างๆ ที่เกี่ยวกับการพัฒนาโปรแกรมเชิงวัตถุเพิ่มมากขึ้น ซึ่งในส่วนการวิเคราะห์และออกแบบมีการใช้ UML ในการทำงาน โดยอาศัยเอสทูลดช่วยสร้างแผนภาพ คือ โปรแกรม Poseidon for UML ส่วนการพัฒนาโปรแกรมซึ่งพัฒนาบนเฟรมเวิร์ก .NET จึงอาศัยโปรแกรม Microsoft Visual Studio 2005 โดยเลือกใช้ภาษา VB ในการพัฒนา ซึ่งสามารถพัฒนาเป็น โปรแกรมพร้อมชุดติดตั้งได้ง่ายและสะดวก

การวิเคราะห์และออกแบบที่ใช้ UML มีการสร้างแผนภาพแบบต่างๆ ดังนี้ แผนภาพยูสเคส แผนภาพแอกทิวิตี แผนภาพคลาส แผนภาพซีเควนซ์ และแผนภาพดีพลอยเมนต์เพื่อเป็นตัวกลางในการติดต่อสื่อสารถ่ายทอดกระบวนการทำงานของโปรแกรมผ่านทั้งมุมมองของผู้ใช้และผู้พัฒนา สำหรับผลลัพธ์ของการพัฒนาโปรแกรม สามารถพัฒนาโปรแกรมออกมาได้เป็นที่น่าพอใจ ตรงกับที่ได้วิเคราะห์และออกแบบไว้ โดยสามารถนำโปรแกรมไปติดตั้งบนเครื่องผู้ใช้ที่ติดตั้งระบบปฏิบัติการวินโดวส์เพื่อใช้งานได้จริง ซึ่งการสร้างไวยากรณ์ XQuery สามารถให้ผู้ใช้เลือกสร้างไวยากรณ์เองผ่านหน้าจอบนพื้นที่ที่กำหนด และยังสามารถสร้างไวยากรณ์ XQuery ผ่านตัวช่วยสร้าง โดยไวยากรณ์ XQuery ที่สร้างขึ้น สามารถบันทึกเป็นไฟล์นามสกุล XQuery เพื่อไว้เปิดใช้งานในภายหลังได้ ส่วนผลลัพธ์ของการค้นหาข้อมูลภายในเอกสาร XML สามารถทำได้สำเร็จเป็นที่น่าพอใจ อีกทั้งเวลาที่ใช้ในการค้นหาสามารถตอบสนองได้ดี และสามารถบันทึกผลลัพธ์เป็นไฟล์นามสกุล XML เพื่อนำไปใช้ประโยชน์อย่างอื่นต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 ปัญหาที่พบและข้อเสนอแนะ

ปัญหาหลักที่พบในการพัฒนาโครงการนี้ คือ การพัฒนาโปรแกรมแบบเชิงวัตถุ ตั้งแต่เริ่มต้นจากการวิเคราะห์ ออกแบบ จนถึงการพัฒนาโปรแกรม เนื่องจากอาจจะยังขาดความรู้ ความเข้าใจ และขาดประสบการณ์กับการพัฒนาโปรแกรมเชิงวัตถุเพื่อใช้งานจริง จึงทำให้โปรแกรมที่พัฒนาขึ้นยังพบปัญหาและอาจเกิดข้อผิดพลาดขึ้นได้ในบางจุด

จากการพัฒนาโครงการนี้ ยังต้องมีข้อปรับปรุงแก้ไขข้อผิดพลาดหรือเปลี่ยนแปลงความต้องการใหม่ๆ เพิ่มขึ้นมาได้อีกมาก ดังนั้น จึงเสนอข้อแนะนำบางส่วนที่อาจจะช่วยให้โปรแกรมนั้น มีความสมบูรณ์และใช้งานได้ดียิ่งขึ้นดังนี้

1. ในหน้าจอสำหรับผู้ที่มีความรู้ไวยากรณ์ XQuery ที่ผู้ใช้ต้องพิมพ์นิพจน์ XQuery เอง ยังขาดตัวช่วยในการพิมพ์นิพจน์ XQuery ให้ง่ายและสะดวกมากยิ่งขึ้น เช่น
 - การแสดงหมายเลขบรรทัด ในกรณีที่ชุดคำสั่งไวยากรณ์ XQuery มีจำนวนบรรทัดมากๆ นั้น การเห็นหมายเลขบรรทัดของคำสั่งจะช่วยให้สามารถตรวจสอบและแก้ไขคำสั่งได้ง่ายขึ้น
 - การช่วยแนะนำการเขียนชุดคำสั่งไวยากรณ์ XQuery (IntelliSense) ซึ่งจะช่วยให้ผู้ใช้ไม่ต้องจดจำรายละเอียดของคำสั่ง เช่น การแนะนำฟังก์ชัน อธิบายรายละเอียดของฟังก์ชัน เป็นต้น
2. ในส่วนของตัวช่วยสร้างชุดคำสั่งไวยากรณ์ XQuery เพิ่มความสามารถในการสร้างนิพจน์อื่น นอกเหนือนิพจน์ FLOWR ซึ่งมีการเพิ่มนิพจน์ Quantified แล้ว แต่ยังคงปรับเปลี่ยนให้มีประสิทธิภาพที่ดีขึ้น
3. การตรวจสอบความถูกต้องของชุดคำสั่งไวยากรณ์ XQuery ให้โดยอัตโนมัติทันที โดยไม่ต้องรอให้ประมวลผล XQuery ก่อน
4. เครื่องมือตรวจสอบข้อผิดพลาด (Debugger) เพื่อจะช่วยให้ผู้ใช้เห็นการทำงานของชุดคำสั่งไวยากรณ์ XQuery ได้ทีละขั้นตอน ซึ่งทำให้ตรวจสอบและแก้ไขข้อผิดพลาดได้เป็นอย่างดี
5. ความสามารถในการเพิ่ม Engine อื่นๆ ได้ เพื่อจะช่วยให้โปรแกรมอาศัยความสามารถของ Engine นั้น ซึ่งอาจมีความสามารถในการประมวลผล XQuery ที่ต่างกันหรือทันสมัยกับข้อกำหนดที่องค์กร W3C กำหนดออกมา

บรรณานุกรม

- กิตติชัย พินิจดำ. 2544. การเขียนโปรแกรม XML และ SOAP สำหรับ BizTalk Server. กรุงเทพฯ : สามย่าน.com.
- เกียรติศักดิ์ อุนรรรม. XML ที่สุดของภาษาสำหรับการพัฒนาระบบข้อมูลข่าวสารในยุค Generation X. [Online]. เข้าถึงได้จาก : http://internet.seed.com/content/IN80/IN80_51.asp. 2548.
- ขนิษฐอรุณ เวียรศิลป์. 2546. การพัฒนาโปรแกรมสำหรับการค้นหาข้อมูลภายในเอกสาร XML ด้วย ไวยากรณ์ XQuery ผ่านเว็บ. รายงานวิชาโครงการพัฒนาระบบงานวิทยาศาสตร์มหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ บัณฑิตวิทยาลัย, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- ประณต บุญไชยอภิสิทธิ์. เอกสาร XML สำหรับสิ่งพิมพ์อิเล็กทรอนิกส์. [Online]. เข้าถึงได้จาก : http://www.tiac.or.th/tiacthai/seminar/2003/annual/presentation/dr_pranot.ppt. 2546.
- รหัสสากลสภาอุตสาหกรรมแห่งประเทศไทย. XML คืออะไร. [Online]. เข้าถึงได้จาก : http://www.eanthai.org/eant2003/th/eansystem/xml/what_xml.asp. 2546.
- ศุภชัย สมพานิช. 2544. เข้าใจและใช้งานภาษา XML ฉบับโปรแกรมเมอร์. พิมพ์ครั้งที่ 2. นนทบุรี : อินโฟเพรส.
- สุรสิทธิ์ คิวประสพศักดิ์ และนันท์นิ แขวงโสภาก. 2546. อินไซด์ Visual Basic .NET ฉบับสมบูรณ์. กรุงเทพฯ : โปรวิชั่น.
- สุวัฒนา สุขสมจินตน์. 2545. คัมภีร์การใช้ XML ฉบับสมบูรณ์. กรุงเทพฯ : ซีเอ็ดบุ๊คเซ็น.
- Altova GmbH. AltovaXML User Manual. [Online]. Available : <http://www.altova.com/manual2006/AltovaXML/>. 2006.
- Boag, S. XQuery 1.0: An XML Query Language. [Online]. Available : <http://www.w3.org/TR/xquery/>. 2005.
- Brundage, M. Introduction to XQuery. [Online]. Available : http://www.qbrundage.com/xquery/xquery_ch03.pdf. 2004.
- Chand, M. XML Programming using VB.NET: Part 1. [Online]. Available : <http://www.vbdotnetheaven.com/Code/Apr2003/005.asp>. 2003.
- Hamilton, K. and Miles, R. 2006. Learning UML 2.0. Sebastopol, California : O'Reilly Media.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Microsoft. **Visual Studio 2005 Document**. [Online]. Available :

<http://msdn2.microsoft.com/library/default.aspx>. 2005.

Siegers, P. **Using XQuery in ASP.NET**. [Online]. Available :

<http://www.topxml.com/xquery/xquery.pdf>. 2004.

W3Schools. **XQuery Tutorial**. [Online]. Available :

<http://www.w3schools.com/xquery/default.asp>. 2005.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

ตัวอย่างเอกสาร XML

ไฟล์ videos.xml

```
<?xml version="1.0"?>
```

```
<result>
```

```
<actors>
```

```
  <actor id="00000003">Jones, Tommy Lee</actor>
```

```
  <actor id="00000006">Ford, Harrison</actor>
```

```
  <actor id="00000009">Smith, Will</actor>
```

```
  <actor id="0000000c">Hackman, Gene</actor>
```

```
  <actor id="0000000f">Bonet, Lisa</actor>
```

```
  <actor id="00000012">Voight, John</actor>
```

```
  <actor id="0000003f">Hamill, Mark</actor>
```

```
  <actor id="00000042">Fisher, Carrie</actor>
```

```
  <actor id="0000001e">Heche, Anne</actor>
```

```
  <actor id="00000021">Schwimmer, David</actor>
```

```
  <actor id="916503208">Pitt, Brad</actor>
```

```
  <actor id="916503209">Freeman, Morgan</actor>
```

```
</actors>
```

```
<videos>
```

```
  <video id="id1235AA0">
```

```
    <title>The Fugitive</title>
```

```
    <genre>action</genre>
```

```
    <rating>PG-13</rating>
```

```
    <summary>Tommy Lee Jones and Harrison Ford are the hunter and the hunted in this fast-paced story of a falsely convicted man who escapes to find his wife's true killer.</summary>
```

```
    <year>1997</year>
```

```
    <director>Andrew Davis</director>
```

```
    <studio>Warner</studio>
```

```
    <user_rating>3</user_rating>
```

```

<runtime>110</runtime>
<actorRef>00000003</actorRef>
<actorRef>00000006</actorRef>
<vhs>13.99</vhs>
<vhs_stock>206</vhs_stock>
<dvd>14.99</dvd>
<dvd_stock>125</dvd_stock>
<beta>1.03</beta>
<beta_stock>12</beta_stock>
<LaserDisk>12.00</LaserDisk>
<LaserDisk_stock>10</LaserDisk_stock>
</video>
<video id="id1244100">
  <title>Enemy of the State</title>
  <genre>action</genre>
  <rating>R</rating>
  <summary>After a chance meeting with an old pal, Robert Deal finds himself in
possession of a disk that contains evidence of an assassination plot by the NSA.</summary>
  <year>1999</year>
  <director>Tony Scott</director>
  <studio>Buena vista</studio>
  <user_rating>5</user_rating>
  <runtime>113</runtime>
  <actorRef>00000009</actorRef>
  <actorRef>0000000c</actorRef>
  <actorRef>0000000f</actorRef>
  <actorRef>00000012</actorRef>
  <vhs>16.99</vhs>
  <vhs_stock>188</vhs_stock>
  <dvd>29.99</dvd>
  <dvd_stock>353</dvd_stock>
  <LaserDisk>12.57</LaserDisk>

```

เอกสารนี้เป็นเอกสารที่ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<LaserDisk_stock>203</LaserDisk_stock>

</video>

<video id="id1256990">

<title>Star Wars</title>

<genre>action</genre>

<rating>PG</rating>

<summary>Luke embarks on an epic journey in a galaxy far far away.</summary>

<year>1977</year>

<director>George Lucas</director>

<studio>Fox Films</studio>

<user_rating>5</user_rating>

<runtime>121</runtime>

<actorRef>0000003f</actorRef>

<actorRef>00000042</actorRef>

<actorRef>00000006</actorRef>

<actorRef></actorRef>

<vhs>19.99</vhs>

<vhs_stock>312</vhs_stock>

<LaserDisk>34</LaserDisk>

<LaserDisk_stock>95</LaserDisk_stock>

</video>

<video id="id124F050">

<title>Six Days Seven Nights</title>

<genre>comedy</genre>

<rating>PG-13</rating>

<summary>A classy magazine editor and her gruff pilot must put aside their mutual dislike for each other to survive on a deserted island.</summary>

<year>1998</year>

<director>Ivan Reitman</director>

<studio>Buena vista</studio>

เอกสารนี้เป็นเอกสารที่<user_rating>2</user_rating>การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<runtime>123</runtime>
<actorRef>00000006</actorRef>
<actorRef>0000001e</actorRef>
<actorRef>00000021</actorRef>
<vhs>16.99</vhs>
<vhs_stock>327</vhs_stock>
<LaserDisk>12.00</LaserDisk>
<LaserDisk_stock>10</LaserDisk_stock>
<beta>10.75</beta>
<beta_stock>251</beta_stock>
</video>
<video id="647599249">
  <studio></studio>
  <actorRef>916503208</actorRef>
  <actorRef>916503209</actorRef>
  <director>David Fincher</director>
  <title>Seven</title>
  <dvd>22</dvd>
  <laserdisk>24</laserdisk>
  <laserdisk_stock>2</laserdisk_stock>
  <genre>horror</genre>
  <rating>R</rating>
  <runtime>130</runtime>
  <user_rating>3</user_rating>
  <summary>A serial killer forces each of his victims to die by acting out one of
the seven deadly sins.</summary>
  <vhs>15</vhs>
  <year>1995</year>
  <vhs_stock>88</vhs_stock>
  <dvd_stock>22</dvd_stock>
</video></videos>

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

การติดตั้งและการถอดถอนโปรแกรม KMITL Stepz XQuery Editor 1.0

ขั้นตอนการติดตั้งโปรแกรม

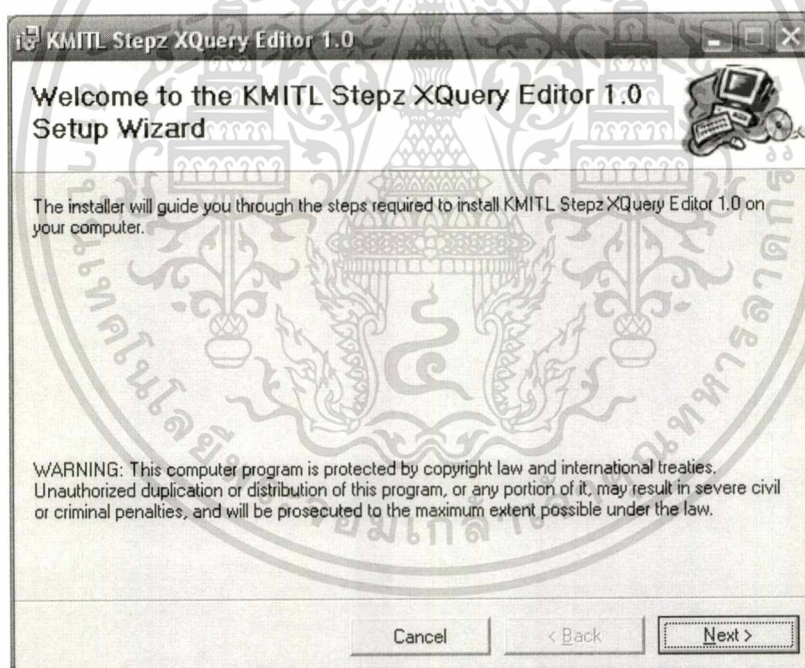
1. ติดตั้งระบบปฏิบัติการวินโดวส์ (แนะนำให้ติดตั้ง Microsoft Windows XP Service Pack 2 ขึ้นไป)

2. ติดตั้ง Microsoft .NET Framework Version 2.0 Redistributable Package (x86) (สามารถหาดาวน์โหลดได้ที่

<http://www.microsoft.com/downloads/details.aspx?FamilyID=0856eacb-4362-4b0d-8edd-aab15c5e04f5&displaylang=en>)

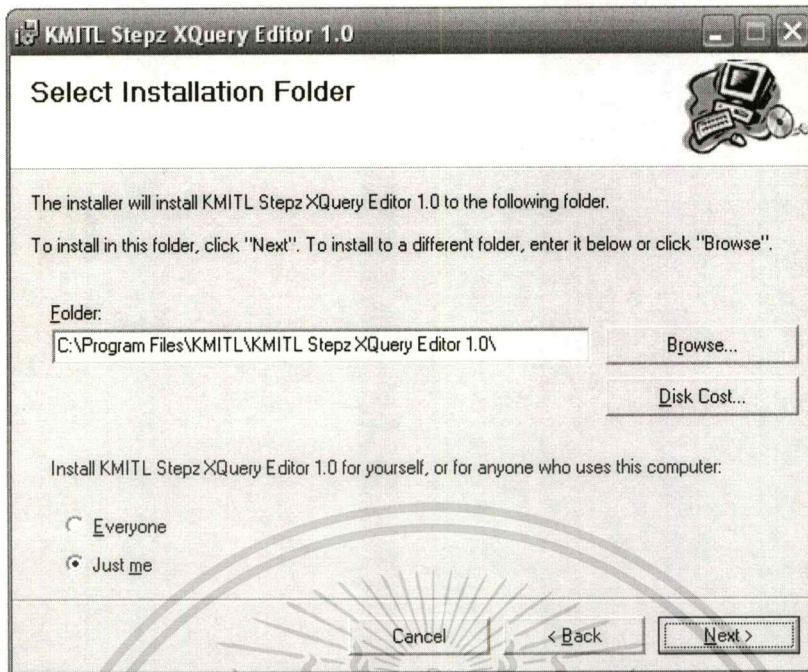
3. ดับเบิลคลิกที่ไฟล์ setup.exe หรือ Stepz-1.0-install.msi จะปรากฏไดอะล็อก ดังภาพที่

ข.1



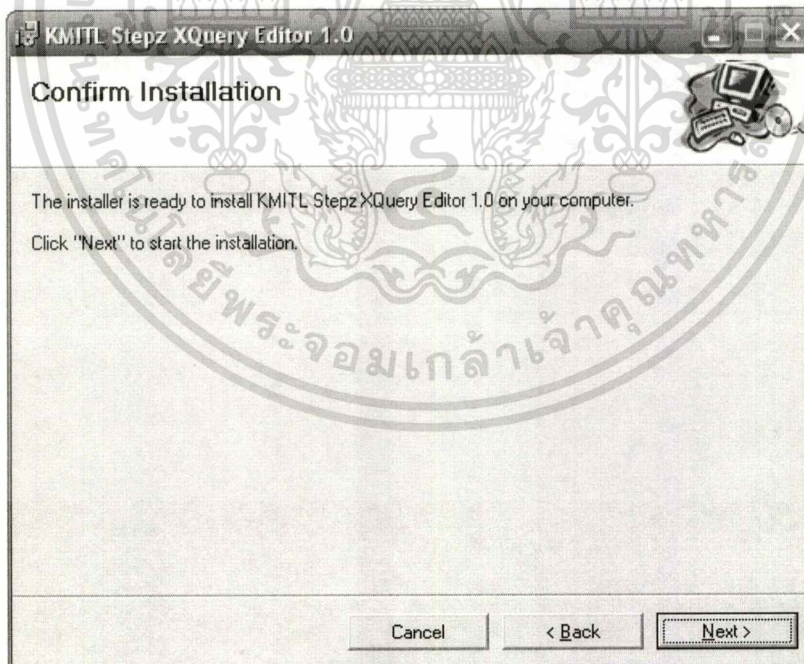
ภาพที่ ข.1 ไดอะล็อกเริ่มต้นการติดตั้งโปรแกรม

4. คลิกปุ่ม Next> เพื่อไปคำสั่งหน้าถัดไป เพื่อระบุตำแหน่งที่จะติดตั้งโปรแกรม หากต้องการเปลี่ยนตำแหน่งที่จะติดตั้งคลิกปุ่ม Browse... ดังภาพที่ ข.2



ภาพที่ ข.2 ไคอะลือกระบวนตำหน่งการติดตั้งโปรแกรม

5. คลิกปุ่ม Next> เพื่อไปคำตั้งหน้าถัดไป เพื่อยืนยันการติดตั้งโปรแกรม ดังภาพที่ ข.3

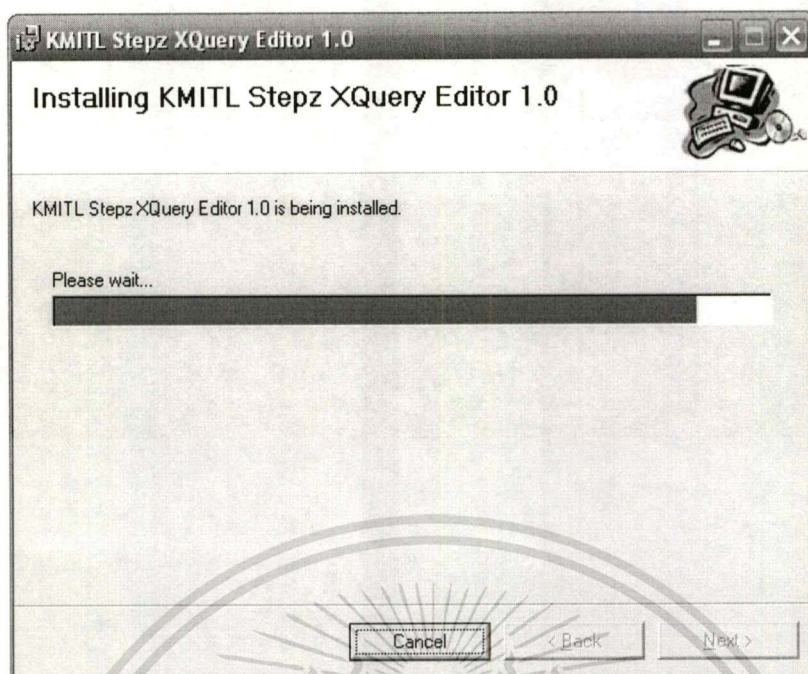


ภาพที่ ข.3 ไคอะลือกยืนยันการติดตั้งโปรแกรม

6. คลิกปุ่ม Next> เพื่อติดตั้งโปรแกรม จะปรากฏไคอะลือกแสดงการทำงาน ดังภาพที่

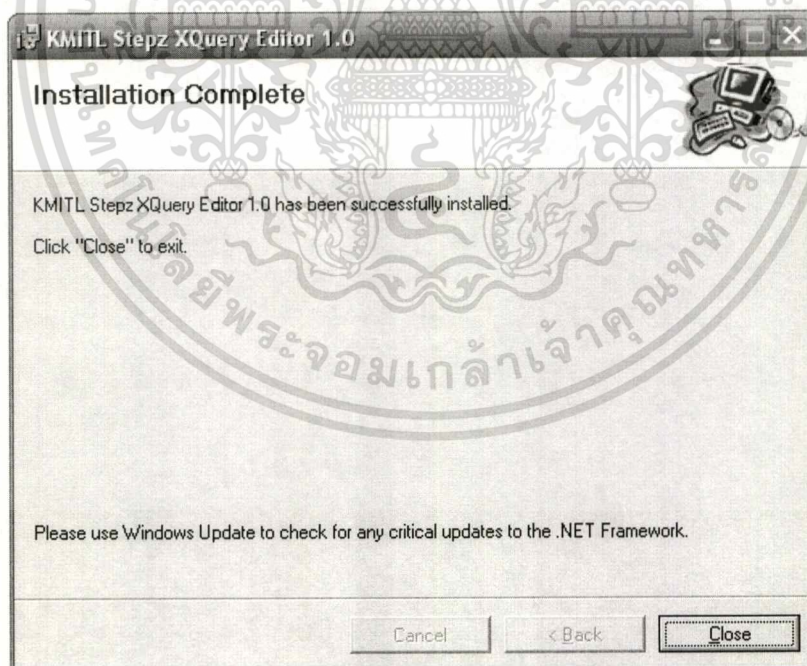
ข.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ ข.4 ไคอะลือกสถานะการดำเนินการติดตั้งโปรแกรม

7. เมื่อเสร็จสิ้นการติดตั้งโปรแกรม จะปรากฏไคอะลือก ดังภาพที่ ข-5



ภาพที่ ข.5 ไคอะลือกการติดตั้งโปรแกรมสมบูรณ์

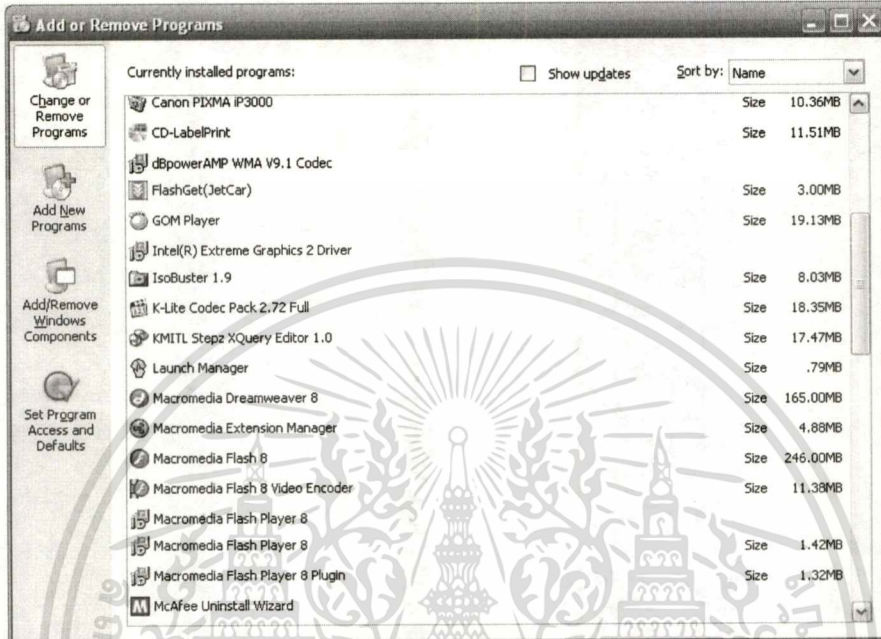
8. คลิกปุ่ม Close เพื่อจบการติดตั้งโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการถอดถอนโปรแกรม

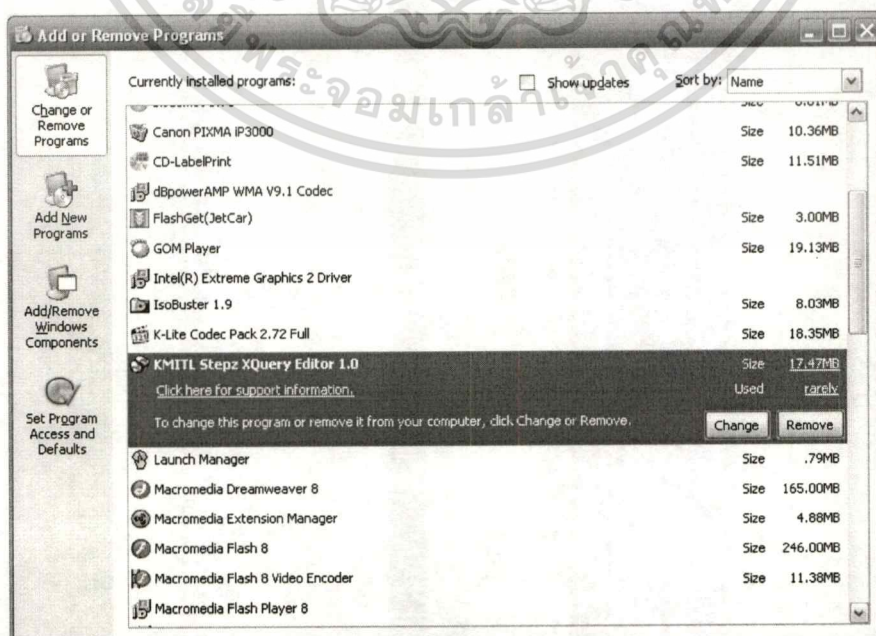
1. เปิดไปที่ Control Panel เลือก Add or Remove Programs จะปรากฏไดอะล็อก ดังภาพ

ที่ ข.6



ภาพที่ ข.6 ไดอะล็อกการเพิ่มและการถอดถอนโปรแกรมบนวินโดวส์

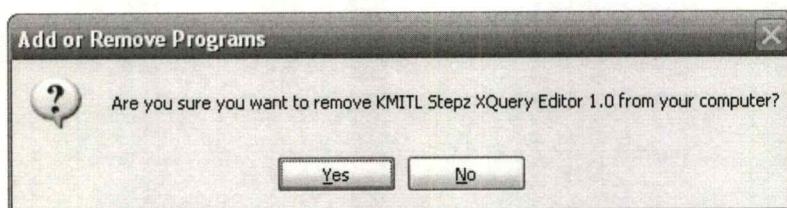
2. ในส่วนของ Change or Remove Programs เลือกไปที่ KMITL Stepz XQuery Editor 1.0 จะปรากฏรายละเอียดขึ้นมา ดังภาพที่ ข.7



ภาพที่ ข.7 ไดอะล็อกรายละเอียดของโปรแกรม KMITL Stepz Editor 1.0

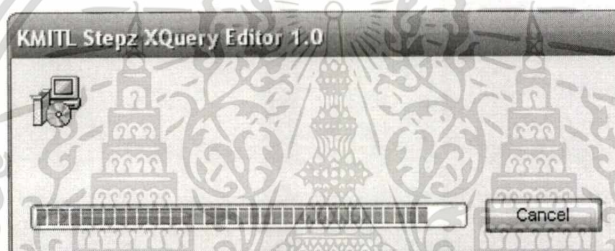
เอกสารนี้เป็นเอกสารที่วางไว้เพื่อความสะดวกในการใช้งานเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลิกปุ่ม Remove เพื่อดำเนินการถอดถอน โปรแกรม จะปรากฏไดอะล็อกถามว่าแน่ใจหรือไม่ที่ดำเนินการถอดถอนโปรแกรม KMITL Stepz Editor 1.0 ดังภาพที่ ข.8



ภาพที่ ข.8 ไดอะล็อกยืนยันการถอดถอนโปรแกรม

4. คลิกปุ่ม Yes เพื่อยืนยันการถอดถอนโปรแกรม หากต้องการยกเลิกคลิกปุ่ม No เมื่อเลือก Yes จะแสดงสถานะการทำงานที่กำลังถอดถอนโปรแกรมออกจากเครื่อง ดังภาพที่ ข.9



ภาพที่ ข.9 ไดอะล็อกสถานะการดำเนินการถอดถอนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายปรีชาพล เหมะสุคนธ์
วัน เดือน ปีเกิด	26 พฤศจิกายน พ.ศ.2524
สถานที่เกิด	กรุงเทพฯ
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตร์บัณฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยราชภัฏจันทรเกษม
ปีที่สำเร็จการศึกษา	2546



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้