NECTEC
Technical Journal
ISSN 1513-2145

วารสารวิชาการเนคเทค ปีที่ 3 ฉบับที่ 10 เดือน มีนาคม 2544 – มิถุนายน 2544 Vol. III, No. 10, March 2001 – June 2001

Short Paper

Tutorial

การรู้จำหน่วยเสียงสระเสียงเดี่ยวสำหรับภาษาไทยโดยการใช้ ทรานส์เฟอร์ฟังก์ชันของอวัยวะกำทอนเสียงบนสเกลบาร์ก

Unmixed Vowel Utterances Recognition in Thai spoken language using Vocal Tract Transfer functions on the Bark scale

นริศ บุญศักดิ์เฉลิม วรา คงคาวิทูร ไกรสิน ส่งวัฒนา

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

3-2 ถนนฉลองกรุง เขตลาดกระบัง กรุงเทพ 10520

โทร: 662 3269967 ; Email: kskraisi@kmitl.ac.th

ABSTRACT - In this paper we address the problem of unmixed vowel utterances recognition in Thai spoken language. The vowel utterances to be recognized are "i", "e", "ε", "ω", "γ", "a", "u", "o", and "⊃". Each utterance is represented by a 18-dimensional feature vector of Critical Band Intensities of the vocal tract transfer function which is determined from Linear Predictive Coding Coefficients. Using the K-Nearest Neighbor rule on feature vectors of sample utterance show the maximum accuracy of 96%.

KEYWORDS - Vowel Recognition, Bark scale, K-Nearest Neighbor

บทคัดย่อ – บทความนี้เสนอวิธีการในการรู้จำหน่วยเสียงสระเสียงเดี่ยวในภาษาไทย และวิธีการปรับปรุงระบบเพื่อให้ได้ประสิทธิภาพสูง โดยหน่วยเสียงสระที่จะทำการรู้จำ คือ เสียง อิ, เอะ, แอะ, อึ, เออะ, อะ, อุ, โอะ และเอาะ แต่ละหน่วยเสียงนั้นจะถูกแทนโดยเวคเตอร์ 18 มิติ โดยแต่ละมิติ คือ Critical Band Intensity ของ Transfer Function ของอวัยวะกำทอนเสียงในการกำเนิดเสียงนั้นๆ ซึ่งคำนวณจากสัมประสิทธิ์ที่ได้จากการทำ Linear Predictive Coding (LPC) การทดลองจะทำการเก็บตัวอย่างเสียงมาทำการคำนวณ Transfer Function ของอวัยวะกำทอนเสียงจาก LPC อันดับต่างๆ แล้วจึงใช้เทคนิค K-Nearest Neighbor ในการแยกแยะเวคเตอร์ ผลการทดลองพบว่าได้ความถูกต้องเฉลี่ยในการรู้จำสูงสุดเท่ากับ 96%

คำสำคัญ - การรู้จำเสียงสระ, สเกลบาร์ก, K-Nearest Neighbor

1. บทนำ

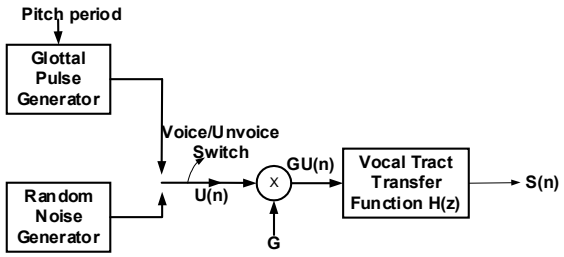
เสียงสระในภาษาไทยประกอบด้วยเสียงสระแท้ทั้งหมด 24 เสียง โดยประกอบไปด้วยเสียงสระเสียงเดี่ยวเสียงสั้นจำนวน 9 เสียง คือ อิ, เอะ, แอะ, อึ, เออะ, อะ, อุ, โอะ และเอาะ เสียงสระเสียงเดี่ยวเสียงยาว 9 เสียง คือ อี, เอ, แอ, อือ, เออ, อา, อู, โอ และออ และเสียงสระผสมอีก 6 เสียง คือ เอียะ, เอีย, เอือะ, เอือ, อัวะ และอัว เสียงสระเสียงเดี่ยวเสียงยาวนั้นเกิดจากการเปล่งหน่วยเสียงสระเสียงเดี่ยวเสียงสั้นในช่วงเวลาที่ยาวขึ้นและมีการหยุดของเสียงที่ช้ากว่าเสียงสระเสียงเดี่ยวเสียงสั้น ส่วนสระผสมนั้นเกิดจากการรวมกันของหน่วยเสียงสระเสียงเดี่ยว 2 ชนิด ดังนั้นในการที่จะรู้จำเสียงสระในภาษาไทยนั้นจะต้องเริ่มต้นจากการรู้จำหน่วยเสียงสระเสียงเดี่ยวให้ได้เสียก่อน

ในบทความนี้นำเสนอวิธีการรู้จำหน่วยเสียงสระเสียงเดี่ยวสำหรับภาษาไทยโดยอาศัย Transfer Function ของอวัยวะกำทอนเสียงบนสเกล Bark ซึ่งสามารถคำนวณหาได้โดยตรงจากสัมประสิทธิ์ที่ได้จากการทำ Linear Predictive Coding สำหรับธรรมชาติของการได้ยินของมนุษย์นั้นเรา

สามารถที่จะคาดคะเนระดับของการตอบสนองต่อเสียงที่ความถี่ใดๆ ของระบบการได้ยินของมนุษย์ได้จาก Critical Band Intensity ดังนั้นในบทความนี้จึงได้ทำการคำนวณ Transfer Function ของอวัยวะกำทอนเสียง บน Critical Band Rate Scale หรือเรียกว่า สเกลบาร์กและใช้ Critical Band Intensity เป็น Feature Vector ในการแยกแยะและจัดกลุ่มเสียงสระแต่ละเสียงด้วยเทคนิค K-Nearest Neighbor

2. Feature Extraction

เนื่องจากความแตกต่างของเสียงสระแต่ละเสียงนั้นเกิดจากความแตกต่างของ Transfer Function ของอวัยวะกำทอนเสียงในการกำเนิดเสียงนั้น ดังในรูปที่ 1 ดังนั้นเราจึงสามารถรู้จำเสียงสระโดยการรู้จำ Transfer Function ของอวัยวะกำทอนเสียงได้



รูปที่ 1. แสดงแผนภาพระบบกำเนิดสัญญาณเสียงพูดของมนุษย์

2.1 การคำนวณ Transfer Function ของอวัยวะกำทอนเสียงจากสัมประสิทธิ์ที่ได้จากการทำ LPC

ในกระบวนการทำ LPC นั้นเริ่มต้นจากสัญญาณเสียงที่เวลา n ซึ่งมีค่าเป็น $s(n)$ นั้นสามารถถูกคาดคะเนได้จากการคำนวณเชิงเส้นของสัญญาณที่ผ่านมาจำนวน p ค่า [4]

$$s(n)\approx\sum_{k=1}^{p}a_k s(n-k) \qquad (1)$$

โดยที่ a_k เป็นสัมประสิทธิ์ของการทำ LPC ที่ทำให้สัญญาณที่ผ่านมา k ค่าในการคำนวณเราสามารถที่จะแปรการประมาณ (1) ให้เป็นสมการโดยการรวมเอาสัญญาณการกระตุ้น $Gu(n)$ ตามรูปที่ 1 เข้าไปด้วยจะได้

$$s(n)=\sum_{k=1}^{p}a_k s(n-k)+Gu(n) \qquad (2)$$

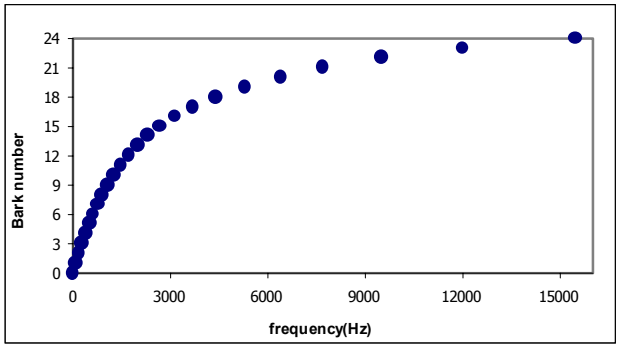
โดยที่ $u(n)$ เป็นสัญญาณการกระตุ้นที่ถูกนอร์มไลซ์แล้ว และ G คือ อัตราการขยาย และเมื่อแปรรูปสมการให้มาอยู่ในรูปฟังก์ชันได้เป็น

$$S(z)=\sum_{k=1}^{p}a_k z^{-k}S(z)+GU(z) \qquad (3)$$

ซึ่งจัดรูปให้สามารถคำนวณหาฟังก์ชันถ่ายโอนของระบบกำทอนเสียงได้

$$H(z)=\frac{S(z)}{GU(z)}=\frac{1}{1-\sum_{k=1}^{p}a_k z^{-k}} \qquad (4)$$

2.2 Critical Band Rate Scale หรือสเกลบาร์ก



รูปที่ 2. แสดงความสัมพันธ์ระหว่างลำดับของ Bark scale กับความถี่ Hz

สเกลที่ใช้ในการวัดระดับความรู้สึกของหูที่มีต่อความถี่ของเสียงนั้น จัดเป็นสเกลที่เรียกว่า Psychoacoustic ซึ่งจะมีความใกล้เคียงกับระบบการได้ยินของมนุษย์มาก โดยค่าความเข้มพลังงานที่แสดงอยู่ในสเกลบนสเกลบาร์กนั้น จะแสดงให้เห็นถึงระดับการตอบสนองต่อเสียงที่ความถี่นั้น ๆ ของระบบการได้ยินของมนุษย์ โดยมีความสัมพันธ์ระหว่างสเกลบาร์กกับสเกลความถี่ปกติเป็นดังกราฟในรูปที่ 2 และสมการ

ประมาณความสัมพันธ์ดังกล่าวในสมการ 5 ซึ่งใช้ได้ดีช่วงความถี่ที่มนุษย์สามารถได้ยิน

$$B = 13\arctan(0.76f/KHz) + 3.5\arctan(f/7.5KHz)^2 \quad (5)$$

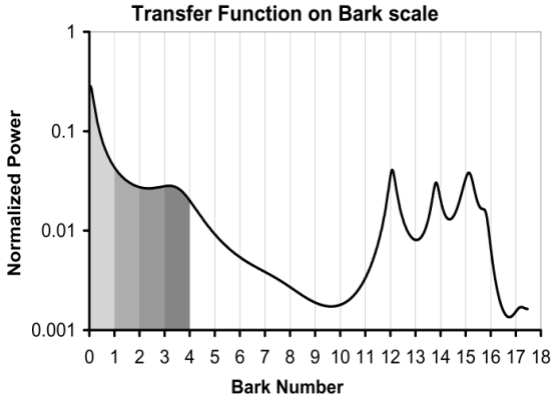
โดยที่ f เป็นความถี่ในหน่วย Hz และ B คือ Bark number ของแถบวิกฤตนั้นๆ

2.3 การใช้ Critical Band Intensity เป็น Feature Vector

ระดับการตอบสนองต่อเสียงที่ความถี่ใดๆ ของระบบการได้ยินเสียงของมนุษย์นั้น สามารถคาดคะเนได้จากการหาความเข้มของสัญญาณเสียงในแต่ละ Critical Band ดังนี้

$$I_B = \int_{B-0.5Bark} \frac{dI}{dB} dB \quad (6)$$

โดยที่ B คือ Bark number ซึ่งขึ้นอยู่กับความถี่ตามสมการที่ 5 ในโดเมนของ Bark ตัวรูปที่ 3 ในบทความนี้เสียงจะถูกรวมที่ความถี่ 11.025 KHz ดังนั้นความถี่สูงสุดของเสียงที่รวมได้คือ 5.5125 KHz ซึ่งอยู่ในบาร์กที่ 18 ดังนั้นเราจึงได้ Critical Band Intensity ในช่วงที่ 1 ถึง 18 ของแต่ละเสียงที่ความถี่ต่างๆ กันไป



รูปที่ คำนวณ แสดง Critical band intensity

3. การแยกแยะโดยใช้เทคนิค K-Nearest Neighbor

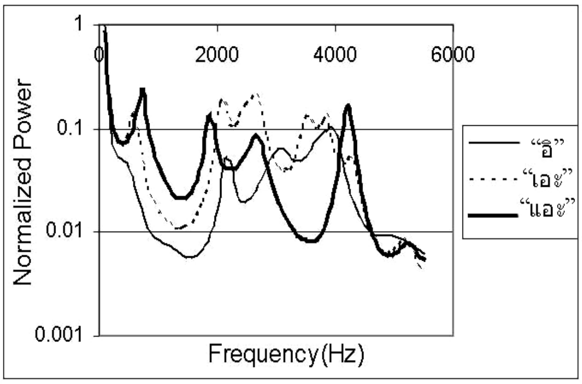
KNN [5] เป็นเทคนิคที่นิยมใช้ในการทำ Pattern Classification โดยในบทความนี้มีเซตของ Feature Vector 18 มิติของ Critical Band Intensity ทั้ง 18 band จากเสียงตัวอย่างจำนวน N เสียง ซึ่งมีเสียงสระ 9 รูปแบบ โดยที่แต่รูปแบบของเสียงสระนั้นจะสามารถอธิบายได้โดยเซตของเวคเตอร์ 18 มิติ

โดยที่ i อยู่ในช่วง 1 ถึง N และเมื่อมีเวคเตอร์เสียงที่ไม่ทราบกลุ่ม $I_{Unknown}=[i_i,i_2,...,i_{18}]$ และต้องการที่จะจัดเข้ากลุ่มใดกลุ่มหนึ่งใน 9 กลุ่ม ซึ่งสามารถที่จะทำโดยการคำนวณระยะห่าง $d(I_{Unknown},I_i)$ ซึ่งเป็นระยะห่างแบบ Euclidean ระหว่างเวคเตอร์ที่ไม่ทราบกลุ่มกับเวคเตอร์ทุกตัวในเซต แล้วจึงทำการคำนวณหาเวคเตอร์ที่ใกล้เคียงที่สุด 9 ตัวกับเวคเตอร์ที่ต้องการจัดกลุ่มและสามารถทำการจัดกลุ่มเวคเตอร์นั้นได้ โดยตรวจสอบดูว่าในเวคเตอร์ทั้ง 9 ตัวที่ใกล้เคียงที่สุดนั้นตรงกับกลุ่มใดมากที่สุดก็จะจัดให้เวคเตอร์ที่ต้องการทราบกลุ่มถูกจัดเข้ากับกลุ่มนั้น

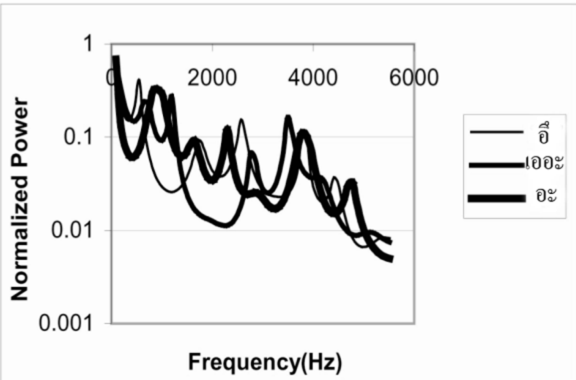
4. การทดลอง

การทดลองในบทความนี้ได้ทำการเก็บตัวอย่างเสียงจากผู้ชาย 5 คน และผู้หญิง 5 คนในห้องทำงานปกติ โดยให้แต่ละคนออกเสียงเป็นคำๆ ที่มีเสียงพยัญชนะและสระรวมอยู่ด้วยกัน เสียงสระที่ประกอบไปด้วยหน่วยเสียงทั้งหมด 9 แบบ ทั้งเสียงนั้นและเสียงบางรวมเป็น 18 เสียงสระรวมกับพยัญชนะทั้งหมด 21 เสียง และเสียงรวมกัน 5 เสียง รวมเป็น 9×18×21 = 21,060 เสียง เทคนิคในการแยกแยะเสียงที่ใช้ในการทดลองนี้ คือ การหาค่าสัมประสิทธิ์ของเสียงตามการคำนวณสัมประสิทธิ์ LPC คือ ใช้ฟรมเสียงขนาด 30 ms และขยับไปครั้งละ 10 ms ในขั้นต้นจะใช้การคำนวณ LPC อันดับที่ 19 ตาม [2] ซึ่งตัวอย่างของ Transfer function ของเสียงถูกส่งออกมาเป็นกลุ่มของค่าหน้า

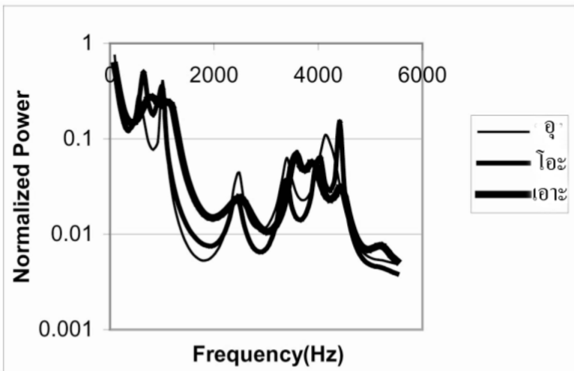
Transfer ของเสียงสระที่ได้แบ่งเป็นกลุ่มเสียงของสระหน้า (อิ, เอะ, แอะ) สระกลาง (อึ, เออะ, อะ) และสระหลัง (อุ, โอะ, เอาะ) [3] ในรูปที่ 2(a), 2(b) และ 2(c) ตามลำดับ



(a) สระหน้า



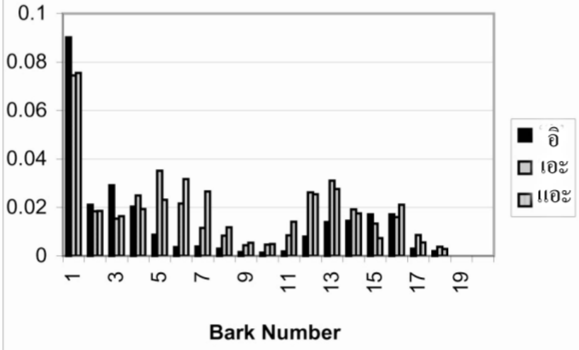
(b) สระกลาง



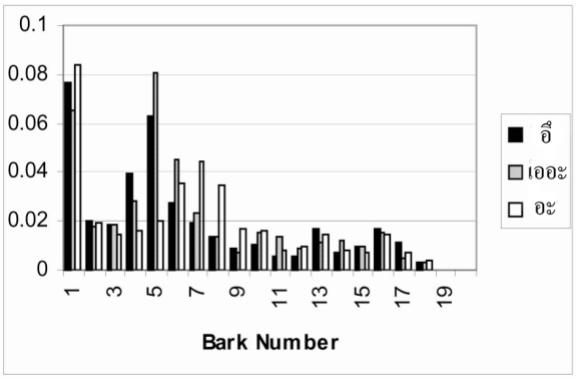
(c) สระหลัง

รูปที่ 2 แสดงตัวอย่าง Transfer Function ของ ... กำหนดเสียง ของเสียงสระเสียงเดี่ยวทั้ง 9 เสียง

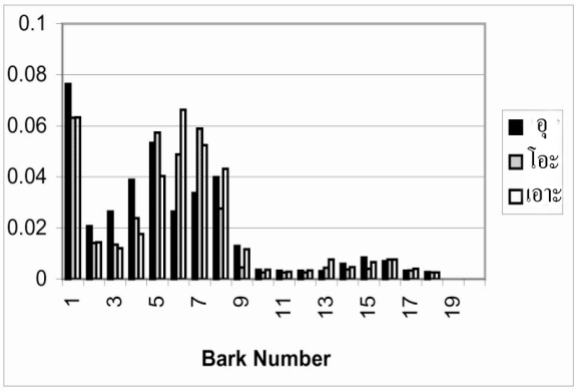
การแบ่งแยกสระหน้า สระกลาง และสระหลังนั้น เป็นการแบ่งตามลักษณะตำแหน่งของลิ้น ในขณะที่กำลังออกเสียงนั้น ๆ ในรูปที่ 4 จะเห็นถึงความแตกต่างของ ... เราจะทำการแปลง ... ให้มีมือขึ้น ... การคำนวณ ... d



ในสระหน้า



(b) *สระกลาง*



(c) *สระหลัง*

รูปที่แสดง Critical Band ของสระ

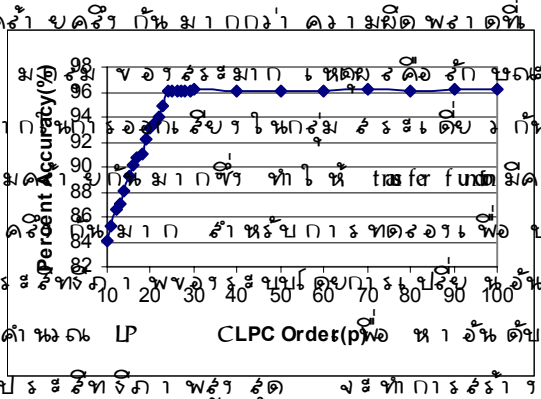
Band Intensity ที่ 18 Band ของเสียงสระเดี่ยวทั้ง 9 เสียง

Feature Vector ที่ได้จากตัวอย่างเสียง 21,600 เสียงนั้นจะถูกใช้เป็น Training Set สำหรับสร้างแบบอ้างอิง และในการทดสอบแบบอ้างอิงนั้นใช้วิธี KNN โดยที่จะไม่ใช้ Nearest Neighbor เพื่อจะไม่ให้เกิดการ self-matching ซึ่งจะทำให้ vector ที่นำเข้าไปทดสอบไม่เป็นส่วนหนึ่งของ training set ซึ่งได้ผลการทดลองดังตารางที่ 1

ตารางแสดงความถูกต้องในการรู้จำเสียงสระเดี่ยว

แบบอ้างอิง		สระหน้า			สระกลาง			สระหลัง			ความถูกต้อง
แบบทดสอบ		"อึ"	"เอะ"	"แอะ"	"อึ"	"เออะ"	"อะ"	"อุ"	"โอะ"	"เอาะ"	%
สระหน้า	"อึ"	2337	5	28	1	10	0	15	4	0	97.38
	"เอะ"	120	2220	40	0	20	0	0	0	0	92.50
	"แอะ"	60	137	2153	10	40	0	0	0	0	89.71
สระกลาง	"อึ"	10	10	0	2046	220	44	0	10	60	85.25
	"เออะ"	0	0	17	66	2284	33	0	0	0	95.17
	"อะ"	0	0	10	40	80	2217	43	0	10	92.38
สระหลัง	"อุ"	0	0	3	0	0	10	2300	57	30	95.83
	"โอะ"	0	0	2	0	80	10	120	2128	60	88.67
	"เอาะ"	0	10	10	20	0	40	22	75	2223	92.63
										เฉลี่ย	92.17

ในตารางที่ 1 นั้น คอลัมน์ทางซ้ายเป็นเสียงสระที่นำเข้ามาทดสอบ ส่วนแถวบนเป็นเสียงสระที่เป็นแบบอ้างอิง ตัวเลขในตาราง คือ จำนวนของเสียงในสระที่ถูกจัดเข้ากับแบบอ้างอิงใดๆ โดยใช้วิธี KNN ส่วนคอลัมน์ขวามือสุดจะเป็นเปอร์เซ็นต์ความถูกต้องของแต่ละเสียงสระ เปอร์เซ็นต์ความถูกต้องเฉลี่ยที่ได้ คือ 92.17% เราสามารถสังเกตเห็นได้ชัดว่าความผิดพลาดที่เกิดขึ้นในการรู้จำส่วนใหญ่เกิดขึ้นภายในกลุ่มสระเดียวกัน คือ ภายในกลุ่มสระหน้า สระกลาง และสระหลัง ซึ่งมีความคล้ายคลึงกันมากกว่าความผิดพลาดที่เกิดขึ้นข้ามกลุ่มของสระมาก เหตุผลคือรูปแบบของช่องปากในการออกเสียงในกลุ่มสระเดียวกันจะมีความคล้ายกันมาก ซึ่งทำให้ transfer function มีความคล้ายคลึงกันมาก สำหรับการทดลองเพื่อปรับปรุงประสิทธิภาพของระบบโดยการเปลี่ยนอันดับในการคำนวณ LPC (LPC Order (p)) เพื่อหาอันดับที่ทำให้ได้ประสิทธิภาพสูงสุด จะทำการสร้างแบบอ้างอิงที่ใช้ LPC อันดับที่ 10 ถึง 100 ได้ความถูกต้องในการรู้จำเสียงดังรูปที่ 6



รูปที่ เทียบเปอร์เซ็นต์ความถูกต้องในการรู้จำเสียง กับอันดับที่ใช้ในการคำนวณ LPC

จากรูปจะเห็นได้ว่า การคำนวณ LPC อันดับที่ต่ำเกินไปจะทำให้ประสิทธิภาพในการรู้จำของระบบต่ำ แต่ถ้าอันดับสูงเกินไปก็จะไม่ทำให้ประสิทธิภาพของระบบดีขึ้นเท่าใดนัก ดังนั้นจากรูปจะเห็นได้ว่า อันดับที่ทำให้ระบบเกิดประสิทธิภาพสูงสุดคืออันดับที่ 24 ให้ความถูกต้องประมาณ 90% ซึ่งมากกว่าการใช้อันดับที่ 19 จากผลการทดลองในตารางที่ 1 ประมาณ 4%

5. สรุป

การแปลง [illegible] และการใช้เทคนิค KNN ในการจัดแยกประเภท [illegible] เสียงวรรณยุกต์ในภาษาไทยได้ความถูกต้องถึง 90% และจากการทดลองเพื่อปรับปรุงประสิทธิภาพของระบบแสดงให้เห็นว่า การใช้อันดับในการคำนวณ LPC ที่ต่ำเกินไปทำให้ประสิทธิภาพของระบบต่ำลงมาก แต่การใช้อันดับที่สูงมากๆ ก็ไม่ได้ช่วยให้ความถูกต้องในการรู้จำมากขึ้นเท่าใดนัก แต่กลับทำให้ความซับซ้อนในการคำนวณมากขึ้น ส่วนจุดที่ทำให้ระบบมีประสิทธิภาพสูงสุดในการทดลองนี้อยู่ที่อันดับที่ 24 ผลของการรู้จำเสียงวรรณยุกต์เดี่ยวนั้น [illegible] วิธีการอื่นๆ เช่น Hid

เอกสารอ้างอิง

[1] ไกรสิน ส่งวัฒนา, อิทธิชัย อรุณศรีแสงไชย, จิตรลดา จารุมิศร์, “แบบจำลองเสียงวรรณยุกต์สำหรับภาษาไทยโดยใช้เทคนิคการควอนไตซ์พิทช์และ Hidden Makov Modeling”, งานประชุมวิชาการทางวิทยาการคอมพิวเตอร์แห่งชาติ 2541, คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

[2] สุนทร อรอินทร์, อัฐ เครือฟัก, “การประมวลผลเสียงพูดโดยการประมาณเชิงเส้น”, วิทยานิพนธ์ ภาควิชาวิศวกรรมไฟฟ้า สาขาวิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 1995

[3] อมร ทวีศักดิ์, “สัทศาสตร์”, สถาบันวิจัยภาษาและวัฒนธรรมเพื่อพัฒนาชนบท มหาวิทยาลัยมหิดล 1999

[4] [illegible] Rabiner, [illegible] “[illegible] Speech Recognition” [illegible] 1993.

[5] [illegible] “Pattern Recognition”, John Wiley, 1993.

[6] D [illegible] “[illegible]”

[7] E [illegible] “P [illegible]”

นริศ บุญศักดิ์ นริม เกิดวันที่ 5 ตุลาคม [illegible] สำเร็จการศึกษาครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิศวกรรมโทรคมนาคม คณะครุศาสตร์อุตสาหกรรม ปีการศึกษา [illegible] ปัจจุบันกำลังศึกษาปริญญาโทคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และทำงานในตำแหน่งวิศวกร บริษัท โยโกกาวา (ประเทศไทย) จำกัด

วรา ตรงคารีทูร เกิดเมื่อวันที่ 5 กุมภาพันธ์ [illegible] สำเร็จการศึกษาวิศวกรรมบัณฑิต ปีการศึกษา [illegible] และวิศวกรรมมหาบัณฑิต ปีการศึกษา [illegible] สาขาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ

ทหารราดกระบัง ปัจจุบัน ทำงานในตำแหน่งวิศวกรการสื่อสารแห่งประเทศไทย

ไกรสิน ส่งวัฒนา เกิดเมื่อวันที่ 16 มิถุนายน 2507, ได้รับปริญญาตรี, ปริญญาโท และปริญญาเอก จาก University of Wisconsin-Madison ประเทศอเมริกา ในปี พ.ศ. 2530, 2532 และ 2536 ตามลำดับ ขณะนี้เป็นผู้ช่วยศาสตราจารย์ประจำภาควิชาโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การวิเคราะห์และเปรียบเทียบภาษาสืบค้นสำหรับ XML ระหว่าง XQuery และ XSLT

เอกพล จีรังสุวรรณ และ สมนึก คีรีโต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

ABSTRACT -- XML is a great technology that changes the way of collecting and using data. It can also merge the usage of database and document together. However, using XML with most efficiency it is necessary to deal with an appropriate query language too. At present, XQuery is the lastest query language for XML that is proposed by W3C, the organization that has the authority to settling many Internet standards. XQuery is being developed continuously and is expected to become the complete standard of query language for XML soon. However, in opinions of authors, XQuery still lacks several features to manage data collecting in document forms. This article is an analysis and comparison between 2 famous XML proposals related to query fields, XQuery and XSLT. XSLT is the standard for transforming XML document that can work with document very well. This article identifies differences, advantages and disadvantages between both languages and hopefully this article can bring some ideas to develop a complete standard query language for XML.

KEY WORDS -- XML, Query Language, XQuery, XSLT

บทคัดย่อ -- XML เป็นเทคโนโลยีซึ่งสร้างรูปแบบใหม่ในการจัดเก็บและใช้งานข้อมูล ทำให้การใช้งานร่วมกันระหว่างฐานข้อมูลและเอกสารเป็นไปอย่างราบรื่น อย่างไรก็ตามการใช้งาน **XML** ให้มีประสิทธิภาพมากที่สุดนั้นจำเป็นต้องทำงานร่วมกับภาษาสืบค้นสำหรับ **XML** ที่เหมาะสมด้วย, ปัจจุบัน **XQuery** นับเป็นภาษาสืบค้นสำหรับ **XML** ล่าสุดที่เสนอโดยหน่วยงาน **W3C** ซึ่งมีหน้าที่กำกับดูแลเกี่ยวกับมาตรฐานต่าง ๆ ของอินเตอร์เน็ต, **XQuery** ยังคงได้รับการพัฒนาอยู่อย่างต่อเนื่องและเป็นที่คาดกันว่า **XQuery** จะกลายเป็นมาตรฐานของภาษาสืบค้นสำหรับ **XML** ที่สมบูรณ์ในอนาคตอันใกล้ อย่างไรก็ตาม ในมุมมองของผู้เขียนมีความเห็นว่า **XQuery** นั้นยังขาดคุณสมบัติที่ดีในการจัดการกับข้อมูล **XML** ที่อยู่ในรูปแบบของเอกสารหลาย ๆ ประการ, บทความนี้นำเสนอการวิเคราะห์และเปรียบเทียบระหว่าง **XQuery** และ **XSLT**, สำหรับ **XSLT** นั้นเป็นมาตรฐานในการแปลงข้อมูลที่ถูกออกแบบมาเพื่อการจัดการกับเอกสารโดยเฉพาะ โดยพยายามชี้ให้เห็นถึงความเหมือนและความแตกต่าง รวมถึงข้อดีและข้อเสียระหว่างภาษาทั้ง 2 แบบ เพื่อเป็นแนวทางในการพัฒนามาตรฐานของภาษาสืบค้นสำหรับ **XML** ที่สมบูรณ์ยิ่งขึ้นต่อไป

คำสำคัญ -- ภาษาสืบค้น, XML, XQuery, XSLT

1. บทนำ

Extensible Markup Language (XML) [7] เป็นภาษา markup เช่นเดียวกับ Hypertext Markup Language (HTML) [8] แต่จะเหนือกว่าตรงที่ XML จะยอมให้ผู้ใช้นิยาม markup หรือ tag ไว้ใช้งานได้เอง ทำให้ไม่ติดกับ รูปแบบและ tag ที่มีจำนวนจำกัดเหมือนกับที่มีอยู่ใน HTML นอกจากนี้ tag ของ XML จะทำหน้าที่ในการขยายความหมายของข้อมูลให้สมบูรณ์ยิ่งขึ้น อันจะทำให้การประมวลผลข้อมูลอัตโนมัติเป็นไปอย่างมี ประสิทธิภาพมากขึ้น ดังตัวอย่างเปรียบเทียบข้อมูลที่อยู่ในรูปแบบของ HTML และ XML ต่อไปนี้

ข้อมูลในรูปแบบของ HTML

```
<b>book</b>
<table>
 <tr><td>year</td><td>1992</td></tr>
 <tr><td>title</td><td>XML Bible</td></tr>
 <tr><td>publisher</td><td>Addison-Wesley</td></tr>
 <tr><td>author</td><td>John Smith</td></tr>
 <tr><td>price</td><td>US$60</td></tr>
</table>
```

ข้อมูลในรูปแบบของ XML

```
<book year="1992">
  <title>XML Bible</title>
  <publisher>Addison-Wesley</publisher>
  <author>
    <last>John</last>
    <first>Smith</first>
  </author>
  <price>US$60</price>
</book>
```

จะเห็นได้ว่าข้อมูลเดียวกัน เมื่อนำเสนอด้วย HTML จะใช้เพื่อการแสดงผลเป็นหลัก และการนำข้อมูลที่อยู่ในรูปของ HTML กลับมาใช้ใหม่หรือนำมาประมวลผล จะเป็นไปได้ยากเพราะ tag <b>, <table>, <tr>, <td> ไม่ได้ช่วยสื่อให้เห็นถึงความหมายของข้อมูลที่อยู่ภายในแต่อย่างไร แต่ XML นั้นอนุญาตให้มี markup ที่สร้างขึ้นเองเพื่อใช้ในการอธิบายความหมายของข้อมูลด้วย เช่น tag <book>, <title>, <publisher> ทำให้เห็นได้ว่า XML นั้นทำให้ข้อมูลมีความหมายสมบูรณ์ยิ่งขึ้นและทำให้ง่ายต่อการนำมาประมวลผล

XML ยังสามารถนำมาใช้ร่วมกับงานทางด้านเอกสาร ทำให้ข้อจำกัดในการใช้งานฐานข้อมูลและเอกสารร่วมกันหมดไป อันจะทำให้สามารถประยุกต์ใช้งานข้อมูลทั้ง 2 รูปแบบได้หลากหลายมากยิ่งขึ้น อย่างไรก็ตามข้อมูลที่เก็บด้วย XML นั้นจะไม่มีประโยชน์เมื่อไม่สามารถนำออกมาใช้ หรือเรียกค้นได้ตามความต้องการ จึงเกิดความจำเป็นในการใช้งานภาษาสืบค้นสำหรับ XML ขึ้น เช่นเดียวกับภาษา Structured Query Language (SQL) [4] ที่ใช้สำหรับฐานข้อมูลแบบสัมพันธ์ (Relational Database) แตกต่างแต่ XML นั้นเป็นฐานข้อมูลที่มีความซับซ้อนมากกว่า และยังสามารถใช้งานร่วมกับข้อมูลที่เป็นเอกสารได้ด้วย

การพัฒนาภาษาสืบค้นสำหรับ XML นั้นเป็นไปอย่างต่อเนื่อง ภาษาสืบค้นหลายๆ แบบถูกนำเสนอออกมา ซึ่งแต่ละแบบก็ล้วนมีจุดเด่นและจุดด้อยแตกต่างกันไป ภาษาสืบค้นสำหรับ XML ในรุ่นหลังๆ ใช้ภาษาสืบค้นในรุ่นแรกๆ เป็นพื้นฐาน คุณสมบัติหลายๆ อย่างได้รับการพัฒนาให้ดีขึ้น ในขณะที่พยายามลดข้อจำกัดและข้อเสียต่างๆลง แต่ปัญหาที่สำคัญที่สุดในการออกแบบก็คือ การทำให้ภาษาสืบค้นสำหรับ XML นี้สามารถใช้งานได้ทั้งฐานข้อมูลและเอกสารร่วมกันอย่างสมบูรณ์ โดยทั่วไป ภาษาสืบค้นที่พัฒนามาในแนวทางของฐานข้อมูลก็จะทำงานกับฐานข้อมูลได้ดีแต่ไม่สามารถทำงานกับเอกสารได้ดีเท่าที่ควร ในขณะที่ภาษาสืบค้นที่พัฒนามาเพื่อเน้นการจัดการกับเอกสาร ก็จะขาดคุณสมบัติในการจัดการฐานข้อมูลที่ดีไป ทำให้ภาษาสืบค้นสำหรับ XML ส่วนใหญ่จะใช้งานได้ดีกับข้อมูลเฉพาะแบบเท่านั้น

ในปัจจุบัน W3C ซึ่งเป็นหน่วยงานที่กำกับดูแลเกี่ยวกับมาตรฐานต่างๆ ของอินเตอร์เน็ตได้เสนอภาษาสืบค้นสำหรับ XML แบบล่าสุดออกมาคือ XQuery [14] ซึ่งได้รับการพัฒนามาอย่างต่อเนื่องโดยกลุ่มนักวิจัยของ W3C, แม้ขณะนี้ XQuery ยังมีสถานะเป็นแค่ Working Draft ซึ่งจะต้องได้รับการพิจารณา และปรับปรุงอีกมาก แต่ก็เป็นที่คาดกันว่า XQuery จะได้รับการพัฒนาต่อจนกลายเป็นมาตรฐานของภาษาสืบค้นสำหรับ XML ที่สมบูรณ์ในอนาคตอันใกล้นี้, XQuery นั้นถูกพัฒนาโดยมีพื้นฐานมาจากภาษา SQL ซึ่งใช้งานได้ง่ายและเป็นที่ยอมรับอย่างแพร่หลาย จึงทำให้ XQuery สามารถทำความเข้าใจและใช้งานได้ง่ายเช่นเดียวกัน อย่างไรก็ตามด้วยเหตุผลนี้จึงทำให้ XQuery ต้องใช้รูปแบบการทำงานที่ไม่ใช่ XML นอกจากนี้ XQuery ยังถูกพัฒนามาในแนวทางของ ฐานข้อมูล จึงทำให้ขาดคุณสมบัติที่ดีในการจัดการข้อมูลที่อยู่ในรูปของเอกสารหลายๆ ประการไป, รายละเอียดและการทำงานของ XQuery จะกล่าวถึงต่อไปในหัวข้อที่ 2, การแนะนำ XQuery

ในบทความนี้ จะกล่าวถึงมาตรฐานอีกหนึ่งมาตรฐานซึ่งมีบทบาทสำคัญในการจัดการกับเอกสารที่อยู่ในรูปของ XML ก็คือ Extensible Stylesheet Language - Transformation (XSLT) [15] , XSLT ถูกออกแบบมาเพื่อการแสดงผลเอกสาร XML, อย่างไรก็ตาม XSLT กลับสามารถทำงานในการสืบค้นข้อมูล XML ได้เป็นอย่างดี [2] นอกจากนี้ยังมีบางบทความชี้ให้เห็นว่าคำสั่งและการทำงานต่างๆ ของ XQuery นั้น ซ้ำซ้อนกับที่มีอยู่แล้วใน XSLT [3] อย่างไรก็ตามในความคิดเห็นของ ผู้เขียน บทความข้างต้นเป็นมุมมองจากทางด้านของ XSLT มากจนเกินไป ผู้เขียนยอมรับว่าความสามารถหลายๆ อย่างของ XQuery คล้ายคลึงหรือซ้ำซ้อนกับ XSLT จริง อย่างไรก็ตามบทความข้างต้นเป็นเพียงการแสดงตัวอย่างเปรียบเทียบให้เห็นเท่านั้น ไม่ได้มีการวิเคราะห์ให้เห็นถึงรายละเอียดและแนวคิดพื้นฐานที่แตกต่างกัน รวมถึงไม่ได้นำเสนอใน มุมมองที่เป็นข้อได้เปรียบของ XQuery และข้อเสียที่ชัดเจนของ XSLT ออกมา, รายละเอียดและการทำงานของ XSLT จะกล่าวถึงต่อไปใน หัวข้อที่ 3, การแนะนำ XSLT

ในความคิดเห็นของผู้เขียน ทั้ง XQuery และ XSLT นั้นแม้จะมีความสามารถบางส่วนที่ซ้ำซ้อนกันบ้าง แต่ก็ยังมีความสามารถและรายละเอียดอีกหลายๆ ส่วนที่ไม่สามารถทำงานทดแทนกันได้ ซึ่งทำให้ภาษาทั้ง 2 แบบมีความสามารถเฉพาะที่แตกต่างกันไป, บทความนี้เป็นการวิเคราะห์ให้เห็นถึงจุดเหมือน และจุดที่แตกต่างที่สำคัญบางประการระหว่างภาษาทั้ง 2 แบบ โดยหวังว่าจะสามารถเป็นแนวทางในการรวมความสามารถและจุดเด่นที่มีอยู่ในทั้ง 2 ภาษานี้เข้าด้วยกัน เพื่อจะนำไปสู่การพัฒนาภาษาสืบค้นสำหรับ XML ที่สมบูรณ์ สามารถเรียกค้นข้อมูลได้ตามความต้องการได้อย่างยืดหยุ่น และสามารถทำงานร่วมกันระหว่างข้อมูลและเอกสารได้อย่างเหมาะสมที่สุด

ในบทความนี้ จะใช้ตัวอย่างเพื่อช่วยให้สามารถเข้าใจการทำงานของภาษาทั้ง 2 แบบได้ดียิ่งขึ้น โดยกำหนดให้เอกสาร XML ตัวอย่างคือ bib.xml โดยมี Document Type Definition (DTD) ดังนี้

```
<!ELEMENT bib (book*)>
<!ELEMENT book (title, (author+ |editor+),
```

```
        publisher, price)>
<!ATTLIST book year CDATA>
<!ELEMENT author (last, first)>
<!ELEMENT editor (last, first, affiliation)>
<!ELEMENT title #PCDATA>
<!ELEMENT last  #PCDATA>
<!ELEMENT first #PCDATA>
<!ELEMENT affiliation #PCDATA>
<!ELEMENT publisher #PCDATA>
<!ELEMENT price #PCDATA>
```

เอกสาร bib.xml มี root element เป็น bib (บรรณานุกรม) โดยใน bib ประกอบไปด้วยหลายๆ book ซึ่งหมายถึงหนังสือแต่ละเล่ม ในหนังสือแต่ละเล่มจะประกอบไปด้วย attribute year ซึ่งแสดงปีที่พิมพ์, ชื่อหนังสือ (title), รายชื่อผู้แต่ง (author) หรือรายชื่อบรรณาธิการ (editor) อย่างใดอย่างหนึ่ง ตามด้วยชื่อสำนักพิมพ์ (publisher) และราคาหนังสือ (price), ชื่อผู้แต่งและบรรณาธิการจะประกอบไปด้วยนามสกุล (last) และ ชื่อ (first) นอกจากนั้นชื่อบรรณาธิการยังประกอบไปด้วยชื่อหน่วยงาน (affiliation) ด้วย

2. แนะนำ XQuery

XQuery เป็นภาษาสืบค้นสำหรับ XML ในรุ่นหลัง ซึ่งได้รับอิทธิพลมาจากภาษาสืบค้นสำหรับ XML ในรุ่นแรกๆ และมาตรฐานอื่นๆ ที่เกี่ยวข้อง อันได้แก่ XPath [10], XQL [5], SQL, XML-QL [1], OQL [6] เป็นต้น ดังมีรายละเอียดต่อไปนี้

XQuery จะใช้แนวคิดของ path expression ในการระบุไปยังข้อมูลตำแหน่งต่างๆ ในเอกสารที่กำลังสนใจ, path expression ของ XQuery จะอ้างอิงมาจากมาตรฐานของ XPath เสริมด้วยคำสั่งหรือฟังก์ชันบางส่วนที่ขอยืมมาจาก XQL, path expression เป็นวิธีที่สะดวกและมีประสิทธิภาพมากในการระบุไปยังข้อมูลที่สนใจในโครงสร้างที่ซับซ้อนของเอกสาร ตัวอย่าง path expression เช่น document ("bib.xml")/bib/book หมายถึง element ที่ชื่อ book ทุกๆ อันที่อยู่ภายใต้ root element ที่ชื่อ bib ของเอกสาร bib.xml

โครงสร้างการทำงานของ XQuery จะประกอบไปด้วยลำดับของประโยค FOR - LET - WHERE - RETURN ซึ่งได้รับอิทธิพลมาจากภาษา SQL ที่เป็นลำดับประโยค SELECT - FROM - WHERE, โดยประโยค FOR จะระบุไปยังข้อมูลที่กำลังสนใจโดยอาศัย path expression และนำข้อมูลนั้นมากำหนดให้กับตัวแปร ในแต่ละรอบของการทำงาน, ประโยค LET จะใช้ในการกำหนดค่าให้กับตัวแปรเพื่อช่วยเสริมการทำงานกับประโยค FOR, ประโยค WHERE จะตรวจสอบเงื่อนไขของข้อมูลที่เก็บอยู่ใน ตัวแปรที่ได้มาจากในส่วนของประโยค FOR, LET ที่อยู่ข้างต้นและส่งผลลัพธ์ที่ตรงตามเงื่อนไขให้กับส่วนประโยค RETURN เพื่อสร้าง ผลลัพธ์ที่ต้องการ

XQuery จะใช้งานตัวแปรเพื่อเชื่อมโยงความสัมพันธ์ระหว่างส่วนของประโยคต่างๆ เข้าด้วยกัน ซึ่งได้รับอิทธิพลมาจาก XML-QL, และ XQuery จะประกอบไปด้วยการใช้งานประโยคคำสั่งต่างๆ, การเรียกใช้งานฟังก์ชัน, การนิยามฟังก์ชัน เช่นเดียวกับที่มีในภาษา OQL

XQuery นั้นถือได้ว่าเป็นภาษาสืบค้นสำหรับ XML ที่มีประสิทธิภาพมากที่สุดในปัจจุบัน อย่างไรก็ตาม XQuery นั้นถูกพัฒนามาในแนวทางของฐานข้อมูล จึงทำให้สามารถจัดการกับข้อมูลที่อยู่ในรูปของฐานข้อมูลได้ดี แต่สำหรับข้อมูลที่อยู่ในรูปของเอกสารนั้นยังมีข้อจำกัดอยู่บางประการ ดังจะได้พิจารณาให้เห็นต่อไป

ตัวอย่างการใช้งาน XQuery เพื่อเรียกค้นข้อมูล เป็นดังนี้

```
<results>
 {
   FOR $b IN document("bib.xml")/bib/book
   WHERE $b/publisher = "Addison-Wesley" AND
         $b/@year > 1991
   RETURN
     <book year={ $b/@year }>
       { $b/title }
     </book>
 }
</results>
```

จากตัวอย่าง เป็นการหารายชื่อหนังสือและปีที่พิมพ์ โดยที่หนังสือเล่มนั้นๆ ต้องพิมพ์โดยสำนักพิมพ์ Addison-Wesley หลังจากปี 1991, ประโยค FOR จะกำหนดค่าของแต่ละ element ที่ชื่อ book ที่คืนมาจาก path expression, document("bib.xml")/bib/book มาเก็บไว้ในตัวแปร $b, ต่อจากนั้นประโยค WHERE จะตรวจสอบเงื่อนไขของค่าที่เก็บอยู่ใน ตัวแปร $b ว่าเป็นไปตามที่ต้องการ ในที่นี้คือมีค่าของ element publisher เป็น "Addison-Wesley" และค่าของ attribute year มากกว่า 1991 (เครื่องหมาย @ แสดงถึงการเป็น attribute), หลังจากนั้นประโยค RETURN ก็จะอาศัยค่าตัวแปร $b เพื่ออ้างอิงถึงค่าของ attribute year และ element title ในการสร้างผลลัพธ์ที่ต้องการ

3. แนะนำ XSLT

ใน HTML นั้น tag ที่ใช้ได้จะมีจำนวนจำกัด เนื่องจาก tag แต่ละตัวจะมีความหมายในแง่ของการแสดงผลที่แตกต่างกันไป เช่น tag ที่ใช้ในการแสดงย่อหน้า, ตาราง หรือ รูปภาพ เป็นต้น แต่ XML จะยอมให้ผู้ใช้นิยาม tag เพื่อใช้งานเองได้ ปัญหาที่ตามมาก็คือ โปรแกรม browser จะไม่รู้จัก tag เหล่านี้ทำให้ไม่สามารถนำมาแสดงผลได้ ดังนั้นเพื่อเป็นการนำเสนอหรือแสดงผลข้อมูล XML จึงมีการพัฒนา XSLT ขึ้น เพื่อใช้ในการแปลงเอกสาร XML, โดยจะแปลง tag ที่ผู้ใช้นิยามขึ้น ให้กลายเป็น tag ที่โปรแกรม browser สามารถเข้าใจ และนำไปแสดงผลได้ ซึ่งโดยทั่วไปก็จะเป็นการแปลงจากเอกสาร XML ให้เป็นเอกสาร HTML เป็นต้น ดังตัวอย่าง

```
<xsl:template match="book">
  <font color="red" size="20">
    <xsl:value-of select="title">
  </font>
```

```
  <table border="1">
    <xsl:apply-templates select="author"/>
  </table>
</xsl:template>

<xsl:template match="author">
  <tr>
    <td><xsl:value-of select="last"/></td>
    <td><xsl:value-of select="first"/></td>
  </tr>
</xsl:template>
```

จากตัวอย่างจะเป็นการแปลงเอกสาร XML ที่มี element ที่ชื่อbook นำมาแสดงผลด้วย HTML โดย เมื่อพบ element ที่ชื่อ book ก็ให้พิมพ์ชื่อหนังสือ (title) ด้วยตัวอักษรสีแดงขนาด 20 และตามด้วยตารางซึ่งแสดงชื่อผู้แต่ง (author) ของหนังสือเล่มนั้นๆ

คำสั่งของ XSLT นั้นจะเป็น element ของ XML ตามปกติ เพียงแต่จะถูกนำหน้าด้วย xsl: (มี prefix ของ namespace [9] เป็น xsl), การทำงานของ XSLT จะประกอบไปด้วยหลาย template, แต่ละ template จะอยู่ภายใน element xsl:template โดยจะมี attribute ที่ชื่อ match ซึ่งจะมีค่าเป็น path expression (XPath) ที่ระบุถึงข้อมูลใดๆ ในเอกสาร XML, เมื่ออ่านข้อมูลภายในเอกสาร XML มา match กับ path expression ของ template ใด template นั้นก็จะถูกเรียกขึ้นมาทำงาน เช่น จากตัวอย่าง เมื่ออ่านข้อมูลภายในเอกสาร XML มาพบ element ที่ชื่อ book, template แรกก็จะถูกเรียกขึ้นมาทำงาน โดยจะพิมพ์ tag <font> และค่าของชื่อหนังสือ ด้วย คำสั่ง xsl:value-of จากนั้นจึงพิมพ์ตาราง <table> โดยเนื้อหาภายใน ตารางนั้นจะส่งการทำงานไปให้ template ที่สองด้วยการส่งค่า element author ที่พบไปให้

นอกจากจะใช้ XSLT เพื่อแปลงเอกสาร XML ให้อยู่ในรูปของ HTML แล้ว XSLT นั้นยังสามารถแปลงเอกสาร XML ให้อยู่ในรูปแบบของเอกสารใดๆ ก็ได้ เช่น ไฟล์ XML เองหรือแม้แต่ไฟล์ข้อความ ทำให้ประโยชน์ของ XSLT ไม่ได้จำกัดอยู่แค่งานแปลงเอกสารเพื่อแสดงผลเท่านั้น แต่สามารถนำมาประยุกต์ใช้งานในการจัดการเอกสารได้อย่างกว้างขวาง

ในที่นี้ได้นำ XSLT มาใช้แทนภาษาสืบค้นในการเรียกดูข้อมูล XML ซึ่งสามารถใช้งานได้ดีในระดับหนึ่ง ดังตัวอย่าง

```
<xsl:template match="/">
  <results>
    <xsl:apply-templates select="bib/book"/>
  </results>
</xsl:template>

<xsl:template match="book[
  publisher = 'Addison-Wesley' and @year > 1991]">
  <book year="{@year}">
    <title>
      <xsl:value-of select="title"/>
    </title>
  </book>
</xsl:template>
```

จากตัวอย่าง จะเป็นการหาหนังสือที่พิมพ์โดยสำนักพิมพ์ "Addison-Wesley" หลังจากปี 1991 เช่นกัน, ในที่นี้ประกอบไปด้วย 2 template โดย template แรกจะมี path expression ที่ match กับ root ของเอกสาร, ภายใน template นี้จะสร้าง element results สำหรับคำตอบ และส่งการทำงานต่อไปให้ template ถัดไปด้วยคำสั่ง xsl:apply-templates เพื่อส่งค่า element bib/book ออกไป, template ต่อมาจะมี path expression ที่รับกันได้กับ element book ที่มีเงื่อนไขเพิ่มเติมอยู่ด้วย หากข้อมูล book ที่ถูกส่งมาจาก template แรก match ตามเงื่อนไขนี้ template นี้ก็จะถูกเรียกให้ทำงานโดยสร้าง element book และข้อมูลอื่นๆ ตามที่ต้องการ

XSLT นั้นถูกออกแบบมาเพื่อการใช้งานกับข้อมูล XML ที่อยู่ในรูปของเอกสารโดยเฉพาะ จึงประกอบไปด้วยคำสั่งและคุณสมบัติมากมายที่สามารถจัดการกับข้อมูลเอกสารและส่วนประกอบอื่นๆ ได้อย่างมีประสิทธิภาพ นอกจากนั้น XSLT ยังสามารถใช้งานในการสืบค้นข้อมูลได้อีก อย่างไรก็ตาม XSLT ก็ยังขาดคุณสมบัติในการเป็นภาษาสืบค้นสำหรับ XML ที่ดีในหลายๆ ประการดังจะได้แสดงให้เห็นต่อไป

4. การเปรียบเทียบระหว่าง XQuery และ XSLT

4.1 โครงสร้างการทำงานพื้นฐาน

การทำงานในการเรียกค้นข้อมูลของ XQuery นั้นจะใช้ประโยค FOR - LET - WHERE - RETURN ในขณะที่ XSLT จะใช้ template, การทำงานของภาษาทั้ง 2 แบบมีจะมีความคล้ายคลึงกันบางอย่าง ซึ่งสามารถแสดงให้เห็นจากตัวอย่างต่อไปนี้

การทำงานของ XQuery

```
FOR $b IN document("bib.xml")/bib/book
RETURN <book year={ $b/@year }>
       { $b/title }
     </book>
```

การทำงานของ XSLT

```
<xsl:template match="/bib/book">
  <book year="{@year}">
    <title>
      <xsl:value-of select="title"/>
    </title>
  </book>
</xsl:template>
```

ภายในประโยค FOR และคำสั่ง xsl:template จะมีส่วนที่ระบุไปยัง path expression เช่นเดียวกัน โดยหากพบข้อมูลตามที่ระบุไว้ ภายในประโยค FOR หรือ template นั้นๆ ก็จะทำงานเพื่อสร้างคำตอบ และจะวนไปจนกว่าจะครบทุกข้อมูลที่พบ จากตัวอย่างข้างต้นจะเห็นได้ว่าโครงสร้างของ XQuery และ XSLT นั้นไม่มีความแตกต่างกัน อย่างไรก็ตามหากการทำงานภายในแต่ละรอบมีรายละเอียดลึกลงไปอีก ก็จะ

สามารถมองเห็นถึงความแตกต่างระหว่างภาษาทั้ง 2 แบบได้ ดังตัวอย่าง

การทำงานของ XQuery

```
FOR $b IN document("bib.xml")/bib/book
RETURN <book year={ $b/@year }>
        { $b/title }
        {
          FOR $a IN $b/author
          RETURN $a
        }
       </book>
```

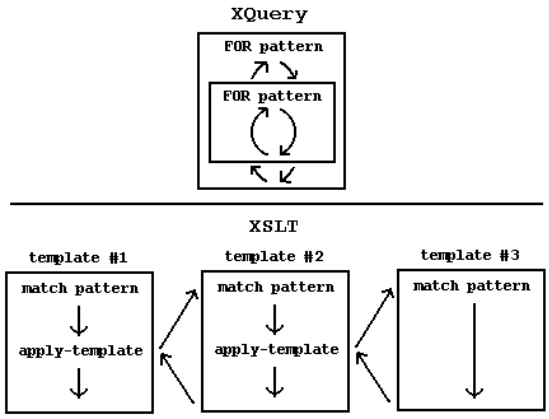
การทำงานของ XSLT

```
<xsl:template match="/bib/book">
  <book year="{@year}">
    <title>
      <xsl:value-of select="title"/>
    </title>
    <xsl:apply-templates select="author"/>
  </book>
</xsl:template>

<xsl:template match="author">
  <author>
    <last>
      <xsl:value-of select="last"/>
    </last>
    <first>
      <xsl:value-of select="first"/>
    </first>
  </author>
</xsl:template>
```

ตัวอย่างนี้จะคล้ายกับตัวอย่างก่อนหน้านี้ เพียงแต่จะเพิ่มการเรียกดูในส่วนของชื่อผู้แต่งขึ้นมา ในหนังสือแต่ละเล่มอาจจะมีผู้แต่งได้หลายคน ดังนั้น XQuery จะใช้ FOR ซ้อนอีกชั้นเพื่อ match ไปยังชื่อผู้แต่งภายใน FOR ครั้งแรกที่ match ไปยังหนังสือแต่ละเล่ม ในขณะที่ XSLT จะแยกการทำงานในส่วนนี้ออกเป็นอีก template แล้วให้ template แรกส่งการทำงานต่อมาให้

XQuery จะทำงานโดยมีประโยค FOR เป็นหลัก ภายในประโยค FOR จะประกอบไปด้วยประโยค อื่นๆ เช่น ประโยค WHERE, ประโยค RETURN และอาจประกอบไปด้วยประโยค FOR อื่นๆ ด้วย, XQuery จะทำงานโดยรวมทุกอย่างไว้ที่จุดเดียวกัน ความซับซ้อนของการทำงานจะอยู่ในทิศทางที่ลึกลงไปในประโยค FOR, ในขณะที่ XSLT จะแยกทำงานย่อยๆ ออกเป็นแต่ละ template และให้ template เหล่านี้ส่งการทำงานไปมาด้วยความสัมพันธ์ของคำสั่ง xsl:apply-templates และ พารามิเตอร์ match ของคำสั่ง xsl:template



รูปที่ 1. แสดงโครงสร้างการทำงานพื้นฐานของ XQuery และ XSLT

แม้การทำงานโดยทั่วไปของ XSLT จะแบ่งการทำงานย่อยๆ ออกเป็น template, อย่างไรก็ตาม XSLT นั้นสามารถมีโครงสร้างในการทำงานเหมือนประโยค FOR ของ XQuery ด้วยคำสั่ง xsl:for-each ดังตัวอย่าง

```
<xsl:template match="/bib/book">
  <book year="{@year}">
    <title>
      <xsl:value-of select="title"/>
    </title>
    <xsl:for-each select="author"/>
      <author>
        <last>
          <xsl:value-of select="last"/>
        </last>
        <first>
          <xsl:value-of select="first"/>
        </first>
      </author>
    </xsl:for-each>
  </book>
</xsl:template>
```

จากตัวอย่างนี้ แทนที่จะใช้คำสั่ง xsl:apply-templates เพื่อส่งการทำงานให้ template อื่น แต่จะใช้คำสั่ง xsl:for-each ทำให้การ match และการทำงานทั้งหมดจะอยู่ภายใน template เดียว ซึ่งมีโครงสร้างการทำงานเป็นเช่นเดียวกับ XQuery

โดยปกติ XQuery จะทำงานโดยประกอบไปด้วยประโยค FOR หลักเพียงส่วนเดียว อย่างไรก็ตาม XQuery สามารถกระจายการทำงานจากส่วนหลักส่วนเดียวไปยังส่วนย่อยอื่นๆ ได้อีก คล้ายแนวคิด template ของ XSLT ด้วยคุณสมบัติของฟังก์ชัน ดังตัวอย่าง

```
FUNCTION My_Function(ELEMENT $b) RETURN ELEMENT {
  <book year={ $b/@year }>
    { $b/title }
  </book>
}

<results>
  {
```

```
  FOR $b IN document("bib.xml")/bib/book
  WHERE $b/@year > 1991 AND
      $b/publisher = "Addison-Wesley"
  RETURN My_Function($b)
 }
</results>
```

ในที่นี้ ภายในประโยค FOR ที่เป็นส่วนหลักจะไม่ทำงานด้วยตัวเองแต่จะไปเรียกฟังก์ชันให้ทำงานแทน, ภายในแต่ละฟังก์ชันก็อาจเรียกไปยังฟังก์ชันอื่นๆ ได้อีก ผ่านการทำงานต่อไปเรื่อยๆ คล้ายแนวคิดของการส่งการทำงานระหว่าง template ของ XSLT นั่นเอง

สรุป -- โครงสร้างพื้นฐานของ XQuery จะทำงานด้วยประโยค FOR ซึ่งมีคำสั่งอื่นๆ อยู่ภายใน นั่นคือ XQuery จะทำงานโดยมีส่วนหลักเป็น ศูนย์กลางเพียงส่วนเดียว แต่อาจกระจายการทำงานออกเป็นส่วนย่อยๆ ได้ด้วยฟังก์ชัน ส่วน XSLT นั้นจะกระจายการทำงานด้วย template โดยแต่ละ template จะมีความสำคัญเท่าเทียมกันไม่มีส่วนใดเป็นส่วนหลัก อย่างไรก็ตาม XSLT นั้นอาจใช้คำสั่ง xsl:for-each เพื่อเลียนแบบการทำงานแบบรวมศูนย์ของ XQuery ได้เช่นกัน จะเห็นได้ว่าภาษาสืบค้นทั้งสองแบบสามารถจำลองโครงสร้างการทำงานของกันและกันได้ อย่างไรก็ตามด้วยคุณสมบัติพื้นฐานบางประการทำให้ภาษาสืบค้นทั้งสองแบบไม่สามารถทดแทนการทำงานกันได้สมบูรณ์ทุกประการดังจะได้กล่าวถึงต่อไป

4.2 การใช้งานตัวแปร

ในการเรียกค้นข้อมูลด้วย XQuery นั้น ข้อมูลในส่วนต่างๆ จะถูกสร้างความสัมพันธ์และอ้างอิงกันด้วยตัวแปร, ในขั้นตอนการทำงานของ FOR ค่า node ของข้อมูลที่สัมพันธ์กับ path expression จะถูกกำหนดให้กับตัวแปรในแต่ละรอบของการทำงานและสามารถอ้างอิงได้ทุกๆ จุดภายในรอบการทำงานนั้น จากตัวอย่าง

```
<results>
 {
  FOR $a IN distinct(document("bib.xml")//author)
  RETURN
    <result>
     { $a }
     {
       FOR $b IN
       document("bib.xml")/bib/book[author = $a]
       RETURN $b/title
     }
    </result>
 }
</results>
```

จากตัวอย่างข้างต้น เป็นการจัดรายชื่อหนังสือโดยแบ่งตามชื่อผู้แต่ง การทำงานจะเริ่มจาก FOR ครั้งแรกเพื่อหาชื่อผู้แต่งแต่ละคนที่ไม่ซ้ำกันออกมาด้วยฟังก์ชัน distinct() และ FOR ภายในเพื่อหารายชื่อหนังสือทุกเล่มที่แต่งโดยผู้แต่งคนนี้ การ FOR ครั้งแรกในแต่ละรอบของการทำงาน ชื่อของผู้แต่งแต่ละคนจะถูกกำหนดให้กับตัวแปร $a ซึ่งสามารถถูกใช้งานได้อย่างอิสระภายใน เช่น การสร้างคำตอบ $a เพื่อแสดงชื่อของผู้แต่งเอง และจาก document("bib.xml")/bib/book [author = $a] เพื่อหาหนังสือที่มีชื่อผู้แต่งเหมือนกับค่าของตัวแปร $a, ภายใน FOR ครั้งที่สอง จะได้ ตัวแปรใหม่อีกตัวคือ $b เพื่ออ้างอิงถึงข้อมูลหนังสือที่มีเงื่อนไขที่ต้องการ

การทำงานโดยทั่วไปของ XSLT จะไม่ได้ใช้ตัวแปรแต่จะอาศัยแนวคิดของ context คือภายใน template ใดๆ การใช้งาน element หรือส่วนประกอบใดๆ จะถูกอ้างอิงกับ node ที่ match ตรงกับเงื่อนไขของ path expression ในรอบนั้นๆ เช่น จากตัวอย่าง

```
<xsl:template match="book">
  <book year="{@year}">
    <title>
      <xsl:value-of select="title"/>
    </title>
    <xsl:apply-templates select="author"/>
  </book>
</xsl:template>

<xsl:template match="author">
  <author>
    <last>
      <xsl:value-of select="last"/>
    </last>
    <first>
      <xsl:value-of select="first"/>
    </first>
  </author>
</xsl:template>
```

จากตัวอย่างข้างต้น, template แรกจะ match กับหนังสือทุกเล่ม โดยผลลัพธ์ที่ต้องการในขั้นตอนนี้คือปีที่พิมพ์และชื่อหนังสือ นอกจากนั้นยังต้องการชื่อผู้แต่งทุกคนซึ่งจะสร้างจาก template ที่สอง โดยในผู้แต่ง แต่ละคนประกอบไปด้วยนามสกุลและชื่อ จากตัวอย่างจะเห็นได้ว่าไม่มีการใช้งานตัวแปรใดๆ เลย, จาก template แรก การอ้างอิงถึงปีที่พิมพ์และชื่อหนังสือจะใช้เพียง @year และ title เท่านั้น เนื่องจากภายใน template นั้นกำลังอยู่ใน context ของ element book ซึ่งกำลังถูก match อยู่ ดังนั้น title จะหมายถึง book/title นั่นเอง และเช่นกันภายใน template ที่สอง last และ first จะหมายถึง author/last และ author/first ตามลำดับ โดย author ในที่นี้จะอ้างอิงมาจาก book ใน template แรก context จะเปลี่ยนไปเมื่อมีการทำงานข้าม template หรือเมื่อใช้คำสั่ง xsl:for-each, จะเห็นได้ว่าปัญหาจะเกิดขึ้นเมื่อมีการเปลี่ยน context ไปแล้วแต่ต้องการใช้งานข้อมูลที่ถูกอ้างอิงโดย context เดิม เช่น หากใน template ที่สองมีความจำเป็นจะต้องใช้ชื่อของหนังสือ ในที่นี้จะไม่สามารถใช้งาน title ได้แล้วเนื่องจากในที่นี้จะหมายความถึง author/title แทน, อย่างไรก็ตาม XSLT นั้นมี คุณสมบัติของตัวแปรเช่นกัน จากตัวอย่าง

```
<xsl:template match="book">
  <xsl:variable name="printed_year">
    <xsl:value-of select="@year"/>
  </xsl:variable>
  <xsl:variable name="book_title">
    <xsl:value-of select="title"/>
```

```
  </xsl:variable>
  <book year="{$printed_year}">
    <title>
      <xsl:value-of select="$book_title"/>
    </title>
  </book>
</xsl:template>
```

จากตัวอย่างข้างต้นเป็นการกำหนดค่าของ attribute year ให้กับตัวแปร printed_year และค่าของ title ให้กับตัวแปร book_title หลังจากนั้นจึงนำตัวแปรทั้งสองมาใช้ในการสร้างผลลัพธ์, ในที่นี้จะเป็นการกำหนด ตัวแปรเพื่อใช้ใน template เดียว ส่วนการส่งค่าของตัวแปรข้าม template ต้องใช้กลไกของการ call template ดังตัวอย่าง

```
<xsl:template match="book">
  <book>
    <xsl:call-template name="PRINT_TITLE">
      <xsl:with-param name="book_title">
        <xsl:value-of select="title"/>
      </xsl:with-param>
    </xsl:call-template>
  </book>
</xsl:template>

<xsl:template name="PRINT_TILTE">
  <xsl:param name="book_title">
  <title>
    <xsl:value-of select="$book_title"/>
  </title>
</xsl:template>
```

ในตัวอย่างนี้เป็นการส่งค่าของตัวแปรข้าม template, template แรกจะ match กับหนังสือทุกเล่มและส่งชื่อหนังสือในรูปของตัวแปรให้กับ template ที่สองเพื่อพิมพ์ค่าให้ ในที่นี้ template ที่สองจะแตกต่างจาก template ธรรมดา คือ template นี้จะต้องถูกระบุชื่อไว้ การเรียกใช้ก็จะใช้คำสั่ง xsl:call-template แทนไม่ใช่ xsl:apply-templates และผ่านค่าของตัวแปรด้วยคำสั่ง xsl:with-param, การใช้งาน template ในลักษณะนี้จะทำงานเหมือนกับฟังก์ชันของ XQuery นั่นเอง

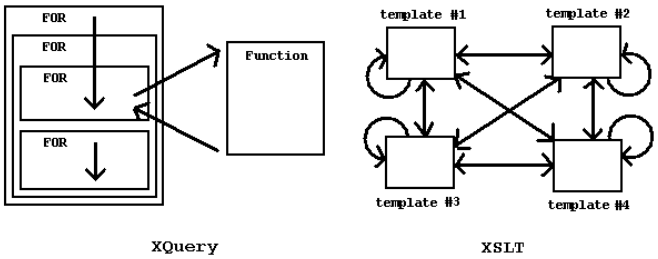
สรุป -- ตัวแปรนับเป็นคุณสมบัติหนึ่งซึ่งจำเป็นต่อการเรียกค้นข้อมูล ซึ่งจะทำให้เกิดการอ้างอิงระหว่างข้อมูลในส่วนต่างๆ ได้ แม้ว่าภาษาสืบค้นทั้งสองแบบจะสามารถใช้งานตัวแปรได้ แต่การใช้งาน XQuery นั้นอยู่บนพื้นฐานของตัวแปร การใช้งานจึงสะดวกและมีความยืดหยุ่นมาก ในขณะที่งานแปลงเอกสารของ XSLT นั้นสะดวกต่อการใช้งาน context มากกว่า ตัวแปรจึงเป็นเพียงคุณสมบัติเสริมซึ่งค่อนข้างยุ่งยากในการใช้งาน และมีข้อจำกัดมากกว่า

4.3 การเรียกค้นข้อมูลที่เป็นเอกสาร

ฐานข้อมูลที่อยู่ในรูปของ XML นั้นแม้จะมีความซับซ้อนได้มากกว่าฐานข้อมูลแบบสัมพันธ์ที่มีเพียง 2 มิติ โครงสร้างของข้อมูลจะสามารถซ้อนกันได้หลายชั้น แต่ละชั้นอาจประกอบด้วยโครงสร้างย่อยอีกมากมาย แต่อย่างไรก็ตามข้อมูลยังคงมีรูปแบบที่แน่นอนตายตัวและจำกัด เช่นจาก ตัวอย่างของข้อมูลหนังสือดังนี้

```
<bib>
  <book year="1994">
    <title>TCP/IP Illustrated</title>
    <author>
      <last>Stevens</last>
      <first>W.</first>
    </author>
    <publisher>Addison-Wesley</publisher>
    <price>65.95</price>
  </book>
  ...
</bib>
```

เนื่องจากโครงสร้างของข้อมูลแบบนี้ ค่อนข้างแน่นอนและจำกัด จึงเหมาะกับโครงสร้างการทำงานของ XQuery, อย่างไรก็ตาม ข้อมูล XML ที่อยู่ในรูปของเอกสาร ความสัมพันธ์ของแต่ละ element ภายในเอกสารจะมีความซับซ้อนมากกว่า, element ต่างๆ สามารถประกอบกันได้อย่างอิสระ ไม่มีรูปแบบที่ตายตัว และสามารถวนซ้ำได้ไม่จำกัดเท่าที่ต้องการ เช่นไฟล์เอกสาร HTML เป็นต้น ทำให้ความสัมพันธ์ระหว่างส่วนต่างๆ ในการเรียกค้นข้อมูลมีความซับซ้อนตามไปด้วย ดังรูปที่ 2



รูปที่ 2. แสดงความสัมพันธ์ในการทำงานของ XQuery และ XSLT

จากรูป การทำงานของ XQuery นั้นจะใช้ประโยค FOR เป็นหลักซึ่งภายในก็อาจจะประกอบไปด้วยประโยค FOR อื่นๆ ได้อีก ยิ่งข้อมูลที่ต้องการเรียกดูมีโครงสร้างที่ลึกเพียงใด ก็จำเป็นต้องมี FOR หลายชั้นตามไปด้วย ความสัมพันธ์ของประโยค FOR แต่ละอันจะอยู่ในทิศทางเดียวคือ จากข้างนอกลึกเข้าไปข้างในเรื่อยๆ แม้ว่า XQuery จะสามารถเรียกใช้ฟังก์ชันอื่นๆ ได้ แต่ก็ไม่ทำให้ทิศทางในการประมวลผลเปลี่ยนแปลงไป แต่ใน XSLT นั้น แต่ละ template จะสามารถเรียกใช้ template อื่นๆ ได้ตามอิสระ ความสัมพันธ์ของข้อมูลในแต่ละส่วนจะเป็นไปได้อย่างหลายหลาก ไม่มีรูปแบบตายตัวและข้อจำกัดใดๆ ซึ่งเหมาะกับการใช้งานกับข้อมูลในรูปแบบเอกสารที่มีโครงสร้างยืดหยุ่น

ตัวอย่างข้อมูลที่อยู่ในรูปของเอกสารที่ไม่มีรูปแบบตายตัวหรือข้อจำกัด เช่น ข้อมูลบทความที่มี DTD ดังนี้

```
<!ELEMENT section (title,
          (paragraph | figure | section)* )>
<!ELEMENT title #PCDATA>
<!ELEMENT paragraph #PCDATA>
```

```
<!ELEMENT figure EMPTY>
<!ATTLIST figure source CDATA>
```

จาก DTD ข้างต้น สามารถอธิบายได้ดังนี้, ภายใน section จะประกอบไปด้วย ชื่อ (title) ตามด้วย ย่อหน้า (paragraph), รูปภาพ (figure) และ section ย่อย ซึ่งสามารถมีจำนวนได้ไม่จำกัด และสามารถเรียงลำดับกันได้อย่างอิสระ ดังนั้นในที่นี้ทั้งจำนวนลำดับชั้นของ section รวมถึงการเรียงตัวของแต่ละ element จะไม่สามารถกำหนดได้ล่วงหน้า

ถ้าต้องการเรียกดูข้อมูล title ของทุกๆ section ที่มี จะสามารถเรียกค้น ข้อมูลด้วย XQuery ได้ดังนี้

```
<results>
 {
   FOR $t IN document("book.xml")//title
   RETURN $t
 }
</results>
```

ในที่นี้จะเป็นการเรียกดูข้อมูลโดยจะยุบโครงสร้างของ section ออกทั้งหมดให้เหลือแต่เฉพาะข้อมูล title ที่ต้องการ ทำให้โครงสร้างของผลลัพธ์ที่ต้องการสามารถถูกกำหนดได้ด้วยประโยค FOR ของ XQuery, อย่างไรก็ตาม หากเปลี่ยนการเรียกดูเป็นชื่อ title ของทุกๆ section โดยที่ยังคงโครงสร้างของแต่ละ section ไว้ด้วย, ในกรณีนี้ประโยค FOR จะไม่สามารถทำงานตามที่ต้องการได้ การเรียกดูข้อมูลในลักษณะนี้จะ เหมาะสมกับโครงสร้าง template ของ XSLT มากกว่าดังตัวอย่าง

```
<xsl:template match="section">
  <section>
    <title>
      <xsl:value-of select="title"/>
    </title>
    <xsl:apply-templates select="section"/>
  </section>
</xsl:template>
```

ในที่นี้ template ที่ match กับ element section จะสร้างคำตอบ title ที่ต้องการจากนั้นจึงส่งการทำงานไปให้ยัง template ถัดไป ซึ่งก็คือ template เดิม โดยจะวนทำงานไปเรื่อยๆ จนกว่าจะหมดโครงสร้างของ section ก็จะได้คำตอบที่ยังคงโครงสร้างของ section ไว้เหมือนในเอกสารต้นฉบับทุกประการ, จะเห็นได้ว่าประโยค FOR ของ XQuery นั้นไม่สามารถทำงานในลักษณะนี้ได้ อย่างไรก็ตามเนื่องจาก XQuery สามารถใช้งานฟังก์ชันได้ และการเรียกดูในตัวอย่างนี้ก็ไม่ซับซ้อนจนเกินไป ดังนั้นจึงสามารถเรียกดูข้อมูลนี้ด้วย XQuery ได้ดังตัวอย่าง

```
FUNCTION print_section(ELEMENT $s) RETURN ELEMENT {
  <section>
    { $s/title }
    {
      FOR $ss IN $s/section
      RETURN print_section($ss)
    }
  </section>
}

<result>
 {
   FOR $s IN document("book.xml")/section
   RETURN print_section($s)
 }
</result>
```

สรุป -- โครงสร้างข้อมูลของฐานข้อมูลจะมีรูปแบบเป็นลำดับชั้นที่แน่นอนและจำกัด เหมาะกับการประมวลผลโดยใช้ FOR ซ้อนกันหลายๆ ชั้นของ XQuery แต่ XSLT นั้นถูกออกแบบมาเพื่อจัดการกับข้อมูลแบบเอกสารซึ่งข้อมูลในแต่ละส่วนจะไม่มีรูปแบบที่ตายตัว การประมวลผลอาจต้องอาศัยการวนกลับไปกลับมา โดยขึ้นอยู่กับลักษณะของข้อมูลที่พบ โดยอาจไม่สามารถคาดการณ์ไว้ล่วงหน้าได้ อย่างไรก็ตามหากข้อมูลมีความซับซ้อนไม่มากเกินไป XQuery ก็สามารถทำได้ด้วยการใช้ความสามารถของฟังก์ชัน โดยเปรียบแต่ละฟังก์ชันเป็น template แต่การเรียกใช้งานก็จะมีความเฉพาะเจาะจงมากกว่า

4. 4 การสร้างส่วนประกอบของผลลัพธ์

ข้อมูลภายในเอกสาร XML จะกระจายไปตามโครงสร้างต่างๆ ของเอกสาร ซึ่งประกอบไปด้วยส่วนประกอบย่อยหลายๆ ส่วนได้แก่ element, attribute, namespace, comment และ processing instruction, การสืบค้นข้อมูลภายในเอกสาร XML ในบางครั้งนอกจากต้องการดึงข้อมูลที่สนใจออกมาแล้วยังต้องมีการจัดโครงสร้างของข้อมูลเหล่านั้นขึ้นใหม่ ดังนั้นภาษาสืบค้นสำหรับ XML จึงต้องสามารถสร้างส่วนประกอบเหล่านี้ได้อย่างอิสระ

4.4.1 การสร้าง element

XQuery จะสร้าง element ได้ 2 วิธี ได้แก่ การใช้ element constructor คือการสร้าง tag เปิดและ tag ปิดเอง และเนื้อหาภายใน element ก็จะประกอบไปด้วยประโยคคำสั่งอื่นๆ เช่น

```
<bib>
 {
   FOR $b IN //book
   RETURN
     <book>
       <title>{ $b/title/text() }</title>
     </book>
 }
</bib>
```

จากตัวอย่างข้างต้น เป็นการเรียกดูชื่อของหนังสือทุกๆ เล่ม ในที่นี้จะใช้ element constructor เพื่อสร้าง element ที่ชื่อ bib, book และ title, วิธีการใช้ element constructor ของ XQuery นี้จะทำให้ผู้ใช้สามารถสร้าง element ได้อย่างอิสระ แต่ข้อเสียของมันก็คือ ผู้ใช้จำเป็นต้องสร้างเนื้อหาภายในของ element นั้นๆ ขึ้นมาเองด้วย จึงทำให้เกิดการสร้าง element ด้วย XQuery อีกวิธีหนึ่งคือ การอ้างอิงมาจาก element ในเอกสาร ต้นฉบับ เช่น

```
<bib>
 {
   FOR $b IN //book
   RETURN
     <book>
      { $b/title }
      { $b/price }
     </book>
 }
</bib>
```

จากตัวอย่างข้างต้น เป็นการสร้าง element title และ price โดยดึงมาจากข้อมูลในเอกสารต้นฉบับโดยตรง วิธีนี้จะทำให้เนื้อหาต่างๆ ภายใน element ที่ถูกดึงมานี้เหมือนกับต้นฉบับทุกประการ เช่น attribute, namespace, sub element เป็นต้น วิธีการนี้จะทำให้ผู้ใช้สามารถสร้าง element ที่มีความซับซ้อนของโครงสร้างในผลลัพธ์อย่างง่ายดาย

การสร้าง element ของ XSLT นั้นจะทำได้คล้ายวิธีแรกของ XQuery เท่านั้นคือไม่สามารถอ้างอิงแล้วดึง element รวมทั้งโครงสร้างภายในของมันทั้งหมดมาได้โดยตรง ดังตัวอย่าง

ตัวอย่างที่ 1

```
<xsl:template match="book">
  <book>
    <title>
      <xsl:value-of select="title"/>
    </title>
  </book>
</xsl:template>
```

ตัวอย่างที่ 2

```
<xsl:template match="book">
  <xsl:element name="book">
    <xsl:element name="title">
      <xsl:value-of select="title"/>
    </xsl:element>
  </xsl:element>
</xsl:template>
```

ตัวอย่างแรกจะเป็นการสร้าง element book และ title ตามปกติซึ่งจะเป็นการระบุอย่างตายตัว ตัวอย่างที่สองจะเป็นการสร้าง element โดยใช้คำสั่งของ xsl คือ xsl:element ซึ่งจะทำให้การกำหนดชื่อของ element มีความยืดหยุ่นมากกว่า

ทั้ง XQuery และ XSLT ต่างก็มีความสามารถในการสร้าง element ที่มีความยืดหยุ่นสูง อย่างไรก็ตาม XQuery นั้นจะได้เปรียบในแง่ที่ใช้ไวยากรณ์ที่กระชับมากกว่า นอกจากนี้ยังสามารถดึง element รวมทั้งเนื้อหาทั้งหมดของมันมาได้โดยตรง ในขณะที่ XSLT จะต้องสร้างเนื้อหาทุกอย่างขึ้นมาเองทั้งหมด ซึ่งทำให้การใช้งานยุ่งยากซับซ้อนจนเกินไป

4.4.2 การสร้าง attribute

XQuery จะสามารถสร้าง attribute ได้ 2 วิธีเช่นกันดังตัวอย่าง
ตัวอย่างที่ 1

```
<bib>
 {
   FOR $b IN //book
   RETURN
     <book year={ $b/@year }>
       <title>{ $b/title/text() }</title>
     </book>
 }
</bib>
```

ตัวอย่างที่ 2

```
<bib>
 {
   FOR $b IN //book
   RETURN
     <book>
       { $b/@year }
       <title>{ $b/title/text() }</title>
     </book>
 }
</bib>
```

ทั้งสองตัวอย่างจะแสดงให้เห็นการสร้าง attribute year ภายใน element book, จากตัวอย่างแรกจะเป็นการสร้าง attribute เข้าไปในส่วน element constructor และวิธีที่สองคือสร้างในส่วนเนื้อหาของ element ด้วยการดึงค่า attribute จากเอกสารต้นฉบับมาโดยตรง

การสร้าง attribute ด้วย XSLT สามารถทำได้ 2 วิธี ดังตัวอย่าง

ตัวอย่างที่ 1

```
<xsl:template match="book">
  <book year="{@year}">
    <title>
      <xsl:value-of select="title"/>
    </title>
  </book>
</xsl:template>
```

ตัวอย่างที่ 2

```
<xsl:template match="book">
  <book>
    <xsl:attribute name="year">
      <xsl:value-of select="@year"/>
    </xsl:attribute>
    <title>
      <xsl:value-of select="title"/>
    </title>
  </book>
</xsl:template>
```

ตัวอย่างแรกจะเป็นการสร้าง attribute year ลงใน element book โดยตรงซึ่งเหมือนกับ XQuery และตัวอย่างที่สองคือ ใช้คำสั่ง xsl:attribute ซึ่งสามารถระบุทั้งชื่อของ attribute และเนื้อหาได้ตามที่ต้องการ

4.4.3 การสร้าง namespace

ทั้ง XQuery และ XSLT ต่างก็สามารถระบุ namespace เพื่อเป็นเงื่อนไขในการเรียกค้นข้อมูล และสร้าง namespace ของผลลัพธ์ได้ตามที่ต้องการ ตัวอย่างต่อไปนี้จะเป็นการค้นหาข้อมูลหนังสือที่อยู่ใน namespace ของ http://www.bn.com/book เพื่อเปลี่ยนให้เป็นข้อมูลที่อยู่ใน namespace ของ http://www.amazon.com/book แทน

การเรียกค้นข้อมูลโดยใช้ XQuery

```
NAMESPACE bn = "http://www.bn.com/book"
NAMESPACE amazon = "http://www.amazon.com/book"

FOR $b IN //bn:book
RETURN
  <amazon:book>
    { $b/@* }
    { $b/* }
  </amazon:book>
```

การเรียกค้นข้อมูลโดยใช้ XSLT

```
<xsl:stylesheet
  version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:bn="http://www.bn.com/book"
  xmlns:amazon="http://www.amazon.com/book">

  <xsl:template match="bn:book">
    <amazon:book>
      <xsl:apply-templates select="*"/>
    </amazon:book>
  </xsl:template>

  ...
</xsl:stylesheet>
```

จะเห็นได้ว่าทั้ง XQuery และ XSLT ก็ต่างทำงานร่วมกับ namespace ได้ดี แต่ก็ยังมีข้อจำกัดบางประการอยู่ทำให้ไม่สามารถสร้างและแก้ไข namespcae ได้อิสระตามใจผู้เรียกค้นข้อมูลได้อย่างสมบูรณ์นัก เช่น ค่าของ namespace ใดๆ ต้องถูกระบุไว้ล่วงหน้าโดยผู้เรียกค้น ยังไม่สามารถระบุค่าของ namespace ที่ถูกอ้างอิงมาจากที่อื่นได้ เป็นต้น

4.4.4 การสร้าง comment และ processing instruction

comment และ processing instruction เป็นส่วนประกอบที่ไม่ใช่ข้อมูลหลักที่กำลังสนใจ อย่างไรก็ตาม ทั้งสองส่วนเป็นสิ่งที่จะทำให้คนอื่นๆ หรือโปรแกรมอื่นๆ สามารถเข้าใจข้อมูลภายในเอกสารนั้นได้ดีมากยิ่งขึ้น จึงถือว่าเป็นส่วนประกอบสำคัญที่จะขาดไม่ได้ในการใช้งานเอกสาร XML, XQuery จะสร้าง comment และ procession instruction ด้วยฟังก์ชัน comment() และ pi() ตามลำดับ ส่วน XSLT จะใช้คำสั่งxsl:comment และ xsl:processing-instruction เช่นเดียวกับการสร้าง element และ attribute

ทั้ง XQuery และ XSLT สามารถสร้าง comment และ processing instruction ได้ อย่างไรก็ตามการใช้ฟังก์ชันของ XQuery นั้นยังมีข้อจำกัดอยู่บ้าง เช่น ไม่สามารถสร้างเนื้อหาของ comment และ processing instruction จากโครงสร้างที่มีความซับซ้อนได้เทียบเท่ากับ การใช้งาน element ในการสร้างเนื้อหาของ XSLT

สรุป -- ทั้ง XQuery และ XSLT ต่างก็มีความสามารถในการสร้าง element, attribute, namespace, comment, processing instruction ได้ ค่อนข้างยืดหยุ่น และมีความสามารถในการทำงานที่ใกล้เคียงกัน อย่างไรก็ตาม ฟังก์ชันการทำงานของ XQuery จะกระชับและมีความสามารถมากกว่า ทำให้ง่ายต่อการใช้งาน ต่างจากคำสั่งของ XSLT ที่จะทำงานในระดับพื้นฐานซึ่งทำให้ต้องใช้ความพยายามในการสืบค้นมากกว่า อย่างไรก็ตามสำหรับการสร้าง comment และ processing instruction ซึ่ง XQuery นั้นจะใช้ฟังก์ชันเพียงอย่างเดียว จึงทำให้ไม่สามารถสร้างเนื้อหา จากโครงสร้างของข้อมูลที่มีความซับซ้อนได้ดีเทียบเท่ากับ XSLT

4.5 การใช้งานร่วมกับเอกสาร XML

เนื่องจากความสามารถในการประยุกต์ใช้งานที่หลากหลายของ XML เกินกว่าที่มาตรฐานดั้งเดิมของ XML จะรองรับได้ทั้งหมด จึงเกิดมาตรฐานต่างๆ ขึ้นมาอีกมากมายเพื่อเสริมความสามารถของ XML เช่น Namespace, XSLT, XML Schema [12, 13] เป็นต้น มาตรฐานเหล่านี้นั้นนอกจากจะสามารถใช้งานกับ XML ได้อย่างสมบูรณ์แล้ว ยังใช้ XML เป็นเครื่องมือในการอธิบายด้วย เช่น XSLT และ XML Schema เป็นต้น ซึ่งการใช้งานสามารถเขียนอยู่ในรูปของเอกสาร XML นั่นเอง เหตุผลหนึ่งก็เพื่อความเป็นอันหนึ่งอันเดียวกันอย่างสมบูรณ์ อย่างไรก็ตาม รูปแบบของ XML นั้นค่อนข้างไม่สะดวกต่อการใช้งาน ซึ่ง XQuery นั้นต้องการเน้นคุณสมบัติในแง่ของความง่ายต่อการใช้งาน จึงทำให้ไวยากรณ์สำหรับ XQuery นั้นเป็นแบบที่สามารถเข้าใจได้ง่าย คือ ไม่ได้อยู่ในรูปแบบของเอกสาร XML นั่นเอง

จุดเด่นซึ่งคือความง่ายของ XQuery ซึ่งได้มาด้วยการไม่ใช้ XML ในการอธิบายนั้นต้องแลกกับข้อด้อยที่สำคัญประการหนึ่งก็คือ การใช้งาน XQuery ร่วมกับเอกสาร XML โดยตรงนั่นเอง เช่น ในกรณีที่ต้องการแทรกผลลัพธ์จากการสืบค้นข้อมูลด้วย XQuery ลงในเอกสาร XML ซึ่งสามารถแสดงให้เห็นได้จากตัวอย่าง

```
<?XML version="1.0"?>
<bib>
  <xql:query>
    <![CDATA[
      FOR $b IN document("bib.xml")//book
      RETURN
        <book year={ $b/@year }>
          { $b/title }
          { $b/price }
        </book>
    ]]>
  </xql:query>
</bib>
```

รายละเอียดของ XQuery ในปัจจุบันยังไม่ได้กล่าวถึงวิธีการในการรวม XQuery เข้ากับเอกสาร XML, ตัวอย่างข้างต้นเป็นหนึ่งในหลายๆ วิธีที่สามารถใช้งานได้ โดยเอาส่วนของการเรียกค้นข้อมูลทั้งหมดของ XQuery ไปไว้ภายใต้ element เฉพาะอันหนึ่งซึ่ง XQuery processor สามารถเข้าใจได้ (ในที่นี้กำหนดให้เป็น element ที่ชื่อ query ภายใต้ prefix ของ namespace เป็น xql) แต่ประโยคคำสั่งของ XQuery นั้นจะประกอบไปด้วยอักขระที่ไม่สามารถใช้งานทั่วไปได้ ส่วนเรียกค้นข้อมูลทั้งหมดจึงต้องอยู่ภายใต้ markup ของ <![CDATA[และ]]> อีกทีหนึ่ง, เมื่อต้องการอ่านข้อมูลจากเอกสารนี้ XQuery processor จะถูกเรียกขึ้นมาทำงานก่อนเพื่อเรียกค้นข้อมูลเหล่านั้นออกมาจริงๆ จากนั้นจึงค่อยส่งการทำงานไปให้โปรแกรมในส่วนอื่นๆ ต่อไป

XSLT นั้นจะไม่มีปัญหาในการใช้งานร่วมกับเอกสาร XML หรือมาตรฐานอื่นๆ เนื่องจากตัว XSLT ก็เป็นเอกสาร XML นั่นเอง

สรุป -- ไวยากรณ์ของ XQuery ไม่ได้อยู่ในรูปแบบ XML จึงทำให้มีข้อได้เปรียบที่สามารถทำความเข้าใจได้ง่าย เขียนได้สะดวก แต่ด้วยความที่ XQuery เองไม่ได้เป็นเอกสาร XML จึงทำให้การใช้งานร่วมกันระหว่าง XQuery และเอกสาร XML หรือมาตรฐาน XML อื่นๆ มีความไม่สะดวกอยู่บ้าง แตกต่างจากไวยากรณ์ของ XSLT ซึ่งเป็นเอกสาร XML และสามารถเข้ากันได้กับเอกสาร XML และมาตรฐานอื่นๆ ได้อย่าง สอดคล้อง

5. บทสรุป

XQuery เป็นผลมาจากการพัฒนาภาษาสืบค้นสำหรับ XML มาอย่างต่อเนื่อง จึงทำให้ XQuery นั้นนับได้ว่าเป็นภาษาสืบค้นสำหรับ XML ที่มีคุณสมบัติและความสามารถที่ค่อนข้างเพียบพร้อม อย่างไรก็ตาม XQuery นั้นถูกพัฒนามาตามแนวทางของฐานข้อมูลซึ่งมีภาษา SQL เป็นพื้นฐาน คุณสมบัติหลายๆ อย่างที่สืบทอดมาจะอ้างอิงกับรูปแบบการทำงานบนฐานข้อมูล ซึ่งบางครั้งไม่สามารถทำงานกับข้อมูลที่อยู่ใน รูปแบบเอกสารที่มีโครงสร้างยืดหยุ่นมากได้ดีเท่าที่ควร นอกจากนั้น XQuery ยังใช้ไวยากรณ์ของภาษาที่ใกล้เคียงกับ SQL เพื่อต้องการให้สามารถใช้งานและเข้าใจได้ง่าย ความกะทัดรัดของประโยคคำสั่งของ XQuery นี้บางครั้งทำให้ไม่สามารถทำงานบางอย่างที่ซับซ้อนเกี่ยวกับ ข้อมูลเอกสารได้เพียงพอ และจุดอ่อนประการสำคัญอีกข้อหนึ่งของ XQuery ก็คือการใช้รูปแบบที่ไม่ใช่เอกสาร XML นั่นเอง ทำให้ XQuery ไม่สามารถใช้งานร่วมกับเอกสาร XML และมาตรฐานอื่นๆ ของ XML ที่มีอยู่แล้วได้อย่างสมบูรณ์

XSLT เป็นมาตรฐานในการแปลงข้อมูลของ XML ซึ่งถูกออกแบบมาเพื่อจัดการกับความซับซ้อนของข้อมูลแบบเอกสารโดยเฉพาะ และยังประกอบด้วยคำสั่ง และการทำงานที่สามารถเป็นประโยชน์ในการสืบค้นข้อมูลได้ อย่างไรก็ตามเนื่องจาก XSLT นั้นไม่ได้ถูกออกแบบมาเพื่อการสืบค้นข้อมูลโดยตรงจึงยังขาดคุณสมบัติที่จำเป็นบางอย่างในการสืบค้นข้อมูลไป นอกจากนั้นรูปแบบการทำงานของ XSLT จะเข้าใจและใช้งานได้ยาก อย่างไรก็ตามจุดเด่นประการสำคัญของ XSLT ก็คือการอ้างอิงอยู่บนมาตรฐานของ XML นั่นเอง ทำให้ XSLT สามารถใช้งานร่วมกับเอกสาร XML และมาตรฐานอื่นๆ ที่เกี่ยวข้องได้อย่างสมบูรณ์

ทั้ง XQuery และ XSLT แม้จะถูกออกแบบมาเพื่อวัตถุประสงค์ในการใช้งานที่แตกต่างกัน แต่กลับมีแนวคิดในการทำงานและความสามารถในการสืบค้นข้อมูลที่คล้ายคลึงกัน อย่างไรก็ตามเนื่องจากการมองข้อมูลในมุมที่แตกต่างกัน จึงทำให้ภาษาทั้ง 2 แบบนี้มีจุดเด่นและจุดด้อยในการสืบค้นข้อมูลที่แตกต่างกัน ซึ่งผู้เขียนเชื่อว่าหากสามารถประยุกต์รวมจุดเด่นของภาษาทั้ง 2 แบบนี้เข้าด้วยกันและกำจัดข้อด้อยที่สำคัญออกไปได้ จะสามารถนำไปสู่การพัฒนาภาษาสืบค้นสำหรับ XML ที่สมบูรณ์และมีประสิทธิภาพมากยิ่งขึ้นได้

ตารางที่ 1. แสดงการเปรียบเทียบคุณสมบัติระหว่าง XQuery และ XSLT โดยอ้างอิงจากเอกสาร XML Query Requirements ของ W3C [11]

คุณสมบัติที่เปรียบเทียบ	XQuery	XSLT
สามารถเข้าใจและใช้งานได้ง่าย	X	-
มีรูปแบบการทำงานที่ใช้ XML	-	X
เป็นภาษาแบบ declarative	X	X
เป็นอิสระจากมาตรฐานอื่นๆ ที่เกี่ยวข้อง	X	X
มีการนิยามข้อผิดพลาด (exception) พื้นฐาน	-	-
รองรับการขยายความสามารถของภาษาในอนาคต	X	X
ใช้งานกับแบบจำลองข้อมูล (data model) ที่จำกัด	X	X
ทำงานร่วมกับ XML Schema ได้	X	-
รองรับการทำงานบนชุดของเอกสาร XML ได้	X	-
สามารถสืบค้นข้อมูลที่อ้างอิงกันระหว่างเอกสาร	X	-
การทำงานร่วมกับ XML Namespace	X	X
มี operation สำหรับข้อมูลพื้นฐานประเภทต่างๆ	-	-
มี Universal และ Existential Quantifiers	X	-
รองรับข้อมูลทั้งแบบโครงสร้างและเรียงลำดับ	X	X
มีฟังก์ชันในการสรุปผลข้อมูล (aggregation)	X	-
สามารถเรียงลำดับข้อมูลจากการสืบค้นได้	X	X
สามารถเปลี่ยนหรือสร้างโครงสร้างข้อมูลได้	X	X
ทำงานร่วมกับมาตรฐาน XML อื่นๆที่มีอยู่แล้วได้	X	X

เอกสารอ้างอิง

[1] A. Deutsch, M. Fernadez, D. Florescu, A. Levy, and D. Suciu, A Query Language for XML. Available http://www.research.att.com/~mff/files/final.html

[2] A. Sahuguet, XSLT as a query language for XML, April 2000. Available http://db.cis.upenn.edu/XSLT/

[3] E. Lenz, XQuery: Reinventing the Wheel?, February 2001. Available http://www.xmlportfolio.com/xquery.html

[4] International Organization for Standardization (ISO), Information Technology-Database Language SQL. Standard No. ISO/IEC 9075:1999.

[5] J. Robie, XQL (XML Query Language), August 1999. Available http://metalab.unc.edu/xql/xql-proposal.html

[6] R. Cattell, The Object Database Standard: ODMG-93, Release 1.2. Morgan Kaufmann Publisher, San Francisco, 1996.

[7] World Wide Web Consortium, Extensible Markup Language (XML) Version 1.0 (Second Edittion). W3C Recommendation, October 6, 2000. Available http://www.w3.org/TR/2000/REC-xml-20001006

[8] World Wide Web Consortium, HTML 4.01 Specification. W3C Recommendation, December 24, 1999. Available http://www.w3.org/TR/html4/

[9] World Wide Web Consortium, Namespaces in XML. W3C Recommendation, January 14, 1999. Available http://www.w3.org/TR/REC-xml-names

[10] World Wide Web Consortium, XML Path Language (XPath) Version 1.0. W3C Recommendation, November 16, 1999. Available http://www.w3.org/TR/xpath.html

[11] World Wide Web Consortium, XML Query Requirements. W3C Working Draft, February 15, 2001. Available http://www.w3.org/TR/xmlquery-req/

[12] World Wide Web Consortium, XML Schema Part 1: Structures. W3C Recommendation, May 2, 2001. Available http://www.w3.org/TR/xmlschema-1/

[13] World Wide Web Consortium, XML Schema Part 2: Datatypes. W3C Recommendation, May 2, 2001. Available http://www.w3.org/TR/xmlschema-2/

[14] World Wide Web Consortium, XQuery: A Query Language for XML. W3C Working Draft, June 7, 2001. Available http://www.w3.org/TR/xquery/

[15] World Wide Web Consortium, XSL Transformations (XSLT) Version 1.0. W3C Recommendation, November 16, 1999. Available http://www.w3.org/TR/xslt.html

เกี่ยวกับผู้เขียน

Somnuk Keretho, Ph.D.

Assistant Professor

Dr. Somnuk Keretho received his B.Eng. and M.Eng. in Electrical Engineering from Kasetsart University in 1981 and 1986, respectively. With the support from Carl Duisberg Gesellschaft Scholarship, he completed his M.Eng. in Computer Applications from Asian Institute of Technology in 1985. As a Fulbright scholar, he got his Ph.D. degree in Computer Science from University of Southwestern Louisiana, U.S.A. in 1992. His current research interest is related to Software Process Improvement with Capability Maturity Model, Object-Oriented Methodology and Unified Modeling Language, Software Components and Multi-tier Web-based Information Systems Development with Java, and eXtensible Markup Language (XML). At present, he is an assistant professor in Computer Engineering and serves as the chairman of Master of Science Program in Information Technology, Department of Computer Engineering, Kasetsart University.

Ekapol Jeerangsuwan

Mr. Ekapol Jeerangsuwan received his B.Eng. in Computer Engineering from Kasetsart University in 1997. He's studying M.Eng in Computer Engineering at Kasetsart University. His current research is about query language of eXtensible Markup Language (XML).

ภาษาสืบค้นสำหรับ XML, Another XML Query Language (AXQL)

เอกพล จีรังสุวรรณ และ สมนึก คีรีโต
ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

ABSTRACT -- The standard of XML improves the way of collecting and using data, both in database and document formats. However, it still lacks mechanism to manage and query data with efficiency. This article proposes another query language for XML that the authors call, Another XML Query Language or AXQL. AXQL is based on previous prominent query languages, XQuery and XSLT. Both languages can work with XML very well. XQuery is easy to understand but it can't deal with complex querying well. XSLT concentrates on the way of processing data that can work with complex querying very well but also makes it hard to use too. AXQL is an attempt to combine their advantages together and get rid of major disadvantages. Hopefully, AXQL can be used to query XML data in both database and complex document forms with efficiency, simplicity, compatibility with other standards of XML and can be easily extended in the future.

KEY WORDS -- XML, Query Language, AXQL, XQuery, XSLT

บทคัดย่อ -- มาตรฐาน XML ทำให้การจัดเก็บและใช้งานข้อมูลที่อยู่ในรูปทั้งฐานข้อมูลและเอกสารมีประสิทธิภาพมากขึ้น อย่างไรก็ตาม XML ยังขาดกลไกในการจัดการและเรียกค้นข้อมูลที่มีประสิทธิภาพ, บทความนี้เป็นการนำเสนอภาษาสืบค้นสำหรับ XML อีกรูปแบบหนึ่งที่ผู้เขียนให้ชื่อว่า Another XML Query Language หรือ AXQL, AXQL ถูกพัฒนาขึ้นโดยมีภาษาต้นแบบ คือ XQuery และ XSLT, ภาษาทั้งสองนี้มีประสิทธิภาพในการจัดการข้อมูล XML มาก อย่างไรก็ตาม XQuery จะเน้นความง่ายในการใช้งานจึงทำให้ไม่สามารถเรียกค้นข้อมูลที่มีความซับซ้อนมากๆ ได้ ในขณะที่ XSLT ซึ่งสามารถทำงานที่มีความซับซ้อนได้ดี แต่จะมีรูปแบบที่ยากต่อการใช้งาน, AXQL เป็นความพยายามในการรวมข้อดีของทั้งสองภาษานี้เข้าด้วยกันและกำจัดข้อเสียที่สำคัญบางประการออกไป โดยหวังว่า AXQL จะสามารถทำงานร่วมกับฐานข้อมูลและเอกสารที่มีความซับซ้อนได้อย่างมีประสิทธิภาพ, ใช้งานได้ง่าย, ทำงานร่วมกับมาตรฐาน XML อื่นๆ ได้อย่างราบรื่น และสามารถขยายความสามารถเพิ่มเติมได้ในอนาคต

คำสำคัญ -- ภาษาสืบค้น, XML, AXQL, XQuery, XSLT

1. บทนำ

มาตรฐาน Extensible Markup Language (XML) [3] ทำให้เกิดรูปแบบใหม่ในการจัดเก็บ และจัดการข้อมูลทั้งที่เป็นฐานข้อมูลและเอกสารที่มีประสิทธิภาพมากยิ่งขึ้น อย่างไรก็ตามมาตรฐาน XML นั้นยังขาดกลไกในการสืบค้นข้อมูล, ภาษาสืบค้นสำหรับ XML รูปแบบต่างๆ ถูกนำเสนอออกมา ซึ่งแต่ละรูปแบบก็ล้วนมีข้อดีและข้อเสียที่แตกต่างกันไป และยังไม่มีภาษาสืบค้นรูปแบบใดได้รับการยอมรับว่าเป็นมาตรฐาน ดังนั้นการพัฒนาภาษาสืบค้นสำหรับ XML จึงยังคงเป็นไปอย่างต่อเนื่องเพื่อค้นหาภาษาสืบค้นสำหรับ XML ที่มีประสิทธิภาพและเหมาะกับการใช้งานมากที่สุด

XQuery [9] เป็นภาษาสืบค้นสำหรับ XML แบบล่าสุดที่เสนอโดยหน่วยงาน W3C ซึ่งมีหน้าที่ดูแลเกี่ยวกับมาตรฐานต่างๆ ของอินเตอร์เน็ต XQuery ถูกพัฒนามาในแนวทางของภาษาสืบค้นสำหรับฐานข้อมูลซึ่งมีภาษา Structured Query Language (SQL) [2] เป็นพื้นฐาน จึงทำให้รูปแบบไวยากรณ์และการใช้งานมีความใกล้เคียงกับ SQL ซึ่งทำให้สามารถเข้าใจและใช้งานได้ง่าย อย่างไรก็ตาม จุดด้อยที่สำคัญประการหนึ่งของ XQuery ก็คือการไม่ใช้รูปแบบ XML ในการอธิบาย ทำให้ไม่สามารถทำงานร่วมกับเอกสาร XML ได้ดีเท่าที่ควร นอกจากนี้ไวยากรณ์

ของ XQuery ซึ่งเน้นที่ความง่ายนั้น บางครั้งมีรูปแบบที่ไม่เหมาะสมในการใช้งานร่วมกับข้อมูลที่อยู่ในรูปของเอกสารที่มีความซับซ้อนสูง [1] Extensible Stylesheet Language (XSLT) [10] เป็นมาตรฐานที่ถูกพัฒนาขึ้นเพื่อใช้ในการแปลงเอกสาร XML เพื่อการแสดงผล อย่างไรก็ตาม ความสามารถในการแปลงข้อมูลของ XSLT นั้นสามารถนำมาใช้ในการสืบค้นข้อมูลได้เป็นอย่างดี, XSLT มีจุดเด่นประการสำคัญคือการใช้งานไวยากรณ์ที่เป็น XML ทำให้สามารถใช้งานร่วมกับเอกสาร XML ได้อย่างสอดคล้อง นอกจากนี้ยังรองรับในการสืบค้นข้อมูลที่มีความซับซ้อนได้เป็นอย่างดี อย่างไรก็ตามเนื่องจาก XSLT ไม่ได้ถูกพัฒนาขึ้นตั้งแต่ต้นเพื่อจุดประสงค์ในการสืบค้นข้อมูล จึงทำให้รูปแบบการใช้งานมีความซับซ้อน, ยากต่อการทำความเข้าใจและใช้งาน และยังขาดคุณสมบัติที่จำเป็นในการสืบค้นข้อมูลอีกหลายๆ ประการ [1]

จะเห็นได้ว่าทั้ง XQuery และ XSLT ต่างก็มีจุดเด่นและจุดด้อยที่สำคัญแตกต่างกันไป ดังนั้นบทความนี้จึงขอเสนอภาษาสืบค้นสำหรับ XML อีกรูปแบบหนึ่งที่เรียกว่า Another XML Query Language (AXQL) ซึ่งได้รวบรวมคุณสมบัติเด่น ของภาษาทั้งสองแบบข้างต้นเข้าด้วยกัน กำจัดจุดด้อยที่เป็นปัญหาในภาษาทั้งสองแบบออกไป, AXQL ถูกออกแบบขึ้นมาโดยมีวัตถุประสงค์หลักคือ จะต้องสามารถเรียกค้นข้อมูลทั้งที่อยู่ในรูปของฐานข้อมูล และเอกสารที่มีความซับซ้อนได้อย่างมีประสิทธิภาพ โดยยังคงสามารถทำความเข้าใจและใช้งานได้ง่าย, สามารถใช้งานร่วมกับเอกสาร XML และมาตรฐานสำหรับ XML แบบอื่นๆ ได้อย่างสอดคล้อง, มีกลไกในการแก้ไขและจัดการข้อมูล นอกจากนี้ยังต้องมีความยืดหยุ่นที่จะขยายความสามารถของภาษาเพิ่มเติมได้ในอนาคต

ในบทความนี้จะแสดงตัวอย่างในการสืบค้นข้อมูลด้วย AXQL โดยจะเรียกค้นข้อมูลจากเอกสาร XML ตัวอย่างคือ bib.xml ซึ่งมี Document Type Definition (DTD) ดังนี้

```
<!ELEMENT bib (book*)>
<!ELEMENT book (title, author+, publisher, price)>
<!ATTLIST book year CDATA>
<!ELEMENT author (last, first)>
<!ELEMENT title #PCDATA>
<!ELEMENT last  #PCDATA>
<!ELEMENT first #PCDATA>
<!ELEMENT publisher #PCDATA>
<!ELEMENT price #PCDATA>
```

เอกสาร bib.xml มี root element เป็น bib (บรรณานุกรม) โดยใน bib ประกอบไปด้วยหลายๆ book ซึ่งหมายถึงหนังสือแต่ละเล่ม ในหนังสือแต่ละเล่มจะประกอบไปด้วย attribute year ซึ่งแสดงปีที่พิมพ์, ชื่อหนังสือ (title), รายชื่อผู้แต่ง (author), ชื่อสำนักพิมพ์ (publisher) และราคาหนังสือ (price), ภายในชื่อผู้แต่งจะประกอบไปด้วย นามสกุล (last) และ ชื่อ (first)

บทความนี้จะแบ่งเนื้อหาของ AXQL เป็นหัวข้อย่อยๆ ดังนี้

หัวข้อที่ 2 แนะนำ path expression
หัวข้อที่ 3 คำสั่งและการทำงานพื้นฐานของ AXQL
หัวข้อที่ 4 การใช้งานด้วยรูปแบบย่อของ AXQL
หัวข้อที่ 5 การสร้างผลลัพธ์จากการสืบค้น
หัวข้อที่ 6 คำสั่งในการตรวจสอบเงื่อนไข
หัวข้อที่ 7 การนิยามและเรียกใช้งานฟังก์ชัน
หัวข้อที่ 8 การเรียงลำดับข้อมูลจากการสืบค้น
หัวข้อที่ 9 การทดสอบชนิดของข้อมูล
หัวข้อที่ 10 การเพิ่มเติม, ลบ และแก้ไขข้อมูล
หัวข้อที่ 11 การพัฒนาระบบการสืบค้นด้วยภาษา AXQL
หัวข้อที่ 12 บทสรุป และตารางเปรียบเทียบคุณสมบัติ

2. แนะนำ Path Expression

path expression เป็นกลไกที่ใช้ในการอ้างอิงไปยังข้อมูลต่างๆ ที่อยู่ภายในเอกสาร XML ได้อย่างกะทัดรัดและมีประสิทธิภาพ, ในปัจจุบันมาตรฐานสำหรับ path expression ที่ประกาศอย่างเป็นทางการนั่นก็คือมาตรฐาน XPath [5] , XPath ถูกใช้อย่างแพร่หลายรวมทั้งใน XQuery และ XSLT, AXQL เองก็อาศัยกลไกของ XPath เช่นเดียวกัน

เอกสาร XML จะถูกมองเป็นแผนภูมิต้นไม้ (tree) ที่ประกอบขึ้นมาจาก node ประเภทต่างๆ 7 ประเภท ได้แก่

1. root node คือ node ที่เป็นจุดอ้างอิงของแต่ละเอกสาร
2. element node คือ node ที่แทน tag หรือ element
3. attribute node คือ node ที่แทน attribute ของ element
4. namespace node คือ node ที่ใช้ในการนิยาม namespace [4]
5. data node คือ node ที่แทนข้อมูลที่อยู่ภายใน element
6. comment node คือ node ที่แสดงความเห็นของผู้เขียน
7. processing instruction node คือ node ที่เป็นคำสั่งพิเศษ

การทำงานของ path expression จะแบ่งเป็น step ซึ่งก็คือ node ในแต่ละระดับ, แต่ละ step จะถูกอ้างอิงมาจาก step ก่อนหน้า โดยจะคั่นระหว่าง step ด้วยเครื่องหมาย "/" ซึ่งหมายถึง child หรือเป็น node ลูกของ node นั้นๆ หรือเครื่องหมาย "//" ซึ่งหมายถึง node ใดๆ ที่อยู่ภายใต้ node นั้นๆ โดยไม่สนใจลำดับขั้น, ตำแหน่งของ node หรือ set ของ node ที่ใช้ในการอ้างอิงขณะใดๆ จะถูกเรียกว่า context ซึ่งหมายถึงสภาวะแวดล้อมในขณะนั้นๆ, path expression ยังสามารถมีเงื่อนไขในการเรียกค้นข้อมูล (predicate) ซึ่งจะทำให้ได้ข้อมูลที่มีความเฉพาะเจาะจงมากยิ่งขึ้นด้วย, ตัวอย่างการใช้งาน path expression เช่น

document('bib.xml') หมายถึง root node ของเอกสาร bib.xml

/ หมายถึง root node ของเอกสารปัจจุบัน

/bib หมายถึง element ที่ชื่อ bib ที่เป็น root element

. หมายถึง context หรือ node ปัจจุบัน

bib หมายถึง element ที่ชื่อ bib ที่อ้างอิงกับ context ขณะนั้น

bib/book หมายถึง element book ที่อยู่ภายใต้ element bib

//book หมายถึง element book ใดๆ ที่อยู่ภายในเอกสาร

book/@year หมายถึง attribute year ของ element book

title/data() หมายถึง data node ของ element title

comment() หมายถึง comment node

book[publisher = 'Addison-Wesley' and @year > 1991] หมายถึง หนังสือทุกๆ เล่มที่พิมพ์โดยสำนักพิมพ์ Addison-Wesley หลังจากปี 1991

3. การสืบค้นข้อมูลด้วย AXQL

ในหัวข้อนี้จะเป็นการแนะนำคำสั่ง และรูปแบบการใช้งานพื้นฐานที่สำคัญบางอย่างของ AXQL, AXQL ถูกออกแบบให้มีรูปแบบในการนำเสนอที่เป็น XML คำสั่งต่างๆ ของ AXQL จะอยู่ในรูปของ element, AXQL สามารถใช้งานร่วมกับข้อมูล XML ตามปกติได้ ดังนั้นเพื่อเป็นการแบ่งแยกระหว่าง element ที่เป็นคำสั่งของ AXQL และ element ที่เป็นข้อมูล XML ชื่อของ element ที่เป็นคำสั่งของ AXQL จะมี prefix หรือ namespace เป็น "axql" ซึ่งจะอ้างอิงไปยัง URI คือ "http://www.cpe.ku.ac.th/AXQL"

3.1. คำสั่ง axql:query

เนื่องจาก AXQL นั้นเป็นเอกสาร XML จึงต้องมี root element ดังเช่นเอกสาร XML ปกติ, root element ของ AXQL นั้นถูกกำหนดให้เป็น axql:query ดังตัวอย่าง

```
<axql:query
  xmlns:axql="http://www.cpe.ku.ac.th/AXQL">
  <!-- AXQL top level element -->
</axql:query>
```

ภายใน tag เปิดของ axql:query จะมีการนิยาม namespace ของ AXQL ด้วยเสมอเพื่อบอกให้รู้ว่าเอกสารนี้เป็นการเรียกค้นข้อมูลโดยใช้ AXQL, ภายใน axql:query จะต้องประกอบไปด้วยคำสั่งของ AXQL ที่เรียกว่า top level element เท่านั้น ซึ่งได้แก่คำสั่ง axql:include, axql:function และ axql:main ซึ่งจะได้กล่าวถึงต่อไป

3.2. คำสั่ง axql:main

ในการเรียกค้นข้อมูลตามปกติ จะประกอบไปด้วย axql:query และ axql:main ซึ่งเป็น top level element ที่เป็นส่วนในการเรียกค้นหลัก ดังตัวอย่าง

```
<axql:query
  xmlns:axql="http://www.cpe.ku.ac.th/AXQL">
  <axql:main>
    <!-- AXQL instructions -->
  </axql:main>
</axql:query>
```

3.3. คำสั่ง axql:for

ภายใน axql:main จะประกอบไปด้วยคำสั่งต่างๆ ของ AXQL, คำสั่งที่ใช้เป็นพื้นฐานในการเรียกค้นข้อมูลของ AXQL คือ axql:for ซึ่งจะมี attribute select ที่มีค่าเป็น path expression ที่จะอ้างอิงไปยังเซตของ node ของข้อมูลที่กำลังสนใจ คำสั่ง axql:for จะวนรอบทำงานคำสั่งที่อยู่ภายในจนกว่าจะครบทุก node ในเซตของ node ที่อ้างอิงถึง ดังตัวอย่าง

```
<axql:query
  xmlns:axql="http://www.cpe.ku.ac.th/AXQL">
  <axql:main>
    <axql:for select="document('bib.xml')//book">
      <book>
        <title><axql:value select="title"/></title>
      </book>
    </axql:for>
  </axql:main>
</axql:query>
```

จากตัวอย่างข้างต้น ภายในคำสั่ง axql:for, path expression จะอ้างอิงไปยังเซตของ element book ทุกๆ อันที่อยู่ภายในเอกสาร bib.xml, ในแต่ละรอบการทำงานของ axql:for จะสร้างผลลัพธ์ของการเรียกค้นคือ element book โดยภายในประกอบไปด้วย element title ที่มีค่าเป็นชื่อของหนังสือแต่ละเล่ม ค่าของชื่อหนังสือนี้จะถูกสร้างขึ้นมาด้วยคำสั่งของ AXQL คือ axql:value ซึ่งจะสร้าง data node ที่มีค่าเป็นข้อมูลที่ถูกอ้างอิงโดย path expression ที่อยู่ภายใน attribute select , คำสั่ง axql:for จะมีผลทำให้ context เปลี่ยนไป จะเห็นได้จาก path expression ภายในคำสั่ง axql:value ระบุเพียง title ซึ่งในที่นี้จะหมายถึง element title ที่อ้างอิงกับ context คือ element book ที่กำลังทำงานอยู่ในรอบนั้นๆ

3.4. คำสั่ง axql:where

การใช้ path expression สามารถระบุเงื่อนไขเพิ่มเติมใน node ที่ต้องการเป็นพิเศษได้ ด้วยการใช้คุณสมบัติ predicate ที่มีอยู่แล้วภายใน path expression เอง หรืออาจใช้คำสั่ง axql:where เพื่อช่วยในการตรวจสอบเงื่อนไขของแต่ละ node, ตัวอย่างต่อไปนี้เป็นการเปรียบเทียบการตรวจ

สอบเงื่อนไขโดยใช้ predicate และคำสั่ง axql:where ซึ่งจะให้ผลลัพธ์ที่เหมือนกันทุกประการ

การตรวจสอบเงื่อนไขโดยใช้ predicate

```
<axql:for select="//book[@year > 1991]">
  <book>
    <title><axql:value select="title"/></title>
  </book>
</axql:for>
```

การตรวจสอบเงื่อนไขโดยใช้คำสั่ง axql:where

```
<axql:for select="//book">
  <axql:where test="@year > 1991">
    <book>
      <title><axql:value select="title"/></title>
    </book>
  </axql:where>
</axql:for>
```

จากตัวอย่างเป็นการเพิ่มเงื่อนไขภายใน element book ที่อ้างอิงมาว่าจะต้องพิมพ์หลังจากปี 1991, คำสั่ง axql:where โดยปกติจะใช้ตรวจสอบเงื่อนไขของ node หลังจากคำสั่ง axql:for โดยหาก node ที่ได้เป็นไปตามเงื่อนไขที่อยู่ใน attribute test ก็จะทำคำสั่งอื่นๆ ต่อไป

3.5. คำสั่ง axql:var

จะเห็นได้ว่า AXQL นั้นใช้แนวคิดของ context จึงทำให้การอ้างอิงไปยัง element ใดๆ ที่เกี่ยวข้องกับ context มีความสะดวกมาก อย่างไรก็ตามมีบางกรณีที่ context ไม่อาจใช้งานได้ เช่น มีการใช้คำสั่งของ AXQL บางคำสั่งที่ทำให้ context เปลี่ยนไป เช่น คำสั่ง axql:for แต่คำสั่งภายในกลับต้องการอ้างอิงถึงข้อมูลที่อยู่ภายใต้ context เดิม ในกรณีเช่นนี้จำเป็นต้องใช้ตัวแปรมาช่วยเก็บค่าของข้อมูลที่ต้องการไว้ก่อน หลังจากนั้นจึงค่อยนำมาใช้ในภายหลัง ดังตัวอย่าง

```
<axql:for select="distinct(//author)">
  <result>
    <author><axql:value select="."/></author>
    <axql:var name="my_var" select="."/>
    <axql:for select="//book[author = $my_var]">
      <title><axql:value select="title"/></title>
    </axql:for>
  </result>
</axql:for>
```

จากตัวอย่างข้างต้น เป็นการเรียกดูข้อมูลผู้แต่งแต่ละคน โดยภายในประกอบด้วยข้อมูลผู้แต่ง และชื่อหนังสือที่แต่งโดยผู้แต่งคนนั้นๆ, ใน path expression ที่ระบุไปยังชื่อผู้แต่ง จะใช้ฟังก์ชัน distinct() เพื่อจัดกลุ่มตามชื่อผู้แต่งโดยกำจัด node ที่มีค่าซ้ำซ้อนกันออกไป โดยภายในจะสร้างผลลัพธ์อยู่ภายใต้ element result ซึ่งประกอบไปด้วย element author นอกจากนั้นคำตอบยังต้องการชื่อหนังสือที่แต่งโดยผู้แต่งคนนั้นๆ อีก ในที่นี้จึงใช้คำสั่ง axql:for อีกครั้งเพื่อวนหาชื่อหนังสือที่ต้องการ แต่ชื่อของผู้แต่งใน element author นั้นจะถูกอ้างอิงไม่ได้เมื่อ context ถูกเปลี่ยนไป จึงต้องมีการเก็บค่า element author นี้ไว้ในตัวแปร $my_var ก่อนด้วยคำสั่ง axql:var

4. รูปแบบย่อของ AXQL

เนื่องจากรูปแบบการใช้งานของ AXQL จะอยู่ในรูปของเอกสาร XML, คำสั่งต่างๆ จะอยู่ในรูปของ element ซึ่งบางครั้งยากต่อการใช้งานและทำความเข้าใจ อย่างไรก็ตามรูปแบบคำสั่งบางประเภทจะถูกใช้งานบ่อยและในสภาวะการทำงานที่เจาะจงบางรูปแบบสามารถจะเขียนย่อได้ ซึ่งจะทำให้สามารถใช้งานได้สะดวกมากยิ่งขึ้น, รูปแบบคำสั่งของ AXQL ที่สามารถย่อได้ มีรูปแบบการใช้งานดังนี้

4.1. รูปแบบย่อของ axql:query และ axql:main

หากภายใน axql:query ประกอบไปด้วย top level element คือ axql:main เพียงอย่างเดียว และภายใน axql:main มีการสร้าง root element ไว้ การทำงานจะสามารถย่อได้ ดังตัวอย่างแสดงรูปแบบเต็มและรูปแบบย่อ ดังนี้

รูปแบบเต็ม

```
<axql:query
  xmlns:axql:="http://www.cpe.ku.ac.th/AXQL">
  <axql:main>
    <results>
      <!-- AXQL insturctions -->
    </results>
  </axql:main>
</axql:query>
```

รูปแบบย่อของ axql:query และ axql:main

```
<results
  xmlns:axql:="http://www.cpe.ku.ac.th/AXQL">
  <!-- AXQL insturctions -->
</results>
```

รูปแบบการใช้งานข้างต้นจะเป็นการละคำสั่ง axql:query และ axql:main ไว้ซึ่งจะทำให้การเขียนมีความกะทัดรัดมากขึ้น อย่างไรก็ตามจำเป็นต้องมีการนิยาม namespace axql ไว้เสมอเพื่อบอกให้รู้ว่าเอกสารนี้เป็นการเรียกค้นข้อมูลด้วย AXQL ไม่ใช่ข้อมูล XML ตามปกติ

4.2. รูปแบบย่อของ axql:for

หากภายใน axql:for มีการสร้าง element หลักเพียง element เดียว การทำงานจะสามารถย่อได้ ดังตัวอย่าง

รูปแบบเต็ม

```
<axql:for select="//book">
  <book>
    <!-- AXQL insturctions -->
  </book>
</axql:for>
```

รูปแบบย่อของ axql:for

```
<book axql:for="//book">
  <!-- AXQL insturctions -->
</book>
```

จากรูปแบบการใช้งานย่อของคำสั่ง axql:for, จะเปลี่ยนสถานะจาก element ไปเป็น attribute axql:for แทน แต่การทำงานยังคงเหมือนเดิม คือ เมื่อพบ node ตรงตามเงื่อนไขที่อ้างอิงโดย attribute axql:for ก็จะวนรอบการทำงานโดยสร้าง element book และเรียกใช้คำสั่งภายในอื่นๆ เพื่อสร้างเนื้อหาของ element book

4.3. รูปแบบย่อของ axql:where

หากภายใน axql:where มีการสร้าง element หลักเพียง element เดียว การทำงานจะสามารถย่อได้ ดังตัวอย่าง

รูปแบบเต็ม

```
<axql:for select="//book">
  <axql:where test="@year > 1991">
    <book>
      <!-- AXQL insturctions -->
    </book>
  </axql:where>
</axql:for>
```

รูปแบบย่อของ axql:where

```
<axql:for select="//book">
  <book axql:where="@year > 1991">
    <!-- AXQL insturctions -->
  </book>
</axql:for>
```

รูปแบบย่อของ axql:for และ axql:where

```
<book axql:for="//book" axql:where="@year > 1991">
  <!-- AXQL insturctions -->
</book>
```

จากตัวอย่างข้างต้น เป็นการใช้รูปแบบการทำงานย่อของ axql:where นอกจากนี้ ทั้งคำสั่ง axql:for และ axql:where นั้นสามารถย่อนำมารวมกันได้ ทำให้มีความกะทัดรัดมาก สามารถเข้าใจได้ง่าย โดยที่ยังคงสามารถทำงานได้เหมือนเดิมทุกประการ

4.4. รูปแบบย่อของ axql:value

คำสั่ง axql:value นั้นใช้เพื่อสร้าง data node เพื่อใช้เก็บข้อมูลภายใน element ซึ่งเป็นรูปแบบการทำงานที่พบค่อนข้างบ่อย, หากภายใน element ที่มีการสร้าง data node นั้นมีส่วนประกอบคือคำสั่ง axql:value เพียงหนึ่งเดียว จะสามารถเขียนในรูปแบบย่อได้ดังนี้

รูปแบบเต็ม

```
<book axql:for="//book">
  <title><axql:value select="title"/></title>
  <price><axql:value select="price"/></price>
</book>
```

รูปแบบย่อของ axql:value

```
<book axql:for="//book">
  <title axql:value="title"/>
  <price axql:value="price"/>
</book>
```

5. การสร้างผลลัพธ์จากการสืบค้นด้วย AXQL

การสืบค้นข้อมูล XML นอกจากเพื่อเรียกค้นข้อมูลที่สนใจแล้ว ยังต้องการข้อมูลเหล่านั้นในรูปแบบการนำเสนอที่แตกต่างกัน, ในหัวข้อนี้จะพูดถึงวิธีต่างๆ ที่ AXQL ใช้ในการสร้างผลลัพธ์ นั่นคือการสร้างโครงสร้างและส่วนประกอบต่างๆ ของข้อมูลตามต้องการ

5.1. การสร้างผลลัพธ์แบบ literal

การสร้างผลลัพธ์แบบ literal นั้นก็คือการสร้าง tag ที่ต้องการลงไปใน AXQL โดยตรง ดังที่ได้พบในตัวอย่างก่อนหน้านี้ เช่น

```
<axql:for select="//book">
  <book year="{@year}">
    <title><axql:value select="title"/></title>
  </book>
</axql:for>
```

จากตัวอย่างเป็นการสร้าง element book, attribute year และ element title แบบ literal คือระบุเข้าไปโดยตรง, ค่าของ attribute year ในที่นี้จะใช้รูปแบบพิเศษในการให้ค่าข้อมูลที่เรียกว่า attribute value template ที่ขอยืมมาจาก XSLT โดยค่าที่อยู่ภายในวงเล็บ { } คือ path expression และค่าที่ได้ออกมาจะเหมือนกับการใช้งานคำสั่ง axql:value นั่นเอง, ในที่นี้จะใช้การสร้างผลลัพธ์แบบ literal ผสมกับคำสั่งของ AXQL เพื่อสร้างข้อมูลบางส่วนที่ไม่สามารถสร้างขึ้นมาได้เองโดยตรง

5.2. การใช้คำสั่ง axql:copy

การสร้างผลลัพธ์แบบ literal นั้น ผู้เรียกค้นจะต้องเป็นผู้สร้าง element ของคำตอบเอง รวมถึงข้อมูลภายใน element นั้นๆ ด้วย ซึ่งบางครั้งคำตอบที่ต้องการอาจจะเป็นเพียงการยกทั้ง sub tree ของเอกสารต้นฉบับมาโดยตรงเลยก็ได้ การสร้าง element รวมถึงเนื้อหาภายในเองจึงเป็นเรื่องที่เกินจำเป็นและเพิ่มความยากในการเรียกค้นจนเกินไป, AXQL มีคำสั่งที่ใช้ในการทำงานนี้โดยเฉพาะคือ axql:copy ดังตัวอย่าง

การสร้าง element และเนื้อหาเอง

```
<axql:for select="//book">
  <book year="{@year}">
    <title><axql:value select="title"/></title>
    <author>
      <last>
        <axql:value select="author/last"/>
      </last>
      <first>
        <axql:value select="author/first"/>
      </first>
    </author>
    <price><axql:value select="price"/></price>
  </book>
</axql:for>
```

การใช้คำสั่ง axql:copy ดึงข้อมูลมาจากเอกสารต้นฉบับ

```
<axql:for select="//book">
  <book>
    <axql:copy select="@year"/>
    <axql:copy select="title"/>
    <axql:copy select="author"/>
    <axql:copy select="price"/>
  </book>
</axql:for>
```

คำสั่ง axql:copy นั้นไม่จำเป็นว่าจะต้องทำงานกับ element เท่านั้นแต่สามารถใช้งานร่วมกับ node ทุกๆ ประเภทภายในเอกสาร ดังจะเห็นได้จากตัวอย่างว่าสามารถใช้คำสั่ง axql:copy กับ attribute year ได้ด้วย, จะเห็นได้ว่าคำสั่ง axql:copy นั้นเพิ่มความสะดวกในการเรียกค้นข้อมูลได้อย่างมาก

5.3. การสร้าง element และ attribute ด้วยคำสั่งของ AXQL

การใช้งาน literal เพื่อสร้าง element และ attribute นั้นจะมีข้อจำกัดที่จะต้องระบุชื่อ, namespace prefix ของ element และ attribute อย่างตายตัว, การใช้คำสั่ง axql:copy แม้จะสะดวกแต่ก็ไม่เปิดโอกาสให้ผู้เรียกค้นสามารถแก้ไขรายละเอียดของข้อมูลได้ตามที่ต้องการ แต่ด้วยคำสั่ง axql:element และ axql:attribute จะทำให้การสร้าง element และ attribute มีความยืดหยุ่นมากยิ่งขึ้นดังตัวอย่าง

```
<axql:for select="//book">
  <axql:element name="book">
    <axql:attribute name="year" value="@year"/>
    <axql:element name="title">
      <axql:value select="title"/>
    </axql:element>
  </axql:element>
</axql:for>
```

จากตัวอย่างเป็นการสร้าง element book, attribute year และ element title ด้วยคำสั่ง axql:element และ axql:attribute ซึ่งแม้จะมีรูปแบบการใช้งานที่ยุ่งยากกว่า แต่ก็ให้อิสระแก่ผู้เรียกค้นในการปรับแต่งข้อมูลได้มากขึ้นเช่นกัน

5.4. การสร้าง namespace

การสร้าง namespace สามารถทำได้ 2 วิธีคือ สร้างโดยตรงภายใน element แบบ literal และสร้างโดยใช้คำสั่ง axql:namespace ดังตัวอย่าง

การสร้าง namespace แบบ literal

```
<axql:for select="//book">
  <bn:book xmlns:bn="http://www.bn.com">
    <bn:title>
      <axql:value select="title"/>
    </bn:title>
  </bn:book>
</axql:for>
```

การสร้าง namespace ด้วยคำสั่ง axql:namespace

```
<axql:for select="//book">
  <bn:book>
    <axql:namespace prefix="bn"
      uri="http://www.bn.com"/>
    <bn:title>
      <axql:value select="title"/>
    </bn:title>
  </bn:book>
</axql:for>
```

จากตัวอย่างจะเป็นการสร้างข้อมูลที่อยู่ภายใต้ namespace เป็น "bn" ซึ่งอ้างอิงกับ URI คือ "http://www.bn.com", โดยปกติการใช้งาน namespace นั้นมักจะถูกนิยามแบบ literal มากกว่า อย่างไรก็ในกรณีที่ต้องการนิยาม namespace ที่มีความยืดหยุ่นมากๆ เช่น สามารถเปลี่ยนแปลงตามสถานการณ์ได้ การใช้คำสั่ง axql:namespace จะมีความเหมาะสมมากกว่า

5.5. การสร้าง node ของข้อมูล

เป็นการสร้างข้อมูลที่แท้จริงที่ถูกเก็บอยู่ภายใน element, การสร้างข้อมูลจะทำได้ 2 วิธีเช่นกัน คือ แบบ literal แต่วิธีนี้ค่อนข้างจะเป็นวิธีที่ไม่มีประโยชน์ในการเรียกค้นข้อมูลเนื่องจากไม่สามารถปรับเปลี่ยนค่าได้ และวิธีที่ใช้คำสั่งของ AXQL ซึ่งจะเห็นได้จากหลายๆ ตัวอย่างที่ผ่านมาคือคำสั่ง axql:value, attribute ที่สำคัญของคำสั่ง axql:value นี้ นอกจาก select ซึ่งเป็น path expression ที่อ้างอิงไปยังข้อมูลที่ต้องการแล้ว ยังประกอบไปด้วย attribute type ซึ่งสามารถระบุประเภทของข้อมูลได้ โดยประเภทของข้อมูลนี้จะสัมพันธ์กับประเภทข้อมูลที่ถูกนิยามโดย XML Schema [7, 8], และ attribute format ซึ่งใช้ในการจัดรูปแบบของข้อมูลที่จะแสดงไปยังผลลัพธ์ตามที่ต้องการ ดังตัวอย่าง

```
<axql:value select="title" type="xsd:string"/>
<axql:value select="price" type="xsd:decimal"
  format="0.00"/>
```

จากตัวอย่างเป็นการสร้าง data node ที่มีข้อมูลเป็นข้อความของชื่อหนังสือ และข้อมูลราคาหนังสือที่มีทศนิยม 2 ตำแหน่ง

5.6. การสร้าง comment และ processing instruction

ทั้ง comment และ processing instruction แม้จะไม่ใช่ข้อมูลที่ผู้ใช้กำลังสนใจโดยตรง แต่ก็เป็นส่วนประกอบสำคัญที่ช่วยอธิบายให้ผู้อื่น หรือโปรแกรมอื่นๆ สามารถทำความเข้าใจและทำงานร่วมกันได้ดียิ่งขึ้น, การสร้าง comment และ processing instruction จะทำได้โดยใช้คำสั่ง axql:comment และ axql:pi ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 1

```
<axql:comment value="This is comment."/>
<axql:pi name="user-parameter" value="none">
```

ตัวอย่างที่ 2

```
<axql:comment>
  This is comment. The book's title is
  <axql:value select="title"/>.
  Printed by <axql:value select="publisher">
  in <axql:value select="@year + 543"> BC.
</axql:comment>
<axql:pi name="user-parameter">
  <axql:value select="string('none')"/>
</axql:pi>
```

จากตัวอย่างข้างต้นเป็นการสร้าง comment และ processing instruction ด้วยค่าของ path expression จาก attribute value และ ด้วยคำสั่งของ AXQL ตามลำดับ, การสร้าง comment และ processing instruction ของคำตอบ ไม่สามารถใส่ลงไปในการเรียกค้นของ AXQL ได้โดยตรง เหมือนการสร้าง element แบบ literal เนื่องจากข้อมูลเหล่านี้จะไม่ถูกสนใจโดย query processor

6. คำสั่งตรวจสอบเงื่อนไข

ในบางครั้งการสร้างผลลัพธ์จากการเรียกค้นจะขึ้นอยู่กับเงื่อนไขหลายๆ ประการ, คำสั่งตรวจสอบเงื่อนไขที่ได้พบมาก่อนหน้านี้ คือ axql:where นั้นใช้ในการตรวจสอบเงื่อนไขว่าจะทำหรือไม่ทำเท่านั้น แต่ในความเป็นจริงเงื่อนไขในการเลือกทำอาจมีมากกว่านี้ ดังจะอธิบายได้จากคำสั่งต่อไปนี้ คือ axql:if และ axql:choose มีรายละเอียดต่อไปนี้

6.1. คำสั่ง axql:if

คำสั่ง axql:if จะมี attribute test ซึ่งค่า expression ภายในจะถูกตีความออกมาเป็น boolean, หากค่าที่ได้เป็น true คำสั่งที่อยู่ภายใน element axql:then จะถูกเรียกให้ทำงาน ไม่เช่นนั้นคำสั่งภายใน element axql:else ก็จะถูกเรียกให้ทำงานแทน ดังตัวอย่าง

```
<axql:for select="//book">
  <axql:if test="price > 100">
    <axql:then>
      <expensive-book title="{title}"/>
    </axql:then>
    <axql:else>
      <cheap-book title="{title}"/>
    </axql:else>
  </axql:if>
</axql:for>
```

จากตัวอย่าง เป็นการทดสอบเงื่อนไขราคาของหนังสือแต่ละเล่ม หากหนังสือมีราคาสูงกว่า 100 ก็จะสร้าง element คำตอบชื่อ expensive-book ไม่เช่นนั้นก็สร้าง cheap-book แทน

6.2. คำสั่ง axql:choose

คำสั่ง axql:if จะมีทางเลือกในการทำงานได้เพียง 2 ทางเลือกเท่านั้น แต่ในบางสถานการณ์ เงื่อนไขและทางเลือกอาจมีมากกว่านั้น ซึ่ง AXQL จะทำงานโดยใช้คำสั่ง axql:choose โดยภายในจะประกอบด้วยหลายๆ ทางเลือกคือคำสั่ง axql:when ที่มีเงื่อนไขในแต่ละกรณีระบุอยู่ด้วย attribute test ไม่เช่นนั้นก็ต้องทำในทางเลือกสุดท้ายในคำสั่ง axql:otherwise ดังตัวอย่าง

```
<axql:for select="//book">
  <axql:choose>
    <axql:when test="price > 100">
      <expensive-book title="{title}"/>
    </axql:when>
    <axql:when test="price > 20">
      <cheap-book title="{title}"/>
    </axql:when>
    <axql:otherwise>
      <sale-book title="{title}"/>
    </axql:otherwise>
  </axql:choose>
</axql:for>
```

จากตัวอย่าง หากราคาของหนังสือสูงกว่า 100 ก็จะสร้าง element ที่ชื่อ expensive-book, ถ้าราคาต่ำว่า 100 แต่ไม่ต่ำไปกว่า 20 ก็จะสร้าง element ที่ชื่อ cheap-book, หากไม่มีเงื่อนไขใดที่ตรงเลย ก็จะทำเงื่อนไขสุดท้าย คือ สร้าง element ที่ชื่อ sale-book

7. การเรียกใช้งานฟังก์ชัน

7.1. การนิยามและเรียกใช้ฟังก์ชัน

การเรียกใช้งานคำสั่งของ AXQL บางครั้งชุดของคำสั่งหนึ่งๆ มีความจำเป็นต้องถูกเรียกใช้มากกว่าหนึ่งครั้ง การเขียนคำสั่งที่ซ้ำซ้อนกันทุกๆ ครั้งไม่ใช่วิธีการที่ดี AXQL จึงมีคุณสมบัติของฟังก์ชันเพื่อช่วยลดความซ้ำซ้อนนี้ นอกจากนั้นฟังก์ชันยังช่วยให้สามารถใช้งานและทำความเข้าใจคำสั่งในการเรียกค้นข้อมูลได้ง่ายยิ่งขึ้น การนิยามฟังก์ชันจะใช้คำสั่ง axql:function ซึ่งเป็น top level element ของคำสั่ง axql:query และเรียกใช้ฟังก์ชันด้วยคำสั่ง axql:call-function ดังตัวอย่าง

```
<axql:query
  xmlns:axql="http://www.cpe.ku.ac.th/AXQL">

  <axql:function name="create-book">
    <book year="{@year}">
      <title><axql:value select="title"/></title>
    </book>
  </axql:function>

  <axql:main>
    <results>
      <axql:for select="//book">
        <axql:call-function name="create-book"/>
      </axql:for>
    </results>
  </axql:main>

</axql:query>
```

จากตัวอย่าง เป็นการนิยามฟังก์ชันชื่อ create-book เพื่อสร้าง element book, title และ attribute year, ภายใน axql:main ซึ่งเป็นส่วนหลักของการทำงาน หลังจากคำสั่ง axql:for เพื่อค้นหาหนังสือทุกๆ เล่มแล้ว จึงส่งการทำงานไปให้ยังฟังก์ชัน create-book แทนด้วยคำสั่ง axql:call-function และระบุชื่อของฟังก์ชันที่เรียกใน attribute name

7.2. การเรียกใช้ฟังก์ชันด้วย pattern

จากตัวอย่างการใช้งานฟังก์ชันที่ผ่านมาเป็นการเรียกใช้งานฟังก์ชันโดยการเรียกจากชื่อของฟังก์ชัน แต่ในบางครั้งข้อมูลที่อยู่ในรูปของเอกสาร XML นั้นจะประกอบด้วย element ที่เป็นไปได้จำนวนมากมาย จนบางครั้งไม่สามารถคาดเดาได้ว่าในกรณีเช่นใด ควรจะต้องเรียกใช้งานฟังก์ชันใด, AXQL ได้อาศัยคุณสมบัติของการเรียกใช้งาน template ใน XSLT มาประยุกต์ใช้งานร่วมกับฟังก์ชัน ทำให้สามารถเรียกใช้ฟังก์ชันโดยพิจารณาจาก pattern ของข้อมูลที่เหมาะสม ดังตัวอย่าง

```
<axql:query
  xmlns:axql="http://www.cpe.ku.ac.th/AXQL">

  <axql:function pattern="book[price > 100]">
    <expensive-book title="{title}"/>
  </axql:function>

  <axql:function pattern="book[price &lt;= 100]">
    <cheap-book title="{title}"/>
  </axql:function>

  <axql:main>
    <results>
      <axql:for select="//book">
        <axql:call-function select="."/>
      </axql:for>
    </results>
  </axql:main>

</axql:query>
```

จากตัวอย่างข้างต้น จะเรียกใช้งานฟังก์ชันจากความสัมพันธ์ระหว่างค่าของ attribute select ใน axql:call-function และ ค่าของ attribute pattern ใน axql:function โดย axql:call-function จะ select "." ซึ่งในที่นี้คือ element book แต่ละเล่มนั่นเอง ส่งไปให้ยังฟังก์ชันต่างๆ ที่เหมาะสม หาก book นั้นมีราคาสูงกว่า 100 ก็จะตรงกับ pattern ของฟังก์ชันแรก แต่หากราคาของ book ต่ำกว่า 100 ก็จะตรงกับ pattern ของฟังก์ชันที่สอง (ภายในค่าของ attribute เครื่องหมาย < จะไม่สามารถใช้งานได้โดยตรง เนื่องจากเป็นเครื่องหมายที่แสดงถึงการเปิด markup ดังนั้นจึงต้องใช้ markup < แทน[1])

ในกรณีที่ข้อมูลสามารถ match กับ pattern ของฟังก์ชันได้มากกว่าหนึ่งฟังก์ชัน ในที่นี้ AXQL จะเลือกใช้งานเพียงฟังก์ชันเดียว โดยดูจากเงื่อนไข ดังต่อไปนี้

- กรณีฟังก์ชันที่ถูกนิยามไว้ในไฟล์อื่นๆ ที่ถูกรวมเข้ามาด้วยคำสั่ง axql:include (จะได้กล่าวถึงต่อไปในหัวข้อ 7.5) ลำดับของไฟล์ที่ถูกรวมเข้ามานี้จะมีความสำคัญแตกต่างกัน โดยไฟล์ที่ถูกรวมเข้ามาในลำดับหลังจะมีความสำคัญมากกว่า แต่ก็ไม่เท่ากับฟังก์ชันที่ถูกนิยามไว้ในเอกสารเองซึ่งจะมีระดับความสำคัญสูงสุด
- ถ้าฟังก์ชันยังคงมีระดับความสำคัญเท่ากัน AXQL จะดูจาก attribute priority ของคำสั่ง axql:function ซึ่งจะมีค่าเป็นตัวเลข ค่าที่มากกว่าจะมีระดับความสำคัญมากกว่า
- ในกรณีที่ไม่สามารถเลือกใช้งานฟังก์ชันที่เหมาะสม จากเงื่อนไขข้างต้นได้ AXQL จะแจ้งความผิดพลาดให้ผู้ใช้ทราบ

7.3. การรับพารามิเตอร์และการส่งค่าคืนของฟังก์ชัน

โดยปกติในภาษาโปรแกรมใดๆ การใช้งานฟังก์ชันจะมีการระบุประเภทของข้อมูลที่จะคืนมาจากการทำงานของฟังก์ชัน, AXQL จะสามารถระบุประเภทของข้อมูล ที่คืนมาจากการเรียกใช้งานฟังก์ชันในส่วนของการนิยามด้วย attribute type ดังตัวอย่าง

```
<axql:query
  xmlns:axql="http://www.cpe.ku.ac.th/AXQL">

  <axql:function name="find-book" type="xsd:any">
    <axql:param name="author">
    <axql:for select="//book[author = $author]">
      <book title="{title}" year="{year}">
    </axql:for>
  </axql:function>

  <axql:main>
    <results>
      <axql:for select="distinct(//auhtor)">
        <result>
          <axql:copy select="."/>
          <axql:call-function name="find-book">
```

[1] แม้เครื่องหมาย < ซึ่งแสดงถึงการเปิด markup จะถูกห้ามใช้ในค่าของ attribute แต่เครื่องหมาย > ซึ่งแสดงถึงการปิด markup สามารถใช้งานได้ตามปกติ โดยเป็นไปตามข้อกำหนดของมาตรฐานเอกสาร XML version 1.0, นอกจากนี้แม้เครื่องหมาย < จะเป็นเครื่องหมายที่ถูกต้องของมาตรฐาน XPath version 1.0 อย่างไรก็ตามเมื่อนำมาใช้ร่วมกับ XML จึงต้องเป็นไปตามข้อกำหนดของมาตรฐาน XML

```
            <axql:with-param name="author"
              select="."/>
          </axql:call-function>
        </result>
      </axql:for>
    </results>
  </axql:main>

</axql:query>
```

จากตัวอย่าง เป็นการนิยามฟังก์ชันชื่อ find-book โดยค่าที่ฟังก์ชันคืนกลับมาจะเป็นประเภทข้อมูล xsd:any ที่ถูกระบุด้วย attribute type, xsd:any เป็นประเภทของข้อมูลชนิดหนึ่งที่นิยามโดย XML Schema หมายถึง sub tree ใดๆ ซึ่งจะมีค่าเท่ากับกรณีที่ไม่ได้ระบุไว้ใน attribute type ในความเป็นจริงการระบุชนิดของข้อมูลเป็น xsd:any ในที่นี้ไม่ได้ช่วยในการตรวจสอบใดๆ เลย โดยปกติมักจะใช้ประเภทของข้อมูลที่ผู้ใช้ได้นิยามไว้ก่อนแล้ว เพื่อตรวจสอบว่าข้อมูลที่คืนกลับมานั้นมีความถูกต้องตามประเภทที่ต้องการจริงๆ, ในตัวอย่างนี้ยังได้แสดงการใช้งานคำสั่ง axql:param และ axql:with-param เพื่อใช้ในการส่งค่าไปให้ยัง function ด้วย, ตัวอย่างข้างต้นเป็นการเรียกดูหนังสือโดยแบ่งตามชื่อของผู้เขียนแต่ละคน ฟังก์ชัน distinct() ใช้เพื่อจัดกลุ่มตามชื่อผู้แต่ง หลังจากนั้นจึงสร้าง element author แล้วจึงเรียกฟังก์ชันเพื่อสร้าง element book ของหนังสือทุกเล่มที่แต่งโดยผู้แต่งคนนี้

7.4. การเรียกใช้ฟังก์ชันภายใน path expression

ในบางครั้งการเรียกใช้งานฟังก์ชันโดยผ่าน element axql:call-function ไม่สะดวกและไม่สามารถทำงานในบางลักษณะได้ AXQL จึงกำหนดให้มีวิธีการเรียกใช้งานฟังก์ชันได้อีกวิธีหนึ่ง คือการเรียกใช้งานฟังก์ชันภายใน path expression ซึ่งมีความยืดหยุ่นในการใช้งานมากกว่า สามารถแสดงได้ดังตัวอย่าง

```
<axql:query
  xmlns:axql="http://www.cpe.ku.ac.th/AXQL">

<axql:function name="depth" type="xsd:integer">
  <axql:param name="e">
  <axql:if test="empty($e/*)">
    <axql:then>1</axql:then>
    <axql:else>
      <axql:value type="xsd:integer"
        select="max(depth($e/*)) + 1">
    </axql:else>
  </axql:if>
</axql:function>

<axql:main>
  <depth>
    <axql:value type="xsd:integer"
      select="depth(document('partlist.xml'))"/>
  </depth>
</axql:main>

</axql:query>
```

จากตัวอย่างข้างต้น เป็นการนิยามฟังก์ชัน depth เพื่อใช้หาระดับความลึกของ sub element โดยจะทำงานในลักษณะ recursive คือมีการวนเรียกเข้าตัวเอง, การใช้งานฟังก์ชัน depth จะถูกเรียกใช้ได้จากภายใน path expression เช่น depth(document('partlist.xml')) หรือ max(depth($e/*)) + 1 เป็นต้น

การรองรับการทำงานของฟังก์ชันแบบ recursive ทำให้ AXQL สามารถเรียกค้นไปในโครงสร้างที่ซับซ้อนของเอกสาร XML ได้อย่างมีประสิทธิภาพ ซึ่งการใช้งานฟังก์ชันแบบ recursive นี้จะสามารถใช้ได้กับทั้งการเรียกใช้งานฟังก์ชันจากชื่อฟังก์ชันโดยตรง และโดยอาศัย pattern ของข้อมูล

7.5. คำสั่ง axql:include

ในบางครั้งต้องการเรียกใช้งานฟังก์ชันที่ถูกนิยามไว้ในไฟล์ AXQL อื่นๆ จึงจำเป็นต้องมีกลไกในการรวมเอกสาร AXQL หลายๆ ไฟล์เข้าด้วยกัน ด้วยคำสั่ง axql:include ซึ่งเป็น top level element ของ axql:query, อย่างไรก็ตามเมื่อรวมเอกสาร AXQL จากหลายๆ แหล่งเข้าด้วยกันแล้วจะต้องมี element axql:main ได้เพียงหนึ่งเดียวเท่านั้น, การใช้งาน axql:include แสดงได้ดังตัวอย่าง

```
<axql:query
  xmlns:axql="http://www.cpe.ku.ac.th/AXQL">

  <axql:include href="library.axql"/>

  <axql:main>
    <!-- AXQL instructions-->
  </axql:main>

</axql:query>
```

7.6. ฟังก์ชันแบบ built-in

การใช้งานฟังก์ชันที่กล่าวมาข้างต้น เป็นการเรียกใช้ฟังก์ชันที่ผู้ใช้เป็นผู้นิยามขึ้นเอง อย่างไรก็ตาม AXQL ได้จัดเตรียมฟังก์ชันแบบ built-in ซึ่งเป็นฟังก์ชันพื้นฐานที่มีความจำเป็นในการใช้งานสำหรับการสืบค้นข้อมูล XML ให้ผู้ใช้สามารถเรียกใช้ได้, ตัวอย่างฟังก์ชันแบบ built-in ที่ได้แสดงให้เห็นก่อนหน้านี้ ยกตัวอย่างเช่น ฟังก์ชัน document() เพื่อคืนค่า root node ของเอกสาร XML, ฟังก์ชัน distinct() เพื่อใช้ในการจัดกลุ่มข้อมูล, ฟังก์ชัน empty() เพื่อใช้ตรวจสอบว่าเป็น node ว่างหรือไม่ เป็นต้น

เนื่องจาก AXQL มีการใช้งาน XPath เป็น path expression, ดังนั้น AXQL จะสามารถเรียกใช้งานฟังก์ชันต่างๆ ที่มีอยู่ภายใน XPath ได้ เช่น ฟังก์ชัน position() เพื่อระบุตำแหน่งของ node ภายใน node list, ฟังก์ชัน id() เพื่อการเชื่อมโยงกันระหว่างข้อมูลในส่วนต่างๆ ด้วยความสัมพันธ์ของ attribute แบบ ID และ IDREF หรือ IDREFS, ฟังก์ชันเกี่ยวกับ data type ประเภทต่างๆ เช่น string(), substring(), boolean(), number() เป็นต้น นอกจากนี้ AXQL ยังมีการนิยามฟังก์ชันแบบ build-in อื่นๆ ที่มีประโยชน์ในการสืบค้นข้อมูลเพิ่มเติมอีก เช่น ฟังก์ชันที่เกี่ยวกับการ

คำนวณค่าสรุปของข้อมูล (aggregation function) เช่นเดียวกับที่มีในภาษา SQL อันได้แก่ฟังก์ชัน sum(), count(), max(), min(), avg() เป็นต้น การใช้งานฟังก์ชันการคำนวณค่าสรุปของข้อมูล แสดงได้ดังตัวอย่าง

```
<axql:for select="distinct(//publisher)">
  <axql:var name="publisher" select="text()"/>
  <axql:var name="books_price"
    select="//book[publisher = $publisher]/price"/>
  <publisher axql:value="$publisher"/>
  <avg_books_price axql:value="avg($books_price)"/>
</axql:for>
```

จากตัวอย่างข้างต้น เป็นการใช้งานฟังก์ชัน distinct() เพื่อจัดกลุ่มข้อมูลตามชื่อสำนักพิมพ์ และใช้ฟังก์ชัน avg() เพื่อหาค่าเฉลี่ยของราคาหนังสือของแต่ละสำนักพิมพ์

8. การเรียงลำดับข้อมูล

จากตัวอย่างการเรียกค้นข้อมูลที่ผ่านมา ลำดับข้อมูลของคำตอบจะเป็นเช่นเดียวกับลำดับที่ในเอกสารต้นฉบับ ในกรณีที่ต้องการให้ผลลัพธ์เรียงลำดับตามข้อมูลที่ต้องการจะเรียกใช้คำสั่ง axql:sort, คำสั่ง axql:sort จะใช้งานร่วมกับคำสั่ง axql:for และ axql:call-function เมื่อเรียกใช้ฟังก์ชันโดยการ match pattern เท่านั้น ตัวอย่างการใช้งาน เช่น

ใช้คำสั่ง axql:sort ร่วมกับ axql:for

```
<axql:for select="//book">
  <axql:sort select="title" type="xsd:string"/>
  <book>
    <title><axql:value select="title"/></title>
  </book>
</axql:for>
```

ใช้คำสั่ง axql:sort ร่วมกับ axql:call-function

```
<axql:call-function select="//book">
  <axql:sort select="title" type="xsd:string"/>
</axql:call-function>
```

จากตัวอย่างข้างต้น ภายในคำสั่ง axql:for และ axql:call-function จะอ้างอิงไปยังเซตของ element book แต่ก่อนที่จะวนรอบแต่ละ node เพื่อสร้างผลลัพธ์หรือส่งไปให้ฟังก์ชันนั้นจะต้องจัดเรียงลำดับการทำงานของข้อมูลตามคำสั่ง axql:sort ซึ่งในที่นี้ ต้องการให้ข้อมูลหนังสือ ถูกจัดเรียงตามชื่อหนังสือ โดยจัดเรียงในลักษณะของข้อความธรรมดาเป็นต้น, การใช้งานคำสั่ง axql:sort กับ axql:for นั้น สามารถเขียนอยู่ในรูปแบบย่อได้ ดังตัวอย่าง

```
<book axql:for="//book" axql:sort="title">
  <title axql:value="title"/>
</book>
```

9. การทดสอบชนิดของข้อมูล

การใช้งาน XML ในยุคแรกๆ นั้น ข้อมูลที่เก็บอยู่ภายในจะอยู่ในรูปของข้อความเท่านั้น อย่างไรก็ตามความจำเป็นในการระบุประเภทของข้อมูลที่จัดเก็บในระดับของเอกสาร XML นั้นนับว่ามีความสำคัญมาก จึงเกิดมาตรฐาน XML Schema ขึ้นทำให้การระบุชนิดของข้อมูลภายในเอกสาร XML มีความชัดเจนยิ่งขึ้น ข้อมูลที่จัดเก็บสามารถมีประเภทข้อมูลที่หลากหลาย นอกจากนั้นยังสามารถตรวจสอบได้ในระดับของเอกสาร XML, AXQL สามารถทำงานร่วมกับ XML Schema เพื่อการตรวจสอบประเภทของข้อมูลที่เป็นคำตอบจากการเรียกค้นว่าเป็นไปตามที่ต้องการหรือไม่ ด้วยคำสั่ง axql:data-type ดังตัวอย่างต่อไปนี้

การนิยามประเภทข้อมูลด้วย XML Schema

```
<xsd:schema
  xmlns:xsd="http://www.w3.org/2000/08/XMLSchema">

  <xsd:complexType name="MyResultType">
    <xsd:element name="results">
      <xsd:complexType>
        <xsd:element name="book" type="BookType"
          minOccurs="0" maxOccurs="unbounded"/>
      </xsd:complexType>
    </xsd:element>
  </xsd:complexType>

  <xsd:complexType name="BookType">
    <xsd:attribute name="year" type="xsd:integer"/>
    <xsd:element name="title" type="xsd:string"/>
  </xsd:complexType>

</xsd:schema>
```

การทดสอบประเภทของข้อมูล

```
<axql:data-type name="MyResultType">
  <results>
    <axql:for select="//book">
      <book year="{@year}">
        <title><axql:value select="title"/></title>
      </book>
    </axql:for>
  </results>
</axql:data-type>
```

จากตัวอย่างข้างต้น เป็นการตรวจสอบประเภทข้อมูลของผลลัพธ์ที่ได้จากการเรียกค้นข้อมูลว่าเป็นประเภท MyResultType ที่ถูกนิยามไว้จาก XML Schema หรือไม่ ถ้าประเภทของข้อมูลไม่ถูกต้อง AXQL processor ก็จะแจ้งเตือนให้รู้ถึงข้อผิดพลาด, คำสั่ง axql:data-type สามารถใช้ร่วมกับการสร้าง element แบบ literal ได้เป็นรูปแบบย่อดังนี้

```
<results axql:data-type="MyResultType">
  <book year="{@year}" axql:for="//book">
    <title axql:value="title"/>
  </book>
</results>
```

10. การแก้ไขข้อมูล

จากคุณสมบัติของ AXQL ที่ได้กล่าวมาทั้งหมดข้างต้นล้วนเป็นคุณสมบัติที่ใช้ในการเรียกค้นข้อมูลทั้งสิ้น อย่างไรก็ตามนอกเหนือจากการใช้งานภาษาสืบค้นเพื่อเรียกค้นข้อมูลแล้ว จุดประสงค์ที่สำคัญอีกประการหนึ่งก็คือเพื่อการจัดการข้อมูลที่ถูกเก็บอยู่ภายในเอกสาร XML เช่น การเพิ่มข้อมูลใหม่, การลบข้อมูล และการแก้ไขข้อมูลเป็นต้น, ในหัวข้อนี้จะเป็นการแนะนำคุณสมบัติของ AXQL ที่ทำหน้าที่ในการจัดการกับข้อมูลในเอกสาร XML บางส่วน ซึ่งได้แก่คำสั่ง axql:insert, axql:delete และ axql:update ดังมีรายละเอียดต่อไปนี้

10.1. คำสั่ง axql:insert

คำสั่ง axql:insert จะใช้ในการแทรก node หรือ sub tree ของข้อมูลเข้าไปในเอกสาร XML, มีพารามิเตอร์ที่สำคัญคือ attribute select เพื่ออ้างอิงไปยัง node ที่จะนำข้อมูลใหม่เข้าไปแทรก, attribute prep เป็นตัวระบุตำแหน่งที่แน่นอนในการแทรกข้อมูลเมื่อเทียบกับ node ที่อ้างอิงมาจาก attribute select, attribute new-node เป็น path expression ที่ระบุถึงข้อมูลใหม่ที่จะนำมาแทรก หรือหากไม่ต้องการอ้างอิงถึงข้อมูลเก่าก็สามารถสร้างข้อมูลใหม่ขึ้นมาได้ภายในคำสั่ง axql:insert ดังตัวอย่าง

ตัวอย่างที่ 1

```
<axql:insert select="//book/price" prep="after">
  <new-price>
    <axql:value select="0.80 * ./data()">
  </new-price>
</axql:insert>
```

ตัวอย่างที่ 2

```
<axql:insert select="//book[position() = last()]"
  prep="after">
  <book year="2000">
    <title>XML Tutorial</title>
    <author>
      <last>Adam</last>
      <first>Smith</first>
    </author>
    <publisher>New-Wave</publisher>
    <price>49.99</price>
  </book>
</axql:insert>
```

คำสั่งในตัวอย่างที่ 1 จะดำเนินการวนรอบหนังสือทุกๆ เล่ม แล้วแทรก element new-price ที่มีค่าเป็น 80% ของราคาเดิมต่อท้ายจาก element price ที่มีอยู่, คำสั่งในตัวอย่างที่ 2 ทำการแทรก element book ใหม่ต่อท้าย element book ที่มีอยู่เดิมอันสุดท้าย

10.2. คำสั่ง axql:delete

คำสั่ง axql:delete จะใช้ลบ node ออกจากเอกสาร XML มีพารามิเตอร์ที่สำคัญคือ attribute select ที่จะชี้ไปยัง node ที่ต้องการจะลบออก ดังตัวอย่าง

```
<axql:insert select="//book/price" prep="after">
  <new-price axql:value="0.80 * ./data()"/>
</axql:insert>
<axql:delete select="//book/price"/>
```

จากตัวอย่าง ในหนังสือทุกๆ เล่ม จะการสร้าง element new-price ขึ้นมาใหม่ จากนั้นจึงลบ element price เดิมทิ้งไป

10.3. คำสั่ง axql:update

คำสั่ง axql:update ใช้ในการแทนที่ node ด้วย node ของข้อมูลใหม่ ดังตัวอย่าง

```
<axql:update select="//price">
  <new-price xql:value="0.80 * ./data()"/>
</axql:update>
```

จากตัวอย่างข้างต้นเป็นการแทนที่ element price ด้วย element new-price

11. การพัฒนาระบบการสืบค้นด้วยภาษา AXQL

เนื่องจาก AXQL เป็นภาษาที่ถูกพัฒนาขึ้นมาโดยมีพื้นฐานเป็นเอกสาร XML จึงทำให้การพัฒนาระบบการสืบค้นด้วยภาษา AXQL (AXQL Query Engine) เป็นไปด้วยความสะดวก เนื่องจากสามารถใช้ parser ของ XML ที่มีอยู่เป็นแล้วพื้นฐานได้ โดยไม่ต้องเขียน parser ขึ้นมาใหม่โดยเฉพาะ ทำให้ลดเวลาในการพัฒนาลง ผู้พัฒนาจะสามารถเน้นความสนใจไปยังขั้นตอนในการทำงานของ query engine ได้อย่างเต็มที่

ตัวอย่างการนำระบบการสืบค้นด้วยภาษา AXQL ไปใช้งาน เช่น การใช้งานร่วมกับโปรแกรมจัดการฐานข้อมูล (Database Management System), เนื่องจากในปัจจุบันข้อมูลเอกสาร XML มีจำนวนมากขึ้นเรื่อยๆ และโปรแกรมจัดการฐานข้อมูลในรุ่นใหม่จะสามารถรองรับการเก็บข้อมูลแบบ XML ได้ อย่างไรก็ตามยังไม่มีกลไกในการสืบค้นข้อมูล XML ที่เป็นมาตรฐาน ดังนั้นผู้ผลิตแต่ละรายก็จะกำหนดวิธีการของตัวเองทำให้ผู้ใช้เกิดความสับสน และต้องผูกติดกับผลิตภัณฑ์ตัวใดตัวหนึ่งจนเกินไป ในที่นี้ AXQL จะสามารถถูกใช้เป็น gateway เพื่อเชื่อมต่อระหว่างผู้ใช้ซึ่งจะสืบค้นข้อมูลด้วยภาษา AXQL และโปรแกรมจัดการฐานข้อมูล ระบบจะเป็นตัวรองรับในการติดต่อกับกลไกที่หลากหลายของโปรแกรมจัดการฐานข้อมูลแบบต่างๆ ให้ ทำให้โปรแกรมที่เขียนมีความยืดหยุ่นสูง และมีความซับซ้อนน้อยลง เป็นต้น ตัวอย่างอื่นที่เห็นได้ชัดในการใช้งาน AXQL เช่น การใช้งานร่วมกับ Web Server เป็นต้น เนื่องจากปัจจุบัน โปรแกรม Web Browser สามารถ

รองรับการทำงานกับ XML ได้ ดังนั้นข้อมูลที่ถูกส่งมาจาก Web Server จึงสามารถเป็นเอกสาร XML ได้ อย่างไรก็ตามข้อมูล XML แบบ static จะยากต่อการจัดการ ซึ่งสามารถแก้ไขได้โดยเก็บข้อมูล XML เหล่านี้ไว้ในฐานข้อมูล จากนั้นเมื่อผู้ใช้ต้องการข้อมูล Web Server ก็จะเรียกโปรแกรมหรือ script ที่ทำงานร่วมกับ AXQL เพื่อเรียกค้นข้อมูลที่ผู้ใช้ต้องการส่งกลับไป, ในกรณีนี้ทั้งข้อมูลและเงื่อนไขในการเข้าถึงข้อมูลของผู้ใช้จะถูกแยกออกจากกัน ทำให้ง่ายต่อการจัดการมากยิ่งขึ้น

12. บทสรุป

AXQL เป็นภาษาสืบค้นสำหรับ XML ที่ได้รับอิทธิพลมาจาก ทั้ง XQuery และ XSLT ทำให้ได้รับการสืบทอดข้อดีหลายๆ ประการจากภาษาทั้งสองมา, AXQL ใช้ไวยากรณ์ที่เป็นแบบ XML และการทำงานเลียนแบบ template ของ XSLT ทำให้สามารถทำงานกับเอกสารที่มีความซับซ้อนสูงได้ดี แต่อาศัยรูปแบบการทำงานที่กะทัดรัดและกระชับของ XQuery จึงทำให้เข้าใจได้ง่าย นอกจากนี้ยังมีรูปแบบการใช้งานแบบย่อซึ่งเป็นรูปแบบการใช้งานหลักๆ ที่มักจะใช้บ่อย จึงทำให้แม้จะอยู่ในรูปแบบของ XML แต่ก็มีความง่ายต่อการใช้งานเหมือน XQuery, สามารถทำงานร่วมกับข้อมูลในแบบเอกสาร และฐานข้อมูลได้ดีเหมือนทั้ง XQuery และ XSLT, สามารถทำงานร่วมกับมาตรฐานอื่นๆ ของ XML ที่มีอยู่แล้วได้อย่างสอดคล้อง, มีคุณสมบัติที่จำเป็นในการจัดการข้อมูลภายในเอกสาร XML เช่น คำสั่งในการเพิ่มข้อมูล, ลบข้อมูล และแก้ไขข้อมูล นอกจากนี้ยังสามารถขยายคุณสมบัติเพิ่มเติมในอนาคตได้โดยง่ายเนื่องจากใช้รูปแบบ XML ที่เป็นมาตรฐานและมีความยืดหยุ่นสูง

ตารางที่ 1 แสดงการเปรียบเทียบคุณสมบัติที่สำคัญๆ โดยสรุประหว่าง XQuery, XSLT และ AXQL ซึ่งจะทำให้เห็นได้ว่า AXQL นั้นมีความสามารถที่ไม่ด้อยไปกว่า XQuery ในขณะที่ได้รับคุณสมบัติที่มีประโยชน์หลายประการมาจาก XSLT, ซึ่งผู้เขียนหวังว่า AXQL จะเป็นทางเลือกหนึ่งที่น่าสนใจในการสืบค้นข้อมูล XML หรือเป็นแนวทางเพื่อนำไปสู่การพัฒนาภาษาสืบค้นสำหรับ XML ที่สมบูรณ์ยิ่งขึ้นต่อไป

ตารางที่ 1. แสดงการเปรียบเทียบระหว่าง XQuery, XSLT และ AXQL โดยอ้างอิงจากเอกสาร XML Query Requirements ของ W3C [6]

คุณสมบัติที่เปรียบเทียบ	XQuery	XSLT	AXQL
สามารถเข้าใจและใช้งานได้ง่าย	X	-	X
มีรูปแบบการใช้งานเป็น XML[2]	-	X	X
เป็นอิสระจากมาตรฐานอื่นๆ ที่เกี่ยวข้อง	X	X	X
มีการนิยามข้อผิดพลาดพื้นฐาน (exception)	-	-	-
รองรับการขยายความสามารถในอนาคต	X	X	X
ทำงานร่วมกับ XML Schema	X	-	X
รองรับการทำงานบนเซตของเอกสาร XML	X	-	X
สามารถสืบค้นข้อมูลที่อ้างอิงกันระหว่างเอกสารได้	X	-	X
ทำงานร่วมกับ XML Namespace	X	X	X
รองรับข้อมูลทั้งแบบโครงสร้างและเรียงลำดับ	X	X	X
มีฟังก์ชันในการคำนวณค่าสรุปของข้อมูล (aggregation function)	X	-	X
เรียงลำดับข้อมูลจากการสืบค้นได้	X	X	X
เปลี่ยนหรือสร้างโครงสร้างข้อมูลได้	X	X	X
ทำงานร่วมกับมาตรฐาน XML อื่นๆ ที่มีอยู่แล้วได้	X	X	X
การแก้ไขและเปลี่ยนแปลงข้อมูล[3]	-	-	X

เอกสารอ้างอิง

[1] เอกพล จีรังสุวรรณ และ สมนึก คีรีโต, การวิเคราะห์และเปรียบเทียบภาษาสืบค้นสำหรับ XML ระหว่าง XQuery และ XSLT, 2544 (กำลังอยู่ในระหว่างการพิจารณาเพื่อขอตีพิมพ์ลงในวารสารวิชาการเนคเทค)

[2] International Organization for Standardization (ISO), Information Technology-Database Language SQL. Standard No. ISO/IEC 9075:1999.

[2] เอกสาร requirements ระบุว่าอาจเป็นรูปแบบที่มนุษย์สามารถใช้งานได้ง่าย หรือเป็นรูปแบบ XML ก็ได้

[3] เป็นคุณสมบัติที่นอกเหนือจากที่ระบุไว้ในเอกสาร requirements

[3] World Wide Web Consortium, Extensible Markup Language (XML) Version 1.0 (Second Edition). W3C Recommendation, October 6, 2000. Available http://www.w3.org/TR/REC-xml

[4] World Wide Web Consortium, Namespaces in XML. W3C Recommendation, January 14, 1999. Available http://www.w3.org/TR/REC-xml-names

[5] World Wide Web Consortium, XML Path Language (XPath) Version 1.0. W3C Recommendation, November 16, 1999. Available http://www.w3.org/TR/xpath

[6] World Wide Web Consortium, XML Query Requirements. W3C Working Draft, February 15, 2001. Available http://www.w3.org/TR/xmlquery-req

[7] World Wide Web Consortium, XML Schema Part 1: Structures. W3C Recommendation, May 2, 2001. Available http://www.w3.org/TR/xmlschema-1/

[8] World Wide Web Consortium, XML Schema Part 2: Datatypes. W3C Recommendation, May 2, 2001. Available http://www.w3.org/TR/xmlschema-2/

[9] World Wide Web Consortium, XQuery 1.0: An XML Query Language. W3C Working Draft, June 07, 2001. Available http://www.w3.org/TR/xquery

[10] World Wide Web Consortium, XSL Transformations (XSLT) Version 1.0. W3C Recommendation, November 16, 1999. Available http://www.w3.org/TR/xslt

ภาคผนวก ก. ไวยากรณ์ของ AXQL

```
<!-- Category: instruction -->
<axql:attribute
  name = { ncname }
  prefix = { prefix }
  value = string-expression>
  <!-- Content: instructions -->
</axql:attribute>

<!-- Category: instruction -->
<axql:call-function
  name = qname
  select = node-set-expression>
  <!-- Content: (axql:sort | axql:with-param)* -->
</axql:call-function>

<!-- Category: instruction -->
<axql:choose>
  <!-- Content: (axql:when+, axql:otherwise?) -->
</axql:choose>

<!-- Category: instruction -->
<axql:comment
  value = string-expression>
  <!-- Content: instructions -->
</axql:comment>

<!-- Category: instruction -->
<axql:copy
  select = expression />

<!-- Category: instruction -->
<axql:data-type
  name = schema-type>
  <!-- Content: instructions-->
</axql:data-type>

<!-- Category: instruction-->
<axql:delete
  select = node-set-expression />

<!-- Category: instruction -->
<axql:element
  name = { ncname }
  prefix = { prefix }>
  <!-- Content: instructions -->
</axql:element>

<!-- Category: instruction -->
<axql:else>
  <!-- Content: instructions -->
</axql:else>

<!-- Category: instruction -->
<axql:for
  select = node-set-expression>
  <!-- Content: (axql:sort*, instructions) -->
</axql:for>

<!-- Category: top-level-element -->
<axql:function
  name = qname
  pattern = pattern
  type = schema-type
  priority = number>
  <!-- Content: (axql:param*, instructions) -->
</axql:function>

<!-- Category: instruction -->
<axql:if
  test = boolean-expression>
  <!-- Content: (axql:then, axql:else?) -->
</axql:if>

<!-- Category: top-level-element -->
<axql:include
  href = uri-reference />

<!-- Category: instruction-->
<axql:insert
  select = node-set-expression
  prep = ("before" | "after" | "in")
  new-node = node-set-expression>
  <!-- Content: instructions -->
</axql:insert>

<!-- Category: top-level-element -->
<axql:main>
  <!-- Content: instructions -->
</axql:main>

<!-- Category: instruction -->
<axql:namespace
  prefix = { prefix }
  uri = { uri-reference } />

<!-- Category: instruction -->
<axql:otherwise>
  <!-- Content: instructions -->
</axql:otherwise>
```

```
<!-- Category: instruction -->
<axql:param
  name = qname
  type = schema-type />

<!-- Category: instruction -->
<axql:pi
  name = { ncname }
  value = string-expression>
  <!-- Content: instructions -->
</axql:pi>

<axql:query>
  <!-- Content: top-level-elements -->
</axql:query>

<!-- Category: instruction -->
<axql:sort
  select = string-expression
  type = schema-type
  order = ("ascending" | "descending") />

<!-- Category: instruction -->
<axql:then>
  <!-- Content: instructions -->
</axql:then>

<!-- Category: instruction-->
<axql:update
  select = node-set-expression
  new-node = node-set-expression>
  <!-- Content: instructions -->
</axql:update>

<!-- Category: instruction -->
<axql:value
  select = string-expression
  type = schema-type
  format = data-format />

<!-- Category: top-level-element -->
<!-- Category: instruction -->
<axql:var
  name = qname
  select = expression>
  <!-- Content: instructions -->
</axql:var>

<!-- Category: instruction -->
<axql:when
  test = boolean-expression>
  <!-- Content: instructions -->
</axql:when>

<!-- Category: instruction -->
<axql:where
  test = boolean-expression>
  <!-- Content: instructions -->
</axql:where>

<!-- Category: instruction -->
<axql:with-param
  name = qname
  select = expression>
  <!-- Content: instructions -->
</axql:with-param>
```

Somnuk Keretho, Ph.D.

Assistant Professor

Dr. Somnuk Keretho received his B.Eng. and M.Eng. in Electrical Engineering from Kasetsart University in 1981 and 1986, respectively. With the support from Carl Duisberg Gesellschaft Scholarship, he completed his M.Eng. in Computer Applications from Asian Institute of Technology in 1985. As a Fulbright scholar, he got his Ph.D. degree in Computer Science from University of Southwestern Louisiana, U.S.A. in 1992. His current research interest is related to Software Process Improvement with Capability Maturity Model, Object-Oriented Methodology and Unified Modeling Language, Software Components and Multi-tier Web-based Information Systems Development with Java, and eXtensible Markup Language (XML). At present, he is an assistant professor in Computer Engineering and serves as the chairman of Master of Science Program in Information Technology, Department of Computer Engineering, Kasetsart University.

Ekapol Jeerangsuwan

Mr. Ekapol Jeerangsuwan received his B.Eng. in Computer Engineering from Kasetsart University in 1997. He's studying M.Eng in Computer Engineering at Kasetsart University. His current research is about query language of XML.

การระบุเอกลักษณ์ไม่เป็นเชิงเส้นด้วยวิธีการค้นหาแบบตาบูสำหรับระบบสองมวลความเฉื่อย

Nonlinear Identification Using Tabu Search for a Two-Inertia System

กองพัน อารีรักษ์ [1] *สราวุฒิ สุจิตจร* [2] *อาทิตย์ ศรีแก้ว* [3] *โยธิน เปรมปราณีรัชต์* [4]
[1] *นักศึกษาบัณฑิตศึกษา* [2] *รองศาสตราจารย์* [3] *อาจารย์*
สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
[4] *รองศาสตราจารย์ ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์*
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ABSTRACT – Tabu search technique is one of artificial intelligent methods appropriate for system identification purposes. This article presents the research results of applying the technique to nonlinear identification problem found in a two-inertia system. The search yields a set of saturation nonlinearity. Review of the search method is given. The test results to find satisfactory search parameters are also presented. These parameters lead to fast and efficient searches in such a way that local optimum can be escaped effectively.

KEY WORDS – Tabu search, nonlinear identification

บทคัดย่อ – วิธีการค้นหาแบบตาบูเป็นวิธีการทางปัญญาประดิษฐ์ ที่เหมาะแก่การนำมาประยุกต์ใช้เพื่อการระบุเอกลักษณ์ระบบ บทความนี้นำเสนอผลงานวิจัยที่ใช้วิธีการดังกล่าวระบุเอกลักษณ์ไม่เป็นเชิงเส้น ที่ปรากฏในระบบสองมวลความเฉื่อย ได้ผลเป็นกลุ่มของลักษณะสมบัติไม่เป็นเชิงเส้นชนิดอิ่มตัว บทความได้ให้การทบทวนวิธีดำเนินงานตามหลักการค้นหาแบบตาบู และแสดงผลทดสอบในการหาพารามิเตอร์กำหนดแก่อัลกอริธึมการค้นหา เพื่อส่งผลให้สามารถค้นหาคำตอบได้รวดเร็ว และหลุดออกจากการล็อคของคำตอบเฉพาะถิ่น ได้ดี

คำสำคัญ – วิธีการค้นหาแบบตาบู, การระบุเอกลักษณ์ไม่เป็นเชิงเส้น

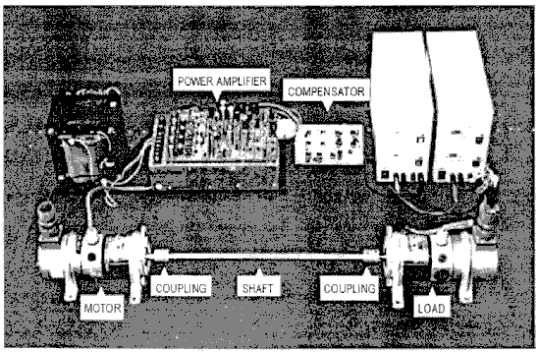
1. คำนำ

อุตสาหกรรมที่ต้องใช้งานระบบขับเคลื่อนทางไฟฟ้ามักประสบปัญหาเนื่องมาจากรีโซแนนท์การบิด (torsional resonance) ซึ่งเป็นการกำทอนเชิงกลในขณะหมุนของมอเตอร์ เพลา และโหลดที่ต่อคู่ควบกัน การบิดตัวของเพลาทำให้เกิดความแตกต่างของการหมุนในตำแหน่งเชิงมุมตลอดแนวเพลา ตำแหน่งเชิงมุมที่เกิดขึ้นนั้นขึ้นอยู่กับความถี่ที่กระตุ้น รวมถึงพารามิเตอร์ทางพลวัตของระบบ และบางความถี่อาจส่งผลให้เกิดมุมของการบิดตัวมีเฟสตรงข้ามกันเป็นผลให้เพิ่มขนาดของการบิดตัว ปรากฏการณ์นี้เรียกว่า รีโซแนนท์การบิด ปรากฏการณ์ดังกล่าวอาจเป็นเหตุให้เกิดความเสียหายต่อโครงสร้างทางกล ระบบมีแนวโน้มที่จะขาดเสถียรภาพได้ง่ายและมีสมรรถนะที่ด้อยลง คณะวิจัยต่างประเทศได้เสนอเทคนิคการแก้ปัญหาการสั่นจากแรงบิดด้วยวิธีการต่างๆ เช่น การควบคุมความเร็วด้วยตัวกรองคาลมาน (Kalman filter) และสถานะอันดับสองเชิงเส้น [9] การใช้ตัวสังเกตบนพื้นฐานการป้อนกลับสถานะ [14] การเลือกอัตราขยายป้อนกลับสถานะที่เหมาะสม [12] การควบคุมแบบปรับตัวโดยใช้ H_∞ [6] การควบคุมความเร็วแบบอัตราขยายอันดับสองเชิงเส้นด้วยการชดเชยแรงบิดโหลดป้อนไปหน้า[10] สำหรับคณะผู้วิจัยไทย[15] ได้เสนอแนวทางแก้ไขปัญหา ด้วยการชดเชยทางพลวัตบนรากฐานของทฤษฎีระบบควบคุมคงทน ออกแบบด้วยการกำหนดตำแหน่งโพล-ซีโร่ เพื่อกำจัดรีโซแนนท์การบิดในระบบ 2 มวลความเฉื่อย โครงสร้างของระบบควบคุมเป็นชนิด 2 ระดับความอิสระ ตัวชดเชยให้สมรรถนะในการกำจัดการกำทอนและให้การตอบสนองที่รวดเร็วน่าพึงพอใจ แต่ในงานวิจัยข้างต้นได้สมมติจุดปฏิบัติงานอยู่ที่ระดับเดียว ดังนั้นระบบจึงนิยามให้เป็นเชิงเส้นได้ แต่การใช้งานจริงมีย่านการใช้งานกว้างและวงจรขยายกำลังมีระบบนิรภัยคอยตัดทอนสัญญาณอินพุต ไม่ให้เกินระดับใดระดับหนึ่งที่อาจกระตุ้นมอเตอร์รุนแรงเกินไปจนเกิดความเสียหาย ตามความเป็นจริงระบบดังกล่าวจึงไม่เป็นเชิงเส้น

งานวิจัยนี้สนใจที่จะศึกษาย่านการทำงานของระบบที่กว้างขึ้น โดยตรวจสอบดูว่ามีข้อจำกัดในทางสมรรถนะและเสถียรภาพต่อระบบอย่างไร เพื่อเป็นแนวทางการตัดสินใจในการขยายย่านการทำงานของระบบควบคุมชุดนี้ ให้ใช้งานในย่านที่กว้างขึ้น ซึ่งจำเป็นที่จะต้องคำนึงถึงความไม่เป็นเชิงเส้นของระบบ ดังนั้นบทความนี้จะนำเสนอวิธีการค้นหาลักษณะสมบัติไม่เป็นเชิงเส้นของระบบดังกล่าวด้วยวิธีการค้นหาแบบตาบู เพราะลักษณะสมบัตินี้จะเป็นประโยชน์ต่อการวิเคราะห์สมรรถนะของระบบต่อไป โดยแบ่งหัวข้อในบทความดังนี้ หัวข้อที่ 2 จะกล่าวถึง วิธีการทดสอบระบบสองมวลความเฉื่อย หัวข้อที่ 3 จะกล่าวถึงหลักการของวิธีการค้นหาแบบตาบู หัวข้อที่ 4 จะกล่าวถึงการทดสอบอัลกอริธึมที่ได้ทำการพัฒนาขึ้น โดยอาศัยหลักการของวิธีการค้นหาแบบตาบู หัวข้อที่ 5 จะแสดงผลการนำอัลกอริธึมที่ผ่านการทดสอบแล้วไปทำการค้นหาลักษณะสมบัติไม่เป็นเชิงเส้นของระบบ หัวข้อที่ 6 จะสรุปปัญหา ข้อเสนอแนะและงานวิจัยที่จะดำเนินงานต่อไปในอนาคตรวมถึงสรุปเนื้อหาของบทความนี้

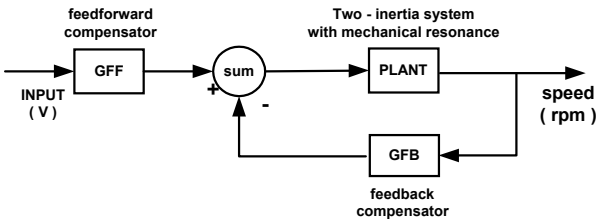
2. การทดสอบระบบสองมวล

การพิจารณาระบบ [15] ที่ได้เสนอแนวทางแก้ไขปัญหาเพื่อกำจัดรีโซแนนท์การบิดในระบบ 2 มวลความเฉื่อย ด้วยการชดเชยทางพลวัตบนรากฐานของทฤษฎีระบบควบคุมคงทน ออกแบบด้วยการกำหนดตำแหน่งโพล-ซีโร่ ตัวชดเชยให้สมรรถนะในการกำจัดการกำทอนและให้การตอบสนองที่รวดเร็วน่าพึงพอใจ แต่การใช้งานของระบบดังกล่าวจำกัดไว้ที่ความเร็วรอบ 143 rpm เนื่องจากถูกจำกัดด้วยความอิ่มตัวแบบไม่เป็นเชิงเส้นของวงจรขยายกำลังสำหรับขับมอเตอร์ งานวิจัยดังกล่าวพิจารณาเฉพาะย่านที่เป็นเชิงเส้นเท่านั้น อย่างไรก็ตามการพิจารณาระบบดังกล่าวในย่านการทำงานที่กว้างขึ้น เป็นสิ่งที่น่าสนใจอย่างมาก เนื่องจากการใช้งานจริงมีย่านการใช้งานที่กว้าง บทความนี้จึงสนใจที่จะศึกษาย่านการทำงานของระบบที่กว้างขึ้น โดยอาศัยตัวชดเชยชุดเดิมจากงานวิจัยที่ได้ทำไว้แล้ว [15] เพื่อเป็นแนวทางการตัดสินใจในการขยายย่านการทำงานของระบบควบคุมชุดนี้ ให้ใช้งานในย่านที่กว้างที่สุดเท่าที่จะทำได้ โดยไม่ส่งผลกระทบต่อสมรรถนะ เสถียรภาพและผลการตอบสนองของระบบ



รูปที่ 1 แสดงระบบสองมวลที่จัดสร้างขึ้นและส่วนประกอบต่างๆ ที่ใช้ในการทดลองของงานวิจัยที่ได้ทำไว้แล้ว [1]

การพิจารณาระบบในย่านการทำงานที่กว้างขึ้น สิ่งแรกที่ควรกระทำ คือ การค้นหาลักษณะสมบัติไม่เป็นเชิงเส้นของระบบว่ามีลักษณะอย่างไรและปรากฏที่ตำแหน่งใดบ้าง ดังนั้นบทความนี้จะนำเสนอวิธีการค้นหาลักษณะสมบัติไม่เป็นเชิงเส้น ด้วยวิธีการค้นหาแบบตาบู ซึ่งจำเป็นต้องอาศัยข้อมูลที่ได้จากการทดสอบระบบ ดังรูปที่ 1 จากรูปพบว่าระบบที่ทำการทดสอบประกอบด้วย ตัวขับดีซี เซอร์โวมอเตอร์ ทำหน้าที่จ่ายไฟให้กับมอเตอร์ และคอยตัดสัญญาณอินพุตไม่ให้เกินระดับใดระดับหนึ่ง เพื่อป้องกันไม่ให้มอเตอร์เกิดความเสียหาย, ชุดควบคุมที่ประกอบด้วยตัวชดเชยในวิถีป้อนกลับ (feedback compensator) มีผลในการกำหนดตำแหน่งโพลและเสถียรภาพของระบบ และตัวชดเชยในวิถีไปหน้าหรือตัวชดเชยอินพุต (input compensator) หรือ พรีฟิลเตอร์ (prefilter) มีผลต่อการปรับปรุงสมรรถนะและผลตอบสนองของระบบตามต้องการ



รูปที่ 2 แสดง บล็อคไดอะแกรมของระบบสองมวลที่สร้างขึ้น

แบบจำลองของระบบสองมวล Gp(s) ได้มาจากการระบุเอกลักษณ์ ARX ส่วนการออกแบบตัวชดเชยในวิถีไปหน้า $G_{FF}(s)$ และตัวชดเชยในวิถีป้อนกลับ $G_{FB}(s)$ อาศัยวิธีการกำหนดตำแหน่งโพล-ซีโร่ แบบคงทนที่มีโครงสร้างแบบ 2 ระดับขั้นเสรี (2 – degree – of – freedom : 2 - DOF) โดยใช้เทคนิคพีชคณิตเชิงเส้น [1]

จากรูปที่ 2 ฟังก์ชันถ่ายโอนของระบบสองมวล, ตัวชดเชยไปหน้าและตัวชดเชยป้อนกลับใช้ค่าต่างๆ ดังนี้

$$G_P(s) = \frac{1.325 * 10\text{^}6}{s\text{^}3 + 13.388 s\text{^}2 + 16.297 * 10\text{^}4 s + 73.117 * 10\text{^}4} \quad (1)$$

$$G_{FF}(s) = 15.093 \frac{s\text{^}3 + 6 * 10\text{^}3 s\text{^}2 + 1.2 * 10\text{^}7 s + 8 * 10\text{^}9}{s\text{^}3 + 7.186 * 10\text{^}3 s\text{^}2 + 19.160 * 10\text{^}6 s} \quad (2)$$

$$G_{FB}(s) = \frac{16.84 * 10\text{^}3 s\text{^}3 + 69.67 * 10\text{^}5 s\text{^}2 + 14.98 * 10\text{^}8 s + 12.07 * 10\text{^}10}{s\text{^}3 + 7.18 * 10\text{^}3 s\text{^}2 + 19.16 * 10\text{^}6 s} \quad (3)$$

2.1 ขั้นตอนการทดสอบระบบสองมวล

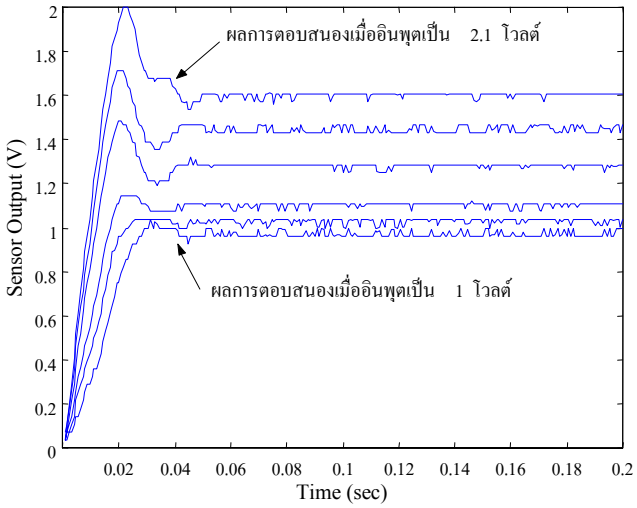
ขั้นตอนที่ 1 ทดสอบระบบที่แรงดันอินพุตเท่ากับ 1 โวลต์ (จุดปฏิบัติงาน) โดยใช้อุปกรณ์ตามรูปที่ 1

ขั้นตอนที่ 2 วัดความเร็วและจับสัญญาณแรงดันที่เอาต์พุต

ขั้นตอนที่ 3 จัดเก็บข้อมูลลงคอมพิวเตอร์

ขั้นตอนที่ 4 ดำเนินการซ้ำตามขั้นตอนที่ 2-3 โดยเปลี่ยนแรงดันอินพุตเป็น 1.1, 1.3, 1.5, 1.7, 1.9 และ 2.1 โวลต์ ตามลำดับ

2.2 ผลการทดสอบระบบสองมวล



รูปที่ 3 แสดงรูปสัญญาณเอาต์พุตจากเซนเซอร์วัดความเร็วที่ได้จากการทดสอบ

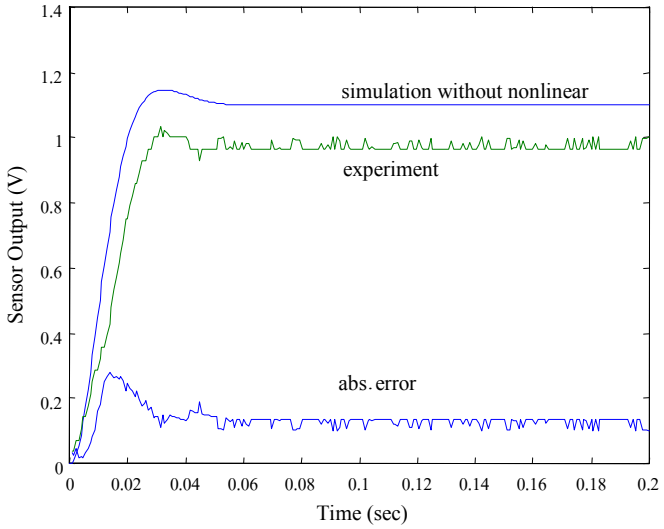
ผลการทดสอบระบบสองมวลความเฉื่อย ดังที่ได้ดำเนินการตามขั้นตอนที่อธิบายไว้ในหัวข้อ 2.1 ได้รับการแสดงไว้ในรูปที่ 3 และอัตราเร็วคงตัวที่วัดได้จากการทดสอบแสดงไว้ในตารางที่ 1 อาจสังเกตจากผลการทดสอบได้ว่า เมื่อมีการขยายย่านการทำงานของระบบดังกล่าว

ตารางที่ 1 แสดงความเร็วที่วัดได้จากการทดสอบ

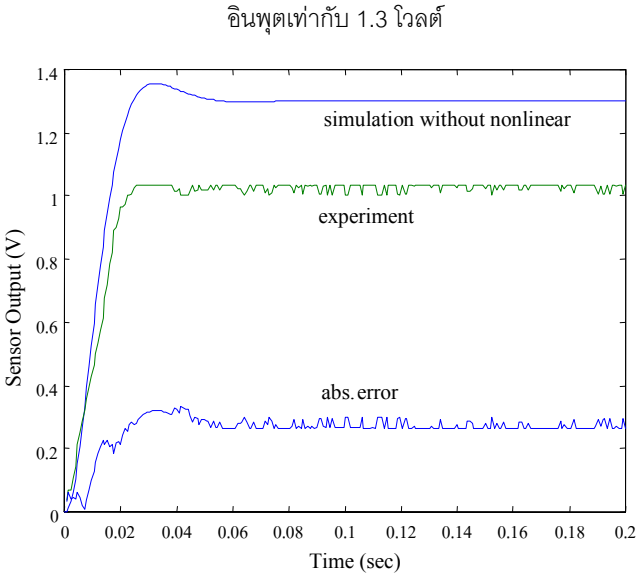
อินพุต (โวลต์)	ความเร็ว (rpm)	เอาต์พุตของเซนเซอร์ (V)
1.0	143	0.95
1.1	144	0.98
1.3	146	1.04
1.5	150	1.10
1.7	179	1.30
1.9	201	1.42
2.1	223	1.60

ระบบมีความไม่เป็นเชิงเส้นค่อนข้างสูง เพื่อเป็นการยืนยันสมมุติฐานดังกล่าวจึงทำการเปรียบเทียบผลการทดลองในรูปที่ 3 กับผลการจำลองสถานการณ์ด้วยคอมพิวเตอร์อาศัยแบบจำลองดังความสัมพันธ์ (1) (2) และ (3) นำมาสร้างเป็นแบบจำลองของระบบรวมที่มีโครงสร้างดังรูปที่ 2 ผลการเปรียบเทียบแสดงได้ดังนี้

อินพุตเท่ากับ 1.1 โวลต์



รูปที่ 4. การเปรียบเทียบผลที่แรงดันอินพุตเท่ากับ 1.1 โวลต์



รูปที่ 5 แสดงการเปรียบเทียบผลที่แรงดันอินพุตเท่ากับ 1.3 โวลต์

จากผลการทดสอบดังรูปที่ 4 และ 5 พบว่าผลการทดสอบจริงกับผลการจำลองด้วยคอมพิวเตอร์มีความคลาดเคลื่อนค่อนข้างสูง สำหรับค่าความคลาดเคลื่อนที่แรงดันอินพุตต่างๆ สรุปได้ดังตารางที่ 2

ตารางที่ 2 แสดงค่าความคลาดเคลื่อนระหว่างผลการทดสอบจริงกับผลการจำลองด้วยคอมพิวเตอร์

อินพุต (โวลต์)	ความคลาดเคลื่อน (sum square error)
1.1	4.9457
1.3	17.8222
1.5	36.8377
1.7	40.5674
1.9	48.6716
2.1	54.9896

จากตารางที่ 2 ค่าความคลาดเคลื่อนมีค่าค่อนข้างสูง สาเหตุเนื่องมาจากแบบจำลองที่ใช้ในการจำลองสถานการณ์เป็นเชิงเส้นมิได้คำนึงถึงความไม่เป็นเชิงเส้นของระบบซึ่งอาจเกิดได้จาก

- การอิ่มตัวของออปแอมป์ในส่วนของตัวชดเชยไปหน้าและตัวชดเชยป้อนกลับ [5]
- การอิ่มตัวของสนามแม่เหล็กในมอเตอร์ [13]
- เนื่องจากอุปกรณ์ที่ใช้ในงานวิจัยเป็น ดีซี เซอร์โวมอเตอร์ 2 ตัวของบริษัท ซันโย เดนกิ (Sanyo Denki Co., Ltd) รุ่น U178T ซึ่งมีทาโคมิเตอร์ต่อคู่ควบอยู่ด้วย ผนวกกับตัวขับ (driver)รุ่น PDT-203-30 ของบริษัทเดียวกัน วงจรขยายกำลังของอุปกรณ์ดังกล่าวมีระบบนิรภัย คอยตัดทอนสัญญาณอินพุตไม่ให้เกินระดับใดระดับหนึ่งเพื่อป้องกันไม่ให้มอเตอร์เกิดความเสียหาย

จากผลการทดสอบและสมมุติฐานข้างต้น สรุปได้ว่า การขยายย่านการทำงานของระบบสองมวลความเฉื่อยของงานวิจัยที่ได้ทำมาแล้ว ระบบดังกล่าวจะต้องคำนึงถึงความไม่เป็นเชิงเส้น ดังนั้นบทความนี้จะนำเสนอวิธีการค้นหาลักษณะสมบัติไม่เป็นเชิงเส้นของระบบด้วยวิธีการค้นหาแบบตาบู ซึ่งเป็นวิธีการที่ใช้เวลาในการค้นหาคำตอบได้รวดเร็วและให้คำตอบได้ใกล้กับจุดคำตอบที่ดีที่สุด (near global optimum) ถ้าเทียบกับวิธีการค้นหาอื่นๆ [16] ลักษณะสมบัติไม่เป็นเชิงเส้นเมื่อหามาได้ ก็จะได้รับการนำไปวิเคราะห์สมรรถนะและเสถียรภาพของระบบได้อย่างถูกต้องแม่นยำ รวมจนถึงนำไปประกอบการพิจารณาปรับปรุงสมรรถนะหากต้องการอีกด้วย

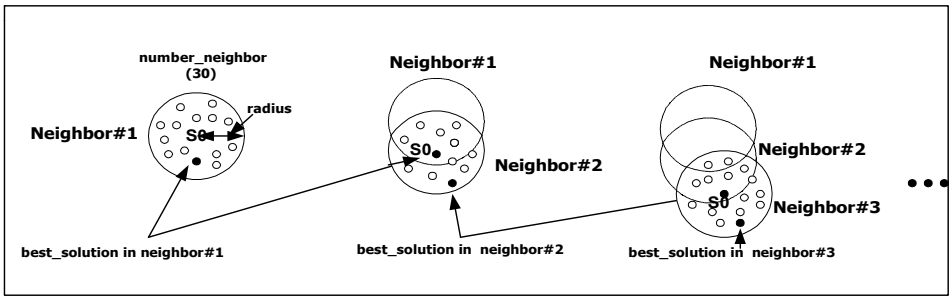
3. หลักการของวิธีการค้นหาแบบตาบู

วิธีการค้นหาแบบตาบู เป็นวิธีการที่นำมาประยุกต์ใช้กับการแก้ปัญหาสำหรับงานที่ต้องการหาคำตอบที่ดีที่สุด (Optimization) ได้อย่างมีประสิทธิภาพ Glover เป็นผู้ริเริ่มเสนอแนวคิดวิธีการค้นหาแบบตาบูไว้เมื่อปี ค.ศ. 1977 ซึ่งได้รับคำอธิบายไว้ใน [8] และหลังจากนั้นก็เป็นที่นิยมใช้กันอย่างแพร่หลาย เนื่องจากวิธีการดังกล่าวสามารถหลีกเลี่ยงคำตอบวงแคบเฉพาะถิ่น (local optimum) และดำเนินการค้นหาคำตอบต่อไปเรื่อยๆจนกระทั่งได้คำตอบที่ใกล้ความเป็นวงกว้าง (near global optimum) [2, 3, 8] นอกจากนั้นเมื่อไม่นานมานี้ได้มีการศึกษาเปรียบเทียบสมรรถนะของเทคนิคการค้นหาแบบ Sequential Quadratic Programming (SQR), Evolutionary Programming (EP) และ Tabu Search (TS) [16] กับปัญหาการหาค่าเหมาะที่สุดภายใต้เงื่อนไขไม่เป็นเชิงเส้น พบว่าวิธีการค้นหาแบบตาบูมีสมรรถนะที่ดีที่สุด ทั้งด้านความแม่นยำในคำตอบและความเร็วในการค้นหา ปัญหาที่ [16] พิจารณานั้นเป็นการค้นหาคำตอบที่ดีที่สุดภายใต้เงื่อนไขบังคับไม่เป็นเชิงเส้น EPใช้เวลาในการค้นหาคำตอบนานที่สุด ส่วน SQP และ TS ใช้เวลา 62.3% และ 18.6% ของเวลาที่ EP ใช้ตามลำดับ

3.1 องค์ประกอบของวิธีการค้นหาแบบตาบู

องค์ประกอบของวิธีการค้นหาแบบตาบูที่แตกต่างจากวิธีการค้นหาแบบอื่นๆ คือ มีเกณฑ์ความเป็นตาบู (tabu list criteria) และ มีเกณฑ์ความปรารถนา (aspiration criteria) ซึ่ง

- "เกณฑ์ความเป็นตาบู" เป็นส่วนที่คอยเก็บข้อมูลของคำตอบในอดีตของกระบวนการค้นหานั้นๆ เพื่อเป็นตัวกำหนดการค้นหาคำตอบว่าจะมีทิศทางไปทางใด หลักการออกแบบเกณฑ์ความเป็นตาบู จะมีลักษณะแตกต่างกันออกไป ขึ้นอยู่กับปัญหาแต่ละชนิด



รูปที่ 6 แสดงภาพอธิบายการทำงาน

- "เกณฑ์ความปรารถนา" เป็นเงื่อนไขที่จะใช้ในบางครั้งที่จำเป็นจะต้องเลือกคำตอบที่อยู่ในเกณฑ์ความเป็นตาบู งานบางชนิดที่ปัญหาไม่ซับซ้อนไม่จำเป็นต้องพึ่งส่วนนี้ก็ได้ เกณฑ์ความเป็นตาบูก็เพียงพอที่จะค้นหาคำตอบที่ดีที่สุดได้

3.2 อธิบายกลไกการทำงานของวิธีการค้นหาแบบตาบู

ความหมายของคำศัพท์ที่ใช้ในการอธิบายการทำงาน

radius = ขอบเขตของการสุ่มในแต่ละรอบของการทำงาน

number_neighbor = จำนวน neighborhood ที่ต้องการสุ่มในแต่ละ search_space

neighbor_list = ส่วนที่เก็บค่า neighborhood ตามจำนวนที่กำหนด (tabu list)

best_neighbor = ค่า neighborhood ที่เป็น local optimum

best_error = ค่าความคลาดเคลื่อนที่เป็น local optimum

overall_neighbor = ค่า neighborhood ที่เป็น near global optimum

overall_best_error = ค่าความคลาดเคลื่อนที่เป็น near global optimum

n = จำนวนรอบในการค้นหาคำตอบ

xlimit = ขอบเขตของพารามิเตอร์แต่ละตัว

S0 = ค่าเริ่มต้นในแต่ละ search_space

S1, S2, S3,…Sn = ค่าที่เก็บไว้ใน neighbor_list

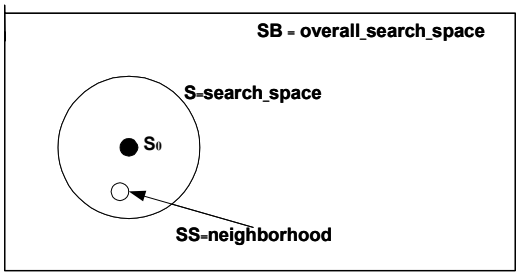
cost = ค่าความคลาดเคลื่อนที่ได้จากฟังก์ชั่นวัตถุประสงค์ (objective function)

หลักการทำงาน

ขั้นตอนที่ 1 โหลดข้อมูลจริงที่ได้จากการทดลอง

ขั้นตอนที่ 2 กำหนดค่า S0 ซึ่งเป็นคำตอบที่เป็นไปได้ทั้งหมดที่อยู่ใน overall_search_space ดังรูปที่ 6 และ7 โดยทำการหาค่า S0 จากการสุ่มคำตอบ

ขั้นตอนที่ 3 เริ่มต้นจากคำตอบที่มีอยู่ โดยกำหนดให้คำตอบที่มีอยู่เป็นคำตอบที่ดีที่สุด best_neighbor = S0 และค่า cost ของคำตอบที่ดีที่สุดกำหนดให้เป็น best_error ซึ่งค่าดังกล่าวได้จากฟังก์ชั่นตรวจสอบค่าความคลาดเคลื่อนหรือฟังก์ชั่นวัตถุประสงค์ (cost หรือ objective function) ในงานวิจัยนี้ ค่า cost คือ ค่าความคลาดเคลื่อนระหว่างข้อมูลที่ได้จากการทดลองจริงกับข้อมูลที่ได้จากการจำลองสถานการณ์ด้วยคอมพิวเตอร์ แสดงด้วยสมการ (4) การค้นหาคำตอบจะทำไปเรื่อยๆจนได้ค่า cost น้อยที่สุดตามที่ได้จัดตั้งค่า cost ไว้ก่อนล่วงหน้า

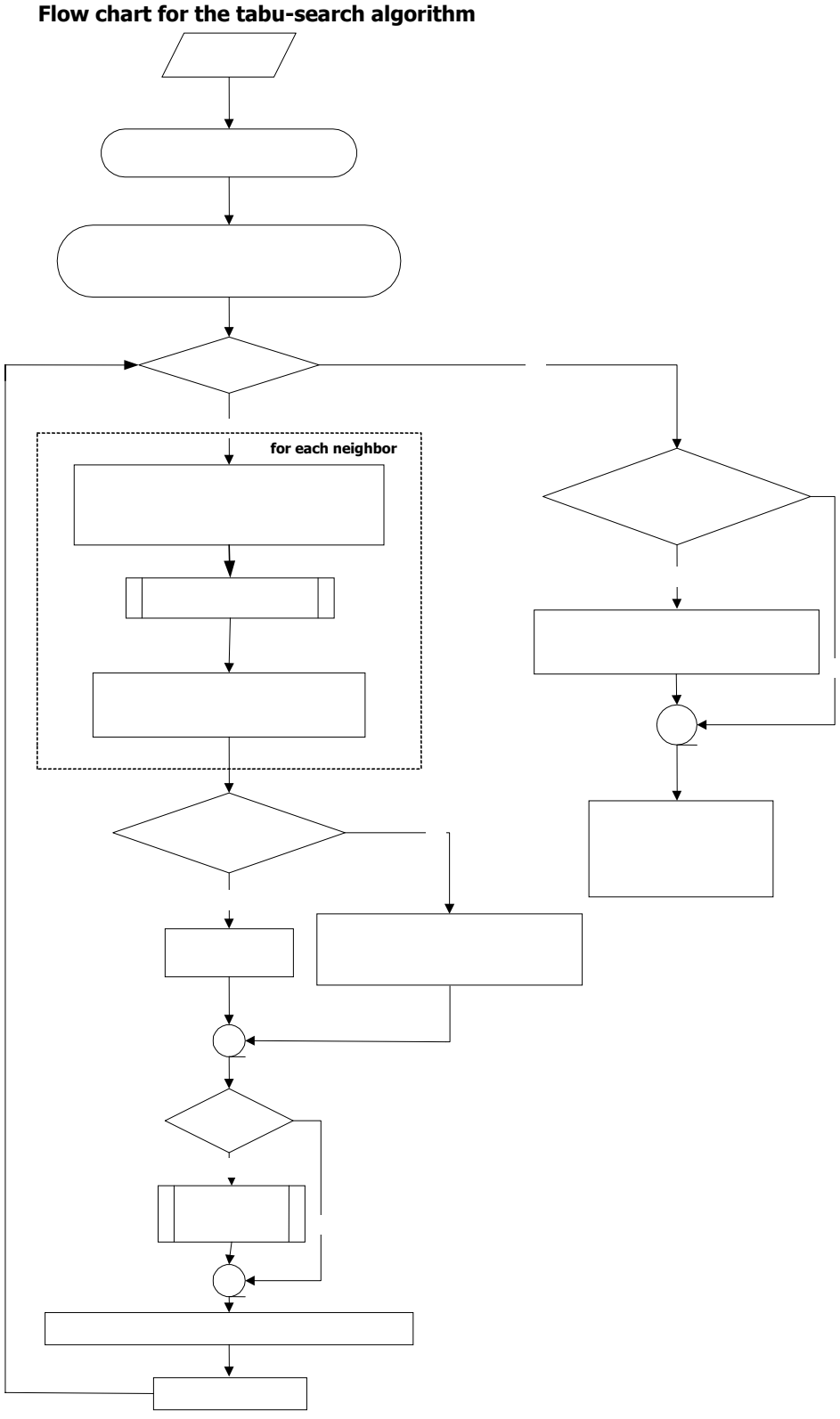


รูปที่ 7 แสดงลักษณะของ space ที่ใช้ในการค้นหาคำตอบ

ขั้นตอนที่ 4 จาก S0 ดำเนินการเคลื่อนย้ายในลักษณะสุ่มเท่ากับจำนวน number_neighbor ในขอบเขตของ search_space ซึ่งค่าดังกล่าวขึ้นอยู่กับค่า radius

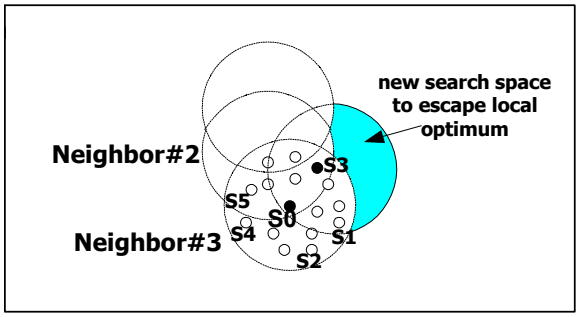
ขั้นตอนที่ 5 คำนวณหาค่า cost ของสมาชิกแต่ละตัวและเลือกค่า cost ที่ดีที่สุด โดยกำหนดให้ค่า cost ดังกล่าวเท่ากับ best_error และ ให้คำตอบนี้เป็น best_neighbor ค่า cost ในที่นี้ต้องมีค่าน้อยกว่าค่า cost ของ S0 ถ้าไม่สามารถหาคำตอบได้ข้ามไปทำในขั้นตอนที่ 7 เพื่อหลีกจากการล็อค

ของคำตอบเฉพาะถิ่น ถ้าสามารถหาคำตอบได้ให้ทำตามขั้นตอนต่อไป นอกจากนี้ในขั้นตอนที่ 5 จะเก็บค่าที่ได้จากการสุ่ม 5 ครั้งล่าสุดไว้ใน neighbor_list เพื่อนำไปใช้ในส่วนของการย้อนรอยการค้นหา (back_tracking) ต่อไป ซึ่งเป็นส่วนที่ได้พัฒนาปรับปรุงขึ้น ให้การค้นหาแบบดั้งเดิมทำงานได้ดีขึ้นในการค้นหาคำตอบ (ในบทความนี้จะขอเรียก back_tracking ต่อจากนี้ไป รายละเอียดอยู่ในขั้นตอนที่ 7)



รูปที่ 8 แสดงแผนภูมิการทำงานของอัลกอริธึมที่อาศัยหลักการทำงานของวิธีการค้นหาแบบตาบู

ขั้นตอนที่ 6 กำหนดให้ S0 เท่ากับ best_neighbor ที่ได้จากขั้นตอนที่ 5 จากนั้นเริ่มทำในขั้นตอนที่ 3 ใหม่



รูปที่ 9 แสดงภาพแสดงการทำงานในช่วง back_tracking

ขั้นตอนที่ 7 จากการเฟ้นสุ่มในขั้นตอนที่ 5 ถ้าไม่มีสมาชิกใดๆที่ให้ค่า cost ดีกว่าค่า cost ของ S0 ให้ทำการกำหนดค่า S0=S3 โดยที่ค่า S3 คือค่าที่ได้จากการเฟ้นสุ่มในอดีต จัดเก็บอยู่ใน neighbor_list ดังที่ได้กล่าวไว้แล้ว การดำเนินการในขั้นตอนนี้เป็นการย้อนรอยการค้นหากลับไปยังค่าที่เคยค้นหามาในอดีต เพื่อจะเริ่มทำการค้นใหม่ในทิศทางที่แตกต่างไปจากเดิม สังเกตได้ว่าการย้อนรอยการค้นหาทำให้เกิดพื้นที่การค้นหาคำตอบใหม่ แสดงได้ดังรูปที่ 9 พื้นที่ดังกล่าวทำให้การดำเนินการค้นหาคำตอบ สามารถหลีกเลี่ยงจากการล็อคของคำตอบวงแคบเฉพาะถิ่นได้อย่างมีประสิทธิภาพ วิธีการนี้จะเรียกโดยรวมว่า back_tracking จากนั้นเมื่อทำตามขั้นตอนนี้เสร็จสิ้นกลับไปทำตามขั้นตอนที่ 3 ใหม่ ภาพรวมของการทำงานค้นหาคำตอบด้วยวิธีตาบูจากที่กล่าวมาข้างต้น แสดงได้ด้วยแผนภูมิในรูปที่ 8

ลักษณะการทำงานของวิธีการค้นหาแบบตาบูข้างต้น ยังมิได้กล่าวถึงเงื่อนไขการยุติการค้นหาคำตอบ โดยทั่วไปกระทำได้สองแนวทางคือ แนวทางแรก กำหนดวงรอบของการค้นหาคำตอบ (max_count) เช่น กำหนดว่าให้ค้นหา 3,000 รอบแล้วยุติ หรือแนวทางที่สอง กำหนดค่าตัวเลขจากฟังก์ชั่นวัตถุประสงค์ (μ) ซึ่งได้มาจากการคำนวณซ้ำๆ เพื่อศึกษาพลวัตของระบบ งานวิจัยนี้เลือกเงื่อนไขการยุติตามแนวทางที่สอง ซึ่งจะกล่าวรายละเอียดต่อไปในหัวข้อที่ 5 ดังนั้นการกำหนดค่า max_count ในแผนภูมิรูปที่ 8 จะกำหนดค่าไว้มากพอสมควรประมาณ 1,000 รอบ เพื่อให้การค้นหาคำตอบดำเนินไปเรื่อยๆจนได้ค่าตัวเลขจากฟังก์ชั่นวัตถุประสงค์ตามที่กำหนดไว้แต่อย่างไรก็ตามการพิจารณาเช่นนี้ต้องมั่นใจว่าอัลกอริธึมที่พัฒนาขึ้นมีความสามารถพอที่จะหลุดจากการล็อคของคำตอบที่เป็นวงแคบเฉพาะถิ่น การทดสอบอัลกอริธึมเพื่อให้เกิดความมั่นใจก่อนนำไปใช้งาน ได้รวบรวมไว้ในหัวข้อที่ 4

4. การทดสอบอัลกอริธึม

จากคำอธิบายหลักการทำงานของอัลกอริธึม แสดงให้เห็นว่าค่า radius และ number_neighbor เป็นค่าพารามิเตอร์ที่สำคัญที่ใช้ในการเฟ้นสุ่มคำตอบ ดังนั้นก่อนที่จะนำอัลกอริธึมมาใช้งานจำเป็นต้องทำการทดสอบอัลกอริธึม เพื่อให้ได้ค่า radius และ number_neighbor ที่เหมาะสม ขั้นตอนการทดสอบกระทำโดยการปรับเปลี่ยนค่าดังกล่าว ดังต่อไปนี้

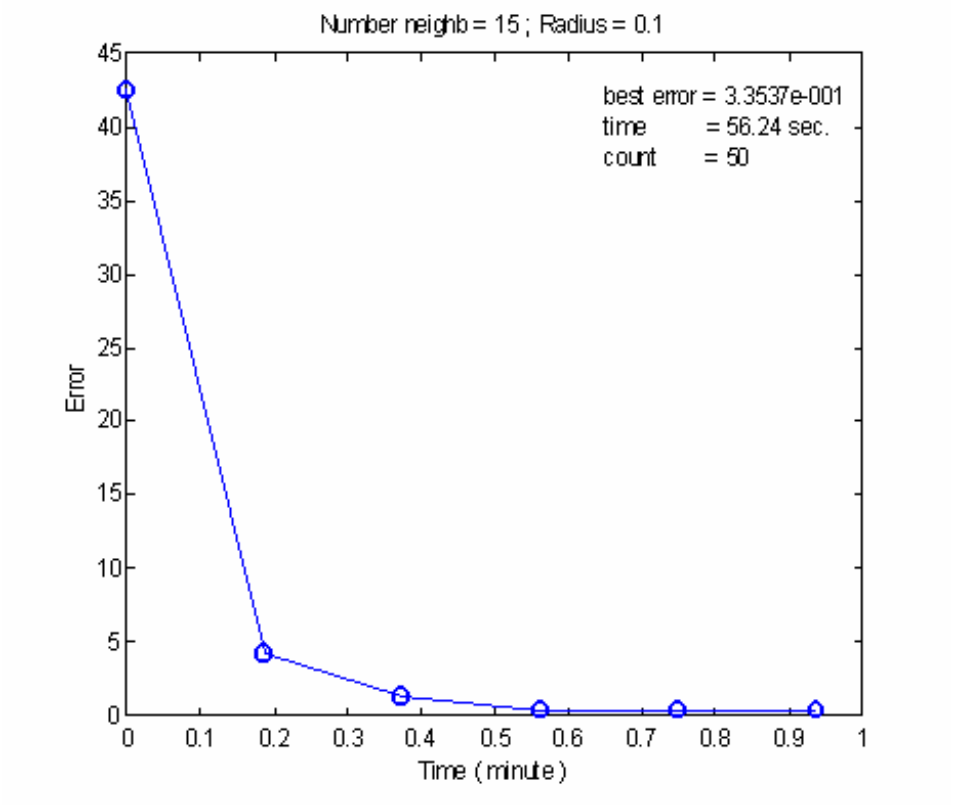
(1) number_neighbor = 15 radius = 0.10
(2) number_neighbor = 30 radius = 0.10
(3) number_neighbor = 50 radius = 0.10
(4) number_neighbor = 15 radius = 0.05
(5) number_neighbor = 15 radius = 0.20

การทดสอบในแต่ละกรณีจะทำการทดสอบกับระบบที่แสดงดังรูปที่ 16 ซึ่งเป็นรูปแบบ Lure's problem [7, 17] โดยกำหนดจุดเริ่มต้น (S0) ที่จุดเดียวกันและจะทำการค้นหาคำตอบไปเรื่อยๆจนกว่าค่าความคลาดเคลื่อนจะลดลงมาอยู่ที่ค่าน้อยกว่า 0.375 จึงจะยุติการค้นหาคำตอบ ในระหว่างการค้นหาจะทำการบันทึกผลและทำการพล็อตกราฟระหว่างค่าความคลาดเคลื่อนกับเวลาที่ใช้ในการค้นหาคำตอบ ซึ่งได้ผลการทดสอบดังที่แสดงไว้ในรูปที่ 10 ถึง 14

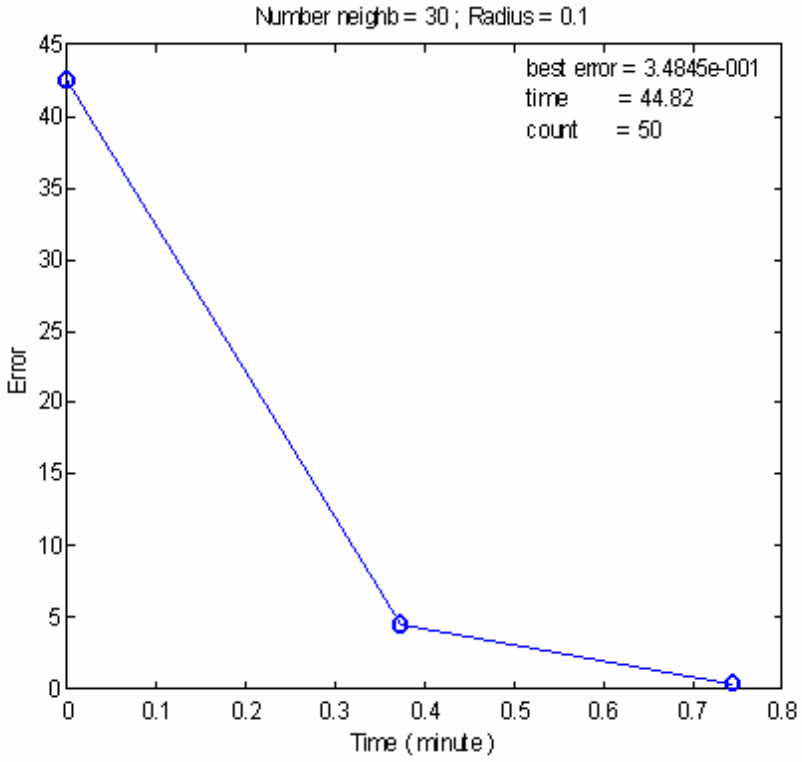
จากรูปที่ 10 ถึง 14 และตารางที่ 3 สรุปได้ว่าค่า number_neighbor = 30 และค่า radius = 0.1 จะให้คำตอบได้ดีที่สุด โดยใช้เวลา 44.82 วินาทีหรือทำการค้นหาคำตอบ 3 รอบ โดย Pentium III 733 MHz ดังนั้นงานวิจัยนี้จึงเลือกใช้ค่าดังกล่าวในการนำไปเฟ้นสุ่มคำตอบต่อไป อย่างไรก็ตามเพื่อเป็นการยืนยันว่าอัลกอริธึมที่ได้พัฒนาขึ้น โดยอาศัยหลักการวิธีการค้นหาแบบตาบู และ back_tracking สามารถค้นหาคำตอบที่ใกล้ความเป็นวงกว้างได้ จึงทำการทดสอบโดยเลือกค่า number_neighbor และค่า radius ที่ได้จากการทดสอบ ทำการค้นหาคำตอบเป็นจำนวน 50 รอบ และทำการบันทึกแนวโน้มการลดลงของค่าความคลาดเคลื่อนเทียบกับเวลา ผลของการทดสอบแสดงดังรูปที่ 15

รูปที่ 15 แสดงให้เห็นว่าอัลกอริธึมที่ได้พัฒนาขึ้นสามารถที่จะค้นหาคำตอบที่ใกล้ความเป็นวงกว้างได้ เนื่องจากผลการทดสอบสามารถหลุดจากจุดคำตอบวงแคบเฉพาะถิ่นได้ถึง 2 ครั้งในช่วงการค้นหาเพียง 50 รอบ

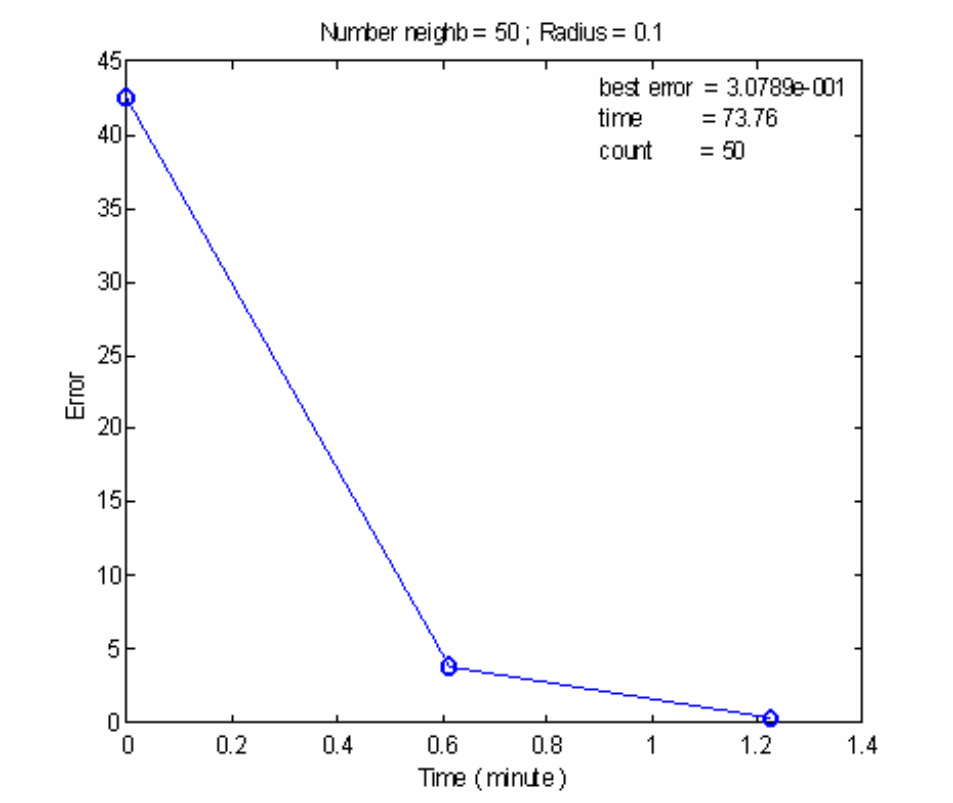
จากที่กล่าวมาข้างต้น ขั้นตอนการทดสอบอัลกอริธึมถือว่าเป็นสิ่งสำคัญในการที่จะนำวิธีการทางปัญญาประดิษฐ์มาประยุกต์ใช้งานเกี่ยวกับการหาคำตอบที่ดีที่สุด เพื่อเป็นการตรวจสอบอัลกอริธึมให้เกิดความมั่นใจได้ว่า คำตอบที่ได้จากการค้นหาจะเป็นคำตอบที่สุด ก่อนที่จะนำอัลกอริธึมดังกล่าวไปใช้งานจริง



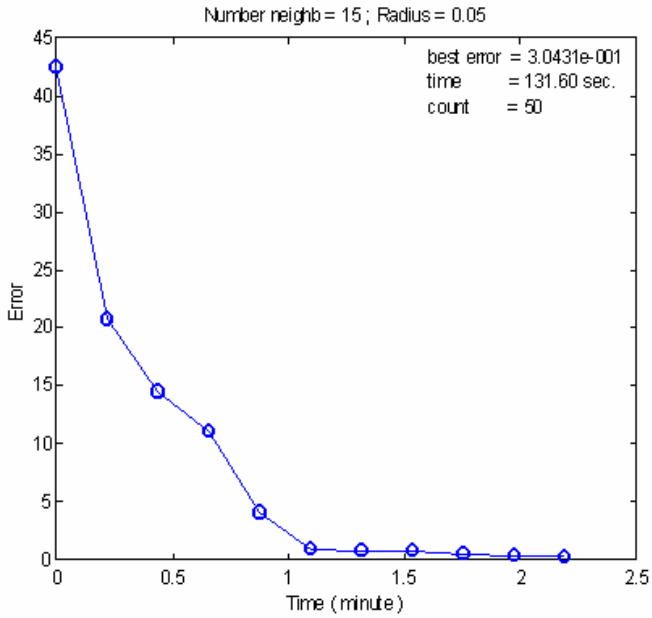
รูปที่ 10. ผลการทดสอบเมื่อ number_neighbor = 15 radius = 0.10



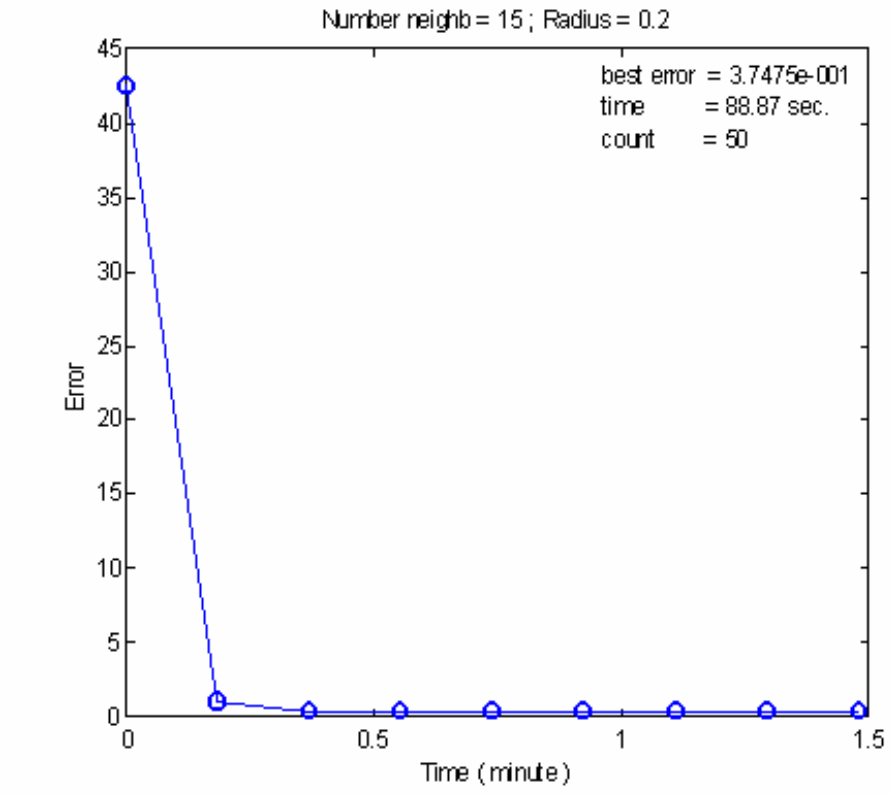
รูปที่ 11 แสดงผลการทดสอบเมื่อ number_neighbor = 30 radius = 0.10



รูปที่ 12 แสดง ผลการทดสอบเมื่อ number_neighbor = 50 radius = 0.10



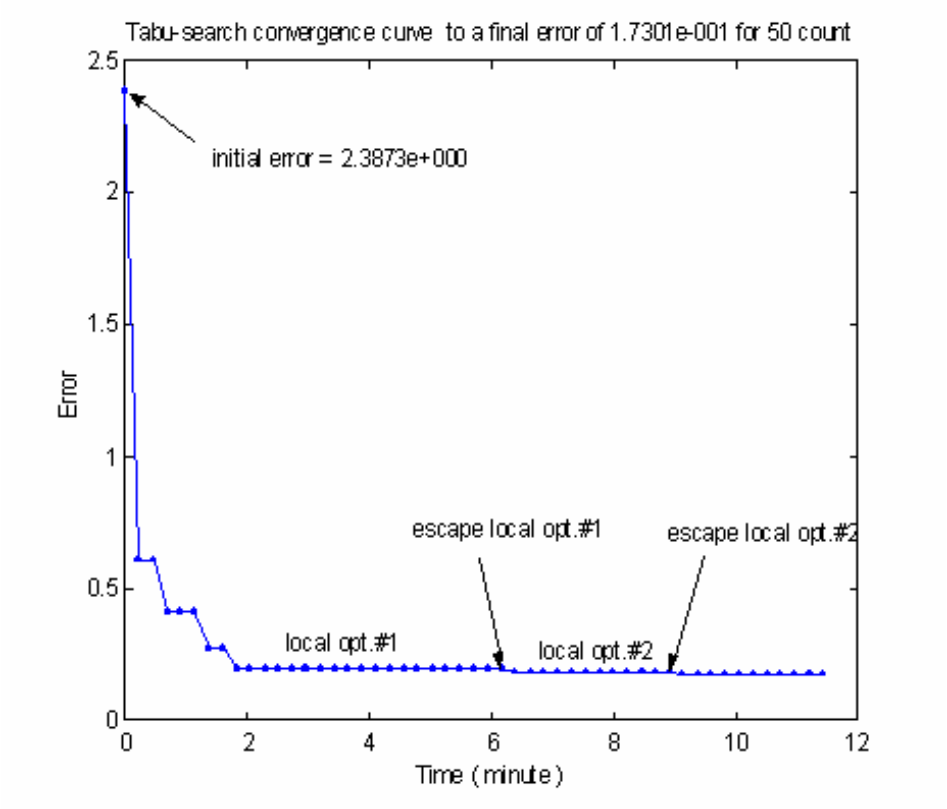
รูปที่ 13 แสดงผลการทดสอบเมื่อ number_neighbor = 15 radius = 0.05



รูปที่ 14. ผลการทดสอบเมื่อ number_neighbor = 15 radius = 0.20

ตารางที่ 3 แสดงผลการทดสอบอัลกอริธึมกรณีต่างๆ

Count	N = 15, R = 0.05	N = 15, R = 0.1	N = 15, R = 0.2	N = 30, R = 0.1	N = 50, R = 0.1
0	4.2555e+001	4.2555e+001	4.2555e+001	4.2555e+001	4.2555e+001
1	2.0674e+001	4.2861e+001	1.1148e+000	4.4967e+000	3.8186e+000
2	1.4575e+001	1.3020e+000	4.3803e-001	3.4845e-001	3.0789e-001
3	1.1119e+001	4.3888e-001	#1	-	-
4	4.0259e+000	#1	#2	-	-
5	9.7949e-001	3.3537e-001	4.1335e-001	-	-
6	7.9695e-001	-	#1	-	-
7	#1	-	#2	-	-
8	4.6597e-001	-	3.7475e-001	-	-
9	3.9030e-001	-	-	-	-
10	3.0431e-001	-	-	-	-
Time (sec.)	131.60	56.24	88.87	44.82 O.K.	73.76

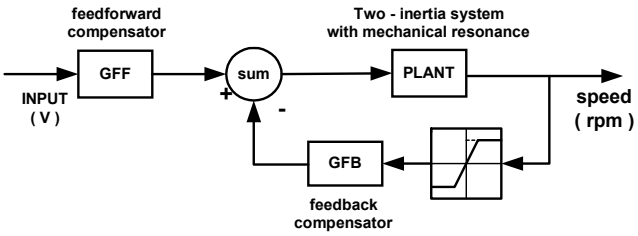


รูปที่ 15 แสดงความสามารถของอัลกอริธึมในการค้นหาคำตอบ

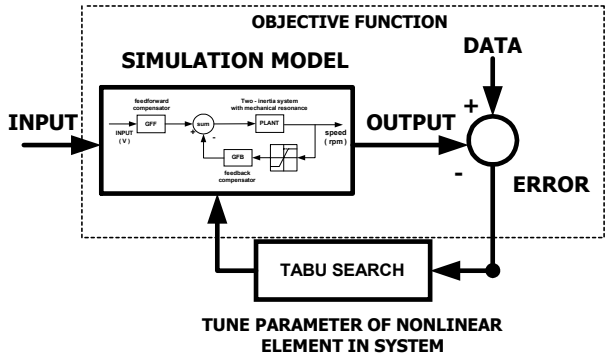
5. การทดลอง

5.1 วิธีการทดลอง

จากผลการทดสอบระบบสองมวลในหัวข้อที่ 2 สามารถตั้งสมมุติฐานได้ว่า ลักษณะสมบัติไม่เป็นเชิงเส้นจะเป็นแบบอิ่มตัว โดยถือว่าตำแหน่งของลักษณะสมบัติไม่เป็นเชิงเส้นจะอยู่ที่ส่วนของสัญญาณป้อนกลับเพียงแค่จุดเดียวตามรูปแบบของ Lure's problem หมายความว่าได้สมมุติให้เป็นลักษณะสมบัติไม่เป็นเชิงเส้นสมมูล (equivalent nonlinearity) ซึ่งแทนความไม่เป็นเชิงเส้นของระบบทั้งหมดไว้ที่ตำแหน่งนี้ตำแหน่งเดียว ดังรูปที่ 16



รูปที่ 16 แสดงสมมุติฐานที่ใช้ในการค้นหาคำตอบ



รูปที่ 17 แสดงแผนภาพแสดงวิธีการทดลอง

รูปที่ 17 ใช้อธิบายการค้นหาพารามิเตอร์ของลักษณะสมบัติไม่เป็นเชิงเส้นซึ่งมีขั้นตอนการดำเนินงานดังนี้ คือ เริ่มจากป้อนอินพุตขั้นบันไดให้กับระบบที่ไม่เป็นเชิงเส้น มีขนาด 1.1, 1.3, 1.5, 1.7, 1.9 และ 2.1 โวลต์ ตามลำดับ จากนั้นนำผลที่เป็นค่าเชิงเลขที่ได้จากการจำลองสถานการณ์ (simulation model) มาเปรียบเทียบกับผลการทดสอบที่ได้บันทึกไว้ และ

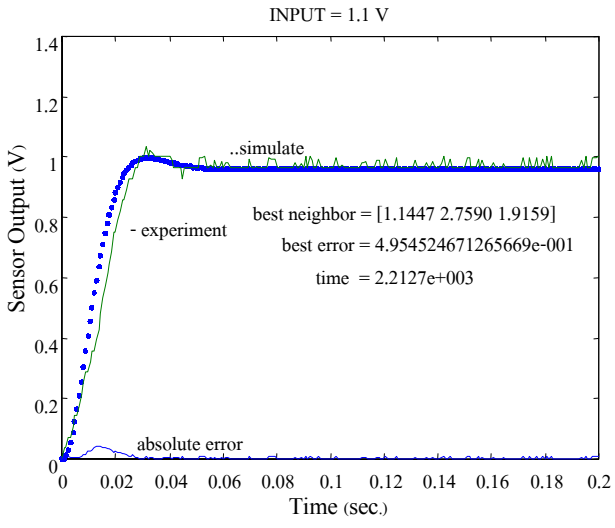
ประเมินค่าความคลาดเคลื่อนที่อาจแสดงได้ด้วยฟังก์ชั่นวัตถุประสงค์ ดังสมการที่ (4)

$$J = \sum e^2(kT) \quad , kT = 0,1,2,... \qquad (4)$$

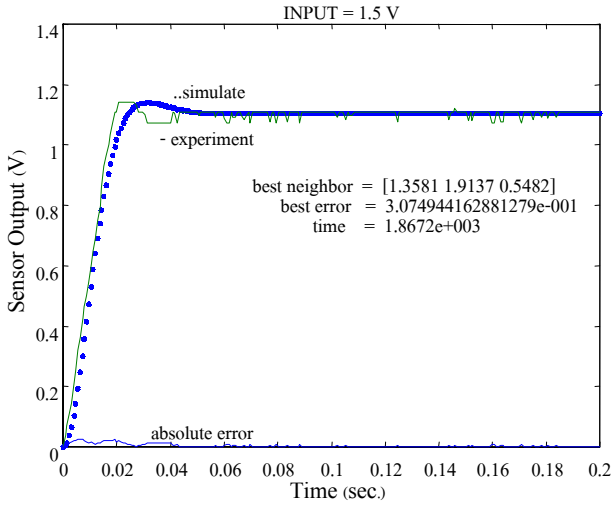
ซึ่ง e(kT) เป็นผลต่างระหว่างความเร็วรอบที่วัดได้โดยเซนเซอร์มีหน่วยเป็นโวลต์กับความเร็วรอบที่ได้จากการจำลองสถานการณ์ ค่าความคลาดเคลื่อนที่ได้มาเป็นตัวกำหนดทิศทางการค้นหาแบบตาบู ว่าจะมีทิศทางไปทางใด เราจะยอมรับผลการค้นหาว่าแม่นยำพอและยุติการค้นหาก็ต่อเมื่อ $J_{MIN} \leq \mu$ ค่า μ ในที่นี้เท่ากับ 0.5 ซึ่งได้มาจากการคำนวณซ้ำๆ เพื่อศึกษาพลวัตของระบบ การจำลองสถานการณ์ในรูปที่ 17 ดำเนินการในโดเมนเวลา ด้วยสมการดิฟเฟอเรนซ์ ที่ได้จากการแปลงแบบจำลองต่อเนื่องด้วยเทคนิคไบลิเนียร์ โปรแกรมจำลองสถานการณ์สร้างขึ้นด้วย MATLAB™

5.2 ผลการทดลอง

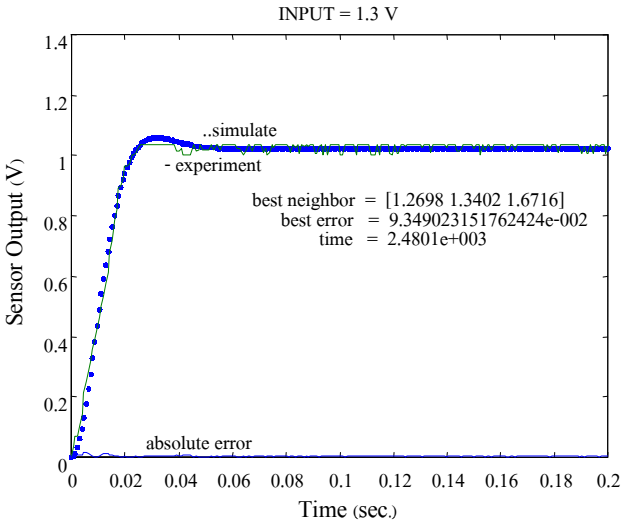
ผลการทดลองที่ได้จากวิธีการค้นหาแบบตาบู พบว่าแบบจำลองที่ได้สามารถอธิบายพฤติกรรมของระบบได้สมจริง เนื่องจากผลที่ได้จากการจำลองสถานการณ์ของระบบดังแผนภาพในรูปที่ 16 มีความใกล้เคียงกับผลการทดสอบจริงเป็นอย่างมาก ซึ่งอาจสังเกตได้ถึงความใกล้เคียงกันของความเร็วในมวลหมุนที่บันทึกได้กับผลการการจำลองสถานการณ์ดังแสดงในรูปที่ 18 ถึง 20



รูปที่ 18 แสดงผลการค้นหาแบบตาบูที่อินพุต 1.1 โวลต์



รูปที่ 19 แสดงผลการค้นหาแบบตาบูที่อินพุต 1.3 โวลต์

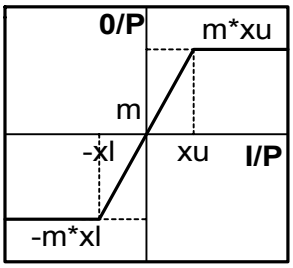


รูปที่ 20 แสดงผลการค้นหาแบบตาบูที่อินพุต 1.5 โวลต์

ตารางที่ 4 แสดงค่าพารามิเตอร์ที่ได้จากการค้นหาแบบตาบู

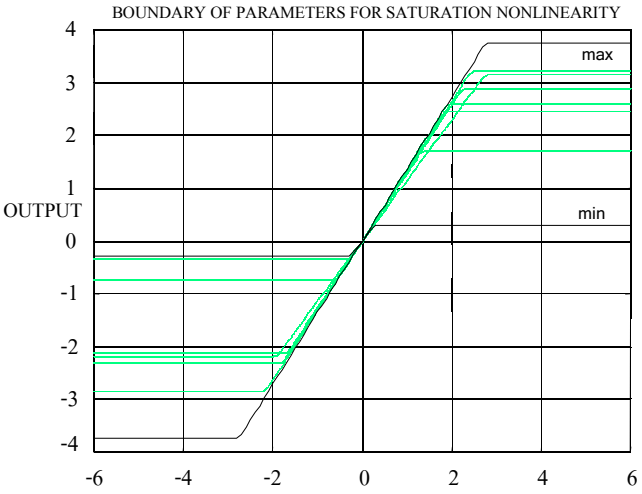
อินพุต (โวลต์)	ค่าพารามิเตอร์		
	m	xu	xl
1.1	1.1447	2.7590	1.9159
1.3	1.2698	1.3402	1.6716
1.5	1.3581	1.9137	0.5482
1.7	1.3128	2.4456	2.1646
1.9	1.3018	1.8827	0.2553
2.1	1.2849	2.2356	1.7925

ผลจากการค้นหาในรูป ค่าพารามิเตอร์ต่างๆ ของลักษณะสมบัติไม่เป็นเชิงเส้นชนิดอิ่มตัวได้รับการรวบรวมไว้ในตารางที่ 4 ซึ่งอาจทำความเข้าใจโดยดูรูปที่ 21 ประกอบได้ดังนี้



รูปที่ 21 แสดงค่าพารามิเตอร์ที่ใช้ในการค้นหาคำตอบ

ค่า m คือค่าความชัน
ค่า xu คือจุดที่เริ่มเกิดการอิ่มตัวทางด้านบวก
ค่า xl คือจุดที่เริ่มเกิดการอิ่มตัวทางด้านลบ



รูปที่ 22 แสดงพารามิเตอร์ที่ได้จากวิธีการค้นหาแบบตาบู ที่ระดับอินพุตต่างๆ

ผลการทดลองที่ได้จากการค้นหาแบบตาบู มีค่าความคลาดเคลื่อนน้อยมาก เมื่อนำค่าความคลาดเคลื่อนกรณีที่ระบบยังไม่มีลักษณะสมบัติไม่เป็นเชิงเส้นมาเปรียบเทียบกับระบบที่มีลักษณะสมบัติไม่เป็นเชิงเส้นพบว่าค่าความคลาดเคลื่อนมีค่าลดลงประมาณ 89-99 % ดังรายละเอียดในตารางที่ 5

ตารางที่ 5 แสดงเปรียบเทียบค่าความคลาดเคลื่อนระหว่างระบบเชิงเส้นกับระบบที่ไม่เป็นเชิงเส้น

อินพุต (โวลต์)	ระบบเชิงเส้น	ระบบไม่เชิงเส้น	%ลดลงของค่าความคลาดเคลื่อน
1.1	4.9457	0.4955	89.9822
1.3	17.8222	0.0935	99.4800
1.5	36.8377	0.3075	99.1653
1.7	40.5674	1.9654	95.1552
1.9	48.4605	3.1829	93.4605
2.1	54.9896	4.2706	92.2339

6. สรุป

จากผลการวิจัยที่ได้นำเสนอ พบว่าวิธีการค้นหาแบบตาบูให้ผลที่มีความถูกต้องสูงและใช้เวลาในการค้นหาคำตอบรวดเร็ว เมื่อนำผลที่ได้เปรียบเทียบกับผลการจำลองสถานการณ์ของระบบที่เป็นเชิงเส้นพบว่ามีค่าความคลาดเคลื่อนลดลงเป็นอย่างมาก ดังตารางที่ 5 ดังนั้นจึงสรุปได้ว่าการขยายย่านการทำงานของระบบสองมวลความเฉื่อยจะต้องพิจารณาลักษณะสมบัติไม่เป็นเชิงเส้น ซึ่งปรากฏอยู่ในรูปลักษณะสมบัติไม่เป็นเชิงเส้นอิ่มตัว ตรงส่วนของระบบวิถีป้อนกลับดังรูปที่ 16 เป็นไปตามรูปแบบของ Lure's problemเมื่อพิจารณาค่าพารามิเตอร์ของลักษณะสมบัติไม่เป็นเชิงเส้น ค่าพารามิเตอร์ดังกล่าวในแต่ละอินพุตไม่สามารถที่จะรวมเป็นพารามิเตอร์ชุดเดียวได้ เนื่องจากระบบมีความไม่เป็นเชิงเส้นค่อนข้างสูง

สำหรับงานวิจัยในอนาคตจะได้นำแบบจำลองที่ได้ไปใช้เพื่อการวิเคราะห์สมรรถนะและเสถียรภาพของระบบ แบบจำลองความไม่เป็นเชิงเส้นปรากฏเป็นตระกูล เพื่อการวิเคราะห์เสถียรภาพจึงกำหนดขอบเขตล้อมรอบกลุ่มของลักษณะสมบัติไม่เป็นเชิงเส้นเป็นขอบเขตบนและขอบเขตล่าง ดังรูปที่ 22 โดยอาศัยข้อมูลที่ได้จากการค้นหาแบบตาบู การวิเคราะห์เสถียรภาพของระบบสามารถทำได้โดยอาศัยวิธีการทางโดเมนความถี่ด้วยวิธีฟังก์ชั่นพรรณนา (Describing function) เกณฑ์ของโพพอฟ (Popov's criterion) และ เกณฑ์วงกลม (Circle criterion) เพื่อเปรียบเทียบผลและให้มั่นใจในเสถียรภาพของระบบขยายย่านการทำงาน

เอกสารอ้างอิง

[1] ชัชชัย อุทัยวสิน. (2543). การกำจัดรีโซแนนซ์การบิดในระบบ 2 มวล โดยใช้เทคนิคการกำหนดตำแหน่งโพล-ซีโร. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศกรรมไฟฟ้า บัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง.

[2] A. H. Mantawy, Y. L. Abdel-Magid, and S. Z. Selim. (1998). Unit Commitment by Tabu Search. IEEE Transaction on Distrib. 145(1):56-64.

[3] A. Kaplan, S. Ozer, and S. Sagiroglu. (1998). Membership Function Optimuization of a Fuzzy Controller using Modified Tabu Search Algorithm. IEEE Transaction on Industrial Electronics. 98(7):64-67.

[4] C. H. Houpis, and G. B.Lamont. (1985). Digital Control System. New York:McGraw–Hill.

[5] D. P.Atherton. (1982). Nonlinear Control Engineering. Student edition. New York: Van Nostrand Reinhold.

[6] H.Hirata, et.al. (1995). Speed Control of DC Motor with Torisional Oscillation and Load Fluctuation. Proc. of Sch. Eng, Tokai University. 35(3):31-41.

[7] H. K.Khalil. (1996). Nonlinear System. Second edition . London :Prentice Hall.

[8] J A Bland G P Dawson. (1991). Tabu Search and Design Optimization. IEEE Transaction on Industrial Electronics. 23 (3):195-201.

[9] J.K. Ji, and S.K.Sul. (1995). Kalman Filter and LQ Based on Controller for Torsional Vibration Suppression in a 2-mass Motor Drive System. IEEE Trans. On IE. 42(6):564-571.

[10] J. K . Ji , et.al. (1993). LQG Based Speed Controller for Torsional Vibration Suppression in 2 – Mass System. Proc. IEEE IECON'93. pp:1157-1162.

[11] J. Ngamwiwit , C. U-thaiwasin , Y. Prempraneerach and S. Sujitjorn. (2000). Torsional Resonance Suppression via PIDA Controller. Proc. IEEE Conf. On Artificial Intelligence & Robotics – TENCON 2000. Kuala Lumpur: Malaysia.

[12] K. Fujikawn, et.al. (1991). Robust and Fast Speed Control for Torsional System Based on State – Space Method. Proc. IEEE IECON'91. pp:687-692

[13] M. Hassul. (1993). Control System Design Using MATLAB. London:Prentice-Hall.

[14] S.H. Song, et.al. (1993). Torsional Vibration Suppression Control in 2 - Mass System by State Feedback Speed Controller. Proc. IEEE CCA'93. pp:129-134.

[15] S. Sujitjorn, C. U-Thaiwasin , and Y.Prempraneerat. (2000). Torsional Resonance Suppression Via Pole – Zero Assignment. Proc. 19th IASTED Int. Conf. On Modelling, Identification, and Control. Innsbruck:Austria.

[16] S. Sujitjorn , and T. Kulworawanichpong. (2001). Optimal Power Flow Using Tabu Search. IEEE Power Engineering Review (submitted)

[17] T. Glad , and L. Ljung. (2000). Control Theory : Multivariable and Nonlinear Methods. London :Taylor & Francis

กองพัน อารีรักษ์ สำเร็จปริญญาตรีในสาขาวิชาวิศวกรรมไฟฟ้า จากมหาวิทยาลัยเทคโนโลยีสุรนารี เมื่อ พ.ศ.2543 ปัจจุบันกำลังศึกษาระดับปริญญาโทที่มหาวิทยาลัยเทคโนโลยี สุรนารี และเป็นผู้ช่วยวิจัยทางด้านระบบควบคุมคงทน โดยได้รับทุนอุดหนุนวิจัยประจำปี 2544 จากสำนักงานคณะกรรมการวิจัยแห่งชาติ

นาวาอากาศโท สราวุฒิ สุจิตจร สำเร็จปริญญาตรีและปริญญาเอกในสาขาวิชาวิศวกรรมไฟฟ้า จากโรงเรียนนายเรืออากาศ และมหาวิทยาลัยเบอร์มิงแฮม ประเทศอังกฤษ เมื่อ พ.ศ.2527 และ 2530 ตามลำดับ ปัจจุบันดำรงตำแหน่งรองศาสตราจารย์และหัวหน้าสาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ดำเนินงานวิจัยทางด้านระบบควบคุมเชิงเส้นและไม่เชิงเส้น การดำเนินกระบวนการทางสัญญาณ การอนุรักษ์พลังงาน และการประยุกต์เทคนิคทางปัญญาประดิษฐ์ อาจารย์สราวุฒิเป็นสมาชิกวิศวกรรมสถานแห่งประเทศไทยฯ สมาคมเทคโนโลยีที่เหมาะสม และ IEEE อีกทั้งได้รับการจารึกชื่อไว้ใน Who's Who in the World และ Who's Who in Science and Engineering.

อาทิตย์ ศรีแก้ว สำเร็จการศึกษาระดับปริญญาตรีในสาขาวิชาวิศวกรรมอิเล็กทรอนิกส์จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เมื่อ พ.ศ. 2537 สำเร็จการศึกษาระดับปริญญาโทและปริญญาเอกสาขาวิศวกรรมไฟฟ้าจาก Vanderbilt University ประเทศสหรัฐอเมริกา เมื่อ พ.ศ. 2540 และ 2543 ตามลำดับ ปัจจุบันดำรงตำแหน่งเป็นอาจารย์ประจำสาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี มีความสนใจงานวิจัยทางด้าน computer and robot vision, image processing, neural networks, artificial Intelligence และ intelligent system

โยธิน เปรมปราณีรัชต์ สำเร็จการศึกษาปริญญาเอกทางวิศวกรรมระบบควบคุม จากมหาวิทยาลัย Nihon ประเทศญี่ปุ่น ปัจจุบันดำรงตำแหน่ง รองศาสตราจารย์ ภาควิชาวิศวกรรมระบบควบคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และเป็นผู้อำนวยการศูนย์ทดสอบผลิตภัณฑ์ไฟฟ้าและอิเล็กทรอนิกส์ (ศทอ.) อาจารย์โยธินฯ สอนและวิจัยทางด้านระบบควบคุม การควบคุมมอเตอร์แบบต่างๆ มากว่า 20 ปี

การจำลองผลระบบพลังงานแสงอาทิตย์

Simulation of a Solar Energy System

เผด็จ เผ่าละออ [1] สราวุฒิ สุจิตจร [2]
[1] ผู้ช่วยวิจัย [2] รองศาสตราจารย์
สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ABSTRACT – This article reports our developed simulation program for a solar energy system. The mathematical models of various components are described. These include solar cells, batteries, motor, and helical pump. The motor and pump is a coupled load. Solar cells are weather dependent sources. Batteries act in two modes of either sources or loads. Due to the component's nonlinearity and insolation characteristic, the simulation program is necessary and useful for energy studies in such a system.

KEY WORDS – solar energy, photovoltaic, simulation

บทคัดย่อ – บทความนี้นำเสนอการจำลองผลระบบพลังงานแสงอาทิตย์ พร้อมรายละเอียดแบบจำลองทางคณิตศาสตร์ของส่วนประกอบต่างๆ ได้แก่ แผงเซลล์แสงอาทิตย์ต่อคู่ควบกับแบตเตอรี่ทำหน้าที่เป็นแหล่งพลังงาน จ่ายให้มอเตอร์และปั๊มพ์หอยโข่งที่เป็นโหลดของระบบ แบตเตอรี่บางขณะเป็นแหล่งพลังงานบางขณะเป็นโหลด แผงเซลล์แสงอาทิตย์เป็นแหล่งพลังงานที่ขึ้นกับสภาพอากาศ เนื่องจากความไม่เป็นเชิงเส้นของส่วนประกอบต่างๆ รวมจนถึงความเข้มแสงอาทิตย์ การมีโปรแกรมจำลองผลจึงจำเป็นและจะมีประโยชน์ต่อการศึกษาด้านพลังงานในระบบเซลล์แสงอาทิตย์

คำสำคัญ – พลังงานแสงอาทิตย์, โฟโตโวลตาอิค, การจำลองผล

1. บทนำ

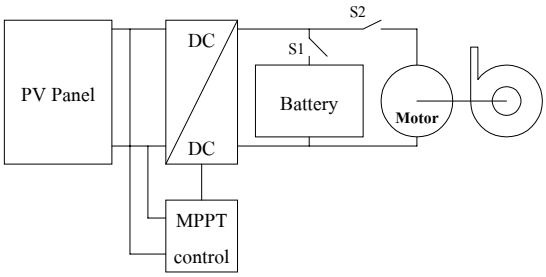
ประเทศไทยอยู่ในภูมิภาคที่ได้รับแสงอาทิตย์ค่อนข้างคงที่ ตลอดทั้งปี นับว่าดีอย่างยิ่งที่จะนำพลังงานจากแสงอาทิตย์ ซึ่งเป็นพลังงานสะอาดมาใช้ประโยชน์ การแปลงพลังงานแสงอาทิตย์ให้เป็นไฟฟ้า อาศัยเซลล์แสงอาทิตย์ที่อยู่ในรูปของแผงเซลล์แสงอาทิตย์ ซึ่งมีลักษณะการใช้งานไม่ง่ายนัก เพราะให้เอาต์พุตที่ไม่เป็นเชิงเส้น ผันแปรไปตามโหลด ปริมาณแสงที่กระทบและอุณหภูมิในสภาพแวดล้อม นอกจากนั้น ยังมีประสิทธิภาพต่ำและราคาแพง แต่เนื่องจากในปัจจุบันแผงเซลล์แสงอาทิตย์เป็นเพียงทางเลือกเดียว สำหรับการแปลงพลังงานแสงอาทิตย์ไปเป็นพลังงานไฟฟ้าเพื่อใช้ประโยชน์ การใช้แผงเซลล์แสงอาทิตย์จึงต้องให้ได้คุ้มค่าต่อการลงทุน

งานวิจัยที่นำเสนอในบทความนี้ สนใจระบบพลังงานแสงอาทิตย์ที่สามารถปฏิบัติงานในพื้นที่ทุรกันดารที่จำเป็น เช่น สถานพยาบาล สถานีอนามัย ที่อาจมีเหตุการณ์ฉุกเฉิน หรือแม้กระทั่งแหล่งท่องเที่ยวตามอุทยานต่างๆ การที่จะบรรลุวัตถุประสงค์ดังกล่าว จำเป็นอย่างยิ่งที่จะต้องมีแหล่งพลังงานสำรองให้แก่แผงเซลล์แสงอาทิตย์ เมื่อพิจารณาถึงความต้องการระบบดังกล่าวสำหรับประเทศกำลังพัฒนา แหล่งพลังงานที่น่าเชื่อถือและราคาถูก คงเป็นแบตเตอรี่รถยนต์ ซึ่งก็จะต้องยอมรับในความยุ่งยากด้านการบำรุงรักษา และลักษณะสมบัติที่ไม่เป็นเชิงเส้นอย่างมากของแบตเตอรี่ การจะทำความเข้าใจให้ได้ถ่องแท้ถึงพลวัตของระบบที่ไม่เป็นเชิงเส้นเช่นนี้ คงต้องอาศัยความรู้ด้านแบบจำลองทางคณิตศาสตร์และการจำลองผล บทความจึงกล่าวถึงรายละเอียดแบบจำลองของอุปกรณ์ต่างๆ ที่ต่อควบกันดังรูปที่ 1 และอธิบายถึงโครงสร้างของโปรแกรมจำลองผลตลอดจนแสดงผลที่ได้จากการใช้งานโปรแกรม ซึ่งพัฒนาขึ้นด้วย MATLAB™

2. แบบจำลองทางคณิตศาสตร์ของอุปกรณ์ในระบบเซลล์แสงอาทิตย์

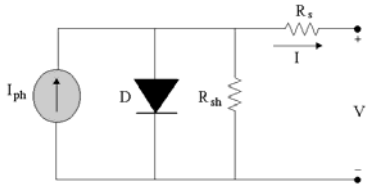
ระบบเซลล์แสงอาทิตย์มีองค์ประกอบดังรูปที่ 1 มอเตอร์ขนาด 2.5 hp พร้อมปั๊มพ์หอยโข่งเป็นโหลดที่ต้องใช้งานทั้งช่วงเวลากลางวันและช่วงเวลากลางคืน โดยสมมติให้ใช้งานในช่วงเวลากลางวันที่มีแสงอาทิตย์ทั้งหมด 8 ชั่วโมง (08.00-16.00น.) และใช้งานในช่วงเวลากลางคืนที่ไม่มี

แสงอีก 6 ชั่วโมง ซึ่งต้องอาศัยแหล่งจ่ายพลังงานจำนวนมาก เป็นแบตเตอรี่ตะกั่ว-กรดซึ่งใช้เป็นแหล่งพลังงานสำรองในช่วงเวลากลางคืนและกลางวันที่ฟ้าปิด ในบางโอกาสแบตเตอรี่เหล่านี้ก็จะเป็นโหลดของแผงเซลล์แสงอาทิตย์กล่าวคือในช่วงชาร์จ แผงเซลล์แสงอาทิตย์เป็นแหล่งพลังงานหลักที่ขึ้นอยู่กับสภาพอากาศ โดยแผงเซลล์แสงอาทิตย์ต้องมีจำนวนเพียงพอที่จะผลิตพลังงานไฟฟ้าเพื่อนำไปชาร์จแบตเตอรี่พร้อมทั้งจ่ายให้แก่มอเตอร์ภายในช่วงระยะเวลาที่มีแสง 8 ชั่วโมง ส่วนช่วงระยะเวลาที่ไม่มีแสง ก็เป็นหน้าที่ของแบตเตอรี่ที่ต้องดิสชาร์จพลังงานไฟฟ้าให้แก่มอเตอร์ โดยแบตเตอรี่จะต้องมีจำนวนเพียงพอที่จะดิสชาร์จได้ตลอดช่วงระยะเวลาที่ไม่มีแสงทั้ง 6 ชั่งโมงนี้ด้วย จากการพิจารณาระบบเซลล์แสงอาทิตย์ในงานวิจัยนี้ จึงประกอบด้วยเซลล์แสงอาทิตย์ทั้งหมด 96 โมดูล ต่อเป็นแผงขนานกัน 12 แผง ซึ่งแต่ละแผงมีเซลล์แสงอาทิตย์ 8 โมดูลต่ออนุกรมกันอยู่ และประกอบด้วยแบตเตอรี่จำนวนทั้งหมด 20 ลูก ต่อเป็นแผงขนานกัน 2 แผง ซึ่งแต่ละแผงมีแบตเตอรี่ 10 ลูกต่ออนุกรมกันอยู่ การใช้งานแผงเซลล์แสงอาทิตย์อย่างคุ้มค่าต้องพึ่งพาชุดควบคุมตามรอยกำลังงานสูงสุด (MPPT control) และดีซี/ดีซี คอนเวอร์เตอร์ เพื่อให้กำลังไฟฟ้าออกมาสูงสุด องค์ประกอบต่างๆ ของระบบมีแบบจำลองทางคณิตศาสตร์ดังที่จะกล่าวถึงในหัวข้อต่อไป



รูปที่ 1. ระบบเซลล์แสงอาทิตย์

2.1 แบบจำลองแผงเซลล์แสงอาทิตย์



รูปที่ 2. วงจรสมมูลของเซลล์แสงอาทิตย์

แผงเซลล์แสงอาทิตย์หมายถึง อุปกรณ์ผลิตพลังงานไฟฟ้าด้วยเซลล์แสงอาทิตย์จำนวนมาก ที่ได้รับการสร้างประกอบขึ้นเป็นแผง วงจรสมมูลของเซลล์แสงอาทิตย์ [1] แสดงได้ดังรูปที่ 2 ปกติความต้านทานขนานรอยต่อ p-n (R_{sh}) มีขนาดใหญ่มาก และความต้านทานในเนื้อสารกึ่งตัวนำและจุดเชื่อมต่อ (R_s) มีค่าน้อยมาก แบบจำลองทางคณิตศาสตร์ที่แสดงความสัมพันธ์ของกระแสและแรงดันของแผงเซลล์แสงอาทิตย์ จึงเขียนแสดงได้ดังนี้

$$I = n_p I_{ph} - n_p I_{rs}\left[\exp\left(\frac{qV}{kTAn_s}\right) - 1\right] \qquad (1)$$

ซึ่ง I_{ph} คือ กระแสโฟโต (A)

I_{rs} คือ กระแสอิ่มตัวย้อนกลับ (A)

n_p คือ จำนวนโมดูลที่ต่อขนานกัน

n_s คือ จำนวนเซลล์ที่ต่ออนุกรมกัน

q คือ ค่าประจุอิเล็กตรอน ($1.6\text{x}10^{-19}$ C)

k คือ ค่าคงที่โบลทซ์มานน์ ($1.38\text{x}10^{-23}$ J/K)

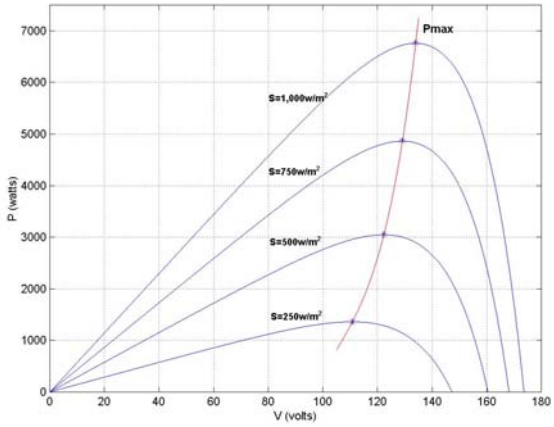
A คือ ค่าคงที่ตัวประกอบของรอยต่อ p-n

T คือ อุณหภูมิของเซลล์ (K)

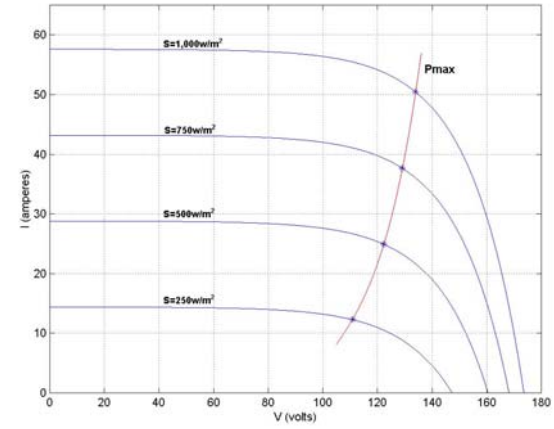
ข้อมูลในการจำลองผล ใช้แผงเซลล์แสงอาทิตย์ของซีเมนส์ SP-75W [2] โมดูลหนึ่งประกอบด้วยเซลล์ทั้งหมด 36 เซลล์ต่ออนุกรมกัน มีค่าพารามิเตอร์ต่าง ๆ ดังนี้

- ความเข้มแสง 1000 W/m^2(peak power)
- อุณหภูมิ 38^{0}C (311 K)
- A=2.46 (มีค่าอยู่ระหว่าง 1-5) [1]
- กำลังไฟฟ้าพิกัด 75 W; กระแสพิกัด 4.4 A
- แรงดันพิกัด 14–17 V; I_{ph}=4.8 A
- แรงดันขณะเปิดวงจร 21.7 V; $R_s \approx 0\ \Omega$

กรณีแผงเซลล์แสงอาทิตย์ SP-75W นี้ ค่า n_s=36 ขณะที่ n_p=1 จากสมการที่ (1) เมื่อแผงเซลล์แสงอาทิตย์ไม่มีโหลด คำนวณได้ว่า I_{rs}=5.1768E-4 A โดยสมมติให้อุณหภูมิของเซลล์แสงอาทิตย์คงที่ตลอด ในการใช้งานให้บรรลุตามสมมติฐานที่ตั้งไว้ จะต้องใช้เซลล์แสงอาทิตย์ทั้งหมด 96 โมดูล โดยแบ่งเป็นจำนวนโมดูลที่อนุกรมในแต่ละแผงเท่ากับ 8 (n_s=8*36) และจำนวนโมดูลที่ต่อขนานกันของเซลล์แสงอาทิตย์เท่ากับ 12 (n_p=12)



รูปที่ 3. ความสัมพันธ์ระหว่างแรงดันกับกำลังไฟฟ้าของแผงเซลล์แสงอาทิตย์



รูปที่ 4. ความสัมพันธ์ระหว่างแรงดันกับกระแสไฟฟ้าของแผงเซลล์แสงอาทิตย์

รูปที่ 3 และ 4 แสดงความสัมพันธ์ระหว่างกำลังไฟฟ้า แรงดัน และกระแสที่ได้จากแผงเซลล์แสงอาทิตย์ผันแปรไปตามความเข้มแสงอาทิตย์ การใช้งานแผงเซลล์แสงอาทิตย์อย่างคุ้มค่า อาศัยกลไกการตามรอยกำลังงานสูงสุดหรือ MPPT โดยหลักการแล้วอุปกรณ์ MPPT จะจัดการให้กำลังไฟฟ้าเอาต์พุตของแผงเซลล์แสงอาทิตย์อยู่ที่ระดับสูงสุดสอดคล้องกับความเข้มแสงอาทิตย์ (S) ในขณะนั้น ในทางปฏิบัติตัวควบคุม MPPT ทำงานควบคู่ไปกับ ดีซี/ดีซี คอนเวอร์เตอร์ ซึ่งจะต้องทำงานด้วยอัตราส่วนการแปลง D (transformation ratio) ที่เหมาะสมกับสภาวะการทำงานของแหล่งพลังงานและโหลด

2.2 แบบจำลองแบตเตอรี่ตะกั่ว-กรด

แบตเตอรี่ตะกั่ว-กรดแม้ว่าจะมีราคาถูก หาได้ง่ายในประเทศ แต่ก็มีลักษณะสมบัติการทำงานที่ไม่เป็นเชิงเส้นอย่างมากทั้งในขณะชาร์จและดิสชาร์จ สมการที่ (2) อธิบายกระบวนการชาร์จและดิสชาร์จ [3] รูปสมการที่ปรากฏขณะนี้ใช้สำหรับช่วงการดิสชาร์จ สำหรับช่วงชาร์จเครื่องหมายลบทางขวาของสมการที่ (2) จะเปลี่ยนเป็นบวกทั้งหมด

$$V_b = V_0 - (R_{tot} \cdot I) - \left[K1 \cdot \frac{I^n}{C} \right] \cdot t - \left[\frac{K2}{C - I^n \cdot t} \right] \quad (2)$$

ซึ่ง V_b คือ แรงดันที่ขั้วของแบตเตอรี่ (V)

V_o คือ แรงดันเริ่มต้นของแบตเตอรี่ (V)

R_{tot} คือ ความต้านทานภายในรวม (Ω)

I คือ กระแสไฟฟ้าที่ไหลในแบตเตอรี่ (A)

C คือ ค่าความจุของแบตเตอรี่ (Ah)

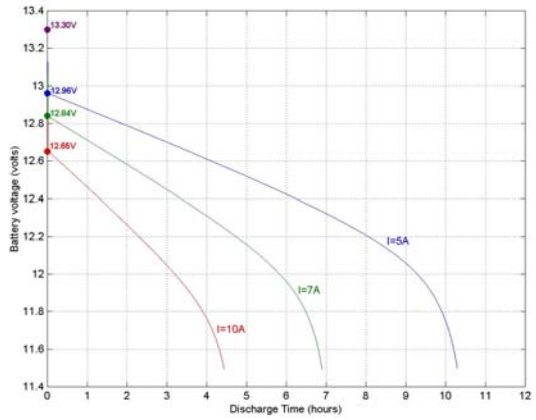
t คือ เวลาที่ใช้ในการดิสชาร์จ (ชาร์จ) แบตเตอรี่ (h)

n คือ Peukert's exponent

K1 คือ ค่าสัมประสิทธิ์จาก Peukert's equation

K2 คือ ค่าสัมประสิทธิ์ที่ทำให้แรงดันตก (เพิ่ม) กระทันหันเมื่อแบตเตอรี่ดิสชาร์จ (ชาร์จ) ใกล้หมด (เต็ม)

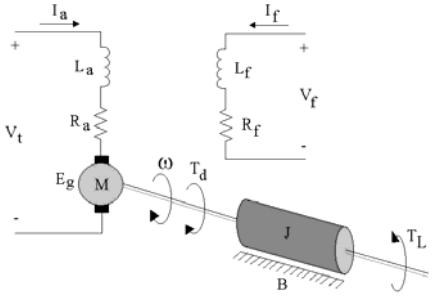
จากสมการที่ (2) จะพิจารณาเฉพาะช่วงสภาวะคงตัวเท่านั้น ซึ่งสภาวะชั่วครู่เกิดขึ้นได้ในจังหวะที่แบตเตอรี่เปลี่ยนการทำงานจากสภาวะชาร์จเป็นดิสชาร์จ และกลับไปมาด้วยช่วงเวลาที่สั้นมากคิดเป็นเพียงมิลลิวินาที จึงไม่คำนึงถึงในที่นี้ ดังนั้น จึงพิจารณาวัฏจักรการทำงานของแบตเตอรี่เป็น 3 ช่วง [4] กล่าวคือ ช่วงชาร์จด้วยกระแสคงที่หรือ CCC (constant current charge) ช่วงชาร์จด้วยแรงดันคงที่หรือ CVC (constant voltage charge) และช่วงดิสชาร์จด้วยกระแสคงที่หรือ CCD (constant current discharge) ข้อมูลในการจำลองผล ใช้แบตเตอรี่ตะกั่ว-กรดของ CELTIC [5] แรงดันปกติ 12 V ค่าความจุพิกัด 70 Ah ค่า R_{tot} ขณะดิสชาร์จและชาร์จเท่ากับ 6.15 E^{-2} Ω และ 6.56 E^{-2} Ω ตามลำดับ ค่า K1 และ K2 ขณะดิสชาร์จและชาร์จมีค่าเท่ากันเท่ากับ 0.90 และ 2.2 ตามลำดับ และค่า n ขณะดิสชาร์จและชาร์จเท่ากับ 1.16 และ 1.00 ตามลำดับ ส่วนค่าช่วงแรงดันขณะดิสชาร์จมีค่าแรงดันสูงสุด 13.3 V และแรงดันต่ำสุด 11.5 V และค่าช่วงแรงดันขณะชาร์จมีค่าแรงดันต่ำสุด 11.8 V และแรงดันสูงสุด 14.1 V นำค่าข้อมูลขณะดิสชาร์จแทนค่าในสมการที่ (2) ซึ่งจะได้ความสัมพันธ์ของแรงดันแบตเตอรี่ขณะดิสชาร์จเทียบกับเวลา เมื่อพิจารณาอัตราการดิสชาร์จคงที่ที่ 5, 7 และ 10 A ตามลำดับ แสดงได้ดังรูปที่ 5 ซึ่งอาจสังเกตเห็นว่า ขณะเริ่มดิสชาร์จแรงดันที่ขั้วของแบตเตอรี่จะตกลงอย่างรวดเร็วจากแรงดันสูงสุด ซึ่งมีสาเหตุจากค่าความต้านทานภายในรวมของแบตเตอรี่ จากนั้นแรงดันจะตกลงอย่างราบเรียบแบบเอ็กซ์โปเนนเชียล ซึ่งเกิดจากค่าความจุของแบตเตอรี่และค่าความต้านทานภายในรวม



รูปที่ 5. แรงดันที่ขั้วของแบตเตอรี่ขณะดิสชาร์จ

2.3 แบบจำลองมอเตอร์ไฟฟ้ากระแสตรงและปั๊มพ์

มอเตอร์ไฟฟ้ากระแสตรงที่พิจารณาในงานนี้เป็นแบบกระตุ้นฟิลด์แยกส่วน โดยมีวงจรสมมูลดังรูปที่ 6 [6] เมื่อใช้งานมอเตอร์ด้วยการปรับแรงดันอาร์เมเจอร์โดยมีความเข้มสนามเต็มพิกัด สามารถอธิบายการทำงานของมอเตอร์ด้วยแบบจำลองดังสมการที่ (3)



รูปที่ 6. วงจรสมมูลของมอเตอร์ไฟฟ้ากระแสตรง

$$\begin{bmatrix} \frac{dI_a}{dt} \\ \frac{d\omega}{dt} \end{bmatrix} = \begin{bmatrix} \frac{-R_a}{L_a} & \frac{-k_b}{L_a} \\ \frac{k_t}{J} & \frac{-B}{J} \end{bmatrix} \begin{bmatrix} I_a \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \\ 0 & \frac{-1}{J} \end{bmatrix} \begin{bmatrix} V_t \\ T_L \end{bmatrix} \quad (3)$$

เมื่อ V_t คือ แรงดันที่ป้อนให้มอเตอร์ทางด้านอาร์เมเจอร์ (V)

I_a คือ กระแสอาร์เมเจอร์ (A)

L_a คือ ความเหนี่ยวนำทางด้านอาร์เมเจอร์ (H)

R_a คือ ความต้านทานอาร์เมเจอร์ (Ω)

ω คือ ความเร็วเชิงมุม (rad/sec)

J คือ โมเม้นท์แรงเฉื่อยของมอเตอร์ ($Kg.m^2$)

B คือ วิสคอสฟริคชั่นของมอเตอร์ (N.m/rad/sec)

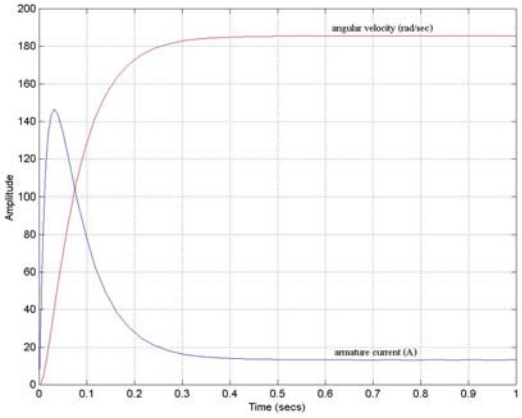
E_g คือ แรงดันย้อนกลับ: $k_b\omega$ (V)

T_L คือ แรงบิดของโหลด (N.m)

ข้อมูลในการจำลองผลของมอเตอร์ [7] มีดังนี้

- ขนาดพิกัด 110 V, 20 A, 2.5 hp , 1800 rpm , 9.89 N-m
- $R_a = 0.6\ \Omega$, $L_a = 8$ mH
- $J = 0.0465$ kg-m^2, $B = 0.004$ N-m.sec/rad
- k_t (N-m/A) = k_b (V/rad/sec) = 0.55 N-m/A

โหลดที่มอเตอร์ขับในที่นี้คือปั๊มพ์หอยโข่ง (helical pump) ซึ่งมีแรงบิดขึ้นอยู่กับความเร็วของมอเตอร์ กล่าวคือ $T_L = k\omega^2$ ซึ่ง k คือค่าคงที่ของปั๊มพ์ มีค่าเท่ากับ $1.898*10^{-4}$ N-m/(rad/sec)^2 [8] เมื่อนำข้อมูลในการจำลองผลของปั๊มพ์และมอเตอร์แทนค่าในสมการที่ (3) เพื่อหาความสัมพันธ์ของค่า I_a และ ω เทียบกับเวลา โดยป้อนแรงดันที่ค่าพิกัด V_t = 110 V ซึ่งกำหนดให้ที่เวลาเป็นศูนย์ ค่า I_a และ ω มีค่าเป็นศูนย์ ซึ่งแสดงได้ดังรูปที่ 7

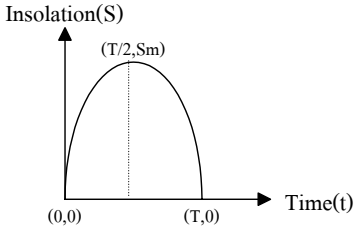


รูปที่ 7. ความสัมพันธ์ของกระแสอาร์เมเจอร์และความเร็วเทียบกับเวลา

จากรูปที่ 7 จะสังเกตเห็นว่ากระแสอาร์เมเจอร์เพิ่มสูงขึ้นอย่างรวดเร็วในขณะเริ่มเดินเครื่อง จากนั้นกระแสอาร์เมเจอร์จะลดลงอย่างรวดเร็วและเข้าสู่สภาวะคงตัวที่13.25 A ส่วนความเร็วเชิงมุมของมอเตอร์ จะเพิ่มขึ้นอย่างรวดเร็วและเข้าสู่สภาวะคงตัวที่ 185.56 rad/sec หากว่าพิจารณาการมินิไมซ์กำลังสูญเสียในมอเตอร์ กระแสอาร์เมเจอร์จะเข้าสู่สภาวะคงตัวที่ 12.49 A เท่านั้น

2.4 แบบจำลองความเข้มแสงอาทิตย์

จากข้อมูลดาวเทียม GMS 4 และ GMS 5 ตั้งแต่เดือนมกราคม 2536 ถึงธันวาคม 2541 ค่าเฉลี่ยความเข้มแสงอาทิตย์ (insolation, S) ทั่วประเทศไทยจากทุกพื้นที่จะมีค่าเท่ากับ 18.2 MJ/m^2/day [9] หรือโดยประมาณเท่ากับ 5 kw.hr/m^2/day โดยงานวิจัย [10] ได้เสนอค่าความเข้มแสงอาทิตย์มีลักษณะเป็นรูปพาราโบลาระฆังคว่ำ ดังรูปที่ 8 ซึ่งเป็นค่าความเข้มแสงที่สอดคล้องกับความเป็นจริงในวันที่ฟ้าเปิด (sunny day)



รูปที่ 8. ลักษณะของรูปพาราโบลาระฆังคว่ำ

เมื่อ A คือพื้นที่ใต้กราฟของรูป และ p คือระยะจากจุดยอดไปตามแกนพาราโบลา จะได้ว่า

$$A = \int_0^T S\,dt = \int_0^T S_m\,dt - \frac{1}{4p}\int_0^T \left(t - \frac{T}{2}\right)^2 = S_m T - \frac{T^3}{48p} \qquad (4)$$

จะได้ค่าความเข้มแสง S สัมพันธ์กับเวลา t ดังสมการที่ (5)

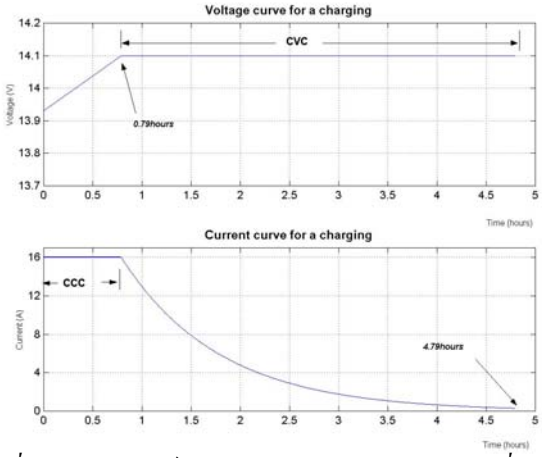
$$S = \frac{3}{2}\left(\frac{A}{T}\right) - \frac{6A}{T^3}\left(t - \frac{T}{2}\right)^2 \qquad (5)$$

โดยพื้นที่ใต้กราฟ A เท่ากับ 5,000 w.hr/m^2/day และใน 1 วัน จะกำหนดให้มีแสงอาทิตย์ทั้งหมด 8 ชั่วโมง (08.00-16.00 น.) ดังนั้น T = 8 hr. ซึ่งจากสมการที่ (5) จึงได้

$$S = 937.5 - 58.59(t-4)^2 \qquad (6)$$

3. การชาร์จแบตเตอรี่

แบตเตอรี่ทั้ง 20 ลูก ของระบบเซลล์แสงอาทิตย์นี้ ต่อเป็นแผงขนานกัน 2 แผง ซึ่งแต่ละแผงมีแบตเตอรี่ 10 ลูก ต่ออนุกรมกันอยู่ การชาร์จแบตเตอรี่แต่ละแผง อาจแบ่งออกเป็น 2 ช่วง ได้แก่ CCC และ CVC ดังรูปที่ 9 โดยช่วง CCC เป็นการชาร์จด้วยอัตรากระแสคงที่ 16 A นาน 0.79 ชั่วโมง [11] ในช่วงนี้ค่าแรงดันแบตเตอรี่จะเพิ่มสูงขึ้นอย่างรวดเร็วจนกระทั่งถึงค่าแรงดันสูงสุด จากนั้นก็ดำเนินการชาร์จในช่วง CVC ต่อทันที ก่อนที่จะเกิดแรงดันเกินในแบตเตอรี่ ในช่วง CVC ต้องทำการลดปริมาณกระแสในการชาร์จแบตเตอรี่ลงแบบเอ็กซ์โปเนนเชียลจนเกือบเป็นศูนย์ เพื่อรักษาแรงดันในช่วง CVC ให้คงที่อยู่ที่ระดับสูงสุดประมาณ 4 ชั่วโมง จึงจะทำให้แบตเตอรี่ได้รับการชาร์จอย่างสมบูรณ์ [11] ในช่วงที่สองนี้อัตราการชาร์จจึงอธิบายได้ด้วยฟังก์ชัน 16·exp(-t) การชาร์จแบตเตอรี่แต่ละแผงจึงต้องการเวลา 4.79 ชั่วโมง แบตเตอรี่ทั้งสองแผงจะเริ่มชาร์จไม่พร้อมกัน โดยจะเริ่มชาร์จเป็นลำดับทีละแผงตามค่าความเข้มแสงที่พอเพียงต่อแบตเตอรี่แผงนั้นๆ

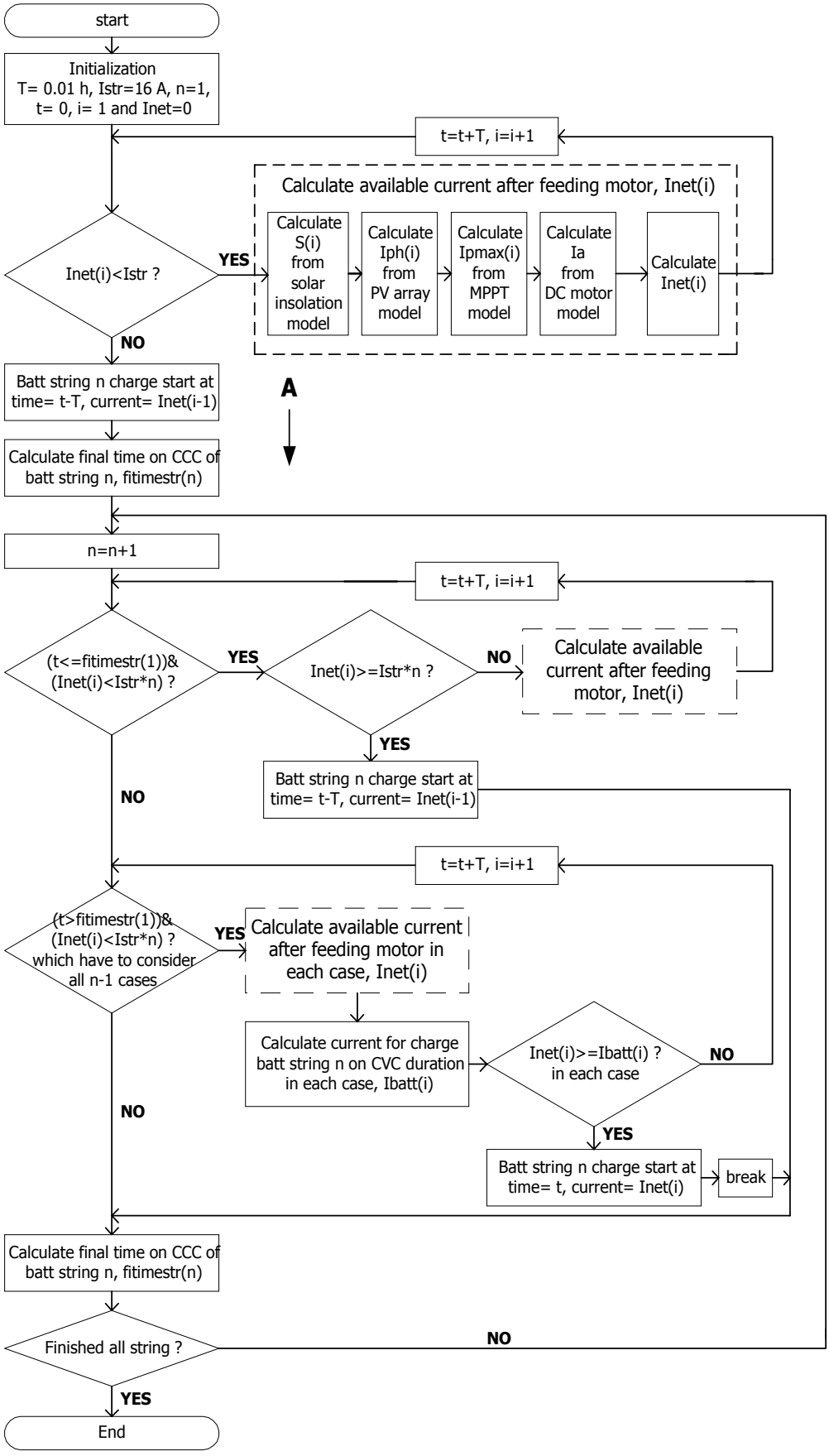


รูปที่ 9. ความสัมพันธ์ของแรงดันและกระแสของแบตเตอรี่ขณะชาร์จ

4. การจำลองผลระบบ

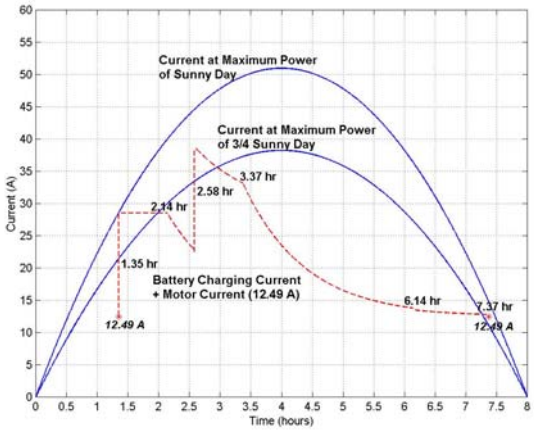
แบบจำลองทางคณิตศาสตร์ของอุปกรณ์ต่างๆ ได้รับการบูรณาการขึ้นเป็นชุดโปรแกรมจำลองผล ที่มีโครงสร้างดังแผนภูมิในรูปที่ 10 ส่วนหนึ่งของโปรแกรมจำลองผล ดำเนินการเกี่ยวกับโหมดการชาร์จแบตเตอรี่ ดำเนินการเริ่มที่ตำแหน่ง A ในแผนภูมิลงไปจนสิ้นสุด

ตัวอย่างหนึ่งของการใช้โปรแกรมจำลองผลนี้ เพื่อศึกษาการใช้พลังงานในระบบแสดงดังรูปที่ 11 ซึ่งให้รายละเอียดการดึงกระแสจากแผงเซลล์แสงอาทิตย์ เส้นโค้งรูปพาราโบลาเส้นบนแสดงปริมาณกระแสซึ่งให้กำลังงานสูงสุดที่จะได้จากแผงเซลล์แสงอาทิตย์ช่วง 8 ชั่วโมงในเวลากลางวัน (08.00-16.00น.) ของวันที่ฟ้าเปิด (ความเข้มแสงอาทิตย์ 5 kw.hr/m^2/day) เป็นกรณีดีที่สุดซึ่งสามารถเกิดขึ้นได้จริงในประเทศไทยช่วงฤดูร้อนและฤดูหนาว เส้นโค้งรูปพาราโบลาเส้นล่างแสดงสิ่งเดียวกันในกรณีความเข้มแสงอาทิตย์เป็น 3/4 ของวันที่ฟ้าเปิดเพื่อเปรียบเทียบ เส้นประแสดงปริมาณกระแสที่จ่ายมอเตอร์ 12.49 A ยืนพื้นรวมกับกระแสที่อาจดึงไปเพื่อชาร์จแบตเตอรี่ ผลดังที่แสดงให้ความหมายว่า การชาร์จแบตเตอรี่ทั้งสองแผงจะทำได้ภายในช่วงเวลากลางวันใน 1 วันนั้นก็เฉพาะกรณีวันที่ฟ้าเปิดเท่านั้น หากเหตุการณ์ไม่เป็นดังนั้นอย่างเช่นความเข้มแสงอาทิตย์อ่อนลงเป็น 3/4 ของวันที่ฟ้าเปิด ก็จะไม่สามารถชาร์จแบตเตอรี่ให้เสร็จสิ้นได้ภายใน 1 วัน ในความเป็นจริงความเข้มแสงอาทิตย์ตลอดวันนั้นอาจไม่สม่ำเสมอ เราก็สามารถสร้างโครงรูปของกราฟความเข้มแสงใดๆ และป้อนเป็นข้อมูลให้โปรแกรมจำลองผลนำไปใช้งานได้ไม่ยาก

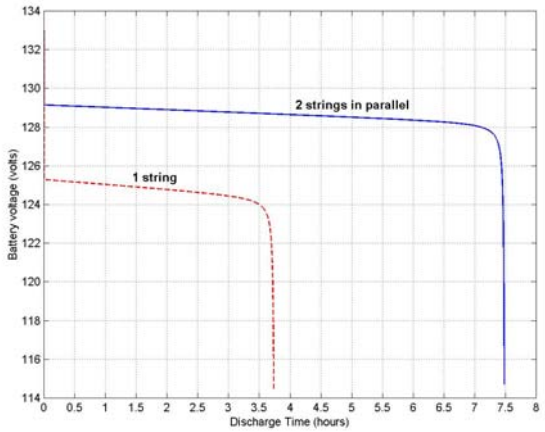


รูปที่ 10. แผนภูมิแสดงโครงสร้างของโปรแกรมจำลองผล

อีกตัวอย่างหนึ่งที่จะสะท้อนให้เห็นถึงความจำเป็นและประโยชน์ของการมีโปรแกรมจำลองผลก็ดังเช่น การวินิจฉัยว่าแบตเตอรี่จะทำงานได้นานกี่ชั่วโมง ในทางปฏิบัติที่คุ้นเคยกัน เราอาจคำนวณคร่าวๆ เช่น แบตเตอรี่ความจุ 70 Ah ดังที่ใช้ในงานวิจัยนี้ หากต้องจ่ายกระแส 12.49 A แก่มอเตอร์ จะทำงานได้ประมาณ 5.6 ชั่วโมง แต่หากพึ่งพาการคำนวณโดยใช้แบบจำลองที่แม่นยำดังที่ปรากฏในงานนี้ จะพิจารณาได้โดยละเอียดถึงลักษณะสมบัติจริงของแบตเตอรี่และของโหลด ดังที่แสดงในรูปที่ 12 เราพบว่าหากมีแบตเตอรี่แผงเดียว การป้อนไฟฟ้าแก่มอเตอร์โดยตรงทำได้เพียง 3.7 ชั่วโมงเท่านั้น หากใช้แบตเตอรี่ 2 แผง การขับมอเตอร์อาจทำได้ถึง 7.5 ชั่วโมง มิใช่ 11.2 ชั่วโมง



รูปที่ 11. ผลการจำลองระบบแสดงการดึงกระแสจากแผงเซลล์แสงอาทิตย์



รูปที่ 12. การดิสชาร์จของแบตเตอรี่ 1 แผง และ 2 แผง

5. บทสรุป

การใช้งานระบบเซลล์แสงอาทิตย์มีความซับซ้อน เนื่องมาจากลักษณะสมบัติอันไม่เป็นเชิงเส้นของแผงเซลล์แสงอาทิตย์และแบตเตอรี่ ประกอบกับแผงเซลล์แสงอาทิตย์มีลักษณะสมบัติการทำงานที่ขึ้นกับสภาพอากาศ จำนวนแบตเตอรี่ที่ใช้สำรองพลังงานขึ้นกับความต้องการในการประกันการทำงานของระบบ ว่าจะให้ใช้งานได้เมื่อแดดอ่อนนานเป็นเวลาเท่าใด ความต้องการในการชาร์จแบตเตอรี่อย่างพอเพียง จึงทวีความซับซ้อนต่อการใช้งานระบบดังกล่าว การทำความเข้าใจพลวัตของระบบจึงยากเกินกว่าที่จะนึกหรือจินตนาการได้ เป็นเหตุให้ต้องพึ่งพาเทคนิคการจำลองผลระบบด้วยคอมพิวเตอร์ ซึ่งต้องใช้แบบจำลองทางคณิตศาสตร์ของอุปกรณ์ต่างๆ ที่ประกอบขึ้นเป็นระบบ ดังที่บทความนี้ได้นำเสนอในรายละเอียด โปรแกรมการจำลองผลด้วย MATLAB™ ที่พัฒนาขึ้น สามารถนำไปใช้ประโยชน์เพื่อการศึกษาการใช้พลังงานในระบบเซลล์แสงอาทิตย์ได้เป็นอย่างดี

6. กิตติกรรมประกาศ

ขอขอบคุณมหาวิทยาลัยเทคโนโลยีสุรนารีที่ให้การสนับสนุนทุนวิจัย

เอกสารอ้างอิง

[1] Hussein, K. H., Muta, I., Hoshino,T., and Osakada, M. (1995). Maximum photovoltaic power tracking : An algorithm for rapidly changing atmospheric condition. *IEE Proc-Gener. Transm. Distrib.*142 (1): 59-64.

[2] Pan, C., Chen, J., Chu, C., and Huang, Y. (1999). A fast power point tracker for photovoltaic power system. *Industrial Electronics Society,1999.IECON'99 Proceedings.* 1: 390-393.

[3] Rynkiewicz, R. (1999). Discharge and charge modeling of lead acid batteries, *Applied Power Electronics Conference and Exposition. IEEE.* 2: 707-710.

[4] Salameh, Z.M., Casacca, M.A., and Lynch, W.A. (1992). A mathematical model for lead-acid batteries. *IEEE Transactions on Energy Conversion.* 7 (1): 93-97.

[5] Protogeropoulos, C., Marshall, R. H., and Brinkworth, B. J. (1994). Battery state of voltage modelling and an algorithm describing dynamic conditions for long-term storage simulation in a renewable system. *Solar Energy.* 53 (6): 517-527.

[6] Ramamurthi, V.P. and Subrahmanyam, V. (1991). Performance of a separately excited dc motor fed from a multiphase chopper. *TENCON'91.1991 IEEE Region 10 International Conference on EC3-Energy,Computer,Communication and Control Systems.* 1: 238-241.

[7] Sousa, C.D. and Bose, K. (1994). A fuzzy set theory based control of a phase-controlled converter dc machine drive. *IEEE Transactions on Industry Application.* 30 (1): 34-44.

[8] Yao, Y. and Ramshaw, R.S. (1995). Optimized dc motor output in a photovoltaic system. *Can.J.Elect. & Comp.Eng.* 20 (2): 79-84.

[9] มหาวิทยาลัยศิลปากร และ กรมพัฒนาและส่งเสริมพลังงาน. (ม.ป.ป.). แผนที่ศักยภาพพลังงานแสงอาทิตย์จากข้อมูลดาวเทียมสำหรับประเทศไทย (Solar Radiation Map of Thailand) [ซีดี]. กรุงเทพฯ: กรมพัฒนาและส่งเสริมพลังงานร่วมกับมหาวิทยาลัยศิลปากร.

[10] Harrington, S., Corporation, K., and Dunlop, J. (1992). Battery charge controller characteristics in photovoltaic systems. *IEEE AES MAGAZINE*: 15-21.

[11] Casacca, M.A., Capobianco, M.R., and Salameh, Z.M. (1996). Lead acid battery storage configuration for improved available capacity. *IEEE Transactions on Energy Convertion.* 11(1): 139-145.

เผด็จ เผ่าละออ สำเร็จปริญญาตรีในสาขาวิชาวิศวกรรมไฟฟ้า จากมหาวิทยาลัยเทคโนโลยีสุรนารี เมื่อ พ.ศ.2540 ภายหลังสำเร็จการศึกษาได้เข้าทำงานในตำแหน่งผู้ช่วยสอนและวิจัย สาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี เป็นเวลา 1 ปี ปัจจุบันกำลังศึกษาระดับปริญญาโทและเป็นผู้ช่วยวิจัยที่มหาวิทยาลัยเทคโนโลยีสุรนารีโดยได้รับทุนอุดหนุนวิจัยประจำปี 2543 ทางด้านการอนุรักษ์พลังงานจากทางมหาวิทยาลัย

นาวาอากาศโท สราวุฒิ สุจิตจร สำเร็จปริญญาตรีและปริญญาเอกในสาขาวิชาวิศวกรรมไฟฟ้า จากโรงเรียนนายเรืออากาศ และมหาวิทยาลัยเบอร์มิงแฮม ประเทศอังกฤษ เมื่อ พ.ศ.2527 และ 2530 ตามลำดับ ปัจจุบันดำรงตำแหน่งรองศาสตราจารย์ และหัวหน้าสาขาวิชาวิศวกรรมไฟฟ้า สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ดำเนินงานวิจัยทางด้านระบบควบคุมเชิงเส้นและไม่เชิงเส้น การดำเนินกระบวนการทางสัญญาณ การอนุรักษ์พลังงาน และการประยุกต์เทคนิคทางปัญญาประดิษฐ์ อาจารย์สราวุฒิเป็นสมาชิกวิศวกรรมสถานแห่งประเทศไทยฯ สมาคมเทคโนโลยีที่เหมาะสม และ IEEE อีกทั้งได้รับการจารึกชื่อไว้ใน Who's Who in the World และ Who's Who in Science and Engineering

Programmers' Receptive Attitude toward Code Sharing and their Use of Computer-Mediated Communication Systems

Chatpong Tangmanee
Faculty of Commerce and Accountancy, Chulalongkorn University

ABSTRACT - One way for programmers to improve their work is to share code with peers. Computer-mediated communication (CMC) systems, such as e-mail, or the World Wide Web (WWW), may facilitate the process of code sharing provided that programmers are receptive toward the process. Upon a mail survey with 730 professional programmers, the study discovered three items of what programmers attain through CMC systems: (1) task-related, (2) socio-emotional and (3) exploring accomplishments. In addition, only among the programmers whose work involves actual code reviews was their receptivity toward code sharing correlated with their use of CMC systems for task-related purposes. The correlation with other accomplishments was not statistically significant. Discussions on the results' conceptual and practical contributions are addressed in the final section.

Keywords - computer-mediated communication, programmers, receptive attitude, code sharing

บทคัดย่อ - วิธีหนึ่งที่โปรแกรมเมอร์สามารถปรับปรุงคุณภาพของซอฟแวร์ที่พัฒนาขึ้น คือการแชร์หรือแลกเปลี่ยนความคิดเห็นเกี่ยวกับโปรแกรมที่พัฒนาขึ้นกับเพื่อนหรือผู้ร่วมงาน การสื่อสารหรือแชร์ความคิดเห็นที่ว่า อาจกระทำผ่านสื่อคอมพิวเตอร์ เช่นการใช้ไปรษณีย์อิเล็กทรอนิกส์หรืออินเตอร์เนต แต่กระนั้นการให้และรับฟังความเห็นผ่านสื่อคอมพิวเตอร์ในลักษณะนี้จะสำเร็จได้จริง ก็ขึ้นอยู่กับความใส่ใจของโปรแกรมเมอร์ต่อการแลกเปลี่ยนข้อเสนอแนะใดๆ กับเพื่อนร่วมงาน จากการเก็บข้อมูลด้วยแบบสอบถามทางไปรษณีย์ จากผู้ประกอบอาชีพโปรแกรมเมอร์จำนวน 730 คน พบว่า การสื่อสารผ่านสื่อคอมพิวเตอร์ของโปรแกรมเมอร์ มีขึ้นเพื่อสนองวัตถุ-ประสงค์ (1) ทางการงาน (2) ส่วนบุคคลและสังคม และ (3) เพื่อเรียนรู้ข้อมูลใหม่ ทั้งนี้เมื่อพิจารณาเฉพาะในกลุ่มของโปรแกรมเมอร์ที่เคยมีประสบการณ์จริงในการรีวิวโค้ด จะพบว่าความใส่ใจของโปรแกรมเมอร์ต่อการแลกเปลี่ยนความคิดเห็นกับเพื่อนร่วมงานมีความสัมพันธ์ทางบวกอย่างมีนัยสำคัญทางสถิติกับการสื่อสารผ่านสื่อคอมพิวเตอร์เพื่อการงานเท่านั้น โดยที่การสื่อสารเพื่อวัตถุประสงค์อื่นไม่มีความสัมพันธ์ทางสถิติกับความใส่ใจต่อเรื่องดังกล่าว การใช้ผลวิจัยทั้งทางทฤษฎี และทางปฏิบัติได้นำเสนอในบทสรุปของงานนี้

คำสำคัญ - การสื่อสารผ่านสื่อคอมพิวเตอร์, โปรแกรมเมอร์, ทัศนคติต่อการพัฒนาระบบสารสนเทศ, การแชร์โปรแกรม

1. Introduction

The goal of this research is to bridge a gap between programmers' communication behavior and their attitude toward a programming practice. Given the wide applications of communication technology to software development processes, it is still largely unknown whether programmers' use of such technology is related to their programming perception. The current study is an attempt to explore this void. In particular, it seeks to understand a connection between the purposes for which programmers are engaged in computer-mediated communication and their receptivity toward code sharing.

2. Computer-mediated communication systems and receptivity toward code sharing

Computer-mediated communication (CMC) systems are defined in this study as programmers' perceptions of a collection of tools that primarily facilitate human communication via computers and communication networks. In addition, neither voice mail nor facsimiles are included because a large number of users do not perceive the communication via these two channels as computer-mediated [10]. Thus, the instances of CMC systems investigated in this research include computer conferencing systems (e.g., e-mail or visual conferences), Internet-based communication (e.g., Internet relay chat (IRC), Internet phone, ICQ or the WWW), group support systems, groupware (e.g., Lotus Notes) and the systems that simply support communication via e-mail, newsgroups and/or listserves.

Code sharing is one of the significant programming practices [13]. Software experts have suggested that programmers share and review their code with peers [6,14]. Reaction from peers could improve the code quality because an overlooked flaw may be detected during the review [8]. The benefits of code sharing is still unrecognized, however, unless programmers are attentive and willing to practice it. Still, to practice code sharing appears to rely on how the programmers interact with peers. This is the role in which CMC systems could help promote their receptivity and thereby provoke them to engage in code sharing. The current research thus defines the receptivity toward code sharing is the extent to which a programmer is willing to share programming code with peers.

3. Scope of the study

As said earlier, the current study attempts to shed light on programmers' computer-mediated communication behavior and their willingness to share code. To achieve the goal, it seeks to answer the following questions:

What do programmers accomplish through the use of CMC systems?

To what extent do the accomplishments relate to receptivity of programmers toward code sharing?

3.1 Accomplishments through CMC systems

The study's first focus is to examine what programmers accomplish through the use of CMC systems. Researchers have acknowledged the use of CMC systems so as to achieve progress or completion in work-related tasks. Sawyer and Guinan [11] examined how one CMC system would affect intra-group conflict-solving process among teams of software developers. "By focusing [on] the work product, and not [on] each other, the product becomes less attached to any one person: it is shared by the team" [11].

In addition to task-related benefits, the use of CMC systems can accommodate the social and emotional needs of programmers. This type of use has been ascertained in various investigations, although only a few have been conducted in the programmer context. For example, workers used e-mail not only to handle work assignments, but also to stay in touch with friends or family, or to meet people with the same interests [10]. With a secured distributed technology, programmers can complete any financial transaction (i.e., purchase a plane ticket or buy stock) over the Internet. They can also be entertained by a variety of information, or download it for their own pleasure [9].

Besides the two major purposes (i.e., task-related and socio-emotional), evidence from the literature suggest that programmers may use CMC systems for other accomplishments. For instance, Rice and Steinfield [10] identified the surveillance purposes of electronic communication. Other writers comment that organizational members may use communication to explore innovative ideas from the environment and, perhaps, share the knowledge with colleagues [5].

3.2 Computer-mediated communication and code sharing practices

Software researchers have suggested that programmers share and discuss their programming code with peers [7, 14]. An overlooked flaw is often uncovered during the discussion [8]. Consequently, the programming code that passes a thorough review will likely be bug-free [7, 13]. The idea of a programmer creating such bug-free software by having a review session with peers is widely accepted in many leading computer companies [3].

In a code review session, a programmer presents code to colleagues and responds to their feedback. At the end of the review, the programmer would learn about overlooked programming bugs and possible solutions.

Kraut and Streeter [15] investigate the extent to which programmers in one organization coordinated their work via various means of communication. E-mail and electronic bulletin boards were reported as critical channels through which the programmers could share their work with colleagues, elicit software requirements from clients, or negotiate with hardware vendors [15]. Another longitudinal study of listserves (one instance of CMC systems) used among program designers exhibits a connection between the design task and different roles of the designers [16].

Empirical evidence seems to confirm that a practice of code sharing involves programmers' communication skills. The use of CMC systems could thus be an alternative. There is no published formal method for programmers to share code via CMC systems. They may, however, utilize several features of CMC systems for the purposes of code-sharing. Examples of the features include (1) posting programming work on a corporate network and making it accessible to only involved members, (2) encouraging members to share feedback through an electronic bulletin board or sometimes to give comments in a company chat room, or (3) attaching a piece of code to e-mail sent to involved parties for subsequent examination. Sawyer and Guinan [11] remark that projecting the code on the wall appears to direct the attention of the members to the code on the screen, not to the programmer who wrote it. This therefore may raise programmer's willingness to share the code.

Despite the promising benefits of sharing code and the potential utility of CMC systems in assisting programmers to do so, the benefit is still unrealized unless the programmers are willing to do it. Johnson [8] reports the results of an informal survey in which 80% of 90 participants admitted that they practiced code reviews irregularly or not at all, although most of the subjects agreed that there are benefits from reviewing code with peers. Consequently, the ways in which the programmers use CMC systems (i.e., what they accomplish through CMC systems) may be associated with the extent to which they are receptive to code sharing.

4. Methodology

4.1 Questionnaire development

The first draft of a questionnaire was developed based upon two strategies. First, the literature on CMC and software engineering was reviewed. It helped the researcher to locate most of the questionnaire scales associated with various accomplishments one may have via the use of CMC systems. Only a few were constructed exclusively for this research because the scales were not found during the literature review. Second, the researcher conducted several interview sessions with actual programmers. This subsequent interview was to ensure that all items in the first draft were clear and understandable to programmers.

A group of scholars pretested the first draft. For the feedback from pretest participants, the questionnaire scales were modified. After the pretest, fifty other actual programmers participated in the questionnaire pilot test, of which the result is to assess the scale reliability and validity. The pilot test's findings ascertained an acceptable level of the questionnaire quality. Due to space constraints, a copy of the questionnaire is excluded from this manuscript but available upon request to the researcher.

4.2 Questionnaire administration

730 professional programmers who are members of the Association of Computer Machinery (ACM) received mail questionnaires. According to Babbie [1], this number of sample size is acceptable to provide a statistically significant finding. For a major drawback of a low response rate in using a mail survey, the researcher made every effort to follow recommendations from survey experts [1, 4] in order to draw a high volume of responses. After a three-month data collection, 438 programmers returned usable questionnaires, amounting to a 60% response rate. An examination of the non-respondents (i.e., the remaining 40%) using the trend projection method [12] detected no bias between the respondents and the non-respondents.

5. Results

5.1 Respondent demographics

Table 1 presents demographic variables of participating programmers. The highlights of these variables are as follows:

Table 1. Shows Characteristics of Participating Programmers

Major Characteristics	Respondents N (%)
Age (N=434)	
20-29 yrs.	18 (4)
30-39	134 (31)
40-49	166 (38)
50+	116 (27)
Gender (N=431)	
Male	381 (88)
Female	50 (12)
Highest Education (N=434)	
Some college	13 (3)
College degrees	105 (24)
Masters or some graduate work	273 (63)
Doctoral or higher	43 (10)
Major (N=429)	
Computer science	239 (56)
Mathematics	55 (13)
Engineering	49 (11)
Management or Business Administration	27 (6)
Information science	16 (4)
Physics	13 (3)
Others (e.g., Education, etc.)	30 (7)
Whether a code review is practiced at work (N=430)	
Yes	119 (28)
No	311 (72)
Work responsibility (N=432)	
Developing in-house systems	181 (42)
Developing packaged software	149 (35)
Installing packaged software in-house	17 (4)
Combination of the above three	23 (5)
Others (e.g., educational software)	62 (14)

- Considering that the sample was selected from regular members of the ACM who described themselves as programmers, it is not surprising that more than half of the respondents (63%) hold master degrees and about the same percentage (65%) are forty years old or higher. It seems that young

programmers or those fresh from college are not ACM members as they may not yet realize the benefits of the memberships.

- The majority (88%) of participants are men. About half (56%) of the respondents hold their highest degree in computer science while other individuals are from adjacent fields.
- This research collected data from actual programmers, instead of from computer-related students. Furthermore, responses to the questionnaire's "work responsibility" item seem to confirm that the participants are in charge of various types of programming projects, ranging from developing in-house software systems (42%) and building packaged software products (35%), to installing packaged software (4%). These findings thus ensure that the survey participants encompass professional programmers holding various actual programming responsibilities, not student programmers working on class assignments.
- Since the focus of this research is on a programmer's willingness to code sharing, whether a survey participant has actually been involved in code reviews may affect their perception. The questionnaire had therefore included one item asking respondents about their involvement. The result shows that about a quarter of the participating programmers (28%) reported that code reviews are practiced at their work.

5.2 What programmers accomplish through CMC systems

Thirty-five items reflecting various activities in which one may be engaged through CMC systems were included in the questionnaire [15]. To uncover the key purposes underlying what programmers accomplish through the use of CMC systems, the thirty-five items were factor-analyzed. Prior to the analysis, however, the items with marginal variance were excluded as they would not serve to differentiate among emerging factors. An objective criterion of a standard deviation of less than one is used to determine which items should be dropped from the analysis. As a result, four of the 35 items were excluded, leaving 31 items for the subsequent analysis.

Based on factor analysis with principle axis extraction and oblique rotation, three meaningful factors that underscore the major accomplishments programmers gain through CMC systems emerged. Table 3 displays the three purpose factors and the items that reflect on each purpose. Also included are weights of the items on the three factors. The three factors together explained about 42% of the variance among the purpose items. According to Table 3, Factor I accounted for 29.1% of the variance. Highest weights of the nine items on the first factor seem to reflect the "task-related" accomplishment programmers attain using CMC systems. Factor II explained 8.3% of the variance. Four items loaded highest on this factor, indicating that programmers use CMC systems for "socio-emotional" benefits. The final factor, Factor III, accounted for 4.5% of the variance. Highest weights of the other four purpose items tend to suggest the "exploring" purpose for using CMC systems. Assessments of the factor structure suggest that the discovery of these three accomplishments programmers gain through CMC systems is conceptually parsimonious and methodologically sound.

5.3 Receptive attitude toward code sharing and accomplishments through CMC systems

The study measured the survey participants' attitude toward code sharing using five-item Likert scales (see the scales in Appendix A). A mean of 4.19 with a standard deviation of 1.1 may indicate that the participating programmers appear to be receptive to code sharing. The receptive attitude becomes statistically different (see Table 2) among those who are involved in code reviews and those who are not. That is, the programmers who have actual experience with code reviews seem more willing to share code than those who do not. It may therefore suggest that any subsequent examination on this receptivity must take into account the significant difference.

Table 2 Shows Comparison of Receptivity toward Code Sharing between Programmers Whose Work Involves Code Reviews and Those Whose Work Does Not.

	Experience in Code Reviews	
Statistics	Do not have the experience	Have the experience
N	308	118
Mean[1]	4.098	4.456
S.D.	1.15	1.05

Note[1]: $t = -2.94$, $df = 424$, $p < .003$

5.4 Correlation between programmers' accomplishments through CMC systems and receptivity toward code sharing

In the previous section was a measurement description of programmers' receptivity toward code sharing. To explore the links between the receptivity and their use of CMC systems, three indexes characterizing the extent to which programmers use CMC systems for task-related, socio-emotional and exploring purposes were constructed. The indexes are derived from arithmetic means of the items that have highest weights for a given purpose. For instance, the index of a programmer using CMC systems for exploring purposes is computed from a mean of the four items that have the highest weights on this purpose (i.e., Factor III).

Table 3 Shows Factor Analysis Results: Purposes for Programmers to Use CMC Systems

Purpose Items	Factors			
	I	**II**	**III**	
Factor I: "Task-Related"				
Discuss work information with co-workers	**.75**	.06	-.01	
Coordinate work with distant colleagues	**.52**	-.01	-.04	
Schedule meetings	**.65**	-.05	-.04	
Give or receive feedback on work assignments	**.74**	.04	.03	
Send confirmation to colleagues/clients	**.64**	.02	.01	
Discuss work with clients	**.54**	.04	.08	
Resolve work conflicts or disagreements	**.60**	.03	.07	
Transfer files	**.54**	.05	.31	
Keep track of what's happening in a company	**.53**	.10	.06	
Factor II: "Socio-Emotional"				
Fill free time	-.02	**.77**	-.02	
Greet people on social occasions	.16	**.52**	.05	
Be entertained (e.g., electronic humor)	.02	**.68**	.02	
Take a break from work	-.02	**.78**	-.06	
Factor III: "Exploring"				
Check out new services/products	.03	-.04	**-.75**	
Stay up-to-date on computer upgrades	.02	.04	**-.73**	
Seek out alternatives to work problems	.05	.03	**-.62**	
Download information	.00	-.02	**-.83**	
Percent of Variance Explained	*29.1%*	*8.3%*	*4.5%*	*= 41.9%*

Using correlation analysis, the study found a slight positive yet significant correlation between task use and the receptivity of programmers toward sharing code ($r = .101$, $p < .038$). However, the correlation of this receptivity with socio-emotional use and exploring use are not statistically significant (see Table 4, a). Nonetheless, the significant difference in receptivity toward sharing code between programmers who have been involved in code reviews and those who have not (see Table 2) suggests further reexamination of the relationship.

An examination of this relationship among the programmers whose work involves code reviews reveals a slightly stronger and significant correlation between the receptivity and task use ($r = .112$, $p < .023$). The interaction effect of involvement in actual code reviews is further confirmed as the study found the insignificant correlation between the receptivity and task use among those whose work does not involve code reviews (see Table 4, b).

Table 4 Shows Correlation of Programmers' Accomplishments through CMC Systems and Receptivity toward Sharing Code

(a)

Accomplishments	Correlation Statistics
Task use	$r = .101$, $p < .038$, $N = 426$
Socio-emotional use	$r = -.020$, $p < .674$, $N = 424$
Exploring use	$r = .089$, $p < .069$, $N = 423$

(b)

Accomplishments	Correlation Statistics	
	Those whose work involves code reviews	Those whose work does not involve code reviews
Task use	$r = .112$, $p < .023$, $N = 117$	$r = .071$, $p < .217$, $N = 305$
Socio-emotional use	$r = -.101$, $p < .281$, $N = 116$	$r = -.001$, $p < .988$, $N = 304$
Exploring use	$r = .135$, $p < .145$, $N = 117$	$r = .087$, $p < .131$, $N = 302$

6. Conclusions

6.1 Programmers' accomplishments via CMC systems

Upon the study's results, programmers appear to achieve three purposes of using computer-mediated communication systems. They are task-related, socio-emotional and exploring purposes. While the first two accomplishments are generally recognized in the literature, the third is noted by few scholars [2]. Consistent with common literature, the findings stress the essential combination of "work" and "play" [5].

Also, researchers have acknowledged exploring functions of communication as the transfer of knowledge between an organization and its environment [2]. This piece of information may help an organization to cope with changes, especially when the organization's environment is undergoing dramatic shift. Given the dynamic and various changes in software environment, it is reasonable to argue that programmers conduct an exploration via CMC systems, perhaps, in search for innovative ideas (e.g., ready-to-use programming applets, or details of product upgrades) so as to survive the turbulent condition.

6.2 Programmers' receptivity toward code sharing and accomplishments through CMC systems

Throughout the entire sample, the degree to which programmers used CMC systems for task-related purposes exhibited a significant positive correlation with their receptivity toward code sharing. However, there existed a crucial difference in this receptivity between programmers whose work involves actual code reviews and those whose work does not. The sample was therefore divided into two groups according to whether a respondent had been involved in code reviews, and the correlation was reexamined. The interaction effect of involvement in code reviews was confirmed because the correlation among those having been involved in code reviews was slightly stronger and the correlation in the other group became non-significant.

Two conclusions may be drawn from the results. First, it appears that programmers' use of CMC systems for task-related purposes and their willingness to share code relate to each other in the same direction. Given that code sharing is one of the programming-related tasks, the finding could have been expected. The more programmers use CMC systems for task-related activities which may include code sharing, the more likely they experience the positive outcomes, which in turn, may increase their receptivity toward it. This speculation may also explain why socio-emotional and exploring uses were not significantly related to the receptivity.

The existence of the correlation among those who are involved in actual code reviews suggests the second conclusion. It seems that the involvement in code reviews mediates the relationship between task use and the receptivity. Only among those who have experience with actual code reviews (e.g., presenting their own code or taking part in a review session) was the correlation confirmed. Those who learn about code reviews and benefits of code sharing but have never been involved in an actual review may have different attitudes, regardless of how they use CMC systems.

7. Implications

The study's results lead to two implications. First, they extend theoretical concepts on CMC and on software engineering. Regarding communication functions, the current study has confirmed the need to incorporate all three major purposes (i.e., task-related, socio-emotional and exploring) into research on electronic communication. Regarding software engineering issues, the results disclose the interaction effect of programmer' involvement in actual code reviews on their willingness to code sharing. It is thus speculated that programmers' greater involvement in code review may result in more receptivity toward code sharing. Alternatively, without participating in real review sessions, programmers may unlikely conceive the benefits of code sharing and thereby their willingness to practice it remain unchanged.

Second, the study offers practical contributions. The results empirically confirmed three accomplishments programmers achieve through CMC systems. Software practitioners may thus expect to witness activities associated with these three types of use from their programming staff. Positive correlation between programmers' task use of CMC systems and their receptivity toward code sharing may imply that encouraging programmers to communicate via computer media for task-related purposes could increase their willingness to share code. The interaction effect of involvement in actual code reviews may remind practitioners that, unless programmers have real experience with the reviews, their task use of CMC systems may remain unrelated to their receptivity toward code sharing.

8. Limitations

Inferences from this research are limited by two major factors. First, the demographics of the participants appear to temper the study's generalizability of results to the programmer population in general. The programmers who participated in this study are dominantly male, between 30-49 years of age and with at least a college degree. This consequently limits the generalization of the study, despite the random sample of about 700 members, and the high percentage of survey returns.

The second limitation is more methodological. This is a cross-sectional study. Further, the phenomenon under study (i.e., CMC system usage) is dynamically changing due to rapid development of computer technology. Hence, the analyses and the conclusions present only a snapshot of how programmers use CMC systems. Additionally, prior to this study, very little was known about correlation between programmers' computer-mediated communication behaviors

and their programming perception. The study's execution was therefore made based upon an exploratory approach. This is the main reason that the investigated relationships were neither hypothesized nor tested. Nevertheless, the study has ascertained the link between programmers' accomplishments via CMC systems and their receptivity toward code sharing.

References

[1] Babbie, E. R. *The practices of social research (6th edition)*. CA: Wadsworth publications, 1992.

[2] Choo, C. W. (1998). *The knowing organization: How organizations use information to construct meaning, create knowledge, and make decisions*. NY: Oxford University Press.

[3] Cusumano, M. A., & Selby, R. W. (1995). *Microsoft secrets*. NY: The Free Press.

[4] Dillman, D. (1978). *Mail and telephone surveys: The total design method*. NY: Wiley.

[5] Heath, R. L. (1994). *Management of corporate communication*. NJ: Lawrance Erlbaum Associate.

[6] Humphrey, W. S. (1995). *A discipline for software engineering*. MA: Addison-Wesley publishing.

[7] Humphrey, W. S. (1997). *Managing technical people*. MA: Addison-Wesley.

[8] Johnson, P. M. (1998). Reengineering inspection. *Communications of the ACM*, *41*(2), 49-52.

[9] Mehta, M. D., & Plaza, D. E. (1997). Pornography in cyberspace: An exploration of what's in USENET. In S. Kiesler (Ed.), *Culture of the Internet* (pp. 53-67). NJ: Lawrance Erlbaum Associates.

[10] Rice, R. E., & Steinfield, C. W. (1994). Experiences with new forms of organizational communication via electronic mail and voice messaging. In J. H. Andriessen & R. A. Roe (Eds.), *Telematics and work* (pp. 106-137). East Sussex, UK.: Lawrence Erlbuam Associates Ltd.

[11] Sawyer, S., & Guinan, P. J. (1994). Team-based software development using an electronic meeting system: The quality pay-off. In M. Lee, B. Barta & P. Jaliff (Eds.), *Software quality and productivity: Theory, practice, education and training* (pp. 78-83). New York, NY: Chapman Hall.

[12] Smith, J. (1997). Nonresponse bias in organizational surveys: Evidence from a survey of groups and organizations working for peace. *Nonprofit and voluntary sector quarterly*, *26*(3), 359-368.

[13] Weinberg, G. M. (1998). *The psychology of computer programming (the silver anniversary edition)*. NY: Dorset House Inc.

[14] Yourdon, E. (1999). *Death march*. NY: Prentice Hall, Inc.

[15] Kraut, R. E., & Streeter, L. (1995). Coordination in software development. *Communications of the ACM*, *38* (3), 69-81.

[16] Ahuja, M. K., & Carley, K. M. (1998). Network structure in virtual organizations. *Journal of computer-mediated communication*, *3*(4), Available at http://www.ascusc.org/jcmc/vol3/issue4/ ahuja.htm.

Appendix A

Sharing Code with Peers

To better understand your use of CMC systems, we would like to ask some questions about your perception toward sharing code with peers. Please indicate, by circling the appropriate number, to what extent you agree or disagree with these statements.

If you have not had experience with each of the following statements, please indicate your assessment based on your opinion, otherwise based on your own experience.

	Strongly disagree			**Neutral**			**Strongly agree**
Reviewing code with peers is not so useful as it may sound.…	-3	-2	-1	0	1	2	3
I feel nervous if I have to present my code to peers……	-3	-2	-1	0	1	2	3
Showing peers my code makes me uncomfortable..……	-3	-2	-1	0	1	2	3
I like to get comments on my code from peers.....……..	-3	-2	-1	0	1	2	3
It is fearful to share code with peers.....…….	-3	-2	-1	0	1	2	3

Chatpong Tangmanee holds a Ph.D. degree in Information Transfer from Syracuse University in New York. He is a full time faculty member in the field of Information Technology at the Faculty of Commerce and Accountancy's Department of Statistics in Chulalongkorn University. His research interests span across multiple areas including human-computer interaction, computer-mediated communication and impacts of interaction technology on individuals, organizations and society. He is also responsive to methodological issues in the topics of information technology and computer-mediated communication. A former subscriber to IEEE, he is currently a member of the Association of Computing Machinery (ACM) and the Institute for Operations Research and the Management Sciences (INFORMS).

Standardization of Information Technology: Experience of Thailand

By
Thaweesak Koanantakool, Ph.D.
Director, National Electronics and Computer Technology Center
National Science and Technology Development Agency,

For AHTS2, Malaysia.
November 2000.

Introduction

This paper describes the experience of Thailand in developing her own IT standards. Major emphasis were mainly placed on character coding initially. Recent development, however, is shifted towards the facilitation of trade (EDI), information security (PKI and smart card) and electronic commerce. There are many more IT-related standards in the area of electronic industry and the Internet which may be the focus of future development.

Early Works

The need for Thai IT standard was identified since early 1970's when main frame computers were employed in many government agencies. IBM-029 keypunch machines and line printers were localized to accommodate Thai characters. De-facto standard for EBCDIC character code for Thai language has been in used since the seventies.

The first formal recognition for Thai IT standard was in 1984, after my publication on the numerous Thai Character codes [1] was reported in the press. The Thai Industry Standards Institute (TISI) later appointed a technical committee to look after the IT standards development. [2]. The very first standard produced by this committee was the TIS 620-2529 (1986): The Character Codes for Thai languages [3]. The standard character chde sets were made into two categories: one for use with the ISO-620 standard (widely known as ASCII and one for use with EBCDIC.

The committee subsequently produced a number of standards which are summarised in the website of NECTEC (see Figure 1) [4].

List of existing standards

- TIS 620-2533 (1990) Standard for Thai Character Codes for Computers
- TIS 988-2533 (1990) Recommendation for Thai Combined Character Codes and Symbols for Line Graphics for Dot-Matrix Printers
- TIS 1074-2535 (1992) Standard for 6-Bit Teletype Codes
- TIS 1075-2535 (1992) Standard for Conversion Between Computer Codes and 6-Bit Teletype Codes
- TIS 1099-2535 (1992) Standard for Province Identification Codes for Data Interchange
- TIS 1111-2535 (1992) Standard for Representation of Dates and Times
- TIS 820-2538 (1995) Layout of Thai Character Keys on Computer Keyboards
- WTT 2.0 (1992) Standard Input and Output Method for Thai language. [5]

]

Figure 1. IT-Standard web page at http://www.nectec.or.th/it-standards/

Recent Developments

The implementation of TIS 620 standard codes for global information interchange is not yet completed. There are several activities which enabled this standard code usable with the global standards. Our recent experience is mainly on the real implementation of the standard code in real systems such as MIME (over the Internet), UNIX Locale (Linux and all other UNIX systems), X-Window, GTK+ and Qt Toolkits.

Other line of developments are dealing with application of IT in business and government. I will cover this in the section "IT Standards for the New Economy".

Using Thai language on Computer Systems in Year 2000

The use of TIS-620 as an 8-bit character set coexisting with the ASCII code have been in used in Windows 95. This is known as Code-page 874. The code was also registered with the IANA (Internet Assigned Number Authority) to gain universal use in the MIME (Multipurpose [Multimedia] Internet Mail Extension) standard, according to RFC2278 standard for the Internet. Similar implementation of TIS 620 on the MaOS platform was also established since 1992, this is knoen as MacThai code.

In September1998, Trin Tantsetthi successfully registered TIS 620 as "tis-620" encoding with the IANA for Thailand [5]. As a result, the non-standard declaration for character encoding names such as "windows-874" or "cp-874" can all be supported by just one declaration "tis-620". New versions of Internet applications such as Mozilla (web browser), Internet Explorer (web browser) and Outlook (email client) now recognize "tis-620" character set.

Some further works are still required in making TIS 620 a usable standard. Certain legacy data have to make do with something else in order to convey Thai messages due to limitation of software features. For example, with Netscape Navigator up to version 4.7, the only way to display character in Thai language is to define fonts with "ISO 8859-1" encoding or "User-defined" encoding. Of course, the end result depends on the information receipient to set up the browser with a proper set of fonts to view Thai messages.

At the time of this article, the attempt of Thailand to implement the TIS 620 standard under the ISO 8859 framework is not yet succesful. Throught several negotiations, the ISO 8859-11 is now schedule for standardization process. It is now at the stage of FDIS.

UCS

Due to Microsoft's initiative to implement UCS in its Office-97 products on the Windows platform, Thai standard codes are made available widely across the world through UCS. The actual Microsoft implementation, however, added some character codes such as Em dash, En dash, ellipses, etc. in the code table. The code is supported natively in text editors and Internet applications.

Public-domain Thai fonts

In 1999, NECTEC released three public-domain Thai fonts. The work is part of its effort to globalize the Thai language on computers. The three fonts are made to display and print Thai texts to match the Roman serif and sans serif styles. Through the public-domain approach, users of freeware and open-source operating systems can now enjoy high-quality display and print out.

Tai Script Study Group

The work on development on Tai script was also on the way. NECTEC set up a joint meeting among Tai script specialists in April 2000. The aim of the meeting was to gather the baseline information about the Tai scripts. The detailed update on this work is reported fully in the companion paper for AFSIT [6].
The study group planned for the following work in the years to come:

1. writing system (consonants, vowels, tones, digits, direction, syllable composition)
2. encoding (alphabetic elements, interchange vs. presentation, composing marks)
3. input method (key availability, input sequence, keyboard layout)
4. output method (display levels, glyph design, combining characters, shaping, font encoding)
5. sorting order (alphabetic order, classes of character weights)
6. word boundary (word/phrase/sentence delimiters, requirements of word boundary analysis)

IT Standards for the New Economy

In addition to the character coding and input/output systems for computers, several new areas in IT standardization took place in Thailand in 1999 and 2000. These activities are:

1. The draft Smart Card standards (volume 1: reference applications, volume 2: PKI applications). The document was produced by Thailand Smart Card Working Group during 1997 and 1998. The working group consisted of more than 70 organizations with more than 150 people attending the series of committee meetings hosted by NECTEC.
2. IT-Model Office Interoperability Specifications (1999). This document was produced by the Government Information Technology Service Program (http://www.gits.net.th/). It describes a number of government applications which are required to adhere to some reference standards in order to be compatible with the emerging GINet (Government Information Networkl) to support e-Government plan of Thailand.
3. The EDI Messaging Working Groups. So far,a number of standard EDIFACT messages are adopted by Thailand though three working groups: EDI Purchasing WG, EDI Customs WG, and EDI Financial WG. At this stage, the Security WG for EDI is in the development stage.

Not only industrial standards for the New Economy are emerging, new laws for IT are also being developed in order to push Thailand into the new millennium. In the year 2000, three laws have been drafted:

- Electronic Transactions Law
- Electronic Signature Law
- National Information Infrastructure Development Law.

The first two laws have been merged into one and is now awaiting for the Senate to consider. The third item is near approval by the cabinet. Three other laws which are being drafted consist of:

- Computer Crime Law
- Private Data Protection Law
- Electronic Funds Transfer Law.

These laws are expected to be sent for the cabinet by early next year.

Conclusion

The struggle to standardize the usage of Thai character code is not yet over. We still have a few important tasks to do. The tasks of developing code-points for Tai scripts of various parts of Asia has been started. We found that there are many other standards required by the IT industry in Thailand and some of them are being developed. These include: EDI standards, smart-card, PKI and e-Government interoperability Guide. The need for governing IT laws are also required by the society.

References

[1] Thaweesak Koanantakool, "Thai Character Codes in Current Use" (in Thai), *Microcomputer Magazine*, No.9, August 1984.
[2] Thaweesak Koanantakool and the Thai API Consortium, "Computer ans the Thai Language" (in Thai), National Electronics and Computer Technology Center, Bangkok, ISBN 974-7570-66-1, October 1991.
[3] Thai Industry Standards Institute: TIS 620-2529 and TIS 620-2530 (1980), ISBN 974-606-153-4, available at http://www.nectec.or.th/it-standards/std620/std620.htm (in Thai).
[4] Thaweesak Koanantakool and Adshariya Agsorn-intara, Character Codes and Input/Output Method for the Thai language, http://thaigate.rd.nacsis.ac.jp/refer/thaiconf/ (in English).

[5] Trin Tantsetthi, "Legitimate Thai Charset on the Internet", http://software.thai.net/alerts/tis-20/index.html, Last update: September 1998.
[6] Theppitak Karoonboonyanan and Thaweesak Koanantakool, "Status of IT Standardization in Thailand (2000)", AFSIT 14, Jahore Baruh, November 2000.

Thaweesak Koanantakool received his Bachelor and Ph.D. degrees in Electronical Engineering from Imperial College of Science and Technology, London University. He had a number of industrial constructs in UK before he came back to Thailand to start his government service career in 1981. He taught in Electronical Engineering with the Faculty of Engineering, Prince of Songkla University. In 1985, he moved to Bangkok, Thammasart University, and was appointed Associate Director of the Information Processing Institute for Education and Development. Since 1994, he became Deputy Director of NECTEC as well as leading the Network/Software Technology labs. Thaweesak introduced the Internet into Thailand and set up the largest academic and research network known as ThaiSarn under NECTEC. He later co-founded the first Internet Service Provider (ISP) owned by Thai government in 1995. The ISP in Thailand Company Limited, at present in the largest ISP in Thailand, has 45% market share (by IP numbers managed). In 1996-1997, Thaweesak led Thailand's Information Superhighway test bed Project funded by NECTEC. The project was a major test bed in Thailand using ATM switches for both local area and wide-areas. In August 1998, Thaweesak was appointed the Director of NECTEC.

โปรโตคอลมาตรฐานสำหรับอินเตอร์เน็ทเทเลโฟนนี*

(Internet Telephony Protocols)

สาธิตพงศ์ พุทธิประเสริฐ[1] สินชัย กมลภิวงศ์[2] ลัญฉกร วุฒิสิทธิกุลกิจ[3]

[1]นิสิตปริญญาโท ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย

[2]อาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

[3]อาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ABSTRACT -- Internet telephony, also known as voice-over-IP (VoIP) or IP telephony (or IPtel in short), offers an opportunity to design global multimedia communication systems which include voice, video, and image, that may eventually replace the existing telephone infrastructure. Many benefits will be gained by end-users from Internet telephony including low service cost (expect to be *few tens Baht per hour* in stead of few tens Baht per minute as offered by PSTN), more features (Web, email integration, voice mail box), mobility (virtual private home). In this paper, we present the fundamental of how Internet telephony works. Two standards of Internet telephony are described: H.323 developed by ITU (International Telecommunications Union) and SIP (Session Initial Protocol) developed by IETF (Internet Engineering Task Force). Their advantages and disadvantages then are compared. A possibility of interconnection scenarios between H.323 and SIP are given. Applying such two standards for mobile phones as well as development tools for IP telephony are presented.

KEYWORDS -- VoIP, IP Telephony, voice over IP, H.323, SIP

บทคัดย่อ -- เทคโนโลยี Internet Telephony (หรือเรียกว่า VoIP (Voice over IP) หรือ IPTel) เป็นเทคโนโลยีที่สามารถรองรับการสื่อสารแบบพหุสื่อ (multimedia) เช่นการสื่อสารด้วยเสียง ภาพ และ วิดีทัศน์ (Video) และเป็นที่คาดว่าเทคโนโลยีนี้จะมาทดแทนระบบเครือข่ายโทรศัพท์ในปัจจุบัน ผู้ใช้จะได้ประโยชน์จากการใช้งาน Internet Telephony หลายประการเช่น ราคาต่ำกว่า (คาดว่าราคาจะตกอยู่ในหลักสิบบาทต่อชั่วโมงแทนที่จะเป็นหลักสิบบาทต่อนาที) การให้บริการได้หลากหลายรูปแบบ เช่น การใช้งานร่วมกับ web email หรือ กล่องไปรษณีย์เสียง (voice mail box) ไม่จำกัดสถานที่ในการใช้งาน (mobility) ในบทความนี้ จะกล่าวถึงหลักการทำงานของ Internet Telephony ของ 2 มาตรฐานหลัก คือ H.323 และ SIP (Session Initial Protocol) H.323 พัฒนาขึ้นโดย ITU (International Telecommunication Union) ส่วน SIP พัฒนาขึ้นโดย IETF (Internet Engineering Task Force) ข้อดีและข้อเสียของระบบ H.323 และ SIP และการเชื่อมต่อระหว่างสองระบบดังกล่าว จะได้กล่าวถึงในรายงาน นอกจากนี้ยังได้กล่าวถึงการประยุกต์ใช้ Internet Telephony สำหรับเครือข่ายไร้สายอีกด้วย

คำสำคัญ -- อินเตอร์เน็ทเทเลโฟนนี, VoIP, voice over IP, H.323, SIP

* *ได้รับสนับสนุนทุนวิจัยจาก NECTEC (รหัสโครงการ NT-B-06-4B-18-313)*

1. บทนำ (Introduction)

ในปัจจุบันเครือข่ายอินเตอร์เน็ตได้มีจำนวนผู้ใช้เพิ่มขึ้นอย่างรวดเร็วเนื่องจากว่าผู้ใช้สามารถค้นหาข้อมูลที่ต้องการจากแหล่งข้อมูลขนาดใหญ่และสามารถติดต่อสื่อสารกับบุคคลจากที่ต่างๆ ได้อย่างสะดวก นอกจากนี้การพัฒนาการให้บริการในรูปแบบใหม่เพื่อรองรับความต้องการของผู้ใช้ได้เพิ่มขึ้นอย่างรวดเร็ว เนื่องมาจากความยืดหยุ่นของเครือข่ายอินเตอร์เน็ตทำให้สามารถพัฒนาการให้บริการได้ง่ายกว่าเครือข่ายอื่นๆ

การให้บริการในรูปแบบหนึ่งที่กำลังได้รับความสนใจจากผู้ให้บริการรายต่างๆ เป็นอย่างมากคือ การใช้งานโทรศัพท์บนเครือข่ายอินเตอร์เน็ตซึ่งใช้โปรโตคอล protocol) IP (Internet Protocol) โดยเรียกว่า Voice over IP (VoIP) Internet telephony หรือ IP telephony (IPTel) ซึ่งจะรวมไปถึงการส่งวีดีโอและข้อมูลด้วย (หรือเรียกว่า การสื่อสารแบบพหุสื่อ (multimedia communication)) การให้บริการแบบนี้จะทำให้ผู้ใช้สามารถใช้โทรศัพท์บนเครือข่ายอินเตอร์เน็ตแทนเครือข่ายชนิดอื่นที่ใช้อยู่ ซึ่งการใช้งานในลักษณะนี้จะเป็นผลดีต่อผู้ใช้คือ จะช่วยประหยัดค่าโทรศัพท์ทางไกล สามารถใช้งานโทรศัพท์พร้อมกับการใช้งานอย่างอื่นบนอินเตอร์เน็ต มีการเชื่อมต่อในการใช้งานที่ยืดหยุ่นกว่าโทรศัพท์ทั่วไป รวมทั้งยังสามารถปรับแต่งการใช้งานโทรศัพท์ได้หลากหลายกว่า เมื่อพิจารณาในแง่ของผู้ให้บริการหรือผู้ดูแลเครือข่าย การพัฒนาปรับปรุงเครือข่ายหรือการให้บริการสามารถทำได้ง่ายกว่า รวมทั้งการดูแลเครือข่ายทำได้สะดวกกว่า เพราะว่าไม่จำเป็นต้องแยกเครือข่ายสำหรับ ข้อมูล เสียงและวิดีโอ ระบบ VoIP จะทำให้ข้อมูลเสียงและวีดีโอสามารถส่งรวมกันไปในเครือข่ายอินเตอร์เน็ตเพียงเครือข่ายเดียว จากข้อดีเหล่านี้ต่างๆ ทำให้ VoIP เป็นที่สนใจอย่างมากและอาจจะเป็นไปได้ว่า VoIP จะเข้ามาแทนที่ระบบโทรศัพท์ในปัจจุบันในอนาคตอันใกล้นี้

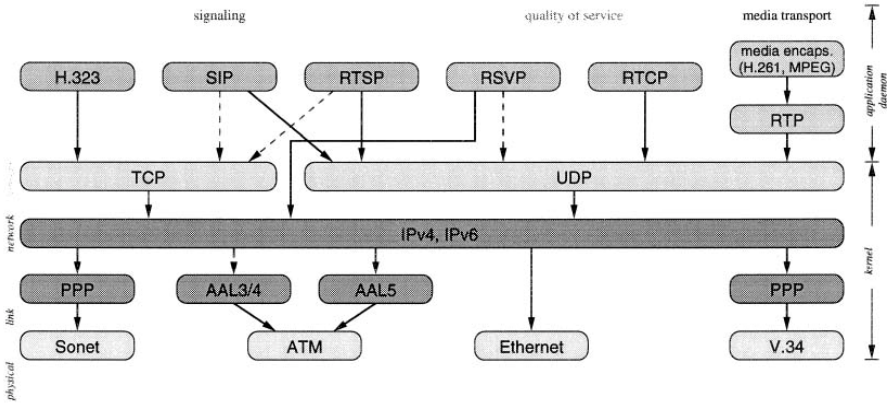
อย่างไรก็ตามความต้องการระดับพื้นฐานของ VoIP คือคุณภาพของการให้บริการ (QOS) ต้องเทียบเท่าคุณภาพในระบบโทรศัพท์หรือดีกว่าซึ่งเป็นปัญหาสำหรับการใช้งาน VoIP ในทางปฏิบัติ เนื่องจากว่าเครือข่ายอินเตอร์เน็ตเป็นเครือข่ายที่ไม่รับประกันในเรื่องคุณภาพของการให้บริการ ปัญหานี้จึงเป็นอุปสรรคสำคัญในการนำ VoIP มาใช้แทนระบบโทรศัพท์ที่มีอยู่

เครือข่ายอินเตอร์เน็ตได้ให้การบริการในการส่งข้อมูลทั่วไปซึ่งไม่ต้องการคุณสมบัติแบบเวลาจริง (real time) แต่สำหรับในกรณีของ VoIP หรือการสื่อสารแบบพหุสื่อต้องการคุณสมบัติแบบเวลาจริง ดังนั้นจึงต้องมีการพัฒนาโปรโตคอลที่สามารถให้คุณสมบัติดังกล่าวในการส่งข้อมูล และที่สำคัญ VoIP ต้องการฟังก์ชันในการสร้างหรือสิ้นสุดการเรียก และหาตำแหน่งที่อยู่ของผู้ใช้ที่ถูกเรียก รวมทั้งฟังก์ชันที่ช่วยให้สามารถให้บริการได้เช่นเดียวกับในระบบโทรศัพท์ ซึ่งในชุดโปรโตคอล TCP/IP (Transport Control Protocol/Internet Protocol) นั้นไม่มีโปรโตคอล ที่ให้ฟังก์ชันดังกล่าว ดังนั้นจึงต้องมีการพัฒนาโปรโตคอลขึ้นใหม่เพื่อรองรับ VoIP ในปัจจุบันโปรโตคอลสำหรับ VoIP มีอยู่ 2 มาตรฐานคือ H.323[2] และ SIP (Session Initial Protocol)[1] โปรโตคอล H.323 เป็นโปรโตคอลที่พัฒนาโดย ITU-T (International Telecommunications Union-Telecommunications section) ส่วน SIP ถูกพัฒนาโดย IETF (Internet Engineering Task Force) โปรโตคอลทั้งสองมีหน้าที่หลักในการสร้าง สิ้นสุด และการเปลี่ยนแปลงรายละเอียดของการเรียก ระหว่างผู้ใช้ VoIP รวมทั้งยังสามารถให้ฟังก์ชันเพิ่มเติมอื่นๆ โปรโตคอลทั้งสองเป็น โปรโตคอลสำหรับ VoIP ซึ่งใช้บริการชุดโปรโตคอล TCP/IP ในชั้นต่ำกว่า และสามารถใช้งานร่วมกับโปรโตคอลอื่น เพื่อให้เกิดการบริการที่มีคุณภาพมากขึ้น

Protocol stack สำหรับ VoIP จะเป็นดังรูปที่ 1. จะเห็นว่าทั้งโปรโตคอล H.323 และ SIP เป็นโปรโตคอลในชั้นแอปปลิเคชัน (application layer) และใช้บริการของโปรโตคอลในชั้นที่ต่ำกว่า SIP สามารถใช้ได้ทั้ง UDP และ TCP ส่วน H.323 ใช้ TCP เท่านั้น แต่เนื่องจากว่าฟังก์ชันของ H.323 และ SIP มีขอบเขตจำกัด ดังนั้นจึงได้นำโปรโตคอลอื่นมาช่วยในการทำงาน ซึ่งได้แก่ RTSP (Real-time Streaming Protocol) [3] RSVP (Resource Reservation Protocol) [7] และ RTP/RTCP [4] ซึ่งทำให้ VoIP สามารถให้บริการได้อย่างมีประสิทธิภาพมากขึ้น โปรโตคอลดังกล่าวเป็นโปรโตคอลในชั้นแอปปลิเคชันซึ่งทำงานอยู่บนชุดโปรโตคอล TCP/IP โปรโตคอลเหล่านี้ไม่ได้เป็นโปรโตคอลเฉพาะสำหรับ VoIP ดังนั้นจะกล่าวถึงอย่างคร่าวๆ ในส่วนนี้ แต่สำหรับโปรโตคอล H.323 และ SIP ซึ่งเป็นโปรโตคอลหลักสำหรับ VoIP จะกล่าวถึงรายละเอียดในหัวข้อถัดไป

2. โปรโตคอล H.323 (H.323 Protocol)

H.323 เป็นมาตรฐานหรือโปรโตคอลสำหรับการสื่อสารแบบพหุสื่อ (multimedia communication) แบบเวลาจริง บนเครือข่าย IP โปรโตคอล H.323ได้ให้รายละเอียดสำหรับขั้นตอนในการสร้างการเรียก (call setup) เอนทิตี้ภายในเครือข่าย H.323 และการทำงานร่วมกันระหว่างเอนทิตี้ภายใน

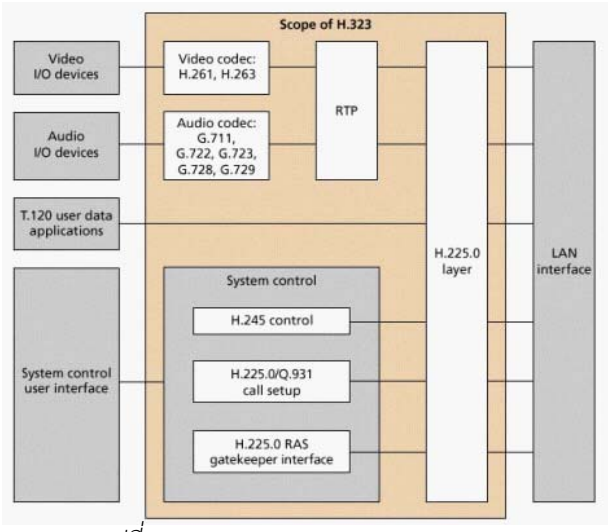


รูปที่ 1. แสดง IP telephony protocol stack

เครือข่าย H.323 ถูกพัฒนาโดย ITU-T โดยเป็นส่วนหนึ่งของมาตรฐาน H.32x ที่เป็นมาตรฐานสำหรับการประชุมแบบพหุสื่อ (multimedia conference) บนเครือข่ายต่างๆ เช่น H.320 สำหรับเครือข่าย ISDN (Integrated Service Digital Networks) H.324 สำหรับเครือข่าย PSTN (Public Switching Telephone Networks) H.323 จะครอบคลุมโปรโตคอลอื่นไว้ คือ H.225.0 สำหรับ call signaling และการจัดรูปแบบแพ็กเกตมีเดีย (media packet format) H.245 สำหรับการแลกเปลี่ยนข้อมูลความสามารถเกี่ยวกับมีเดีย (media capability exchange) และการควบคุมช่องสัญญาณมีเดีย (media channel control) H.450.x เป็นขั้นตอนสำหรับสร้างการบริการเพิ่มเติม (supplementary service) และ H.235 เป็นมาตรฐานเกี่ยวกับความปลอดภัย เป็นต้น รวมทั้งยังได้อ้างอิงถึงมาตรฐานในการเข้ารหัสสำหรับสัญญาณเสียง เช่น G.711 G723.1 G.729 และสัญญาณวีดีโอเช่น H.261 และ H.263

2.1 สถาปัตยกรรมของ H.323 (H.323 Architecture)

โปรโตคอล H.323 ครอบคลุมและอ้างอิงถึงโปรโตคอลอื่นๆ เช่น H.225.0/Q.931[5] H.245[6] และH.225.0/RAS[5] เพื่อช่วยในการทำงานของโปรโตคอล H.323 โดยสถาปัตยกรรมของโปรโตคอล H.323 สำหรับ endpoint (หรือ terminal) จะเป็นดังรูปที่ 2.



รูปที่ 2. แสดง H.323 terminal architecture

ขอบเขตของ H.323 ดังแสดงในรูปที่ 2. จะจำกัดอยู่ที่มาตรฐานในการบีบอัดข้อมูล (compression) รูปแบบแพ็กเกตมีเดีย (media packet format) การส่งสัญญาณ (signalling) และ การควบคุมการรับส่งข้อมูล (flow control) ซึ่งจะมีรายละเอียดดังนี้

- การเข้ารหัส/ถอดรหัสสัญญาณวิดีโอ (video codec) ทำหน้าที่ในการเข้ารหัสสัญญาณวีดีโอสำหรับการส่ง และถอดรหัสสัญญาณวีดีโอที่ได้รับซึ่งจะถูกนำไปแสดงผลต่อไป มาตรฐานการเข้ารหัส/ถอดรหัสที่ endpoint จำเป็นต้องเข้า/ถอดรหัสได้คือ H.261 ที่ระดับความละเอียด QCIF (Quarter Common Intermediate Format) ส่วน H.263 ซึ่งให้คุณภาพที่ดีกว่าเป็นตัวเลือกที่อาจจะรองรับหรือไม่ก็ได้ รายละเอียดเกี่ยวกับการเข้า/ถอดรหัสสัญญาณวีดีโอจะตกลงกันระหว่าง endpoint ในช่วงของการแลกเปลี่ยนความสามารถ (capability exchange) โดยการ

ใช้โปรโตคอล H.245 มาตรฐานการเข้ารหัส/ถอดรหัสที่จะใช้จะต้องรองรับโดยทุกๆ endpoint ที่เข้าร่วมในการสื่อสาร

- การเข้ารหัสสัญญาณเสียง (audio codec) ทำหน้าที่ในการเข้ารหัสเสียงจากไมโครโฟน หรือแหล่งกำเนิดอื่นสำหรับการส่ง และถอดรหัสสัญญาณที่ถูกเข้ารหัสที่รับได้ซึ่งจะถูกส่งต่อไปยังลำโพง มาตรฐานที่ endpoint จำเป็นต้องรองรับ คือ G.711 สำหรับ G.722 G.728 G.729 MPEG-1 และ G.723.1 เป็นตัวเลือกที่อาจจะรองรับหรือไม่ก็ได้ มาตรฐานการเข้ารหัส/ถอดรหัสที่ใช้จะต้องรองรับโดยทุก endpoint ซึ่งจะทำการตกลงกันโดยใช้โปรโตคอล H.245
- ช่องสัญญาณส่งข้อมูล (data channel) มาตรฐานที่ใช้ คือ T.120 สำหรับการสร้างการประชุมข้อมูล (data conferencing) รายละเอียดหรือพารามิเตอร์อาจจะตกลงกันโดยใช้โปรโตคอล H.245
- RTP (real time transport protocol) ทั้งสัญญาณเสียงและวีดีโอจะถูกส่งโดยบรรจุในแพ็กเกต RTP ซึ่งเป็นโปรโตคอลที่ใช้สำหรับการส่งข้อมูลแบบเวลาจริง บนเครือข่าย IP โดย RTP จะทำงานร่วมกับ RTCP ซึ่งทำหน้าที่ในการควบคุมการส่งข้อมูลโดย RTP ดังที่ได้กล่าวมาแล้ว

System Control Unit เป็นส่วนที่ทำหน้าที่เกี่ยวกับ การส่งสัญญาณ และ flow control ประกอบด้วยส่วนต่างๆ ดังนี้

- H.245 เป็นโปรโตคอลควบคุมมีเดีย (media control protocol) ทำหน้าที่ในการแลกเปลี่ยนความสามารถ (capability exchange) ตกลงรายละเอียดของช่องสัญญาณ (channel negotiation) เปลี่ยนโหมดของมีเดีย (switching of media mode) และการสร้างช่องสัญญาณทางตรรก (logical channel) สำหรับการส่งเสียงหรือวิดีโอ โปรโตคอลนี้จะใช้ TCP ในการส่งแมสเสจโดยใช้ช่องสัญญาณในการส่งของตัวเอง
- H.225.0/Q.931 เป็นโปรโตคอล H.225.0 ในส่วนที่ทำหน้าที่สร้างการเชื่อมต่อ(connection establishment) ซึ่งถูกดัดแปลงมาจาก โปรโตคอล Q.931 โดยโปรโตคอลนี้จะใช้ TCP ในการส่งแมสเสจผ่านช่องสัญญาณของตัวเอง
- H.225.0 RAS เป็นโปรโตคอล H.225.0 ที่ทำหน้าที่ในการควบคุมการยอมรับ (admission control) การลงทะเบียน (registration) และการรายงานสถานะ โปรโตคอลนี้จะใช้ระหว่าง endpoint และ gatekeeper เพื่อใช้สำหรับการควบคุมดูแลโดย gatekeeper โดยแมสเสจในโปรโตคอลจะใช้ UDP
- H.225.0 layer โปรโตคอล H.225.0 เป็นโปรโตคอลสำหรับ call-signaling ซึ่งมีหน้าที่ดังที่กล่าวมาข้างต้น นอกจากนั้นยังทำหน้าที่ในการแปลงมีเดีย (วิดีโอ เสียง และข้อมูล) และข้อมูลสำหรับการ ควบคุม (control data) ที่จะถูกส่งให้อยู่ในรูปแพ็กเกตที่เหมาะสมเพื่อส่งต่อให้กับ network interface และทำหน้าที่รับข้อมูลทั้งหมด (media data และ control data) จาก network interface เพื่อส่งให้ส่วนอื่นต่อไป

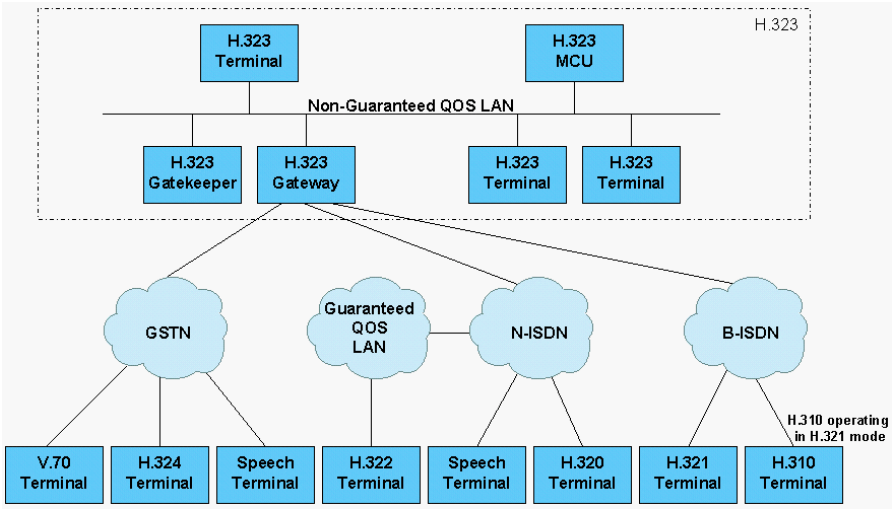
2.2 เอนทิตี้และฟังก์ชันของ H.323 (H.323 entities & functions)

ในมาตรฐาน H.323 ได้อธิบายถึงเอนทิตี้ที่เป็นองค์ประกอบเครือข่าย H.323 ซึ่งได้แก่ เทอร์มินัล MCU (Multipoint Control Unit) เกทเวย์ และ gatekeeper การเชื่อมต่อระหว่างเอนทิตี้ภายในเครือข่าย H.323 กับเครือข่ายอื่นดังแสดงในรูปที่ 3. รายละเอียดของแต่ละเอนทิตี้มีดังนี้

2.2.1 H.323 เทอร์มินัล (H.323 terminal)

เทอร์มินัลเป็น endpoint ของเครือข่ายซึ่งอาจจะเป็นคอมพิวเตอร์หรือชุดอุปกรณ์ที่สามารถใช้งานโปรโตคอล H.323ได้ เทอร์มินัลต้องสนับสนุนการสื่อสารโดยใช้เสียง ส่วนสัญญาณวีดีโอและข้อมูลเป็นตัวเลือก ซึ่งฟังก์ชันหลักของเทอร์มินัลมีดังนี้

- ทำหน้าที่ในการติดต่อกับผู้ใช้ โดยรับคำสั่งและแสดงผลให้กับผู้ใช้
- จัดการในการส่ง call signaling ให้กับ voice gateway
- ส่งหมายเลขโทรศัพท์ (มาตรฐาน E.164) และหมายเลข IP ของผู้ใช้ ให้กับ gatekeeper ซึ่งเป็นหมายเลขที่ใช้อ้างอิงถึงในการเชื่อมต่อ ในการส่งหมายเลขดังกล่าวจะบรรจุอยู่ในแมสเสจ ARQ ของโปรโตคอล H.225.0/ RAS ซึ่งอาจจะมีหมายเลข alias address ส่งไปพร้อมกัน
- ทำการแปลงแพ็กเกตที่ได้รับจากเครือข่าย โดยผ่านกระบวนการของโปรโตคอลในชั้นต่างๆ ตามลำดับ (IP->UDP->RTP) เป็นเฟรมเสียงแล้วทำการถอดรหัส G.xxx ให้อยู่ในรูปของ PCM (Pulse Code Modulation) สตรีมเพื่อทำการส่งให้กับซาวน์การ์ดเพื่อแสดงผล ต่อไป
- ทำการเข้ารหัส G.xxx ให้กับ PCM สตรีมจากซาวน์การ์ดและทำการรวมเป็นแพ็กเกต แล้วแปลงแพ็กเกตเป็นแพ็กเกตที่ส่งในเครือข่ายโดยผ่านกระบวนการของโปรโตคอลในชั้นต่างๆ (RTP->UDP->IP) แล้วจึงทำการส่งผ่านเครือข่ายในรูปของแพ็กเกต



รูปที่ 3.แสดงเครือข่าย H.323 เอนทิตี้และฟังก์ชัน

2.2.2 H.323 เกทเวย์ (H.323 Gateway)

เกทเวย์เป็นเอนทิตี้ที่ทำหน้าที่ในการเชื่อมต่อระหว่างเครือข่าย H.323 กับเครือข่ายอื่นซึ่งอาจจะไม่จำเป็นต้องมีในกรณีที่ไม่มีการเชื่อมต่อกับเครือข่ายชนิดอื่นๆ การเชื่อมต่อเครือข่าย H.323 กับเครือข่ายอื่นโดยใช้เกทเวย์จะมีลักษณะดังรูปที่ 3. เกทเวย์ทำหน้าที่เสมือนเป็น endpoint ของเครือข่ายหนึ่งในการเชื่อมต่อระหว่างเครือข่าย โดยจะทำหน้าที่ในการเชื่อมต่อระหว่างเครือข่ายดังนี้

- สร้างการเชื่อมต่อกับเทอร์มินัล PSTN ในระบบแอนาลอก
- สร้างการเชื่อมต่อกับเทอร์มินัลที่รองรับมาตรฐาน H.320 บนเครือข่าย switched circuit ที่เป็น ISDN
- สร้างการเชื่อมต่อเทอร์มินัลที่รองรับมาตรฐาน H.324 บน เครือข่าย PSTN

เนื่องจากเกทเวย์สามารถให้การเชื่อมต่อระหว่าง H.323 กับเครือข่ายอื่น ดังนั้นฟังก์ชันของเกทเวย์ จึงเป็นฟังก์ชันในการแปลงข้อมูลระหว่าง 2 เครือข่ายคือ

- รับและประมวลผลการเรียกที่มาจากเทอร์มินัลในเครือข่ายอื่นไปยังเทอร์มินัล H.323 เกทเวย์จะต้องทำการแปลง การส่งสัญญาณและ control ต่างๆ จากเครือข่ายอื่นมาเป็นของ H.323 เช่น จากขั้นตอนในการสร้างการสื่อสารจาก H.242 เป็น H.245 รวมทั้งทำหน้าที่สร้างและสิ้นสุดการเรียก หรืออาจจะมองได้ว่าเกทเวย์จะทำหน้าที่แทนเทอร์มินัลในเครือข่ายอื่นโดยเสมือนกับเป็นเทอร์มินัลในเครือข่าย H.323
- รับและประมวลผลการเรียกจากเทอร์มินัล H.323ไปยังเครือข่ายอื่น เกทเวย์จะต้องทำการแปลงการส่งสัญญาณ และ control ต่างๆ ตาม H.323 ให้เป็นมาตรฐานในเครือข่ายอื่น เช่น แปลงจากโปรโตคอล H.245 เป็น H.242 รวมทั้งสร้างและสิ้นสุดการเรียก หรืออาจจะมองว่าเกทเวย์จะทำหน้าที่แทนเทอร์มินัลในเครือข่าย H.323 เสมือนกับเป็นเทอร์มินัลในเครือข่ายอื่น
- ทำหน้าที่ในการดูแลกระบวนการการเรียก (call) และการส่งสัญญาณ (signaling) ว่ามีความผิดพลาดเกิดขึ้นหรือไม่ ซึ่งการทำงานจะอยู่ภายใต้การควบคุมของ gatekeeper
- ทำการเข้ารหัส G.xxx ให้กับสัญญาณ PCM จากเครือข่ายอื่น แล้วทำการรวมสัญญาณที่ถูกเข้ารหัสให้เป็นแพ็กเกตเพื่อส่งไปในเครือข่าย IP โดยผ่านกระบวนการแปลงของโปรโตคอลในชั้นต่างๆ เพื่อให้อยู่ในรูปแพ็กเกตที่สามารถส่งไปในเครือข่าย ดังในหัวข้อที่ 2
- ทำการแปลงแพ็กเกตจากเครือข่าย IP กลับแพ็กเกตเสียง แล้วทำการถอดรหัส G.xxx และแปลงสัญญาณ PCM เพื่อส่งให้กับเครือข่าย ภายนอก ดังในหัวข้อที่ 2.1

สำหรับฟังก์ชันเพิ่มเติมของเกทเวย์ไม่ได้มีการกำหนดไว้แน่นอนขึ้นอยู่กับผู้ออกแบบ เช่น จำนวนของเทอร์มินัลที่รองรับจำนวนการเชื่อมต่อกับเครือข่าย switched circuit และจำนวนการประชุมที่สนับสนุน เป็นต้น เมื่อเกทเวย์มีการพัฒนามากขึ้นจะทำให้เครือข่าย H.323 สามารถเชื่อมต่อกันกับเครือข่ายชนิดอื่นๆ ได้มากขึ้น

2.2.3 H.323 Gatekeeper

Gatekeeper ทำหน้าที่ในการดูแลและให้บริการกับเอนทิตี้อื่นภายในโซน โดยโซนจะประกอบไปด้วย gatekeeper 1 ตัวและเอนทิตี้อื่นๆ ทั้งหมดที่ลงทะเบียนกับ gatekeeper ถึงแม้ว่า gatekeeper เป็นเอนทิตี้ที่ไม่จำเป็นต้องมีในเครือข่าย H.323 แต่ gatekeeper ก็เป็นเอนทิตี้ที่สำคัญมาก ด้วยเหตุผลต่างๆ ดังนี้

- เครือข่ายขนาดใหญ่สามารถแบ่งเป็นหลายโซน ซึ่งแต่ละโซนจะอยู่ภายในการดูแลของ gatekeeper เพื่อความสะดวกในการดูแลรักษาเครือข่าย
- gatekeeper สามารถให้ความปลอดภัยในการเข้าถึงเครือข่ายได้โดยการให้บริการ authentication สำหรับแต่ละการเรียก หรือแต่ละเอนทิตี้
- gatekeeper เป็นศูนย์กลางในการ authentication authorization และ admission (เรียกรวมกันว่า AAA) ของโซน
- gatekeeper สามารถจัดการควบคุมแบนด์วิดท์(bandwidth management) เช่นการจำกัดจำนวนของการเชื่อมต่อ

เพื่อเป็นการรักษาแบนวิดท์สำหรับการใช้งานอย่างอื่น เช่น อีเมล์ การโอนย้ายไฟล์ gatekeeper ไม่สามารถเป็น endpoint ของการเชื่อมต่อได้ เมื่อมี gatekeeper ทุกเอนทิตี้จะต้องทำการลงทะเบียนกับ gatekeeper ดังนั้น gatekeeper จึงทำหน้าที่เป็นศูนย์กลางของการเรียกทั้งหมดภายในโซนและอาจจะให้บริการเพิ่มเติมกับโซนได้ สำหรับฟังก์ชันหลักที่จำเป็นของ gatekeeper ตามมาตรฐาน H.323 มี 4 ฟังก์ชันดังนี้

- การแปลงแอดเดรส (Address translation) gatekeeper จะทำหน้าที่ในการแปลง alias addressให้เป็น transport address เอนทิตี้จะทำการส่ง alias พร้อมกับการลงทะเบียนโดยใช้แมสเสจ RRQ ซึ่งอาจจะสามารถปรับเปลี่ยนในภายหลังได้
- การควบคุมการยอมรับ (Admission control) เมื่อเอนทิตี้ภายในโซนต้องการสร้างการเรียก จะต้องทำการขออนุญาตไปยัง gatekeeper โดยใช้แมสเสจ ARQ gatekeeper อาจจะอนุญาตหรือไม่ก็ได้โดยจะทำการตรวจสอบจากเงื่อนไขต่างๆ เช่น แบนด์วิดท์ แหล่งกำเนิดการเรียก (call) และ authentication เป็นต้น
- การควบคุมแบนด์วิดท์ (Bandwidth control) gatekeeper สามารถรองรับการควบคุมแบนด์วิดท์ได้ เอนทิตี้จะทำการร้องขอแบนด์วิดท์ที่ต้องการโดยใช้แมสเสจ BRQ และ gatekeeper จะทำการตรวจสอบค่าแบนด์วิดท์ที่ร้องขอกับเงื่อนไขที่กำหนดไว้สำหรับการจัดการแบนด์วิดท์ (bandwidth management) แล้วจึงจะอนุญาตหรือไม่อนุญาตด้วยการส่งแมสเสจ BCF หรือ BRJ ตามลำดับ
- การจัดการโซน (Zone management and Directory service) จากรูปที่ 8 โซนจะประกอบด้วยเทอร์มินัล เกทเวย์และ MCU ทั้งหมดที่ลงทะเบียนกับ gatekeeper 1 ตัว gatekeeper ทำหน้าที่ในการดูแลและจัดการให้กับทุกเอนทิตี้ที่อยู่ในภายในโซน โดยการใช้ฟังก์ชันข้างต้นและการให้บริการอื่นๆ รวมทั้งการให้บริการ directory service ของโซน

นอกจากฟังก์ชันหลักดังกล่าวแล้ว gatekeeper อาจจะให้ฟังก์ชันเพิ่มเติมอื่นๆ มีดังนี้

- การควบคุมการส่งสัญญาณ (call control signaling) gatekeeper อาจจะช่วยในการประมวลผลแมสเสจ Q.931 ที่ส่งระหว่าง เทอร์มินัลได้ในระหว่างการสร้างการเรียก
- การตรวจสอบการเรียก (call authorization) gatekeeper อาจจะปฏิเสธการสร้างการเรียก จากเทอร์มินัลด้วยเหตุผลบางอย่าง เช่น จำกัดการเข้าถึงจากเทอร์มินัลหรือเกทเวย์บางตัว จำกัดการเข้าถึงในบางช่วงเวลา เป็นต้น ซึ่งเงื่อนไขในการตรวจสอบจะอยู่นอกเหนือขอบเขตของ H.323
- การจัดการแบนด์วิดท์ gatekeeper จะปฏิเสธการเรียกจาก เทอร์มินัลในกรณีที่มีแบนด์วิดท์ไม่เพียงพอ รวมถึงในกรณีที่มีการร้องขอการเพิ่มแบนด์วิดท์ สำหรับเงื่อนไขจะอยู่นอกเหนือขอบเขตของ H.323
- การจัดการการเรียก (call management) gatekeeper อาจจะทำการเก็บรักษารายการการเรียกที่เกิดขึ้นเพื่อใช้ในการระบุเทอร์มินัลที่ถูกเรียกว่าว่างหรือไม่ หรือเพื่อให้ข้อมูลกับฟังก์ชันในการจัดการแบนด์วิดท์
- การตรวจสอบผู้ใช้ (authenticating users) สามารถจำกัดการ เข้าถึงของผู้ใช้ได้ตามเงื่อนไขที่กำหนดไว้
- การจัดการบริการ(managing services) gatekeeperจะทำหน้าที่ในการจัดการให้บริการต่างๆ แก่ผู้ใช้
- การจัดการฐานข้อมูลของสมาชิก (managing subscriber databases) gatekeeper ทำหน้าที่ดูแลและจัดการเกี่ยวกับฐานข้อมูลของสมาชิกที่ได้ลงทะเบียนไว้กับ gatekeeper
- การหาตำแหน่งของสมาชิก (locating subscribers) gatekeeper ทำหน้าที่ในการหาตำแหน่งของสมาชิกได้โดยการค้นหาจากข้อมูลของสมาชิก ที่สมาชิกได้จากการลงทะเบียน
- การรวบรวมข้อมูลสำหรับการเก็บค่าบริการ (collecting charging information) gatekeeper จะทำการเก็บข้อมูลที่จำเป็นสำหรับการ

คิดค่าบริการของการเรียก โดยที่การเรียก (call) ต้องถูกจัด เส้นทางผ่าน gatekeeper

- การควบคุมเกทเวย์ (managing gateway) gatekeeper จะควบคุมการทำงานของเกทเวย์ เช่นควบคุมการสร้างการเรียก ของเกทเวย์ระหว่างเครือข่าย
- การช่วยในการสร้างการเรียก (assisting in call setup) เมื่อการเรียกถูกจัดเส้นทางผ่าน gatekeeper จะช่วยในการสร้างการเรียก เช่นอาจจะทำการจัดเส้นทางให้กับการเรียก ไปยังเกทเวย์ที่เหมาะสม

การติดต่อสื่อสารระหว่างเอนทิตี้กับ gatekeeper จะใช้โปรโตคอล H.225.0 /RAS ส่วนแมสเสจ call signaling (H.225.0/ Q.931) และ media control (H.245) อาจจะผ่าน gatekeeper หรือไม่ก็ได้ขึ้นอยู่กับการลงทะเบียนของเอนทิตี้ และเงื่อนไขของ gatekeeper สำหรับ gatekeeper อาจจะถูกรวมอยู่ในเกทเวย์และ MCUได้ โดยที่ต้องแยกทางตรรก (logical) จาก endpoint

2.2.4 Multipoint Control Unit (MCU)

MCU ทำหน้าที่ในการสนับสนุนการประชุมแบบหลายจุด (multipoint conference) ระหว่างเทอร์มินัล 3 เทอร์มินัลขึ้นไป MCU เป็นเอนทิตี้ที่จะมีหรือไม่ก็ได้ MCU ประกอบด้วย multipoint controller (MC) และ multipoint processor (MP) ในการประชุมจะต้องมี MC ส่วน MP อาจจะมีหรือไม่มีก็ได้ หรืออาจจะมีมากกว่าหนึ่งก็ได้ MC เป็นส่วนที่ทำหน้าที่ในจัดการเกี่ยวกับการส่งสัญญาณในการควบคุมมีเดีย (media control signaling) ให้กับแต่ละเทอร์มินัล โดยที่ทุกเทอร์มินัลต้องมีช่องสัญญาณ H.245 เชื่อมต่อกับ MC แบบจุดถึงจุด (point-to-point) ส่วน MP จะทำหน้าที่ในการจัดการกับมีเดียสตรีมโดยทำหน้าที่ในการผสม (mixing) สวิตช์ (switching) และประมวลผลมีเดียที่ใช้การประชุมภายใต้การ ควบคุมของ MC

2.3 Multipoint Conference

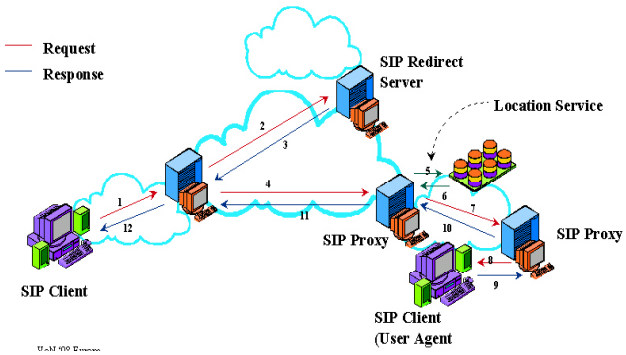
การประชุมแบบหลายจุด (multipoint conference) คือการสื่อสารที่มีผู้เข้าร่วมมากกว่า 2 ซึ่งจำเป็นต้องมี MC อยู่เป็นอย่างน้อย สำหรับแบบจำลองที่ใช้มี 3 แบบ

- Centralize Model ในแบบจำลองนี้จำเป็นต้องมี MCU อยู่ทุก เทอร์มินัลที่เข้าร่วมในการประชุมต้องมีช่องสัญญาณ H.245 เชื่อมต่อแบบจุดถึงจุด (point-to-point) กับ MCU ซึ่ง MC จะหน้าที่ควบคุมการประชุมโดยใช้ฟังก์ชันของ H.245 ส่วน MP จะทำหน้าที่รับมีเดียสตรีมจากทุกเทอร์มินัลทำการรวมสัญญาณเสียง เลือกสัญญาณวีดีโอที่ตรงกัน และประมวลผล แล้วทำการส่งกลับไปให้กลับเทอร์มินัลอื่นๆ ทุกเทอร์มินัล
- Decentralized Model ในแบบจำลองนี้ เทอร์มินัลจะมัลติคาสท์สัญญาณเสียงและวีดีโอให้กับเทอร์มินัลอื่นๆ โดยไม่ผ่าน MCU แต่การควบคุมยังคงถูกควบคุมโดย MC ผ่านทางช่องสัญญาณ H.245 ที่เชื่อมต่อกับเทอร์มินัลแบบจุดถึงจุด (point-to-point) เทอร์มินัลที่ได้รับสัญญาณจะทำหน้าที่ในการประมวลผลสัญญาณเอง โดยอาจจะใช้ฟังก์ชัน MP ของแต่ละเทอร์มินัลช่วยทำหน้าที่ในการประมวลผลมีเดียสตรีม
- Hybrid Model แมสเสจ H.245 รวมทั้งสัญญาณเสียงหรือวีดีโอจะถูกส่งและประมวลผลผ่านMCUโดยใช้การเชื่อมต่อแบบจุดถึงจุด (point-to-point) ส่วนสัญญาณที่เหลือจะถูกส่งโดยเทอร์มินัลแบบมัลติคาสท์ให้กับเทอร์มินัลอื่นๆ

3. โปรโตคอล SIP (SIP: Session Initial Protocol)

SIP เป็นโปรโตคอลใช้งานสำหรับ IP Telephony ที่กำหนดโดย IETF (Internet Engineering Task Force) SIP เป็นโปรโตคอลในชั้นแอปปลิเคชันซึ่งทำหน้าที่ในการสร้าง สิ้นสุด และเปลี่ยนแปลงแก้ไข เซสชันของพหุสื่อ (multimedia session) หรือ การเรียก ซึ่งรวมถึง Internet telephony การประชุมแบบพหุสื่อ (multimedia conference) และแอปปลิเคชันอื่นที่คล้ายคลึงกัน SIP เป็นโปรโตคอลไคล์เอนท์-เซิร์ฟเวอร์ (client-server) โดยใช้การส่งข้อมูลในรูปของตัวอักษร(text based) เช่นเดียวกับโปรโตคอล HTTP (Hypertext Transfer Protocol) รวมทั้งยังมีกลไกที่คล้ายคลึงกันทำให้สามารถใช้เฮดเดอร์และกลไกที่มีอยู่บางอย่างของ HTTP ได้ สำหรับฟังก์ชันที่ SIP สนับสนุนมีดังนี้

- *User location* การกำหนด endpoint ที่ใช้ในเซสชันการสื่อสาร
- *User capabilities* การกำหนดมีเดียและพารามิเตอร์ของมีเดียที่ใช้ในการสื่อสาร
- *User availability* การกำหนดความต้องการของผู้ถูกเรียกว่าต้องการเข้าร่วมในเซสชันหรือไม่
- *Call setup* การสร้าง การเรียก และกำหนดพารามิเตอร์ของการเรียก
- *Call handling* การจัดการกับ การเรียก รวมทั้งการโอนย้าย การเรียก และการสิ้นสุดการเรียก



รูปที่ 4. แสดงสถาปัตยกรรมของ SIP

SIP ถูกพัฒนาโดย IETF โดยเป็นส่วนหนึ่งของสถาปัตยกรรมควบคุมและข้อมูลพหุสื่อ (multimedia data and control architecture) ซึ่งรวมถึงโปรโตคอล เช่น RSVP RTP RTSP และ SDP (Session Data Protocol) เป็นต้น โดย SIP สามารถใช้งานหรือทำงานร่วมกับโปรโตคอลเหล่านี้เพื่อประสิทธิภาพที่ดีขึ้น แต่ฟังก์ชันและการทำงานของ SIP ไม่ขึ้นอยู่กับโปรโตคอลเหล่านี้ สำหรับในปัจจุบัน SIP ได้ถูกพัฒนาอยู่ในเวอร์ชัน 2

3.1 สถาปัตยกรรมและองค์ประกอบของ SIP (SIP architecture & Components)

SIP เป็นโปรโตคอลไคลเอนท์-เซิร์ฟเวอร์ ไคลเอนท์จะทำหน้าที่ส่งคำร้องขอให้กับเซิร์ฟเวอร์เพื่อทำการประมวลผลแล้วจึงตอบสนองกับมายังไคลเอนท์ ในการส่งแมสเสจร้องขอ แมสเสจอาจจะถูกส่งผ่านเซิร์ฟเวอร์หลายตัวจนกระทั่งถึงเซิร์ฟเวอร์ที่สามารถตอบสนองคำร้องขอของไคลเอนท์ได้ ในระบบ SIP จะมีองค์ประกอบที่ทำหน้าที่ของไคลเอนท์และเซิร์ฟเวอร์ องค์ประกอบเหล่านี้จะทำการติดต่อสื่อสารกันโดยใช้แมสเสจ SIP ซึ่งมีสถาปัตยกรรมดังรูปที่ 4.

ใน SIP จะแบ่งองค์ประกอบเป็น 2 ชนิดหลักคือ user agent และ network server ดังรายละเอียดต่อไปนี้

- User agent เป็น endpoint ที่ทำหน้าที่แทนผู้ใช้ในการติดต่อสื่อสารเนื่องจากว่าผู้ใช้ต้องสามารถเริ่ม การเรียก หรือตอบสนองต่อการเรียก ที่เข้ามา ดังนั้น user agent ควรจะสามารถทำหน้าที่เป็นได้ทั้งไคลเอนท์และเซิร์ฟเวอร์ในกรณีที่มีการเริ่ม การเรียก ผู้ใช้จะทำหน้าที่เป็น ไคลเอนท์เพื่อทำการร้องขอไปยังผู้ถูกเรียกซึ่งจะทำหน้าเป็นเซิร์ฟเวอร์ในการตอบสนองการร้องขอ โดยทั่วไป user agent จึงประกอบด้วยส่วนที่ทำหน้าที่เป็นไคลเอนท์และเซิร์ฟเวอร์ดังนี้
  1. User agent client (UAC) จะทำหน้าที่ในการเริ่ม การเรียก โดยการส่งส่งแมสเสจร้องขอไปยังผู้ถูกเรียกโดยผ่านทาง network server
  2. User agent server (UAS) จะทำหน้าที่ในการรับคำร้องขอ และตอบสนองต่อคำร้องขอโดยจะรอการตอบสนองจากผู้ใช้ ซึ่งการตอบสนองอาจจะเป็นการยอมรับหรือปฏิเสธ การเรียก ในกรณีที่ผู้ใช้มีการใช้งานเทอร์มินัลหลายตัว ผู้ใช้ยังอาจจะกำหนดให้ UAS ทำการ redirect ไปยังที่ UAS อื่นที่ผู้ใช้ใช้งานอยู่จริง
- Network server เป็นเซิร์ฟเวอร์ภายในเครือข่ายซึ่งจะทำหน้าที่ในการจัดการกับแมสเสจที่ได้รับ โดยอาจจะได้รับจาก user agent หรือ network server อื่นๆ การจัดการกับแมสเสจจะขึ้นกับชนิดของเซิร์ฟเวอร์ ซึ่งมี 2 ชนิดคือ
  1. Proxy server เซิร์ฟเวอร์จะทำการกำหนดเอนทิตี้ที่จะรับ ข้อมูลต่อไป โดยอาจจะเป็น UAS หรือ network server ก็ได้ จากนั้นเซิร์ฟเวอร์จะเป็นผู้ทำการร้องขอไปยังเอนทิตี้นั้น พร้อมกับข้อมูลตอบสนองให้กับ UAC (หรืออาจจะเป็น network server อื่นที่ส่งข้อมูลร้องขอมา) เพื่อระบุว่ากำลังรอการตอบสนองจากผู้ถูกเรียก เมื่อเซิร์ฟเวอร์ได้รับการตอบสนองจากผู้ถูกเรียกหรือ UAS เซิร์ฟเวอร์จึงจะส่งแมสเสจตอบสนองต่อกลับไปให้กับ UAC ดังรูปที่ 22. เซิร์ฟเวอร์ชนิดนี้จะทำหน้าที่เป็นทั้งไคลเอนท์และเซิร์ฟเวอร์ ในกรณีที่ส่งแมสเสจร้องขอจะเป็นไคลเอนท์ส่วนในกรณีที่ส่งข้อมูลตอบสนองจะเป็นเซิร์ฟเวอร์
  2. Redirect server เมื่อเซิร์ฟเวอร์ได้รับแมสเสจร้องขอแล้วจะกำหนดเอนทิตี้ที่จะรับข้อมูลต่อไป จากนั้นเซิร์ฟเวอร์จะส่งแอดเดรสของเอนทิตี้นั้นไปให้กับ UAC หรือ network server ที่ส่งข้อมูลร้องขอมา เมื่อ UAC (หรือ network server) ได้รับแอดเดรส

แล้วจึงจะทำการส่งคำร้องไปยังเซิร์ฟเวอร์นั้นด้วยตนเอง ดังรูปที่ 4.

เนื่องจากว่าผู้ใช้อาจจะมีการเปลี่ยนเทอร์มินัลที่ใช้งานได้ network server จึงจะต้องสามารถกำหนดเอนทิตี้ที่รับข้อมูลเพื่อให้สามารถส่งแมสเสจให้กับผู้ถูกเรียกได้ โดย network server จะทำการติดต่อกับ location server เพื่อกำหนดเอนทิตี้ต่อไปที่จะรับแมสเสจ location server จะทำหน้าที่ในการหาตำแหน่งปัจจุบันของผู้ถูกเรียกโดยการกำหนดเอนทิตี้ที่จะรับแมสเสจต่อไปแล้วส่งแอดเดรสของเอนทิตี้นี้ให้กับ network server ข้อมูลของ location server จะได้รับจาก registrar ซึ่งทำหน้าที่ในการรับข้อมูลเกี่ยวกับตำแหน่งของผู้ใช้แล้วส่งข้อมูลนี้จะให้กับ location server ในการให้ข้อมูลของผู้ใช้กับ registrar จะทำได้โดยใช้แมสเสจ REGISTER เพื่อบอกตำแหน่งที่อยู่ของผู้ใช้ โดยทั่วไปแล้ว registrar จะถูกรวมเข้ากับ network server

3.2 ชื่อและแอดเดรส (Addressing & Naming)

ในระบบ SIP การส่งแมสเสจระหว่างเอนทิตี้จะต้องระบุ SIP URL เพื่อใช้อ้างอิงถึงผู้ใช้SIP URL จะประกอบด้วย SIP แอดเดรส รูปแบบของแอดเดรสจะอยู่ในรูปของ name@domain โดยอาจจะเป็น user@domain user@address phone-number@gateway และ user@host แอดเดรสนี้จะถูกใช้อ้างอิงถึงผู้ใช้ทั้งผู้เรียกและผู้ถูกเรียกในการส่งแมสเสจ ตัวอย่างของ SIP URL เช่น SIP://j.doe@example.com โดยที่ URL นี้จะอยู่ในส่วนเฮดเดอร์ของแมสเสจ ในการส่งแมสเสจไปยัง SIP URL ที่ระบุไว้จะต้องมีการแปลง SIP แอดเดรสให้อยู่ในของ User@host โดยอาจจะผ่านการแปลงมากกว่าหนึ่งครั้งจนกระทั่งได้ตำแหน่งที่อยู่ของผู้ใช้ ในการแปลงแอดเดรสอาจจะใช้ DNS (Domain Name Service) หรือ LDAP (Lightweight Directory Access Protocol)

3.3 Locating Server

ในการส่งแมสเสจจะใช้ SIP URL อ้างอิงถึงในการส่ง โดยจะต้องมีการแปลงส่วน domain ของ SIP แอดเดรสไปเป็นหมายเลข IP ซึ่งเป็น แอดเดรสของ SIP server ที่สามารถค้นหาตำแหน่งของผู้ใช้ต่อไปได้ การแปลง SIP แอดเดรสอาจจะทำโดย UAC หรือ UAC จะส่งแมสเสจให้กับเซิร์ฟเวอร์ที่กำหนดซึ่งเซิร์ฟเวอร์จะเป็นผู้ที่ทำหน้าที่ในการแปลง SIP แอดเดรสแทน ในการแปลง SIP แอดเดรสนี้สามารถใช้ DNS เข้ามาช่วยได้

3.4 Locate User

จากข้างต้นเมื่อได้ตำแหน่งของเซิร์ฟเวอร์ที่สามารถส่งข้อมูลให้กับผู้ถูกเรียกแล้วต่อไปจะเป็นการหาตำแหน่งของผู้ถูกเรียก เมื่อ SIP server ได้รับแมสเสจร้องขอแล้ว เซิร์ฟเวอร์จะต้องทำการค้นหาผู้ใช้ที่อ้างอิงถึงใน SIP แอดเดรส โดยการร้องขอข้อมูลไปยัง location server ซึ่งจะตอบกลับด้วยรายการตำแหน่งที่เป็นไปได้ของผู้ถูกเรียก เมื่อ SIP server ได้ข้อมูลเกี่ยวกับตำแหน่งของผู้ถูกเรียกแล้ว ถ้าเป็น proxy server จะทำส่งแมสเสจร้องขอต่อไปยังตำแหน่งต่างๆ ตามรายการที่ได้รับจาก location serverไว้ โดยอาจจะส่งแบบ sequential หรือ parallel ส่วนถ้าเป็น redirect server จะส่งรายการตำแหน่งของผู้ถูกเรียกไปให้ผู้เรียกผ่านโดยใช้เฮดเดอร์ contact เพื่อให้ผู้เรียกส่งแมสเสจร้องขอไปเอง สำหรับตำแหน่งของผู้ใช้จะต้องทำการลงทะเบียนกับ registrar โดยใช้เฮดเดอร์ REGISTER รวมทั้งยังอาจจะอัพโหลด script ของผู้ใช้เองเพื่อเก็บไว้ที่เซิร์ฟเวอร์สำหรับจัดการกับการเรียกตามความต้องการของผู้ใช้

3.5 ความเชื่อถือได้ (Reliability)

ในระบบ SIP จะมีกลไกเรื่องความเชื่อถือได้ (reliability) ไม่ว่าจะใช้ โปรโตคอล UDP หรือ TCP โดยการใช้เมธอด Ack ไคลเอนท์จะส่ง แมสเสจร้องขอใหม่ตามช่วงเวลาที่กำหนดจนกระทั่งได้รับแมสเสจตอบจากเซิร์ฟเวอร์ ทางด้านเซิร์ฟเวอร์ก็จะส่งแมสเสจตอบจนกระทั่งได้รับ แมสเสจ Ack จากไคลเอนท์จึงทำให้การร้องขอที่สมบูรณ์ต้องใช้การ แลกเปลี่ยนแมสเสจ 3 แมสเสจ เซิร์ฟเวอร์อาจจะตอบสนองต่อ Ack โดยการส่งแมสเสจตอบสุดท้ายไปให้กับไคลเอนท์ซึ่งอาจจะไม่จำเป็นต้องมีก็ได้ สำหรับการส่งมีเดียสตรีมเซิร์ฟเวอร์จะยอมให้มีการส่งเมื่อได้รับ Ack จากไคลเอนท์เท่านั้นด้วยกลไกนี้จึงทำให้เกิดความเชื่อถือได้ในการ แลกเปลี่ยนแมสเสจโดยไม่จำเป็นต้องอาศัยกลไกของโปรโตคอลในชั้นต่ำกว่า เช่น TCP

3.6 ความสามารถในการขยาย (Protocol xtension)

SIP สามารถรองรับคุณลักษณะใหม่ที่เพิ่มเติมขึ้นสำหรับ เมธอด เฮดเดอร์ และ status code ได้ดังนี้

- เมธอด เซิร์ฟเวอร์จะส่งแมสเสจแสดงความผิดพลาด (error message) กลับมาให้ไคลเอนท์ถ้าเมธอดที่ร้องขอมาเซิร์ฟเวอร์ไม่เข้าใจ และจะบอกเมธอดที่เซิร์ฟเวอร์เข้าใจโดยใช้เฮดเดอร์ Public และ Allow ไคลเอนท์อาจจะส่งแมสเสจร้องขอเพื่อขอทราบเมธอดที่เซิร์ฟเวอร์สนับสนุนโดยใช้ตัวเลือกที่เฮดเดอร์ (header option)

- เฮดเดอร์ เมื่อเอนทิตี้ได้รับเฮดเดอร์ที่ไม่เข้าใจ ก็จะละทิ้งเฮดเดอร์นั้น ในกรณีที่ไคลเอนท์จำเป็นต้องการใช้เฮดเดอร์บางเฮดเดอร์ ไคลเอนท์จะส่งแมสเสจเพื่อร้องขอเฮดเดอร์ที่จำเป็นต้องใช้ไปโดยระบุในเฮดเดอร์ Require หากมีเฮดเดอร์ที่เซิร์ฟเวอร์ไม่สามารถให้การสนับสนุนได้เซิร์ฟเวอร์จะตอบปฏิเสธกลับมา
- status code ได้แบ่งเป็นคลาสต่างๆ เช่นเดียวกับ response code ของโปรโตคอล HTTP ซึ่งไคลเอนท์ต้องเข้าใจในความหมายในแต่ละคลาสเพื่อที่จะได้ทราบผลของการร้องขอว่าสำเร็จหรือไม่ สำหรับ status code ในแมสเสจตอบจะมีข้อความต่อหลังซึ่งจะเป็นความหมายของ code ซึ่งสามารถอ่านเข้าใจได้ โดยถ้าไคลเอนท์ไม่เข้าใจในรายละเอียดของ code ทั้งหมด ไคลเอนท์จะตีความหมายเป็น X00 เมื่อ X เป็นตัวเลขตัวแรกของ status code และนอกจากนั้นอาจจะนำ PEP (protocol extension protocol) มาปรับปรุงใช้งานกับ SIP ได้

ในกรณีมีการส่งแมสเสจผ่านหลายเซิร์ฟเวอร์ จะใช้เฮดเดอร์ Via เพื่อระบุเซิร์ฟเวอร์ที่เป็นทางผ่านของแมสเสจทั้งหมด สำหรับใช้ในการส่งแมสเสจตอบสนองกลับไปให้ผู้เรียก ในระหว่างการส่งแมสเสจร้องขอและแมสเสจตอบสนองจะมีการตกลงเกี่ยวกับพารามิเตอร์ของเซสชันด้วย ซึ่งรายละเอียดจะอยู่ในส่วนของ message body เช่นในกรณีของการสื่อสารโดยใช้เสียง พารามิเตอร์จะเป็น IP แอดเดรส พอร์ตสำหรับ RTP และการเข้า/ถอดรหัสเสียง หลังการสร้าง การเรียก เสร็จสมบูรณ์ ช่องสัญญาณสำหรับ RTP จะถูกสร้างขึ้นทำให้ทั้งสองฝ่ายสามารถสื่อสารกันได้ รวมทั้งยังอาจจะเชิญผู้อื่นมาเข้าร่วมในเซสชันนี้ได้ ในกรณีที่ต้องการเปลี่ยนพารามิเตอร์ของเซสชัน สามารถทำได้โดยส่งแมสเสจร้องขอใหม่อีกครั้งโดยใช้เมธอด Invite ซึ่งมี call-id เดิม ไปยังผู้ร่วมเซสชันพร้อมทั้งค่าพารามิเตอร์ของเซสชันใหม่ที่ต้องการใช้ รายละเอียดในส่วนนี้จะอยู่ในส่วนของ message body ซึ่งโดยทั่วไปจะใช้โปรโตคอล SDP ในการอธิบายความหมาย

4. การเปรียบเทียบระหว่าง SIP และ H.323 (SIP and H.323 comparisons)

ในปัจจุบันมีโปรโตคอลสำหรับ IP telephony คือ H.323 ซึ่งพัฒนาโดย ITU และ SIP ซึ่งพัฒนาโดย IETF ดังที่ได้กล่าวมาแล้วโปรโตคอล H.323 ได้ถูกพัฒนาขึ้นก่อนจึงทำให้มีการใช้งานโปรโตคอลนี้มากกว่า SIP ซึ่งถูกพัฒนาขึ้นในภายหลัง แต่อย่างไรก็ตามโปรโตคอลทั้งสองก็มีทั้งข้อดีและข้อเสีย ในที่นี้จะพิจารณาในแง่ของความซับซ้อน (complexity) ความสามารถในการขยายขนาดของเครือข่าย (scalibility) ความสามารถในการเพิ่มเติมคุณลักษณะของโพรโทคอล (extensibility) ฟังก์ชันและการบริการ (functionality and services) คุณภาพการให้บริการ (QoS) และการทำงานร่วมกัน (interoperability)

การเปรียบเทียบระหว่าง H.323 และ SIP ในแง่ต่างๆ สามารถสรุปได้ใน [13] จากการเปรียบเทียบในข้างต้นจะเห็นว่า SIP สามารถให้บริการได้คล้ายกับ H.323 แต่มีความซับซ้อนน้อยกว่า ความสามารถในการเพิ่มเติมคุณลักษณะของโปรโตคอล (extensibility) มากกว่า และสามารถรองรับขนาดของเครือข่ายได้ใหญ่กว่า (scalibility) เมื่อพิจารณาถึงการพัฒนาของทั้งสองโปรโตคอล การพัฒนาจะเป็นในลักษณะที่ เรียนรู้ซึ่งกันและกัน เช่น H.323 เวอร์ชัน 3 จะมีความใกล้เคียงกับ SIP จึงอาจจะเป็นไปได้ว่าโปรโตคอลทั้งสองอาจจะมีประสิทธิภาพในการทำงานใกล้เคียงกันแต่อย่างไรก็ตาม H.323 เป็นมาตรฐานที่เกิดขึ้นก่อน ดังนั้นในปัจจุบันจึงมีการใช้งาน H.323 มากกว่า ในขณะที่ SIP เป็น โปรโตคอลใหม่จึงยังมีการใช้งานน้อยกว่า สำหรับ H.323 มีข้อได้เปรียบคือ ITU-T ซึ่งเป็นผู้พัฒนา H.323 เป็นผู้กำหนดมาตรฐานต่างๆ ในระดับล่าง ลงไปถึงในชั้นกายภาพ (physical layer) ในขณะที่ IETF ซึ่งเป็นผู้พัฒนา SIP จะเกี่ยวข้องเฉพาะในชั้นเครือข่าย (network layer) ขึ้นไป H.323 จึงอาจจะมีความเข้ากันได้หรือประสิทธิภาพในการทำงานร่วมกับเครือข่ายได้ดีกว่า SIP ส่วนข้อเสียของ H.323 คือ H.323 ค่อนข้างจะอ้างอิงไปทางแบบจำลอง circuit-switch จึงทำให้มีราคาสูงและยุ่งยากในการใช้งานจริง ในขณะที่ SIP สามารถใช้งานได้ง่ายและมีราคาถูกกว่า ตารางที่ 2 สรุปการเปรียบเทียบข้อจำกัดระหว่าง H.323 และ SIP[14] จากการพิจารณาข้างต้นจะเห็นว่าโปรโตคอลทั้งสองต่างก็มีทั้งข้อดีและข้อเสียจึงไม่อาจบอกได้ชัดว่าโปรโตคอลใดจะเป็นโปรโตคอลที่เหมาะสมสำหรับ IP telephony มากที่สุด และนอกจากนั้นก็ยังมีแนวโน้มว่าอาจจะการใช้ โปรโตคอลทั้งสองทำงานร่วมกัน ตารางที่ 3 [14] แสดงองค์ประกอบเปรียบเทียบระหว่าง H.323 และ SIP จะเห็นได้ว่ามีข้อแตกต่างกันหลายประการ

ตารางที่ 1. แสดง สรุปการเปรียบเทียบการทำงานระหว่าง H.323 และ SIP

	H.323 v1	H.323 v2	H.323 v3	SIP
FUNCTIONALITY				
CALL CONTROL SERVICES:				
Call Holding	No	Yes	Yes	Yes
Call Transfer	No	Yes	Yes	Yes
Call Forwarding	No	Yes	Yes	Yes
Call Waiting	No	Yes	Yes	Yes
ADVANCED FEATURES:				
Third Party Control	No	No	No	Yes
Conference	Yes	Yes	Yes	Yes
Click-for-Dial	Yes	Yes	Yes	Yes
Capability Exchange	Yes&Better	Yes &Better	Yes &Better	Yes
QUALITY OF SERVICE				
Call Setup Delay	6~7 RT	3~4 RT	2~3 RT	2~3 RT
RELIABILITY:				
Packet Loss Recovery	Through TCP	Through TCP	Better	Better
Fault Detection	Yes	Yes	Yes	Yes
Fault Tolerance	N/A	N/A	Better	Good
MANAGEABILITY				
Admission Control	Yes	Yes	Yes	No
Policy Control	Yes	Yes	Yes	No
Resource Reservation	No	No	No	No
SCALABILITY				
Complexity	More	More	More	Less
Server Processing	Stateful	Stateful	Stateful or stateless	Stateful or Stateless
Inter-Server Communication	No	No	Yes	Yes
FLEXIBILITY				
Transport Protocol Neutrality	TCP	TCP	TCP/UDP	TCP/UDP
Extensibility of Functionality	Vendor Specified			Yes, IANA
Ease of Customization	Harder	Harder	Harder	Easier
INTEROPERABILITY				
VersionCompatibility	N/A	Yes	Yes	Unknown
SCN Signaling Interoperability	Better	Better	Better	Worse
EASE OF IMPLEMENTATION				
Protocol Encoding	Binary	Binary	Binary	Text

ตารางที่ 2. แสดง สรุปการเปรียบระหว่าง H.323 และ SIP

Capability	Protocol	
	H.323	SIP
Cost	High	Low
Complexity	High	Low
Maturity	Good	Poor
Scope of Definition	Full	Limited
Interoperability	Good	Some
SS7 Compatibility	Poor	Poor
Similar to ISDN	Yes	No

ตารางที่ 3. แสดงสรุปการเปรียบเทียบองค์ประกอบระหว่าง H.323 และ SIP

Requirement	The H.323 Way	The SIP Way
Call signaling	**H.225/Q.931**	**SIP INVITE Request/Response, ACK, BYE**
Media Negotiation	**H.245 Capability exchange or Q.931 Fast Start method**	**SDP embedded in SIP INVITE Req/Rsp, ACK**
Registration, Location and admission service	**H.225 RAS Protocol to gatekeeper**	**SIP REG request, to a variety of location services. Proxies locate things in processing an INVITE.**
Media transmission	**RTP**	**RTP**
Centralized Call Control	**Gatekeeper-routed call signaling**	**Use of SIP proxies**

5. การโปรแกรมสำหรับเทเลโฟนนี

(Telephony Application Programming Interface: TAPI)

ในการพัฒนาโปรแกรมสำหรับ IP telephony ในปัจจุบันมีเครื่องมือ (tool) ช่วยในการเขียน เช่น API ต่างๆ ตัวอย่างเช่น JTAPI (Java Telephony API) และ TAPI (Telephony API) รวมทั้งยังได้มีผู้ผลิต Protocol stack ทำให้การสร้างแอปปลิเคชันสำหรับ IP telephony ทำได้สะดวกขึ้น เนื่องจากเครื่องมือเหล่านี้ได้ซ่อนรายละเอียดบางอย่างพร้อมทั้งได้ห้องค์ประกอบร่วมสำหรับแอปปลิเคชันในลักษณะเดียวกันซึ่งช่วยลดภาระในการพัฒนาโปรแกรมได้มาก

5.1 JTAPI (Java Telephony API)

JTAPI เป็นอินเตอร์เฟซในการเขียนโปรแกรมสำหรับ computer telephony ซึ่งถูกพัฒนาโดยบริษัท SUN (Sun Micro-System Co., LTD.) JTAPI ถูกออกแบบมาเพื่อให้สามารถเขียนโปรแกรมได้ง่ายแต่มีคุณลักษณะต่างๆ ครบ ซึ่งสามารถใช้ในการเขียนโปรแกรมได้หลากหลายตั้งแต่โปรแกรม call center ไปจนถึงการพัฒนาเว็บเพจ และยังสนับสนุนการควบคุมการเรียกจากแบบบุคคลที่หนึ่งและสาม (first และ third party call control) แอปปลิเคชันที่ถูกพัฒนาจาก JTAPI สามารถในไปใช้ในแพลตฟอร์มและระบบโทรศัพท์ต่างๆ กันได้ เนื่องจาก JTAPI ได้ใช้ภาษา Java ซึ่งโปรแกรมที่พัฒนาโดยภาษานี้จะสามารถรันได้โดยไม่ขึ้นกับแพลตฟอร์มใดๆ

ความต้องการของผู้พัฒนา JTAPI คือต้องการทำให้การพัฒนาโปรแกรมทำได้ง่าย และโปรแกรมสามารถทำงานได้บนแพลตฟอร์มต่างๆ รวมทั้งเครือข่ายโทรศัพท์ต่างๆ โดยการเขียนโปรแกรมเพียงครั้งเดียว JTAPI ช่วยให้ผู้พัฒนาไม่จำเป็นรู้หรือเข้าในกลไกบางอย่างของ IP telephony สำหรับการสร้างแอปปลิเคชันได้ในระดับกว้างตั้งแต่เดสทอปแอปปลิเคชันไปจนถึง call center รวมทั้ง JTAPI ยังได้ลดความแตกต่างระหว่างการควบคุมการเรียกจากบุคคลที่หนึ่งและสาม รวมทั้งความแตกต่างระหว่างการควบคุมการเรียก (call control) และการควบคุมสื่อมีเดีย (media control) นอกจากนี้ผู้พัฒนาโปรแกรมสามารถที่พัฒนาโดยใช้ JTAPI อยู่บน telephony API อื่นๆ ได้ เช่น TAPI และ TSAPI

5.2 TAPI (Telephony API)

TAPI เป็น API บนระบบปฏิบัติการไมโครซอฟท์วินโดว์ (Microsoft Windows) ซึ่งทำให้แอปปลิเคชันสามารถเรียกใช้ฟังก์ชันการทำงานในระบบโทรศัพท์ได้ TAPI ได้ถูกพัฒนาโดยไมโครซอฟท์เพื่อรองรับเทคโนโลยี IP telephony ซึ่งจะช่วยให้การพัฒนาแอปปลิเคชันสำหรับ IP telephony ทำได้สะดวกและรวดเร็วขึ้น TAPI ได้ให้อินเตอร์เฟซของฟังก์ชันการควบคุมการเรียก (call control) สำหรับโปรโตคอลสื่อสารชนิดต่างๆ ทำให้ผู้ใช้สามารถเรียกใช้ได้โดยไม่ต้องสร้างขึ้นมาเอง ในปัจจุบัน TAPI เป็นเวอร์ชัน 3.0 ซึ่งใช้ในระบบปฏิบัติการวินโดว์ 2000 (Windows 2000) เวอร์ชัน 3 นี้มีการพัฒนาจากเวอร์ชัน 2 คือ TAPI 2.1 API เป็น COM ทำให้แอปปลิเคชัน TAPI ถูกพัฒนาจากภาษาใดก็ได้ เช่น C++ หรือ Visual Basic และยังใช้ Active Directory ของวินโดว์ 2000 ช่วยทำให้การจัดการเกี่ยวกับเครือข่ายภายในองค์กร์ทำได้อย่างมีประสิทธิภาพมากขึ้น นอกจากนั้น TAPI 3.0 ยังสนับสนุนในเรื่องของคุณภาพการให้บริการ (QoS) รวมทั้งสนับสนุนมาตรฐาน H.323 ซึ่งพัฒนาโดย ITU

6. การเชื่อมต่อระหว่างโปรโตคอล H.323 และ SIP

(SIP and H.323 Interconnection)

ในการเปรียบเทียบระหว่าง H.323 และ SIP [8] จะเห็นทั้งสอง โปรโตคอลต่างก็มีทั้งข้อดีข้อเสีย และโปรโตคอลทั้งสองยังมีแนวโน้มในการพัฒนาให้มีประสิทธิภาพที่ใกล้เคียงกัน (เช่น กำลังมี H.323 Version 4 [11]) ทำให้ระบุได้ยากว่าโปรโตคอลใดเหมาะที่จะใช้มากกว่า ถึงแม้ว่า SIP จะมีข้อดีกว่ามากโดยเฉพาะเรื่องความไม่สลับซับซ้อน แต่โปรโตคอล H.323 ได้ถูกนำมาใช้งานจริงแล้วในปัจจุบัน เช่น Microsoft Netmeeting ดังนั้นการเชื่อมต่อระหว่างโปรโตคอลทั้งสองจึงเป็นสิ่งจำเป็น[12] เพื่อให้สามารถเกิดการเชื่อมต่อระหว่างแอปปลิเคชันหรือผลิตภัณฑ์ได้อย่างครอบคลุมไม่ว่าจะใช้โปรโตคอล H.323 หรือ SIP

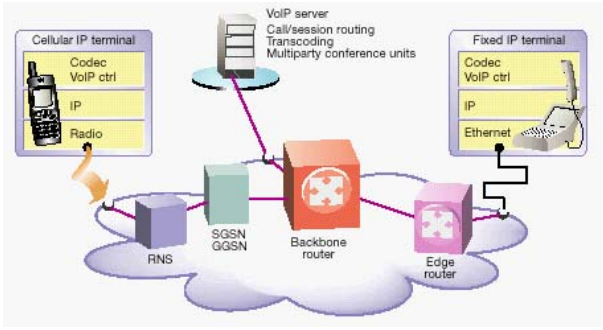
โปรโตคอลทั้งสองทำงานอยู่บนโปรโตคอล IP และใช้ RTP ในการส่งข้อมูลมัลติมีเดีย ดังนั้นในการเชื่อมต่อระหว่างโปรโตคอลทั้งสอง จึงเชื่อมต่อเฉพาะในส่วนที่เป็น การส่งสัญญาณ และส่วนที่ให้รายละเอียดของเซสชันการสื่อสาร(session description) เอนทิตี้ที่ทำหน้าที่ในการเชื่อมต่อจะเรียกว่า SIP-H.323 signaling gateway (SGW) โดยจะเรียกเครือข่ายที่ต้องเชื่อมต่อโดยผ่าน SGW ว่าเครือข่ายภายนอก (foreign network) เช่น เมื่อพิจารณาเครือข่ายที่ใช้ SIP เครือข่าย H.323 จะเป็นเครือข่ายภายนอก ในการเชื่อมต่อจะเป็นลักษณะที่เทอร์มินัลไม่จำเป็นต้องรู้ว่าเทอร์ปลายทางเป็นเทอร์มินัลในเครือข่ายภายนอกหรือไม่โดยที่ผู้ใช้อาจจะเปลี่ยนโปรโตคอลที่ใช้ได้โดยที่ยังคงใช้แอดเดรสเดิม ซึ่งในวิธีนี้ต้องการการลงทะเบียนข้อมูลของเทอร์มินัลหรือผู้ใช้ให้กับเครือข่ายภายนอก

ในการเชื่อมต่อระหว่างเครือข่าย H.323 และ SIP SGW จะต้องทำหน้าที่ในการแปลงข้อมูลจากทั้งสองเครือข่าย โดยส่วนที่จำเป็นต้องมีการแปลงมีดังนี้

- การสร้างการเรียก (call setup) SGW ต้องทำหน้าที่ในการแปลง ขั้นตอนในการสร้าง การเรียก ระหว่างโปรโตคอลทั้งสอง ข้อมูลที่ใช้ได้แก่ แอดเดรสปลายทางในการส่ง การส่งสัญญาณ ความสามารถทางด้านมีเดีย (media capability) และแอดเดรสที่ใช้ในการรับมีเดีย SGW จะนำข้อมูลเหล่านี้จากเครือข่ายหนึ่งแล้วส่งไปในอีกเครือข่ายหนึ่งในรูปแบบและตามขั้นตอนที่ถูกต้อง H.323 ข้อมูลเหล่านี้จะกระจายอยู่ในแมสเสจตามขั้นตอนต่างๆ แมสเสจ SETUP TermCapSet เป็นต้น แต่สำหรับ SIP ข้อมูลจะอยู่ในแมสเสจเดียว คือแมสเสจ INVITE หรือแมสเสจที่ใช้ในการตอบ (response) ดังนั้นในการแปลงจาก SIP ไปเป็น H.323 จึงทำได้โดยตรง โดยการนำข้อมูลจากแมสเสจ แล้วแบ่งเป็นขั้นตอนต่างๆ ใน H.323 แต่การแปลงในทางกลับกันจะมีความซับซ้อนมากกว่า เพราะว่าจะต้องทำการรวบรวมข้อมูลจาก แมสเสจ H.323 ในหลายขั้นตอนเพื่อรวมเป็นเพียงแมสเสจเดียวใน SIP
- การลงทะเบียนของผู้ใช้ (user registration) SGW จำเป็นต้องหาตำแหน่งที่ถูกต้องของผู้ถูกเรียกหรือปลายทางได้โดยจะต้องทำการแปลงเป็น IP แอดเดรส ซึ่งข้อมูลที่ใช้ในการแปลงจะได้จากการลงทะเบียนของผู้ใช้จากทั้งสองเครือข่าย ข้อมูลเหล่านี้จะรับผิดชอบโดย registrar ใน SIP และ gatekeeper ใน H.323 โดยที่ SGW จะต้องสามารถนำข้อมูลเหล่านี้มาใช้ได้
- คำอธิบายรายละเอียดของเซสชัน (session description) ใน SIP โดยทั่วไปแล้วจะใช้ SDP ในการให้รายละเอียดของเซสชันหรือความสามารถทางด้านมีเดีย (media capability) แต่สำหรับ H.323 จะใช้ชุดของ Capability Descriptor ในโปรโตคอล H.245 SGW ต้องสามารถแปลงข้อมูลระหว่างทั้งสองโปรโตคอลได้ ในการแปลงจาก SDP ไปเป็น H.245 สามารถทำได้ไม่ยาก เนื่องจากว่า Capability Descriptor สามารถจะให้รายละเอียดได้มากกว่าการใช้ SDP ทำให้ในการแปลงจาก H.245 ไปเป็น SDP ทำได้ยากกว่า วิธีหนึ่งที่สามารถทำได้คือการส่งแมสเสจ SDP หลายแมสเสจใน ส่วน message body ของแมสเสจ INVITE หรือตอบสนอง(response) โดยที่แมสเสจ SDP แต่ละแมสเสจใช้แทนแต่ละ Capability Descriptor

7. ไอพีเทเลโฟนนีสำหรับเครือข่ายไร้สาย (IP Telephony over Wireless Networks: VoIPoW)

ในโทรศัพท์เคลื่อนที่รุ่นที่ 2 มีข้อจำกัดในการใช้งานอินเตอร์เน็ตหลายประการโดยเฉพาะ bandwidth และคุณภาพการบริการ ในโทรศัพท์รุ่นที่ 3 นี้ การทำงานร่วมกับ IP Network ได้รับการพัฒนาที่ดีขึ้น ซึ่งโทรศัพท์มือถือจะทำหน้าที่คล้ายกับ Mobile IP Device อย่างไรก็ดี การใช้งาน IP บนเครือข่ายไร้สายจะทำให้เกิด overhead สูงมาก การใช้งาน IP สำหรับมือถือจะเรียกว่า Voice over IP over Wireless (VoIPoW) รูปที่ 5. แสดงตัวอย่างโครงสร้างของ VoIPoW มาตรฐาน H.322 และ SIP สามารถรองรับการใช้งานของ VoIPoW



รูปที่ 5. แสดงการเชื่อมโยงระหว่างไอพีเทเลโฟนนีชนิดเคลื่อนที่และไม่เคลื่อนที่

การทำงานในรูปที่ 5. [9] เป็นดังนี้ โทรศัพท์เคลื่อนที่สามารถใช้ในโปรแกรมประยุกต์ต่างๆหรือทำการสนทนาทางโทรศัพท์กับโทรศัพท์ปลายทางได้ (ทั้งชนิดเคลื่อนที่และไม่เคลื่อนที่) เสียงพูดจะถูกแปลงให้อยู่ในรูปของเพ็กเกตโดย Voice Codec (เช่น GSM) จากนั้นจะถูกส่งไปให้ระดับ IP และถูกส่งต่อไปในระดับ Radio-access network (RAN) จากนั้นจะถูกส่งผ่านคลื่นวิทยุไปยังสถานีฐาน (RNS: Radio Network Service) เพื่อส่งต่อไปยังสถานีแม่ (SGSN/GGSN) หลังจากนี้ข้อมูลดังกล่าวจะถูกส่งเข้าไปยัง data network (เช่น IP Network, ATM Network) ข้อมูลจะถูกส่งไปจนถึงปลายทางตามกลไกปรกติของเครือข่าย กล่าวโดยสรุปแล้วระดับชั้น IP จะไม่ถูกกระทบจากการเปลี่ยนแปลงระดับใช้ล่างลงไป เช่น RAN ดังนั้นการประยุกต์ใช้ VoIP สำหรับโทรศัพท์เคลื่อนที่สามารถกระทำได้ดีเช่นเดียวกับโทรศัพท์ทั่วไป

8. สรุป

โปรโตคอลอินเตอร์เน็ตเทเลโฟนี จะถูกนำมาใช้เป็นโปรโตคอลหลักในการสื่อสารทางด้านโทรศัพท์ในอนาคตอันใกล้นี้ โดยอุปกรณ์การสื่อสารจะทำงานเป็น mobile IP device ขณะนี้มี 2 มาตรฐานหลักคือ SIP ซึ่งเป็นข้อกำหนดของ IETF และ H.323 ซึ่ง เป็นข้อกำหนดของ ITU จากการศึกษาเบื้องต้นพบว่าขณะนี้อุปกรณ์ในท้องตลาดส่วนใหญ่จะทำงานตามมาตรฐาน H.323 เนื่องจากเป็นมาตรฐานที่กำหนดไว้อย่างสมบูรณ์ ขณะที่ SIP ยังเป็นมาตรฐานที่ค่อนข้างใหม่ บางส่วนยังอยู่ในช่วงของการพัฒนา แต่อย่างไรก็ดี H.323 มีความซับซ้อนและยุ่งยากในการพัฒนากว่า SIP มาก น่าจะเป็นข้อจำกัดที่สำคัญในการพัฒนา นอกจากนี้รูปแบบของ SIP กำลังได้รับการพัฒนาร่วมกับโปรแกรมอื่น เช่น Java จึงเป็นไปได้ว่า SIP อาจจะมีการนำไปใช้กว้างขวางกว่าในอนาคตอันใกล้นี้ ข้อดีอีกประการหนึ่งของ SIP คือ การใช้งานทางด้าน Mobile device เนื่องจาก SIP ถูกออกแบบให้มี mobility ที่สูงกว่า H.323 อย่างไรก็ตามการรองรับการใช้งานทั้ง 2 มาตรฐานเป็นสิ่งจำเป็น การทำงานในลักษณะ Dual Mode น่าจะเป็นแนวทางของอุปกรณ์โทรศัพท์ทั้งชนิดแบบเคลื่อนที่ และไม่เคลื่อนที่

รายการคำย่อ

ASN.1	Abstract syntax notation
CGI	Common gateway interface
CPL	Call processing language
DNS	Domain name system
DSP	Digital signal processing
G.711	Pulse Code Modulation (PCM) 48 to 64 kbps
G.723.1	Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s
G.729	C source code and test vectors for implementation verification of the G.729 8 kbit/s CS-ACELP speech coder
H.225.0	Call signaling protocols and media stream packetisation for packet-based multimedia (includes Q.931 and RAS)
H.235	Security and encryption for H-series multimedia terminals
H.245	Control protocol for multimedia communications
H.261	Video codec for audiovisual services at p x 64 Kbit/s
H.263	Video coding for low bit rate communications
H.323 ver 1	Visual telephone system and equipment for LAN that provide a non-guaranteed QoS
H.323 ver 2-4	Packet-based multimedia communications systems
H.450.x	Supplementary services for multimedia (call transfer, diversion, hold, park and pickup, call waiting, message waiting)
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IPDC	IP device control
IPTel	IP Telephony หรือ Internet Telephony
IMTC	International Multimedia Teleconferencing Consortium
ITU	International Telecommunication Union
JTAPI	Java telephony application programming interface
LDAP	Lightweight directory access protocol
MC	Multipoint controller
MCU	Multipoint control unit
MGCP	Media gateway control protocol (RFC 2705)
MP	Multipoint processor
PER	Packet encoding rule
PSTN	Public switching telephone network
QCIF	Quarter common intermediate format
QoS	Quality of Service
RTP	Real-time transport protocol (RFC 1889)
RTCP	Real-time transport control protocol (RFC 1889)
RTSP	Real-time Streaming Protocol (RFC 2326)
SDP	Session description protocol (RFC 2327)
SGW	Signaling gateway
SGCP	Simple gateway control protocol
SIP	Session initial protocol (RFC 2543)
T.120	Data protocols for multimedia conferencing
TAPI	Telephony application programming interface
URL	Uniform resource locator
VoIP	Voice over IP
VoIPoW	Voice over IP over Wireless

แหล่งข้อมูลเพิ่มเติม

1. Free h.323: http://www.h323.org/

2. The OpenH323 Project: http://www3.openh323.org/
3. Open Phone: http://openphone.org/
4. OpenH323 Gatekeeper: http://www.willamowius.de/
5. Vovida Open Source Networks: http://www.vovida.com/opensource.html
6. H.323 Corner: http://www.meetingbywire.com/netmeetingh323.htm
7. Network Packetizer: http://www.packetizer.com/
8. Java H.323 Engine: http://www.alphaworks.ibm.com/tech/j323engine
9. IETF IPTEL Working Group Homepage: http://www.bell-labs.com/mailing-lists/iptel/
10. SIP Open Source: http://siphon.sourceforge.net/

เอกสารอ้างอิง

[1] M.Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP : session initial protocol", RFC 2543, Internet Engineering Task Force, Mar 1999

[2] International Telecommunication Union, "Packet based multimedia communication system", recommendation H.323, Telecommunication Standardization Sector of ITU, Feb. 1998

[3] H. Schulzrinne, A. Rao, R. Lanphier, "Real time streaming protocol (RTSP)", RFC 2326, Internet Engineering Task Force, Apr. 1998

[4] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: a transport protocol for real-time application", RFC 1889, Internet Engineering Task Force, Jan 1996

[5] International Telecommunication Union, "Media stream packetization and synchronization on non-guaranteed quality of service LANs", recommendation H.225.0, Telecommunication Standardization Sector of ITU, Nov. 1996

[6] International Telecommunication Union, "Control protocol for multimedia communication", recommendation H.245, Telecommunication Standardization Sector of ITU, Nov. 1998

[7] R. Braden, L. Zhang "Resource Reservation Protocol (RSVP)-version1 functional specification", RFC 2205, Internet Engineering Task Force, Sep 1997

[8] K. Singh, H. Schulzrinne, "Interworking between SIP/SDP and H.323", Internet Draft, Internet Engineering Task Force, Jan 2000,

[9] G. AP Eriksson, B. Olin, K. Svanbro and D. Turina, "The challenges of voice-over-IP-over-wireless," Ericsson Review No. 1, 2000, pp.20-31.

[10] C. Andersson and P. Svensson, "Mobile Internet—An industry-wide paradigm shift?," Ericsson Review No. 4, 1999, pp.207-213

[11] I. Sebestyen, "ITU-T Standards Update on H.323, H.248," IMTC Fall Forum, October 2000.

[12] R. Baird, "SIP-H.323 Interconnecting Issues," IMTC Fall Forum, November 2000.

[13] Nortel Networks., Inc., "A Comparison of H.323v4 and SIP," Tdoc S2-000505, 3GPP2, January 2000.

[14] A. Percy, "IP Telephony Inter-Gateway Protocols," Brooktrout Technology, Inc. www.brooktrout.com

สินชัย กมลภิวงศ์ สำเร็จการศึกษาปริญญาตรีและปริญญาโททางวิศวกรรมไฟฟ้า จากมหาวิทยาลัยสงขลานครินทร์ และปริญญาเอกทางด้านวิศวกรรรมไฟฟ้าและสื่อสารจาก The University of New South Wales ประเทศออสเตรเลีย ปัจจุบันเป็นอาจารย์ประจำและดำรงตำแหน่งผู้ช่วยศาสตราจารย์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ งานวิจัยที่สนใจคือ ATM/Broadband Networks IP Networks and Multimedia Communications Modeling and Simulation in Computer Networks ปัจจุบัน เป็นกรรมการใน Computer Communications Society สาขาประเทศไทย และเป็นสมาชิกสมาคมวิชาการ IEEE ACM และ Computer Society

ลัญฉกร วุฒิสิทธิกุลกิจ จบปริญญาตรีทางวิศวกรรมไฟฟ้า สาขาสื่อสาร จากจุฬาลงกรณ์มหาวิทยาลัย จบปริญญาโท และปริญญาเอกทางด้านวิศวกรรรมโทรคมนาคมจาก University of Essex ประเทศอังกฤษ งานวิจัยหลักประกอบด้วยสองส่วนสำคัญ ส่วนแรกคือ การออกแบบและพัฒนาระบบสื่อสาร โทรคมนาคมความเร็วสูง เช่น เทคโนโลยี WDM ATM MPLS และ TCP/IP เป็นต้น ส่วนที่สองคือ การ

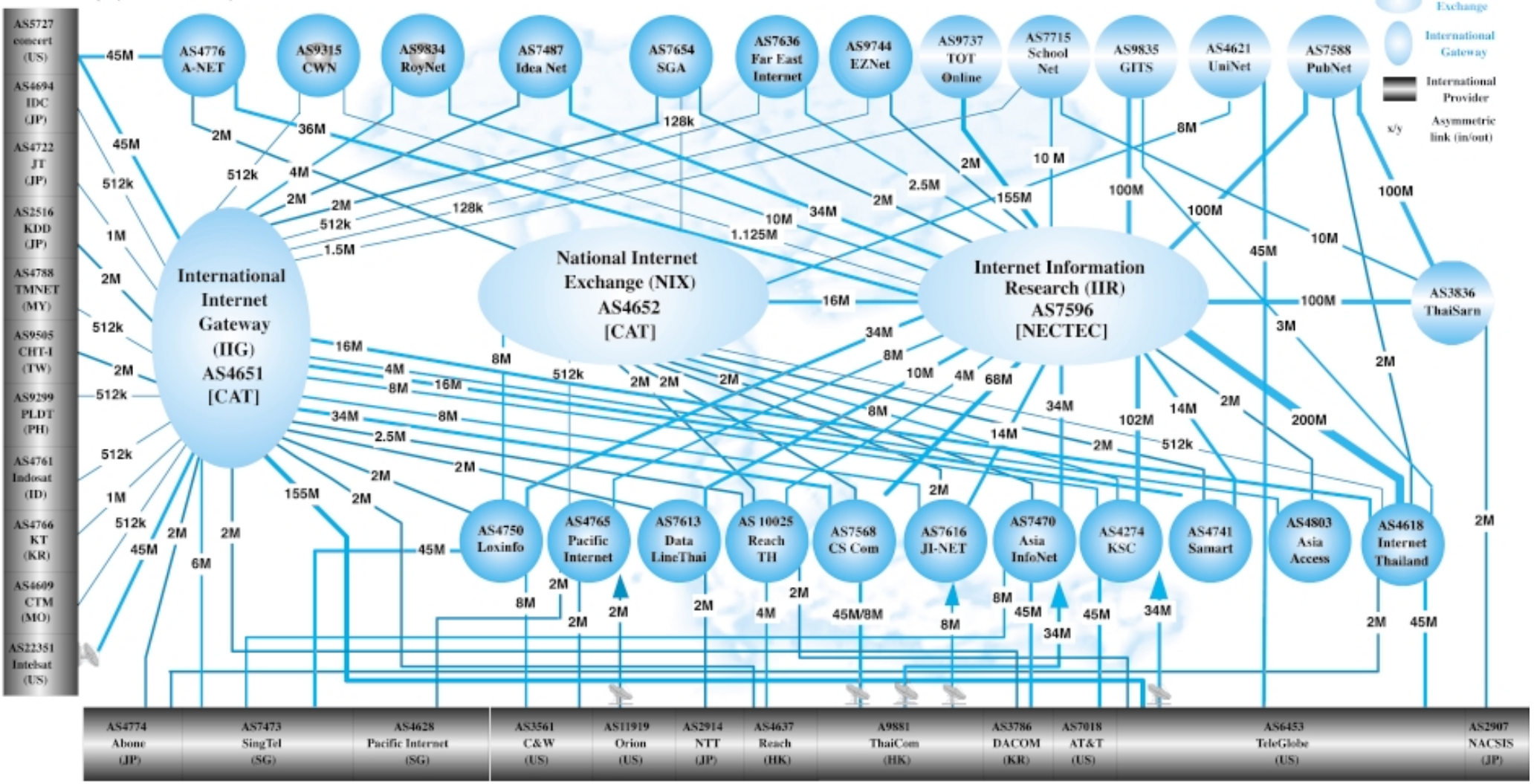
พัฒนาระบบสื่อสารไร้สาย เช่น Wireless ATM, Mobile IP, GSM, CDMA, IMT2000 และ MAC protocol แบบต่าง ๆ ในปัจจุบันได้เริ่มมีการศึกษากรรมวิธีการเข้ารหัสช่องสัญญาณแบบเทอร์โบ ปัจจุบัน ดร.ลัญฉกร วุฒิสิทธิกุลกิจ ดำรงตำแหน่ง ผู้ช่วยศาสตราจารย์ ห้องปฏิบัติการวิจัยโทรคมนาคม ภาควิชาวิศวกรรมไฟฟ้า สาขาสื่อสาร คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

สถิตพงศ์ พุทธิประเสริฐ เกิดเมื่อวันที่ 16 เมษายน พ.ศ.2520 จบการศึกษาระดับปริญญาตรี สาขาโทรคมนาคม จากภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปัจจุบันกำลังศึกษาต่อในระดับปริญญาโทในสาขาและภาควิชาเดียวกัน โดยมีความสนใจในเรื่องระบบเครือข่ายคอมพิวเตอร์ และ อินเตอร์เน็ทเทเลโฟนนี

Internet Connectivities in Thailand (January 2002)

http://www.nectec.or.th/internet/map/

Total International bandwidth :
690.5 Mbps (into Thailand) and
575.5 Mbps (out from Thailand)



DISCLAIMER

Chart Date: 2002-01-03

InetGIF ThaiSam