

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

การพัฒนาระบบสารสนเทศการให้บริการลูกค้า บริษัท ทีโอที จำกัด (มหาชน) จังหวัดมหาสารคาม โดยการชำระเงินผ่าน PayPal และ แจ้างผลบนระบบปฏิบัติการแอนดรอยด์ ได้ทำการศึกษา ค้นคว้า รวบรวมข้อมูล และทฤษฎีที่เกี่ยวข้อง ดังต่อไปนี้

1. การให้บริการ บริษัท ทีโอที จำกัด (มหาชน)
2. ระบบปฏิบัติการ Android
3. การชำระค่าบริการผ่าน Paypal
4. การวิเคราะห์และออกแบบระบบ
5. ซอฟต์แวร์ที่ใช้ในการพัฒนา
6. แนวคิดและทฤษฎีเกี่ยวกับเว็บเซอร์วิส (Web Service) แบบ REST
7. การประเมินเครื่องมือในการศึกษา
8. งานวิจัยที่เกี่ยวข้อง

การให้บริการ บริษัท ทีโอที (จำกัด) มหาชน

1. บริการโทรคมนาคมระหว่างประเทศ (International Telecommunication Services)

บริษัท ทีโอที จำกัด (มหาชน) ก่อตั้งเมื่อวันที่ 24 กุมภาพันธ์ พ.ศ.2497 โดยแปลสภาพจากองค์การโทรศัพท์แห่งประเทศไทย เป็น บริษัท จำกัด (มหาชน) เมื่อวันที่ 31 กรกฎาคม พ.ศ. 2545 ตลอดระยะเวลาที่ผ่านมา ทีโอที ได้มีส่วนสำคัญในการวางรากฐานระบบสื่อสารโทรคมนาคมไทยด้วยประสบการณ์อันยาวนาน

ทีโอที พร้อมให้บริการสื่อสารโทรคมนาคมแบบครบวงจรทั้งในและระหว่างประเทศ เพื่อตอบสนองความต้องการของลูกค้าให้ได้รับความพึงพอใจสูงสุด ภายใต้วิสัยทัศน์ “การมุ่งสู่ความเป็นผู้ให้บริการสื่อสารโทรคมนาคมแห่งชาติที่สนองต่อความต้องการของลูกค้า และสาธารณชน อย่างใกล้ชิด ทั้งด้านเศรษฐกิจ และสังคมความด้วยนวัตกรรมที่ทันสมัย พร้อมด้วยบริการที่หลากหลาย” รวมถึงบริการ International Voice and Data ซึ่งเป็นบริการโทรออกต่างประเทศที่ครอบคลุมทั่วโลก โดยผ่านรหัส 007 และ 008 รวมถึงการรับส่งข้อมูลระหว่างสาขาผ่านบริการสื่อสารข้อมูลระหว่างประเทศหลากหลายบริการ ได้แก่ บริการ IPLC , บริการ International IP-VPN และ บริการ International Ethernet

นอกจากนี้ ยังมีบริการ IIG ที่ทำให้สามารถเชื่อมต่อและแลกเปลี่ยนข้อมูลอินเทอร์เน็ตผ่านอินเทอร์เน็ตเกตเวย์ระหว่างประเทศ จากประเทศไทยไปยังศูนย์กลางการ

แลกเปลี่ยนข้อมูลอินเทอร์เน็ตของโลก (Internet Hub) ที่สามารถส่งผ่านข้อมูลได้อย่างรวดเร็ว และปลอดภัย

ทีโอที ได้มุ่งมั่นพัฒนาโครงข่ายระหว่างประเทศมาเป็นลำดับ นอกเหนือจากการเชื่อมต่อวงจรกับประเทศเพื่อนบ้านแล้ว ทีโอที ยังได้จัดตั้ง PoP (Point of Presence) สำหรับบริการ Voice ที่ฮ่องกง และสหรัฐอเมริกา รวมถึงได้จัดตั้ง PoP สำหรับบริการ Data ที่ ฮ่องกง สิงคโปร์ สหรัฐอเมริกา และอังกฤษ นอกจากนี้ ทีที ยังได้เข้าร่วมลงทุนสร้างเคเบิลใต้น้ำระหว่างประเทศหลายเส้นทาง เช่น เส้นทางสิงคโปร์-ฮ่องกง-ญี่ปุ่น-สหรัฐอเมริกา เพื่อใช้เป็นเส้นทางหลักในการเชื่อมต่ออินเทอร์เน็ตในทวีปเอเชีย และส่งผ่านข้อมูลการติดต่อสื่อสารไปยังสหรัฐอเมริกา รวมถึงสามารถเชื่อมต่อได้กับทุกทวีปทั่วโลก

2. บริการสื่อสารทางเสียงระหว่างประเทศ (International Voice Services)

2.1 บริการ ทีโอที 007 “007 เสียงคมชัด ทางลัดธุรกิจ” เป็นบริการโทรศัพท์ระหว่างประเทศ ที่มีคุณภาพเสียงคมชัดระดับดีเลิศ เหมาะสำหรับลูกค้าองค์กร ที่ต้องการความคล่องตัวในการติดต่อธุรกิจอย่างมีประสิทธิภาพ สามารถติดต่อผ่านโทรศัพท์ และโทรสารได้ถึง 229 ปลายทางทั่วโลก

2.1.1 คุณสมบัติของโครงข่าย ให้บริการด้วยเทคโนโลยี TDM (Time Division Multiplexing) ผ่านโครงข่าย PSTN โดย ทีโอที ทำการเชื่อมต่อโดยตรง (Direct Link) กับผู้ให้บริการโทรคมนาคมรายใหญ่ในต่างประเทศมากกว่า 22 บริษัท

2.1.2 วิธีการใช้งาน กด 007 + รหัสประเทศ + รหัสเมือง หรือ รหัสโทรศัพท์เคลื่อนที่ + เลขหมายปลายทางสามารถใช้บริการ 007 ได้จาก

- 1) โทรศัพท์ประจำที่ของ TOT, TRUE และ TT&T
- 2) โทรศัพท์เคลื่อนที่ระบบ TOT3G , 470 MHz, AIS, DPC และ

DTAC

- 3) โทรศัพท์สาธารณะของ ทีโอที
- 4) บัตร TOT Card, บัตร TOT Prepaid (ผ่านบริการ PIN Phone

108) และบัตร Family Card ลักษณะเด่นของบริการ 007

- 5) เสียงคมชัด โทรติดง่าย
- 6) สะดวกกับการชำระค่าบริการในบิลเดียวกันเลขหมายที่ท่านใช้

บริการ 007

2.2 บริการ ทีโอที 008 “008 โทรประหยัด ชัดทั่วโลก” เป็นบริการโทรศัพท์ระหว่างประเทศ ราคาประหยัด โทรไกลถึง 228 ปลายทางทั่วโลก เหมาะสำหรับลูกค้าทั่วไปหรือนักท่องเที่ยวจะโทรเข้าเบอร์บ้าน หรือมือถือทุกช่วงเวลาาราคาเดียว ตลอด 24 ชั่วโมง

2.2.1 คุณสมบัติของโครงข่ายให้บริการด้วยเทคโนโลยี Voice over IP (VoIP) ทำให้ลดต้นทุน ช่วยให้ผู้ใช้บริการได้ในราคาประหยัดด้วยคุณภาพเสียงคมชัดตามมาตรฐานสากล

2.2.2 วิธีการใช้งานกด 008 + รหัสประเทศ + รหัสเมือง หรือ รหัสโทรศัพท์เคลื่อนที่ + เลขหมายปลายทางสามารถใช้บริการ 008 ได้จาก

- 1) โทรศัพท์ประจำที่ของ TOT และ TT&T
- 2) โทรศัพท์เคลื่อนที่ระบบ TOT3G, 470 MHz, AIS, DPC และ DTAC
- 3) โทรศัพท์สาธารณะของ ทีโอที
- 4) บัตร TOT Card, บัตร TOT Prepaid (ผ่านบริการ PIN Phone 108) และบัตร Family Card

2.2.3 ลักษณะเด่นของบริการ 008

- 1) คิดอัตราค่าโทรไปยังเบอร์โทรศัพท์บ้านหรือมือถือในต่างประเทศ ทุกช่วงเวลาราคาเดียวตลอด 24 ชั่วโมง
- 2) สะดวกกับการชำระค่าบริการในบิลเดียวกันเลขหมายที่ท่านใช้บริการ 008

3. บริการเสริมทางเสียงระหว่างประเทศ (International Value added Services)

3.1 บริการ IFS เป็นบริการเรียกเก็บเงินปลายทางอัตโนมัติ โดยบริษัทที่ขอใช้บริการจะเป็นผู้รับภาระค่าใช้จ่ายโทรศัพท์ให้กับลูกค้าที่โทรเข้ามาติดต่อธุรกิจในประเทศไทย ในระยะแรกเปิดให้บริการ IFS กับประเทศมาเลเซียและมีแผนขยายไปยังประเทศอื่นต่อไปในอนาคตเหมาะสำหรับองค์กรธุรกิจชั้นนำ เช่น โรงแรม สายการบิน บริษัทนำเข้า-ส่งออก สถาบันการเงิน เป็นต้น

3.1.1 วิธีการใช้บริการ

- 1) สมัครใช้บริการได้ที่ ศูนย์บริการลูกค้า ทีโอที ทุกแห่ง จากนั้นท่านจะได้รับเลขหมาย IFS จาก ทีโอที
- 2) แจ้งหมายเลข IFS ให้ลูกค้าของบริษัทฯ ซึ่งอยู่ในต่างประเทศ (มาเลเซีย) ทราบโดยกดหมายเลข ดังนี้ 1800 + 80 + เลขหมาย IFS ของท่าน

3.1.2 ลักษณะเด่นของบริการ

เป็นบริการที่สามารถสร้างความประทับใจให้กับลูกค้าคนสำคัญของคุณ เนื่องจากคุณจะเป็นผู้รับภาระค่าใช้จ่ายบริการแทนลูกค้าในการโทรติดต่อธุรกิจ/สอบถามข้อมูล

3.2 บริการ HC-IDD

เป็นบริการโทรกลับบ้านแบบเรียกเก็บเงินปลายทาง โดยผ่านโอเปอเรเตอร์ที่ให้บริการด้วยภาษาของประเทศนั้น ๆ สะดวกด้วยการหักค่าใช้จ่ายบริการจากบัตรเครดิต ปัจจุบันเปิดให้บริการโทรไปยังประเทศต่าง ๆ ได้แก่ สหรัฐอเมริกา เยอรมนี อิตาลี ญี่ปุ่น จีน และฮ่องกง เป็นต้น

เหมาะสำหรับ นักท่องเที่ยว หรือนักธุรกิจชาวต่างชาติ ที่ต้องการโทรกลับบ้านได้ โดยไม่ต้องกังวล กับปัญหาเรื่องภาษา และการชำระค่าบริการ

3.2.1 วิธีการใช้บริการ

- 1) กดเลขหมาย 1800 200 200 จากโทรศัพท์สาธารณะหรือ โทรศัพท์ประจำที่ของ ทีโอที
- 2) แจ้งเลขหมายที่ต้องการติดต่อแก่อโอเปอเรเตอร์เพื่อให้ต่อสายไปยังผู้รับปลายทาง

3.2.2 ลักษณะเด่นของบริการ

เป็นบริการที่ช่วยอำนวยความสะดวกให้กับนักท่องเที่ยวหรือนักธุรกิจชาวต่างชาติ ให้สามารถโทรกลับบ้านได้ โดยไม่ต้องกังวลกับปัญหาเรื่องภาษา และการชำระค่าบริการ

3.3 บริการ Collect Call

เป็นบริการโทรไปต่างประเทศแบบเรียกเก็บเงินปลายทาง โดยผ่านโอเปอเรเตอร์ที่ให้บริการด้วยภาษาของประเทศนั้น ๆ ปัจจุบันได้ให้บริการมากกว่า 20 ปลายทางทั่วโลกเหมาะกับนักท่องเที่ยวชาวต่างชาติ นักธุรกิจหรือชาวต่างชาติที่อาศัยอยู่ในประเทศไทยที่ประสงค์จะโทรกลับบ้านของตนเอง โดยไม่ต้องกังวลกับปัญหาเรื่องภาษาและการชำระค่าบริการ

3.3.1 วิธีการใช้บริการ

- 1) กดเลขหมาย “Collect Call” ตามกำหนดของแต่ละประเทศ
- 2) แจ้งเลขหมายปลายทางที่ต้องการเรียกไปภายหลังได้ยินเสียงตอบรับจาก โอเปอเรเตอร์

3.3.2 ลักษณะเด่นของบริการ

เป็นบริการที่ช่วยอำนวยความสะดวกให้กับนักท่องเที่ยวหรือนักธุรกิจชาวต่างชาติ ให้สามารถโทรกลับบ้านได้โดยไม่ต้องกังวล กับปัญหาเรื่องภาษาและการชำระค่าบริการ

3.4 บริการ Calling Home

เป็นบริการโทรกลับเมืองไทย แบบเรียกเก็บเงินปลายทางอัตโนมัติของ ทีโอที และชำระค่าบริการโดยการตัดมูลค่าจากบัตร TOT Prepaid (ใช้ผ่านบริการ PIN Phone 108) ในระยะแรกเปิดให้บริการโทรกลับจากประเทศมาเลเซีย และจะขยายการให้บริการไปยังประเทศต่าง ๆ ในอนาคตเหมาะสำหรับคนไทย หรือชาวต่างชาติที่เดินทางไปต่างประเทศ แล้วต้องการโทรกลับประเทศไทย

3.4.1 วิธีการให้บริการ

กดเลขหมาย 1800 800 066 หรือ 1800 180 066 แล้วปฏิบัติตามขั้นตอนของเสียงตอบรับอัตโนมัติ

3.4.2 ลักษณะเด่นของการบริการ

ผู้ใช้บริการจะได้รับความสะดวก ในการโทรศัพท์กลับมายังประเทศไทย โดยไม่ต้องกังวลเรื่องภาษาและการชำระค่าบริการ เนื่องจากชำระค่าบริการโดยตัดมูลค่าจากบัตร TOT Prepaid

4. บริการวงจรเช่าระหว่างประเทศ (TOT International Private Leased Circuit : TOT IPLC)

4.1 TOT International Private Leased Circuit (TOT IPLC)

เป็นบริการสื่อสารข้อมูลความเร็วสูง ผ่านโครงข่ายเคเบิลใยแก้วนำแสง ให้บริการเชื่อมต่อตรงแบบจุดต่อจุด (point-to-point) รองรับการส่งทั้งภาพและเสียงรวมถึงข้อมูลมัลติมีเดียอื่น ๆ มี Bandwidth ให้เลือกใช้บริการได้หลายระดับ ตั้งแต่ 64 Kbps เป็นต้นไปทีโอที สามารถให้บริการเชื่อมต่อจากทุกพื้นที่ในประเทศไทยไปยังทุกภูมิภาคทั่วโลก ภายใต้ความร่วมมือกับพันธมิตรผู้ให้บริการโทรคมนาคมชั้นนำในต่างประเทศ (Global Network Operator) เหมาะสำหรับองค์กรหรือธุรกิจที่มีสาขาอยู่ต่างประเทศ และต้องการใช้งานโครงข่ายที่มีประสิทธิภาพและมีความปลอดภัยสูง เช่น องค์กรของรัฐ บริษัทข้ามชาติ สถาบันการเงิน ธนาคาร กลุ่มโรงแรม และโรงงานอุตสาหกรรม เป็นต้น

4.2 คุณสมบัติของโครงข่าย

4.2.1 ใช้โครงข่าย SDH และ DQDM ที่รองรับการเชื่อมต่อทั้งแบบ Electrical (V.35,G703) และ Optical (G.957) ตามมาตรฐาน ITU

4.2.2 สามารถให้บริการได้ทั้งแบบ Half circuit และ Full circuit โดยมีจุดเชื่อมต่อผ่านชายแดนหรือเชื่อมต่อผ่าน PoP (Point of Presence) ของ ทีโอที ในต่างประเทศไปยังปลายทางต่าง ๆ โดยร่วมมือกับพันธมิตรระดับชั้นนำของโลก

4.3 ลักษณะเด่นของการบริการ

- 4.3.1 เป็นบริการแบบ Dedicated circuit ทำให้สามารถใช้งาน Bandwidth ได้อย่างเต็มประสิทธิภาพ
- 4.3.2 มีความปลอดภัยในการใช้งานสูง
- 4.3.3 เป็นวงจรแบบ Protection จึงมั่นใจได้ว่าผู้ใช้บริการจะสามารถใช้งานได้ตลอดเวลา
- 4.3.4 สามารถให้บริการได้อย่างเบ็ดเสร็จบนพื้นฐานของ OSS (One-Stop-Shop)

5. บริการอีเทอร์เน็ตระหว่างประเทศ (TOT International Ethernet)

เป็นบริการวงจรสื่อสารข้อมูลความเร็วสูงที่ถูกรออกแบบเพื่อให้่ง่ายในการเชื่อมต่อระบบ LAN ระหว่างสำนักงานเข้าด้วยกัน โดยผ่านโครงข่ายระหว่างประเทศของ ทีโอที ที่มีประสิทธิภาพและความปลอดภัยในการใช้งานสูง มีราคาประหยัด สามารถเลือกใช้บริการได้ที่ Bandwidth ตั้งแต่ 2 Mbps. เป็นต้นไป

เหมาะสำหรับกลุ่มลูกค้าองค์กร ธนาคาร โรงงานอุตสาหกรรม และบริษัทข้ามชาติ ที่มีสาขาอยู่ในต่างประเทศ ที่ต้องการดูแล บริหารจัดการระบบเครือข่ายระหว่างสาขาด้วยตนเอง

5.1 คุณสมบัติของโครงข่าย

- 5.1.1 ให้บริการผ่านโครงข่าย MPLS/VPN (L2) ที่ครอบคลุมพื้นที่ให้บริการทั่วประเทศ และเชื่อมโยงไปยัง PoP (Point of Presence) ของ ทีโอที ในต่างประเทศที่เป็น ศูนย์กลางโทรคมนาคมของโลก
- 5.1.2 สามารถให้บริการไปได้ทั่วโลก โดยร่วมมือกับพันธมิตรระดับชั้นนำของโลก ทำให้สามารถตอบสนองทุกความต้องการของลูกค้าด้วยคุณภาพระดับสากล
- 5.1.3 ให้บริการเชื่อมต่อ Interface แบบ T1-E1 และ Optical
- 5.1.4 รองรับการสื่อสารข้อมูลแบบ IP Protocol ทำให้สามารถรับส่งได้ทั้ง ข้อมูลภาพ เสียง และวีดีโอ

5.2 ลักษณะเด่นของการบริการ

- 5.2.1 มีราคาประหยัดกว่าเมื่อเทียบกับบริการอื่น ๆ ที่มีขนาด Bandwidth เท่ากัน
- 5.2.2 รูปแบบการเชื่อมต่อไม่ซับซ้อนทำให้ง่ายในการปรับเปลี่ยน Bandwidth ได้ตามความต้องการโดยไม่ส่งผลกระทบต่อการใช้งานอยู่
- 5.2.3 ให้บริการวงจรแบบ Protection ทำให้มั่นใจว่าผู้ใช้บริการจะสามารถใช้งานได้ตลอดเวลา

6. บริการต่อผ่านอินเทอร์เน็ตระหว่างประเทศ (TOT International Gateway : TOT IIG)

เป็นบริการเชื่อมต่อ และแลกเปลี่ยนข้อมูลอินเทอร์เน็ตผ่านอินเทอร์เน็ตเกตเวย์ระหว่างประเทศ จากประเทศไทยไปยังศูนย์กลางการแลกเปลี่ยนข้อมูลอินเทอร์เน็ตของโลก (Internet Hub) ทำให้ลูกค้าสามารถเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต ทั้งใน เอเชีย อเมริกา และ ยุโรป อย่างมีประสิทธิภาพเหมาะสำหรับลูกค้าที่เป็นผู้ให้บริการอินเทอร์เน็ต ผู้ให้บริการโทรคมนาคม และองค์กรต่าง ๆ ที่มีเครือข่ายเป็นของตัวเอง ลูกค้าสามารถขอใช้บริการได้ตามความต้องการใช้งานที่ Bandwidth ตั้งแต่ 2 Mbps เป็นต้นไป ทั้งในรูปแบบของ Committed/Fixed Bandwidth และ Burstable Bandwidth

6.1 คุณสมบัติของโครงข่าย

6.1.1 โครงข่ายหลักของอินเทอร์เน็ตเกตเวย์ระหว่างประเทศของ ทีโอที เป็นวงจรมัลติพลาเน็ตระบบเคเบิลใยแก้วนำแสงภาคพื้นดิน (Terrestrial) และ เคเบิลใต้น้ำ (Submarine cable) ที่มีขนาดวงจรมหาศาล เพียงพอในการรองรับการเติบโตของบรอดแบนด์อินเทอร์เน็ตทั้งในประเทศไทย และต่างประเทศ

6.1.2 รองรับการเชื่อมต่อระบบสื่อสารสัญญาณ SDH และ Ethernet ได้ทั้งแบบ Electrical และ Optical โดยลูกค้าสามารถเชื่อมต่อวงจรมายังจุดเชื่อมต่อของ ทีโอที ซึ่งครอบคลุมทุกพื้นที่ภายในประเทศหรือให้ ทีที เชื่อมต่อไปยังสถานที่ของลูกค้า

6.1.3 มี PoP (Point of Presence) ใน 5 ประเทศเพื่อเชื่อมต่อเข้ากับ IP-Port, Peering และ Internet Exchange (IX) กับผู้ให้บริการอินเทอร์เน็ตชั้นนำที่มีเครือข่ายครอบคลุมและหลากหลาย

ระบบปฏิบัติการ Android

1. ระบบปฏิบัติการแอนดรอยด์

1.1 ความเป็นมาของระบบปฏิบัติการแอนดรอยด์

แอนดรอยด์ (Android) กูเกิลแอนดรอยด์ (Google Android) หรือระบบปฏิบัติการแอนดรอยด์ (Android Operating System) เป็นชื่อเรียกชุดซอฟต์แวร์หรือแพลตฟอร์ม (Platform) สำหรับอุปกรณ์อิเล็กทรอนิกส์ที่มีหน่วยประมวลผลเป็นส่วนประกอบ เช่น คอมพิวเตอร์ โทรศัพท์ (Telephone) โทรศัพท์เคลื่อนที่ (Cell phone) อุปกรณ์เล่นอินเทอร์เน็ตขนาดพกพา เป็นต้น แอนดรอยด์ถือกำเนิดอย่างเป็นทางการในวันที่ 5 พฤศจิกายน 2550 โดยบริษัทกูเกิล จุดประสงค์ของแอนดรอยด์มีจุดเริ่มต้นมาจากบริษัท Android Inc. ที่ได้นำเอาระบบปฏิบัติการลินุกซ์ (Linux) ซึ่งนำไปใช้งานกับเครื่องเซิร์ฟเวอร์ (Server) เป็นหลัก นำมาลดทอนขนาดตัว (แต่ไม่ลดทอนความสามารถ) เพื่อให้เหมาะสมแก่การนำไปติดตั้งบนอุปกรณ์พกพา ที่มีขนาดพื้นที่จัดเก็บข้อมูลที่จำกัด โดยที่แอนดรอยด์จะใช้สัญลักษณ์หุ่นยนต์ เป็นหุ่นยนต์ที่คอยช่วยเหลืออำนวยความสะดวกแก่ผู้ที่พกพาในทุกที่ทุกเวลา

กูเกิ้ลแอนดรอยด์เป็นชื่อเรียกอย่างเป็นทางการของแอนดรอยด์ เนื่องจากปัจจุบันบริษัทกูเกิ้ลเป็นผู้ที่ถือสิทธิบัตรในตราสัญลักษณ์ชื่อและรหัสต้นฉบับ (Source Code) ของแอนดรอยด์ภายใต้เงื่อนไขการพัฒนาแบบ GNL โดยเปิดให้นักพัฒนา (Developer) สามารถนำรหัสต้นฉบับไปพัฒนาปรับแต่งได้อย่างเปิดเผย (Open Source) ทำให้แอนดรอยด์มีผู้เข้าร่วมพัฒนาเป็นจำนวนมากและพัฒนาไปได้อย่างรวดเร็ว

แอนดรอยด์เปิดตัวอย่างเป็นทางการเมื่อวันที่ 5 พฤษภาคม พุทธศักราช 2550 ปัจจุบันมีผู้ร่วมพัฒนากว่า 52 องค์กร ประกอบด้วยบริษัทซอฟต์แวร์ บริษัทผู้ผลิตอุปกรณ์ บริษัทผู้ผลิตชิ้นส่วนอิเล็กทรอนิกส์ บริษัทผู้ให้บริการเครือข่าย และบริษัทที่เกี่ยวข้องกับการสื่อสาร

ประเภทของชุดซอฟต์แวร์ เนื่องจากแอนดรอยด์นั้นเปิดให้นักพัฒนาเข้าไปชมรหัสต้นฉบับได้ ทำให้มีผู้พัฒนาจากหลายฝ่ายนำเอารหัสต้นฉบับมาปรับแต่งและสร้างแอนดรอยด์ในแบบฉบับของตนเอง จึงแบ่งประเภทของแอนดรอยด์ออกได้เป็น 3 ประเภท ดังนี้

1. Android Open Source Project (AOSP) เป็นแอนดรอยด์ประเภทแรกที่ กูเกิ้ลเปิดให้สามารถนำต้นฉบับแบบเปิดไปติดตั้งและใช้งานในอุปกรณ์ต่างๆ โดยไม่ต้องเสียค่าใช้จ่ายใดๆ
2. Open Handset Mobile (OHM) เป็นแอนดรอยด์ที่ได้รับการพัฒนาร่วมกับกลุ่มบริษัทผู้ผลิตอุปกรณ์พกพาที่เข้าร่วมกับกูเกิ้ลในนาม Open Handset Alliances ซึ่งบริษัทจะพัฒนาแอนดรอยด์ในแบบฉบับของตนเอง โดยรูปร่างหน้าตาการแสดงผลและฟังก์ชันการใช้งานจะมีความเป็นเอกลักษณ์และมีลิขสิทธิ์เป็นของตนเอง พร้อมได้รับสิทธิในการ มีบริการเสริมต่างๆ จากกูเกิ้ลที่เรียกว่า Google Mobile Service (GMS) ซึ่งเป็นบริการเสริมที่ทำให้แอนดรอยด์มีประสิทธิภาพเป็นไปตามจุดประสงค์ของแอนดรอยด์แต่การจะได้มา 38 ซึ่ง GMS ผู้ผลิตจะต้องทำการทดสอบระบบและขออนุญาตกับทางกูเกิ้ลก่อนนำออกสู่ตลาดได้
3. Cooking หรือ Customize เป็นแอนดรอยด์ที่นักพัฒนานำเอารหัสต้นฉบับจากแหล่งต่างๆ มาปรับแต่งในแบบฉบับของตนเอง โดยจะต้องทำการปลดล็อคสิทธิ์ (Unlock) การใช้งานอุปกรณ์ จึงจะสามารถติดตั้งได้ โดยแอนดรอยด์ประเภทนี้ถือเป็นประเภทที่มีความสามารถมากที่สุดเท่าที่อุปกรณ์เครื่องนั้นๆ จะรองรับได้เนื่องจากได้รับการปรับแต่งให้เข้ากับอุปกรณ์นั้นๆ จากผู้ใช้งานจริง สิทธิในการใช้งานระบบปฏิบัติการทั่วไปที่มีการใช้งานและการเข้าถึงส่วนต่างๆ ภายในระบบเพื่อความปลอดภัยของระบบและผู้ใช้ งาน อุปกรณ์ที่ติดตั้งบนระบบแอนดรอยด์จึงมีการจำกัดสิทธิ์ไว้ (เว้นแต่ได้ทำการปลดล็อคสิทธิ์) สามารถแบ่งสิทธิ์ของผู้ใช้ในการเข้าถึงระบบได้ดังนี้
 - 3.1 สิทธิ root สิทธิการใช้งานระดับราก ซึ่งถือว่าเป็นรากฐานของระบบจึงมีความสามารถในการเข้าถึงทุกๆ ส่วนของระบบ
 - 3.2 สิทธิ ADB (Android Develop Bridge) นักพัฒนาสามารถเข้าถึงส่วนต่างๆ ของระบบได้ผ่านสิทธิ์นี้

3.3 Application & System สิทธิ์ของโปรแกรมในการเข้าถึงระบบและสิทธิ์ของระบบในการเข้าถึงอุปกรณ์โดยสิทธิ์เหล่านี้ ตัวระบบจะเป็นตัวจัดมอบและถอนสิทธิ์ตามเงื่อนไขที่กำหนดซึ่งจะถูกแบ่งย่อยออกเป็นหลายหัวข้อ (<http://developer.android.com/reference/android/Manifest.permission.html>)

3.4 End-user ผู้ใช้งานขั้นสุดท้าย ซึ่งก็คือผู้ใช้งานทั่วไป การเข้าถึงส่วนต่างๆ ของระบบผ่านช่องทางสิทธิ์ที่โปรแกรมได้รับอีกที โดยจะถูกจำกัดไม่ให้เข้าถึงในส่วนที่เป็นอันตรายต่อแก่นระบบและอุปกรณ์

2. ประวัติแอนดรอยด์

ประวัติแอนดรอยด์ไม่ได้ถูกเริ่มต้นพัฒนาที่ Google แต่เริ่มต้นกับบริษัท Android Inc. โดยผู้ที่เริ่มต้นพัฒนาและก่อตั้งแอนดรอยด์นั้น คือ Andy Rubin (co-founder of Danger Inc.) และมีผู้ร่วมก่อตั้งอีก 4 คน คือ Rich Miner (co-founder of Wildfire Communications, Inc.), Nick Sears (once VP at T-Mobile), and Chris White (one of the first engineers at WebTV) ซึ่งการที่ Android Inc. ได้ขายลิขสิทธิ์แอนดรอยด์ให้กับ Google นั้น เป็นเพราะว่า Andy Rubin เองน่าจะเล็งเห็นถึงความสามารถของทรัพยากรบุคคลและเทคโนโลยีต่างๆ ที่มีใน Google ซึ่งในขณะนั้นเอง Android ถูกเข้าใจว่าเป็นเพียงระบบปฏิบัติการบนอุปกรณ์เคลื่อนที่เท่านั้น ซึ่งเหตุการณ์เริ่มต้นทั้งหมดนี้เกิดขึ้นในปี ค.ศ. 2005

ต่อมาในปลายปี ค.ศ. 2007 ก็เกิดปรากฏการณ์ของแอนดรอยด์แพลตฟอร์มอีก 39 ครั้งซึ่งในครั้งนี้ได้มีการเปิดเผยถึงระบบปฏิบัติการบนโทรศัพท์เคลื่อนที่ที่เป็นในลักษณะ Open Platform ซึ่งผู้ใช้สามารถพัฒนาซอฟต์แวร์มาใช้งานเองได้ออกมาจาก กลุ่มบริษัทพัฒนาเทคโนโลยีเกี่ยวกับอุปกรณ์เคลื่อนที่ 34 บริษัทที่มีการเรียกตัวเองว่า Open Handset Alliance และจากการประกาศออกมาในครั้งนี้ทำให้ทั่วโลกเริ่มจับตากับการเคลื่อนไหวของแอนดรอยด์มากขึ้น และในปีนี้เองยังมีการประกาศเป็นครั้งแรกของ Android SDK อีกด้วย

2.1 ข้อเด่นของแอนดรอยด์

เนื่องจากระบบปฏิบัติการแอนดรอยด์มีการเจริญเติบโตอย่างรวดเร็วและมีส่วนแบ่งตลาดของอุปกรณ์ด้านนี้ ขึ้นทุกขณะ ทำให้กลุ่มผู้ใช้งาน และกลุ่มนักพัฒนาโปรแกรม ให้ความสำคัญกับระบบปฏิบัติการแอนดรอยด์เพิ่มมากขึ้นเมื่อมองในด้านของกลุ่มผลิตภัณฑ์ บริษัทที่มีการพัฒนาผลิตภัณฑ์รุ่นใหม่ ได้มีการนำเอาระบบปฏิบัติการแอนดรอยด์ไปใช้ในพร้อมทั้งยังมีการปรับแต่งให้ระบบปฏิบัติการมีความสามารถ การจัดวาง โปรแกรม และลูกเล่นใหม่ ๆ ที่แตกต่างจากคู่แข่งในท้องตลาด โดยเฉพาะอย่างยิ่ง กลุ่มสินค้าที่เป็น มือถือรุ่นใหม่ (SmartPhone) และอุปกรณ์จอสัมผัส (Touch Screen) โดยมีคุณลักษณะแตกต่างกันไป เช่น ขนาดหน้าจอ ระบบโทรศัพท์ ความเร็วของหน่วยประมวลผล ปริมาณหน่วยความจำ แม้กระทั่งอุปกรณ์ตรวจจับต่างๆ (Sensor) หากมองในด้านของการพัฒนาโปรแกรม ทางบริษัท กูเกิ้ล ได้มีการพัฒนา Application Framework ไว้สำหรับนักพัฒนาใช้งาน ได้อย่างสะดวก และไม่เกิด

ปัญหาเมื่อนาซุดโปรแกรมที่พัฒนาขึ้นมา ไปใช้กับอุปกรณ์ที่มีคุณลักษณะต่างกัน เช่นขนาดจอ อุปกรณ์ ไม่เท่ากัน ก็ยังสามารถใช้งานโปรแกรมได้เหมือนกัน

2.2 ข้อดีของแอนดรอยด์

2.2.1 มีลักษณะเป็นโอเพนซอร์สซึ่งทำให้แอนดรอยด์แพลตฟอร์มได้รับความนิยมอย่างรวดเร็ว

2.2.2 มีการเชื่อมต่อที่กลมกลืนกันระหว่าง เว็บแอปพลิเคชัน และโทรศัพท์

2.2.3 มีเครื่องมือที่ใช้พัฒนาซอฟต์แวร์ที่ง่ายและมีประสิทธิภาพ

2.3 ข้อจำกัดของแอนดรอยด์

2.3.1 อุปกรณ์ที่พัฒนาขึ้นมาอาจจะทำงานไม่รองรับกับระบบแอนดรอยด์ได้ครบทุกฟังก์ชัน เพราะ Google ไม่ได้มีส่วนร่วมในการควบคุมการผลิต Hardware โดยตรง

2.3.2 ในบางครั้งแอนดรอยด์แพลตฟอร์มมีการทำงานที่ยังไม่เสถียร

2.4 แผนกลยุทธ์ของแอนดรอยด์

2.4.1 กรกฎาคม ค.ศ. 2005 -Google เริ่มถือลิขสิทธิ์ใน Android. Inc.

2.4.2 12 พฤศจิกายน ค.ศ. 2007 -SDK ตัวแรกของแอนดรอยด์ถูกเปิดเผย

2.4.3 มกราคม ถึง สิงหาคม ค.ศ. 2008 -มีการจัดการแข่งขัน Android Developer Challenge

2.4.4 18 สิงหาคม ค.ศ. 2008 -มีการพัฒนาตัว SDK 0.9 beta

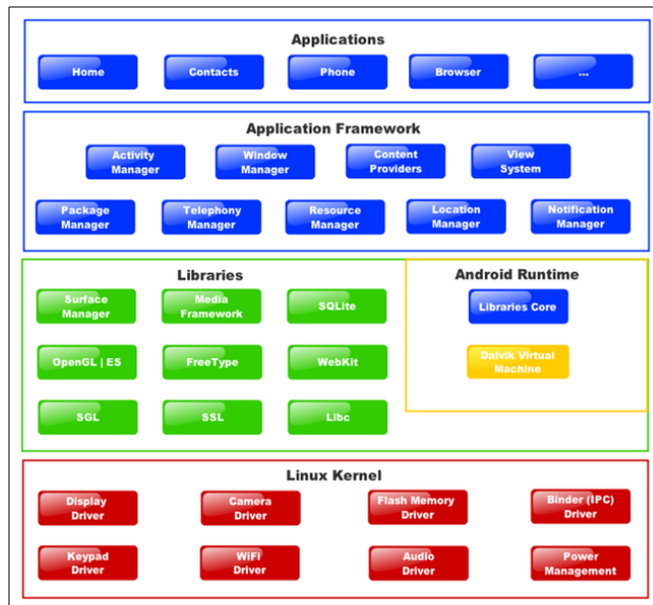
2.4.5 กันยายน ค.ศ. 2008 -Android 1.0 (pre) SDK ถูกผลิตออกมา

2.4.6 ไตรมาสที่ 4 ค.ศ. 2008 -อุปกรณ์ Android 1.0 เริ่มมีการขาย

2.4.7 ไตรมาสที่ 4 ค.ศ. 2008 -มีการเผยแพร่ Source Code ของ

3. สถาปัตยกรรมแอนดรอยด์

การที่จะเริ่มต้นพัฒนาซอฟต์แวร์บนแอนดรอยด์แพลตฟอร์มจำเป็นต้องทำความเข้าใจกับโครงสร้างพื้นฐานต่างๆ เพื่อที่จะได้นำมาประยุกต์ใช้กับซอฟต์แวร์ โดย สิ่งสำคัญสิ่งหนึ่งที่ทำให้เข้าใจรูปแบบของระบบการทำงานของแอนดรอยด์แพลตฟอร์มก็คือ สถาปัตยกรรมของแอนดรอยด์



ภาพที่ 1 สถาปัตยกรรมแอนดรอยด์ (Android Architecture)
ที่มา : คมสันต์ ผุยลานวงศ์. (2555 : 8)

3.1 ลิ눅ซ์เคอร์เนล

ส่วน Kernel ซึ่งเป็นระดับล่างสุดของแอนดรอยด์นั้นจะถูกสร้างขึ้นมาจาก Standard Linux 2.6.24 Kernel และได้ส่งผลให้นักพัฒนาสามารถ Patch Kernel ของลินุกซ์ได้เช่นเดียวกับการ Patch บนระบบปฏิบัติการลินุกซ์โดยทั่วไป แต่ถึงแม้ว่าแอนดรอยด์จะนำ Kernel ของ Linux มาใช้ในการพัฒนาระบบ ตัวแอนดรอยด์แพลตฟอร์มนั้นก็ไม่ถือว่าเป็นระบบปฏิบัติการที่เป็นลินุกซ์โดยสมบูรณ์ เนื่องจากบนแอนดรอยด์แพลตฟอร์มนั้น จะไม่รองรับการใช้งาน glibc เหมือนใน Linux ซึ่งส่งผลให้แอนดรอยด์ไม่มีฟังก์ชันในการทำงานต่างๆ สมบูรณ์เทียบเคียงได้กับลินุกซ์ จึงทำให้สรุปเหตุผลที่แอนดรอยด์เลือกใช้ Linux Kernel ได้ดังนี้

3.1.1 ตัวระบบลินุกซ์เป็นโอเพนซอร์ส ซึ่งเป็นส่วนสำคัญหนึ่งที่ทำให้ระบบแอนดรอยด์แพลตฟอร์มนั้น อยู่ภายใต้เงื่อนไขการใช้งานซอฟต์แวร์โอเพนซอร์ส และทำให้แอนดรอยด์แพลตฟอร์มมีการเผยแพร่ได้รวดเร็วอีกด้วย

3.1.2 รองรับฟังก์ชันการทำงานในส่วนของ Driver ได้อย่างดี ซึ่งทำให้แอนดรอยด์แพลตฟอร์มรองรับกับการใช้งานของชิ้นส่วนต่างๆ ที่มีบนอุปกรณ์สื่อสารเคลื่อนที่

3.1.3 มีการใช้งานระบบ Permission-Based Security Model เพื่อที่แอนดรอยด์แพลตฟอร์มจะมีการกำหนดสิทธิ์ในการใช้งานทรัพยากรของระบบได้ ซึ่งจะทำให้ระบบแอนดรอยด์แพลตฟอร์มมีความปลอดภัยสูงอีกด้วย โดยนอกจากนี้เราสามารถจะดาวน์โหลดตัว Android Kernel มาทำการศึกษาได้จาก <http://git.android.com>

3.1.4 Media Framework จะช่วยรองรับเกี่ยวกับการทำงานกับไฟล์ในฟอร์แมตต่างๆไม่ว่าจะเป็นในด้าน Video, Audio หรือว่าจะเป็น Frame-Format ดังรูปที่มีการนำภาพถ่ายที่มีชนิดไฟล์เป็น JPG



ภาพที่ 2 การแสดงผลไฟล์รูปภาพ

ที่มา : คมสันต์ พุยลานวงศ์. (2555 : 9)

3.1.5 SQLite เป็นไลบรารี ที่เก็บฟังก์ชันการใช้ระบบฐานข้อมูลบนแอนดรอยด์แพลตฟอร์ม ซึ่ง SQLite นี้เป็นระบบฐานที่ได้รับความนิยมในอุปกรณ์ที่มีหน่วยความจำน้อย, ใช้งานง่าย และเหมาะสมกับการใช้งานกับฐานข้อมูลแบบ Standalone ด้วยการทำงานของไลบรารี SQLite บนแอนดรอยด์แพลตฟอร์มนั้นจะช่วยให้นักพัฒนาสามารถสร้างฐานข้อมูลส่วนตัวเอาไว้ใช้กับโปรแกรมได้โดยง่ายไม่ว่าจะผ่านทางภาษา Sql หรือด้วยเมธอดที่มีไว้ในไลบรารี

3.1.6 WebKit เป็น Open Source Browser ที่ช่วยให้แอนดรอยด์แพลตฟอร์ม รองรับการใช้งาน JavaScript, AJAX และ CSS รวมทั้งสามารถเปิดหน้าเว็บไซต์ได้อย่างมีประสิทธิภาพ เช่นเดียวกันกับ Safari ที่ทำงานอยู่บน OS X นอกจากนั้นแล้วการที่แอนดรอยด์แพลตฟอร์มใช้ Webkit ยังส่งผลให้การทำงานของซอฟต์แวร์ที่ Google ได้พัฒนาขึ้นมาอย่าง Chrome และ Gear ร่วมกันได้ดีอีกด้วย ซึ่งการทำงานหลักของไลบรารีนี้ จะประกอบไปด้วยการแสดงผลของหน้าเว็บ และการจัดการกับ Cookies

3.2 แอนดรอยด์รันไทม์

ในส่วนของแอนดรอยด์รันไทม์ นั้นจะมีการทำงานควบคู่ไปกับส่วนของ Libraries โดยจะมีหน้าที่หลักคือการ Interpret โค้ดจาวาถูก Compile แล้วมาทำงาน ซึ่งในส่วนนี้จะมีระบบการทำงานย่อยที่แบ่งได้อีกสองส่วนคือ

3.2.1 Dalvik Virtual Machine จะสามารถเปรียบได้กับ Java Virtual Machine ในจาวา แต่ว่าในส่วนของ Dalvik Virtual Machine นั้นได้ถูกพัฒนาขึ้นมาโดย

Google ซึ่งจะแตกต่างจากจาวาทั่วไปตรงที่จะทำการรันไฟล์ .dex ไฟล์ ที่ถูกแปลงมาจาก .class เพื่อให้เหมาะสมกับอุปกรณ์ที่ใช้งานแอนดรอยด์แพลตฟอร์ม ที่มี หน่วยความจำน้อย

3.2.2 Core Libraries เป็นองค์ประกอบที่ทำให้แอนดรอยด์สามารถใช้งานฟังก์ชันพื้นฐานของจาวา ซึ่งมีพื้นฐานมาจาก Java SE 5 แต่ฟังก์ชันพื้นฐานจะถูกลดให้เหลือเพียงเท่าที่จำเป็น อย่างเช่น Input/output (java.io), Security (java. Security) และ Utility (java.util) เป็นต้น

3.3 แอปพลิเคชันเฟรมเวิร์ค

ในส่วนนี้เป็นระดับที่รวบรวม Service ทั้งหมดที่แอนดรอยด์แพลตฟอร์ม มีให้พร้อมสำหรับซอฟต์แวร์เรียกใช้ ซึ่งในระดับนี้เองที่ทำให้นักพัฒนาซอฟต์แวร์สามารถเรียกใช้ APIs และเข้าถึงข้อมูลต่างๆได้ง่าย ไม่ว่าจะเป็น Activity Manager ที่เป็นตัวกำหนดการทำงานในแต่ละ Activity, Content Providers ที่ทำให้ซอฟต์แวร์แต่ละตัวสามารถส่งผ่านข้อมูลกันได้ และ View System ที่รวบรวม UI ชนิดต่างๆเอาไว้เพื่อให้นักพัฒนาเรียกใช้ เป็นต้น ซึ่งในส่วนของ Application Framework ยังสามารถแบ่งได้อีก 2 ส่วนคือ

3.3.1 Core Platform Service หมายถึง Service ที่ใช้ในการจัดการเกี่ยวกับการรันโปรแกรมและควบคุมการทำงานของโปรแกรม ซึ่งจะมีส่วนประกอบเช่น Activity Manager จะเป็นส่วนที่ใช้ควบคุมการทำงานของแต่ละ Activity ซึ่งจะอธิบายในหัวข้อต่อไป Content Providers จะเป็น Service หนึ่งที่ทำให้ซอฟต์แวร์บนแอนดรอยด์แพลตฟอร์มนั้นสามารถส่งผ่านข้อมูลถึงกันได้ เช่น การไปดึงข้อมูลในส่วนของ Contact List มาเพื่อใช้งานในซอฟต์แวร์อื่นๆ เป็นต้น View System จะเป็นส่วนที่รวบรวม User Interface ต่างๆไว้ให้เรียกใช้ ไม่ว่าจะเป็น Text View และ Button เป็นต้น นอกจากนี้ Service ข้างต้นแล้ว ในส่วนนี้ยังประกอบไปด้วย Package Manager, Window Manager และ Resource Manager

3.3.2 Hardware Service หมายถึง Service ที่เกี่ยวข้องกับการใช้งาน Hardware ต่างๆ โดยในประเภทนี้จะมีส่วนประกอบเด่นๆดังนี้ Telephony Service ที่จะใช้ควบคุมฟังก์ชันที่ใช้โทรศัพท์ Bluetooth Service ที่ใช้ควบคุมการใช้งาน Bluetooth นอกจากนี้แล้วก็ยังมี Location Service, Wi-Fi Service, USB service และ Sensor Service อีก เป็นต้น

3.4 แอปพลิเคชันเลเยอร์

คือระดับที่หมายถึง ซอฟต์แวร์ หรือ โปรแกรมที่ใช้อยู่โดยทั่วไป อย่างเช่น โปรแกรมรับส่งอีเมล, SMS, แผนที่, Browser เครื่องมือจัดการสมุดโทรศัพท์ และโปรแกรมหลักอื่นๆ ซึ่งในขั้นนี้ยังรวมถึงซอฟต์แวร์ที่เราได้ทำการพัฒนาขึ้นมาด้วย

4. แอนดรอยด์ Building Blocks

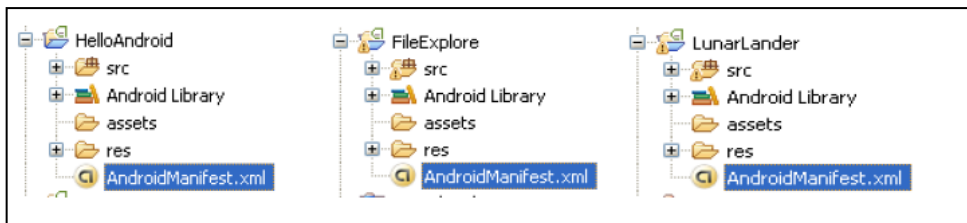
จากการที่ซอฟต์แวร์ที่ทำงานบนแอนดรอยด์แพลตฟอร์มนั้นเกิดจากการทำงาน ที่รวมกันขึ้นมาจาก Service ต่างๆเข้าด้วยกันในระดับ Application Platform ซึ่งผู้พัฒนา

ซอฟต์แวร์จะต้องคำนึงถึงการทำงานของซอฟต์แวร์เพื่อที่จะเรียกใช้งาน Service ต่าง ๆ เหล่านี้ จาก Android APIs ให้เหมาะสมกับซอฟต์แวร์ โดยรายละเอียดต่อไปนี้เป็นลักษณะ ต่าง ๆ ของแต่ละ Android APIs เพื่อการนำไปใช้ในการทำงานที่แตกต่างกัน

4.1 AndroidManifest.xml

จากการที่ซอฟต์แวร์ที่ทำงานบนแอนดรอยด์แพลตฟอร์มนั้นเกิดจาก การทำงานที่รวมกันขึ้นมาจาก Service ต่างๆเข้าด้วยกันในระดับ Application Platform ซึ่งผู้พัฒนาซอฟต์แวร์จะต้องคำนึงถึงการทำงานของซอฟต์แวร์เพื่อที่จะเรียกใช้งาน Service ต่างๆ เหล่านี้จาก Android APIs ให้เหมาะสมกับซอฟต์แวร์โดยรายละเอียดต่อไปนี้เป็นลักษณะ ต่างๆของแต่ละ Android APIs เพื่อการนำไปใช้ในการทำงานที่แตกต่างกัน ซึ่งการที่ได้ นำองค์ประกอบที่เรียกใช้ Android APIs ต่างๆนั้นก็คือ BuildingBlock 2.4.1

AndroidManifest.xml ตัวไฟล์ AndroidManifest.xml เป็นไฟล์ที่จะใช้ควบคุมการทำงานของซอฟต์แวร์ที่ได้พัฒนาขึ้นมาให้ทำงานไปตามลำดับ เป็นไฟล์ที่ใช้สำหรับประกาศเพื่อเรียกใช้งาน ส่วนประกอบอื่นๆ หรือ Service ต่างๆให้ทำงานร่วมกัน และยังเป็นไฟล์ที่เปรียบเสมือนแหล่งอ้างอิงของซอฟต์แวร์ที่พัฒนาขึ้นมาอีกด้วย ซึ่งไฟล์ AndroidManifest.xml จะมีรายละเอียดที่สำคัญในส่วนต่างๆ ดังนี้



ภาพที่ 3 แสดงตำแหน่งของ AndroidManifest.xml บน Project

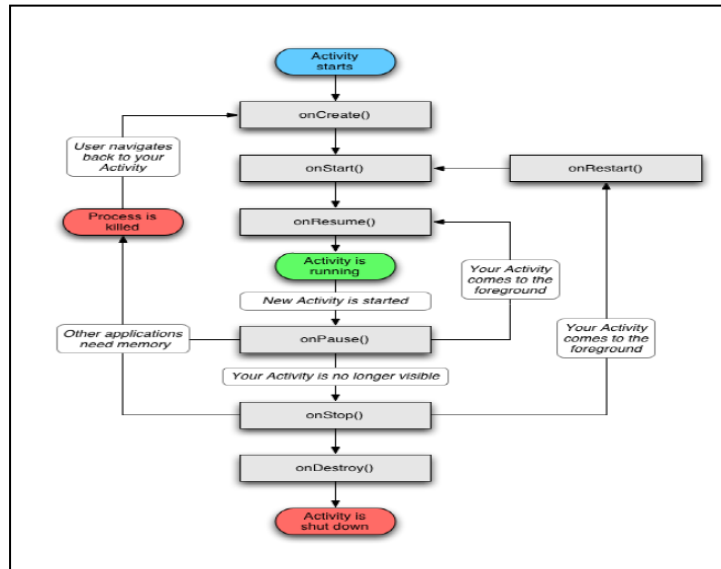
ที่มา : คมสันต์ ผุยสถานวงศ์. (2555 : 12)

และสิ่งสำคัญอีกสิ่งหนึ่งของ Android Manifest คือการประกาศ intent-filters โดยจะใช้เป็นตัวเรียกให้ Activity อื่นที่อยู่ในซอฟต์แวร์ทำงาน การทำงานของ intent-filters จะเริ่มต้นขึ้นเมื่อมีการเรียกใช้ intent หลังจากนั้นจะทำการเปรียบเทียบกับ intent-filters เพื่อดำเนินการต่อไป นอกจากนี้แล้วในการทำงานด้านอื่นๆของ Building Block ด้วย

4.2 Activity

โดยทั่วไปความหมายของ Activity ใน แอนดรอยด์แพลตฟอร์มนั้นจะหมายถึง วัตถุหนึ่งๆ ที่มีการทำงานหลายๆอย่าง และเป็นการทำงานที่มีการแสดงผลรวมอยู่ด้วย ซึ่งในแอปพลิเคชันหนึ่งๆนั้นอาจจะประกอบไปด้วยหลายๆ Activity ได้ และในหนึ่งแอปพลิเคชันนั้นจำเป็นต้องมีอย่างน้อย 1 Activity ซึ่งในการทำงานกับ Activity นั้น จะมีลักษณะการทำงานเป็นวงจร จึงทำให้การทำงานของซอฟต์แวร์บนระบบแอนดรอยด์แพลตฟอร์มมีระเบียบ

แบบแผน Activity Lifecycle เมื่อ Activity เริ่มมีการทำงานจะถูกเก็บไว้ในส่วนของ System ที่เรียกว่า Activity stack ซึ่งเมื่อ Activity เริ่มทำงานก็就会被จัดเก็บไว้ในส่วนบนสุดของ Stack นั้น แต่ในส่วนของ Activity ที่ถูกเรียกใช้งานก่อนหน้าก็就会被จัดเก็บเอาไว้ใน Stack ชั้นต่อไป จากการทำงานของวงจรเบื้องต้นดังกล่าวนี้ จะสามารถศึกษาได้จากวงจรที่เรียกว่า Activity Lifecycle ดังแผนภาพที่ 1



แผนภาพที่ 1 แสดง Activity Lifecycle ของ Android
ที่มา : คมสันต์ ผุยถานวงศ์. (2555 : 13)

การชำระค่าบริการ

1. ความรู้เกี่ยวกับ Paypal

Paypal คือ ระบบการรับเงิน โอนเงินออนไลน์ ที่เป็นนิยมในการซื้อ ขายสินค้าออนไลน์ ในเว็บไซต์ e-commerce ระดับโลกมากมาย ที่อำนวยความสะดวก พร้อมด้วยระบบที่มีความปลอดภัย เชื่อถือได้ วิธีการใช้งานเริ่มต้น เพียงมีบัตรเครดิต หรือบัตรเสมือนจริง ก็สามารถใช้บริการ และเป็นสมาชิกของ paypal ได้

1.1 สิทธิประโยชน์ของการใช้ PayPal

- 1.1.1 รวดเร็ว ชำระเงินได้ในทันที เร็วกว่าส่งเช็คหรือธนบัตร
- 1.1.2 เป็นส่วนตัว PayPal จะไม่เปิดเผยข้อมูลทางการเงินของคุณให้กับ

ผู้ขาย

1.1.3 ใช้ได้ทั่วโลก PayPal ได้รับการยอมรับทั่วโลก และสามารถชำระเงินได้ในประเทศหรือต่างประเทศ

- 1.1.4 ทำได้ง่ายๆ ชำระเงินให้ใครๆ ได้ด้วยไม่กี่คลิก

1.1.5 นำเชื่อถือ ระบบป้องกันการฉ้อโกงระดับชั้นนำของธุรกิจด้านนี้ช่วยให้คุณมั่นใจและปลอดภัย

1.1.6 คุ่มค่า ชำระเงินได้ฟรี และ PayPal เหมาะสมกับธุรกิจทุกประเภท

1.2 การชำระเงิน

กระบวนการชำระเงินของ PayPal แสดงดังภาพที่ 4



ภาพที่ 4 กระบวนการชำระเงินของ PayPal

1.3 จุดเด่นของ Paypal

1.3.1 ชำระเงินได้รวดเร็วทันใจ หลายล้านคนทั่วโลกเลือกใช้ PayPal ด้วยเหตุผลง่ายๆ ข้อเดียวคือ เพราะใช้ง่าย แค่ใช้ที่อยู่อีเมลล์และรหัสผ่านก็สามารถชำระเงินได้เสร็จเรียบร้อยเร็วกว่าหยิบกระเป๋าสตางค์เสียอีก

1.3.2 ชำระเงินได้ไม่ว่าจะอยู่ที่ไหน ซื้อมีได้ทุกที่ไม่ว่าจะอยู่ที่ไหน และเลือกซื้อสินค้าที่ถูกใจโดยไม่ต้องต่อแถว นี่คือนิสัยที่ง่ายและปลอดภัยกว่าในการชำระเงินเมื่อคุณอยู่นอกบ้าน

1.3.3 ชำระเงินให้ใครก็ได้ ที่ไหนก็ได้ทั่วโลก ชำระค่าสินค้าหรือบริการได้ง่ายและปลอดภัยกว่า โดยใช้แค่ที่อยู่อีเมลล์ ไม่ว่าผู้รับจะอยู่ที่ไหนก็สามารถรับเงินจากคุณได้อย่างสะดวกสบาย

1.3.4 เรียกเก็บเงินได้ง่ายๆ การเรียกเก็บเงินเป็นการแจ้งเตือนแบบกันเอง หลังจากทำงานเสร็จ โดยอีกฝ่ายจะได้รับอีเมลล์แจ้งให้ชำระเงินด้วยวิธีใดก็ได้ตามที่สะดวกแบบไม่ต้องคิดมาก

2. ความรู้ในการใช้เทคโนโลยี Push Notification โดยใช้บริการ Google Cloud Message

2.1 การทำงานของระบบ Push

ทุกโปรแกรมจะไปลงทะเบียนตัวเองไว้กับระบบเล็กๆระบบหนึ่งที่มีชื่อว่า Push จะเป็นคนกลางที่รับหน้าที่ รับ/ส่งข้อมูลแทนทุกๆโปรแกรม จากที่ 10โปรแกรมต่างคนต่างส่งข้อมูล 10 โปรแกรมก็ส่งไป 10ชุด Push จะเหมาไปส่งแค่ครั้งเดียว หมายความว่าส่งไปแค่ครั้งเดียว 2KB แต่เป็นการเช็คพร้อมกันทั้ง 10โปรแกรม

เมื่อข้อมูลถูกส่งไปยัง Server กลางของระบบ Push ทาง Server ของระบบ Push ก็จะไปเช็คไปยัง Server ของแต่ละโปรแกรม แล้วฝากข้อมูลขากลับมาที่ระบบ Push หากโปรแกรมไหนมีข้อมูลกลับมา ระบบ Push ก็จะแจ้งเตือน (Notification) ให้เราทราบ

2.2 Notification

ความหมายจาก พจนานุกรมแปล อังกฤษ-ไทย อ. สอ เสถบุตรพจนานุกรม อังกฤษ-ไทย อ. สอ เสถบุตร notification[n.] แจ้ง, บอกกล่าว คำอ่าน โนทิฟิเค-ฉัน คำพ้องความหมาย (Synonym) advise, attend, heed, inform, note, regard

ความหมายจาก พจนานุกรมแปล อังกฤษ-ไทย NECTEC's Lexitron Dictionary notification[n.] การแจ้ง (แจ้งความ, แจ้งการเกิด, อัตราเกิดของโรค ฯลฯ) การแจ้งความ, การประกาศ [syn.] manifesto

2.3 Push Notification

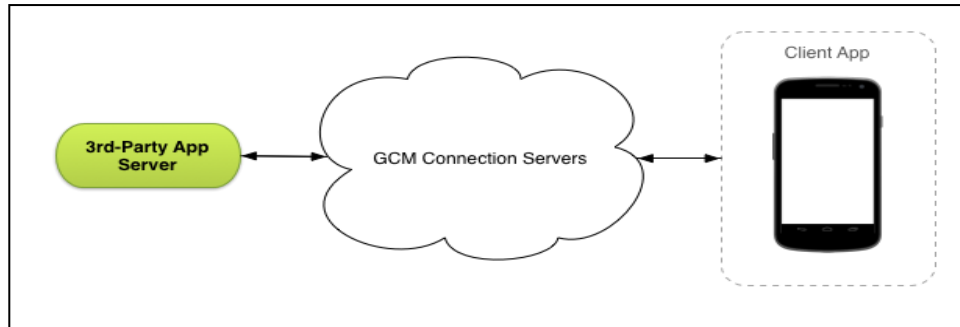
เป็นความสามารถหนึ่งของ Mobile Services ในการ Push Alert แจ้งไปยังเครื่อง Android Client ที่ติดตั้ง App ที่เราเขียนขึ้น และถ้าเราใช้ Android กับ Mobile Services ซึ่ง Android นั้นเป็นของ Google จึงจะต้องใช้บริการของ Google Cloud Messaging (GCM) ทำงานคู่กับ Mobile Services ของ Windows Azure และก่อนการใช้งานจะต้องทำการ Register บริการที่ Google APIs ของ Google ก่อน

Push Notification ก็เหมือนกับระบบ กันเตือนผู้ใช้ซึ่งเป็นของโทรศัพท์และอุปกรณ์ประเภทอื่นๆ เมื่อมีเหตุการณ์อะไรบางอย่างเกิดขึ้น ซึ่งเหตุการณ์ต่างๆ นี้เช่นการที่มีเมลล์ใหม่เข้ามา มีคนส่ง Message มาให้ มีการคนอัปเดตข้อความใน Facebook

2.4 GCM (Google Cloud Message)

2.4.1 การทำงานของ GCM

Google Cloud Messaging for Android (GCM) คือ บริการฟรี ที่ Google จัดให้ ให้เราสามารถส่งข้อมูลจากเซิร์ฟเวอร์ ไปยังเครื่อง Android ต่างๆได้ รวมถึงส่งข้อมูลจากเครื่อง Android กลับมา การทำงาน ของ GCM ดังแสดง ภาพที่ 5



ภาพที่ 5 การทำงาน ของ GCM

2.4.2 คุณสมบัติของ GCM มีอะไรบ้าง

1) ต้องใช้ 3rd-Party เป็น Server (Backend) สำหรับส่งข้อความไปให้เครื่อง user ใช้ GCM Cloud Connection Server (XMPP รูปแบบหนึ่ง) ในการรับข้อมูลจากเครื่อง user App Android ที่จะทำการรับส่งข้อมูล ไม่จำเป็นต้องเปิดโปรแกรมตลอดเวลา ระบบจะรู้เองว่า มีข้อมูลหรือต้องส่งข้อความตอนไหน ผ่าน Intent broadcast

2) GCM เป็นแค่ raw data เท่านั้น จะเอาไปใช้ทำอะไรก็ได้

3) ต้องใช้ Android Version 2.2 ขึ้นไป และต้องติดตั้งแอป Google Play หรือหากใช้ Emulator ก็ต้องสร้างแบบ Google APIs

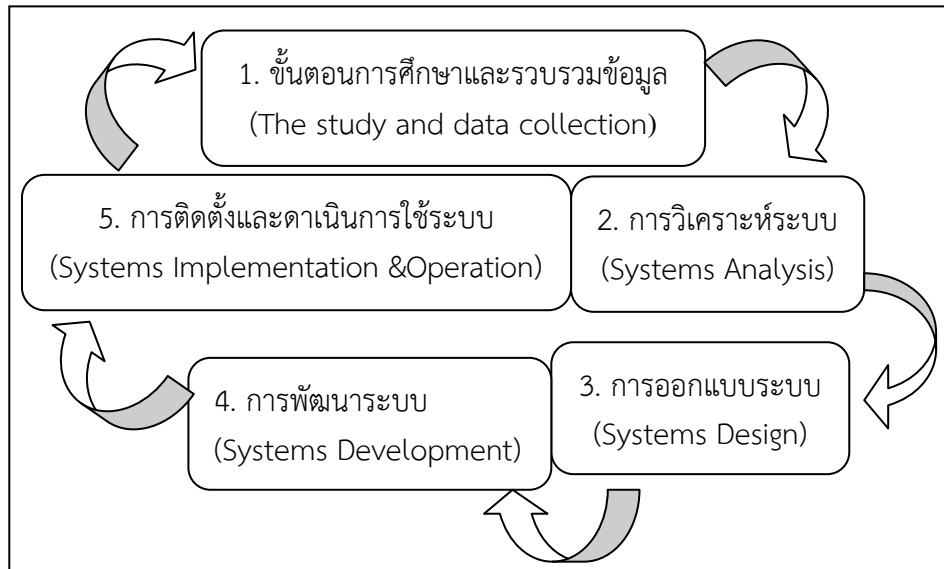
4) ต้องทำการเชื่อมต่อ Google Account ก่อน หากเป็นเครื่องก่อน 3.0 แต่ว่าตั้งแต่ 4.0.4 ใช้งานได้เลย

การวิเคราะห์และออกแบบระบบ

1. วงจรการพัฒนา ระบบ SDLC

วงจรการพัฒนา ระบบ (System Development Life Cycle : SDLC) คือ กระบวนการทางความคิด (Logical Process) ในการพัฒนาระบบสารสนเทศเพื่อแก้ปัญหา และตอบสนองความต้องการของผู้ใช้ได้ โดยระบบที่จะพัฒนา อาจเริ่มด้วยการพัฒนาระบบใหม่ หรือนำระบบเดิมที่มีอยู่แล้วมาปรับเปลี่ยน ให้ดียิ่งขึ้น ภายในวงจรนี้จะแบ่งกระบวนการพัฒนา ออกเป็นระยะ ได้แก่ ขั้นตอนการสำรวจระบบ (System investigation) ขั้นตอนการวิเคราะห์ระบบ (System Analysis) ขั้นตอนการออกแบบและพัฒนาระบบสารสนเทศ (System design) ระยะการสร้างและพัฒนา (Implementation Phase) การทดลองใช้และติดตั้งระบบ (System implementation) และการบำรุงรักษาระบบและการประเมินผล (System maintenance and review) โดยแต่ละระยะจะประกอบไปด้วยขั้นตอนต่าง ๆ แตกต่างกันไป ตาม

Methodology ดังนั้น การศึกษาครั้งนี้ได้ประยุกต์ใช้วงจรการพัฒนากระบวนสารสนเทศ (System development life cycle : SDLC) (Stair 1996 : 411-412) ซึ่งประกอบด้วยกิจกรรมต่าง ๆ 5 ขั้น ดังนี้



แผนภาพที่ 2 แสดงวงจรการพัฒนากระบวน SDLC

1.1 ขั้นตอนการศึกษาและรวบรวมข้อมูล (The study and data collection)

การศึกษาและรวบรวมข้อมูล (The study and data collection) เป็นสิ่งสำคัญที่จะทำให้การพัฒนากระบวนสารสนเทศเป็นไปด้วยดีตามแผน เพราะหากไม่มีการวางแผนที่ดีอาจเกิดความล่าช้าของการพัฒนากระบวนสารสนเทศได้ และเป็นขั้นตอนแรกสำหรับเตรียมความพร้อมในการพัฒนากระบวนสารสนเทศ ด้วยการตั้งประเด็นคำถามที่ว่า มีความต้องการอะไรบ้างในกระบวนสารสนเทศที่จะพัฒนาขึ้น

1.1.1 กำหนดโอกาสของกระบวนสารสนเทศในการใช้งาน (Identify Opportunity) การศึกษาระบบงานปัจจุบันเป็นการศึกษาปัญหาที่เกิดขึ้น จากการใช้งานระบบที่มีอยู่ จุดบันทึกถึงความต้องการสารสนเทศที่จะต้องปรับปรุงเพิ่มเติม อาจสำรวจจากการสัมภาษณ์ แบบสอบถาม

1.1.2 วิเคราะห์ความเป็นไปได้ (Analyze Feasibility) กำหนดขอบเขตของการพัฒนาแอปพลิเคชัน การกำหนดรายละเอียดและขั้นตอนของการดำเนินงาน และระยะเวลาที่ใช้ และหากผลการสำรวจ พบว่าระบบงานนั้นมีความเป็นไปได้ที่จะพัฒนานักวิเคราะห์ระบบ จะวิเคราะห์ต้นทุน/ผลตอบแทน (Cost-Benefit Analysis) ในการศึกษา เพื่อหาแนวทางที่เหมาะสมของกระบวนสารสนเทศที่จะพัฒนาขึ้น

1.1.3 พัฒนาแผนการทำงาน (Develop Work plan) การจัดทำข้อเสนอโครงการสำหรับพัฒนาระบบสารสนเทศฯ ซึ่งเป็นการเก็บรวบรวมข้อมูลตามความต้องการของผู้บริหารทุกระดับ และบุคลากรระดับปฏิบัติการ จากนั้นนำมาวิเคราะห์เพื่อวางแผนถึงทางเลือกต่าง ๆ ของระบบสารสนเทศที่จะพัฒนาขึ้น พร้อมทั้งสรุปผล และนำเสนอแนวทางที่เหมาะสมที่สุด เพื่อนำไปวิเคราะห์และออกแบบระบบต่อไป

1.2 การวิเคราะห์ระบบ (Systems Analysis)

ผู้ศึกษาได้วิเคราะห์และออกแบบเชิงวัตถุโดยใช้ยูเอ็มแอล (Unified Modeling Language : UML) การวิเคราะห์และออกแบบระบบเชิงวัตถุเป็นวิธีที่นิยมกันมากในปัจจุบัน และมีแนวโน้มที่จะทดแทนการออกแบบระบบแบบเดิม กระบวนการพัฒนาระบบตามแบบวิธี Rational Unified Process หรือ Rational Objectory Process เป็นกระบวนการที่ครอบคลุมกระบวนการพัฒนาระบบทั้งหมด โดยการพิจารณาทั้งงานด้านการบริหารและงานด้านเทคนิค กระบวนการพัฒนาจะมีลักษณะการทำซ้ำ (Iterative) และการเพิ่มขึ้น (Incremental) ดังนั้นงานที่ทำจะไม่มีมากในคราวเดียวกันในตอนสุดท้ายของโครงการ แต่จะมีการแบ่งงานออกเป็นช่วง ๆ (Phase) ในช่วงของการสร้างระบบ (Construction Phase) การทดสอบ และการรวบรวมส่วนย่อยเข้ากับระบบรวม จะมีการทำซ้ำหลาย ๆ ครั้ง เพื่อจะให้ได้โปรแกรมที่มีคุณภาพ และตรงตามความต้องการ ในการทำซ้ำแต่ละรอบจะประกอบด้วย การวิเคราะห์ (Analysis) การออกแบบ (Design) การเขียนโปรแกรมที่ใช้ในการสร้างและการพัฒนาซอฟต์แวร์ (Implement) และการทดสอบระบบ (Testing) (ชาติ และเทพฤทธิ, 2544 : 38 - 80)

การวิเคราะห์การพัฒนาแอปพลิเคชันการสอบรายวิชาเทคโนโลยีเว็บ เป็นขั้นตอนที่สำคัญมากที่จะทำให้การพัฒนาระบบมีประสิทธิภาพ โดยการวิเคราะห์จะเกี่ยวข้องกับการออกแบบผังรายละเอียดต่าง ๆ ของการดำเนินงานและสร้างผังการทำต่าง ๆ ได้แก่ Use-Case Diagram Activity Diagram SequenceDiagram Class Diagram และพจนานุกรมข้อมูล (Data Dictionary) (ธีรพล ด่านวิริยะกุล. 2549 : 21-30)

1.3 การออกแบบระบบ (Systems Design)

การออกแบบระบบ (Systems Design) เป็นการนำข้อมูลจากการวิเคราะห์ระบบที่เป็นแนวคิด (Concept) มาออกแบบให้เห็นรูปร่างของระบบสารสนเทศ โดยนักวิเคราะห์ระบบจะออกแบบระบบทีละส่วน โดยเริ่มจากส่วนที่เป็นผลลัพธ์ (Output) ก่อน เพราะผลลัพธ์นั้นเกิดจากการนำข้อมูลเข้าระบบแล้วไปประมวลผล ดังนั้น การออกแบบผลลัพธ์หรือส่วนแสดงผลจะทำให้ทราบถึงการออกแบบในส่วนอื่น ๆ ที่เกี่ยวข้อง

1.4 การพัฒนาระบบ (Systems Development)

การพัฒนาระบบ (Systems Development) หลังจากทีนักวิเคราะห์ระบบได้ออกแบบระบบใหม่และจัดการสิ่งซึ่งอุปกรณ์ต่าง ๆ เสร็จเรียบร้อยแล้ว ในขั้นตอนนี้คือการนำระบบที่ได้ออกแบบมาแล้วมาพิจารณาเพื่อสร้าง Program Software ที่จะใช้งานโดยนักเขียน-โปรแกรม จะเขียนโปรแกรมตามที่นักวิเคราะห์ระบบได้ออกแบบไว้ เมื่อสร้างระบบเสร็จเรียบร้อยแล้ว จะต้องนำ Software ที่สร้างไว้แล้วมาทดสอบ

1.5 การติดตั้งและดำเนินการใช้ระบบ (Systems Implementation & Operation) การติดตั้งและดำเนินการใช้ระบบ (Systems Implementation & Operation) เมื่อดำเนินการสร้างระบบและทำการตรวจสอบแก้ไขเสร็จเรียบร้อยแล้ว ระบบงานจะถูกส่งมอบและทำการติดตั้งระบบ (Installed System) ลงบนมือถือ เพื่อนำไปใช้ในการปฏิบัติงาน ควรมีการประเมินและสร้างการยอมรับระบบงานใหม่ให้กับบุคลากรที่ใช้ระบบสารสนเทศ ซึ่งการดำเนินการใช้ระบบ

2. การวิเคราะห์และออกแบบเชิงวัตถุโดยใช้ยูเอ็มแอล (Unified Modeling Language : UML)

การวิเคราะห์และออกแบบระบบเชิงวัตถุเป็นวิธีที่นิยมกันมากในปัจจุบัน และมีแนวโน้มที่จะทดแทนการออกแบบระบบแบบเดิม กระบวนการพัฒนาระบบตามแบบวิธี Rational Unified Process หรือ Rational Objectory Process เป็นกระบวนการที่ครอบคลุมกระบวนการพัฒนาระบบทั้งหมด โดยการพิจารณาทั้งงานด้านการบริหารและงานด้านเทคนิค กระบวนการพัฒนามีลักษณะการทำซ้ำ (Iterative) และการเพิ่มขึ้น (Incremental) ดังนั้นงานที่ทำจะไม่มีมากในคราวเดียวกันในตอนสุดท้ายของโครงการ แต่จะมีการแบ่งงานออกเป็นช่วง ๆ (Phase) ในช่วงของการสร้างระบบ (Construction Phase) การทดสอบ และการรวบรวมส่วนย่อยเข้ากับระบบรวม จะมีการทำซ้ำหลาย ๆ ครั้ง เพื่อจะให้ได้โปรแกรมที่มีคุณภาพ และตรงตามความต้องการ ในการทำซ้ำแต่ละรอบจะประกอบด้วยการวิเคราะห์ (Analysis) การออกแบบ (Design) การเขียนโปรแกรมที่ใช้ในการสร้างและการพัฒนาซอฟต์แวร์ (Implement) และการทดสอบระบบ (Testing) โดยสามารถแสดงได้ดังนี้ (ชาลี และเทพฤทธิ์, 2544 : 38 - 80)

2.1 ช่วงของการพัฒนาระบบ

2.1.1 อินเซพชันเฟส (Inception Phase) เป็นการเก็บข้อมูลพื้นฐานเกี่ยวกับระบบที่ต้องการ โดยจะมีความเกี่ยวข้องกับฟังก์ชันการทำงานต่าง ๆ ความสามารถ ประสิทธิภาพเทคโนโลยีที่ใช้ และคุณสมบัติอื่น ๆ อีกทั้งยังเป็นการกำหนดแนวคิดเพิ่มเติมและแสดงวิธีที่ใช้ในการพัฒนาในขั้นตอนต่อไป และแสดงวิธีการที่ทำให้ระบบมีความสามารถมากขึ้น โดยผลลัพธ์ที่ได้จากกระบวนการนี้จะปรากฏอยู่ในรูปของงานโดยรวม ซึ่งแสดงว่าจะต้องสร้างอะไรขึ้นมาบ้างกำหนดว่าจะสร้างได้อย่างไร และมีการทำงานอย่างไร กระบวนการนี้จำเป็นต้องมีทักษะในการวิเคราะห์ระบบให้ออกมาอยู่ในรูปของฟังก์ชันหลักของระบบ และผู้ติดต่อกับระบบ

(Actor) ซึ่งอธิบายอยู่ในรูปของมุมมองการใช้งาน (Use Case View) และยังต้องมีการวางแผนด้านงบประมาณค่าใช้จ่ายในการพัฒนาระบบ ความสามารถทางการตลาดการวิเคราะห์ความเสี่ยง และผลิตภัณฑ์ของคู่แข่งในกรณีการพัฒนาระบบเพื่อธุรกิจ

2.1.2 อีลาโบริชันเฟส (Elaboration Phase) จะประกอบไปด้วยรายละเอียดของการวิเคราะห์ระบบ การกำหนด และวางแผนก่อนการทำงานขั้นตอนต่าง ๆ ได้แก่

1) แผนผังที่แสดงภาพในเชิงสถิตยของระบบ (Static Diagram) โดยจะแสดงถึงการมีอยู่ของคลาส และความสัมพันธ์ระหว่างคลาส แต่จะไม่แสดงถึงกิจกรรมที่จะเกิดขึ้น ซึ่งมี 2 แผนผังที่ใช้งาน คือ แผนผังการใช้งานของระบบ (Use Case Diagram) และแผนผังอธิบายความสัมพันธ์ของเอนทิตี (Entity) ต่าง ๆ ของระบบ (Class Diagram)

2) แผนผังที่แสดงภาพในเชิงกิจกรรมของระบบ (Dynamic Diagram) โดยเป็นการแสดงถึงสิ่งที่เกิดขึ้นจากกิจกรรมของ Class ต่าง ๆ ที่มีอยู่ในระบบ ซึ่งมีแผนผังที่ใช้งาน คือ แผนผังแสดงการทำงานระหว่างออบเจกต์ (Sequence Diagram) และ แผนผังแสดงสถานะ (State chart Diagram) ซึ่งแสดงสถานะต่าง ๆ ที่ คลาสหนึ่งคลาสจะเป็นได้ในระหว่างช่วงชีวิตในการตอบสนองต่อเหตุการณ์ (Event) ที่เกิดขึ้น

3) คอนสตรัคชันเฟส (Construction Phase) เป็นการพัฒนาระบบจริงขึ้น โดยเป็นการเขียนโปรแกรม ซึ่งมีการพัฒนาแบบทำซ้ำและเพิ่มขึ้นเรื่อย ๆ ซึ่งกระบวนการที่ทำซ้ำจะประกอบด้วย ขั้นตอนการวิเคราะห์ ออกแบบ เขียนโปรแกรม และการทดสอบ จากนั้นทำการรวมเป็นระบบใหญ่ขึ้นจนได้ระบบที่ต้องการผลลัพธ์ของการทำงานช่วงนี้คือ ระบบที่ต้องการ

4) ทรานซิชันเฟส (Transition Phase) เป็นกระบวนการของการส่งผลิตภัณฑ์ไปสู่ผู้ใช้งานจริง รวมไปถึงการหาลาดหรือ การแพ็คเกจ (Packaging) และการบำรุงรักษาและการสอนการใช้โปรแกรมและจัดทำคู่มือการใช้โปรแกรม

3.1 ส่วนประกอบของ UML ประกอบด้วยส่วนต่าง ๆ ดังนี้

3.1.1 มุมมอง (View) เป็นระบบงานทั้งหมดอาจมีหลายส่วนที่ต้องพิจารณา เพราะอาจมีขอบข่ายงานที่กว้างขวางและซับซ้อนการอธิบายกระบวนการทำงานต่าง ๆ ของระบบไม่สามารถอธิบายได้เพียงแค่มุมมองเดียว ดังนั้นการมองระบบควรจะต้องเป็นมุมมองต่าง ๆ กัน เช่น มุมมองด้าน Functional, Nonfunctional มุมมองขององค์กร เป็นต้น ซึ่งแต่ละไดอะแกรมสามารถที่จะมีมุมมองของผู้ใช้งานระบบ ผู้เขียนโปรแกรมพัฒนาระบบ ซึ่งแต่ละมุมมองทำให้ผู้ทำระบบเข้าใจระบบในแง่มุมมองที่ต่าง ๆ กัน มุมมองต่าง ๆ ของ UML มีดังนี้

3.1.2 มุมมองการใช้งาน (Use Case View) เป็นการมองระบบจากผู้ใช้ภายนอกหรือผู้ระบบซึ่งไดอะแกรมที่ใช้อธิบาย คือ ยูสเคสไดอะแกรม (Use-Case Diagram) หรือบางครั้งแอกทิวิตีไดอะแกรม (Activity Diagram) ตัวอย่างผู้ใช้งาน เช่น ลูกค้า ผู้ออกแบบ

ผู้ทดสอบระบบนักเรียน อาจารย์ เป็นต้น ยูสเคส (Use Case) ในยูสเคสไดอะแกรมเป็นตัวกำหนดเป้าหมายของระบบ จึงเป็นตัวกลางของมุมมองอื่น ๆ ที่จะต้องมีการทำงาน ต่าง ๆ ครอบคลุมที่กำหนดไว้ในยูสเคสไดอะแกรม

3.1.3 มุมมองทางตรรกะ (Logical View) ใช้อธิบายว่าสามารถที่จะจัดการทำงานของระบบให้เป็นไปตามที่ต้องการได้อย่างไรและมีบริการอะไรให้กับผู้ใช้งาน Logical View ต่างจาก Use Case View เนื่องจากเป็นมุมมองของผู้ออกแบบและพัฒนาระบบ โดยจะแสดงในรูปแบบของโครงสร้างแบบสถิต (Static) เช่น คลาส ออบเจกต์ (Object) ความสัมพันธ์ระหว่างการทำงานร่วมกันแบบไดนามิก (Dynamic Collaboration) ซึ่งเกิดเมื่อออบเจกต์ส่งเมสเสจระหว่างการทำงาน

3.1.4 มุมมองในการนำไปใช้ (Deployment View) เป็นการแสดงการจัดระบบในระดับกายภาพ (Physical) ให้เหมาะสม เช่น การเชื่อมต่อระหว่างคอมพิวเตอร์ และโหนดต่าง ๆ และรวมถึงการแมพ (Map) คอมพิวเตอร์ต่าง ๆ ในระดับโครงสร้างทางกายภาพ เช่น ลำดับของหรือโปรแกรมในแต่ละเครื่องคอมพิวเตอร์ใช้สำหรับผู้พัฒนาระบบ ผู้ร่วมพัฒนาระบบ ผู้ทดสอบระบบอธิบายโดยดีพลอยเมนต์ไดอะแกรม (Deployment Diagram)

3.1.5 มุมมองของกระบวนการ (Process View) ไดอะแกรมเป็นกราฟซึ่งแสดงโดยสัญลักษณ์ที่จัดเรียงขึ้น เพื่อใช้อธิบายระบบในมุมมองต่าง ๆ ในระบบหนึ่ง ๆ จะประกอบไปด้วยหลาย ๆ ไดอะแกรม แต่ละไดอะแกรมยังสามารถมองได้หลาย ๆ มุมมองด้วย

4.1 ไดอะแกรมใน UML ประกอบด้วย

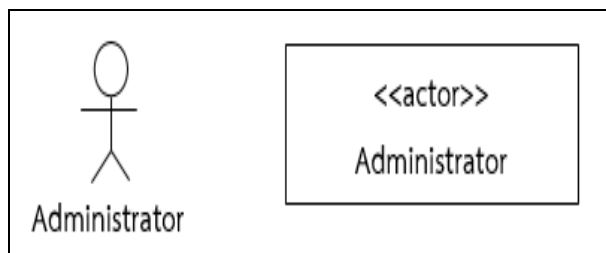
4.1.1 ยูสเคสไดอะแกรม (Use-Case Diagram) สิ่งที่สำคัญในการสร้างยูสเคส คือการค้นหว่าระบบทำงานอะไรได้บ้าง โดยไม่สนใจว่าข้างในสิ่งที่ระบบต้องทำมีกลไกการทำงานอย่างไรหรือใช้เทคนิคการสร้างอย่างไรเปรียบเสมือนเป็น “กล่องดำ” (Black Box) ยูสเคสไดอะแกรมจะแสดงความสัมพันธ์ระหว่างผู้ใช้งานกับระบบ ซึ่งจะมีแอกเตอร์ (Actor) กับระบบ โดยติดต่อผ่านยูสเคสต่าง ๆ ที่เกี่ยวข้องและจะใช้ในการสื่อสารกับผู้ใช้ เพื่ออธิบายถึงฟังก์ชันการทำงานหลักของระบบยูสเคสไดอะแกรม ก็คือ การทำงานต่าง ๆ ที่ผู้ใช้ต้องการ ซึ่งจะได้มาจากการสอบถามจากผู้ใช้

1) ยูสเคส (Use Case) คือ ความสามารถหรือฟังก์ชันที่ระบบซอฟต์แวร์จะต้องทำได้ เช่น ค้นหาข้อมูลของนักศึกษา คุณสมบัติของยูสเคส จะต้องถูกกระทำโดยแอกเตอร์ และแอกเตอร์เป็นผู้ติดต่อกับระบบตามยูสเคสที่กำหนดไว้ ยูสเคสรับข้อมูลจากแอกเตอร์ และส่งข้อมูลให้แอกเตอร์นั่นคือ แอกเตอร์กระทำกับยูสเคสโดยการส่งข้อมูลเข้าสู่ระบบตามยูสเคสหรือรอรับค่าที่ระบบจะส่งกลับให้ยูสเคส ถือว่าเป็นการรวบรวมเอา คุณลักษณะความต้องการในระบบอย่างสมบูรณ์เปรียบเสมือนเป็นการสรุปความต้องการของ ผู้ใช้ออกเป็นข้อ ๆ อย่างครบถ้วน โดยการเขียนยูสเคสใช้สัญลักษณ์รูปวงรี และคำอธิบาย ฟังก์ชันการทำงานอยู่ในวงรีนั้น ดังภาพที่ 4 แสดงยูสเคสการสร้างบัญชีชื่อผู้خبล็อก (Blog)



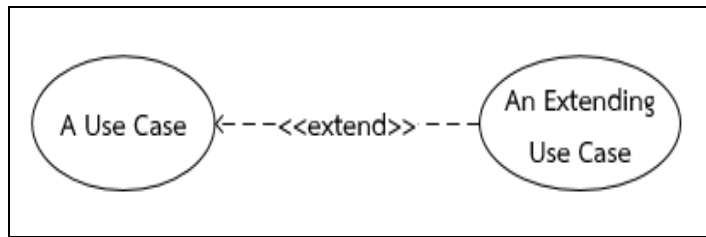
ภาพที่ 6 ตัวอย่างยูสเคสการสร้างบัญชีชื่อผู้ใช้บล็อก (Blog)
ที่มา : อีรพล ด่านวิริยะกุล. 2549 : 24

2) แอคเตอร์ (Actor) คือ ผู้ที่กระทำกับยูสเคสนั้น ๆ เช่น นักศึกษา อาจารย์เจ้าหน้าที่ไม่ใช่ส่วนประกอบของระบบแต่เป็นส่วนที่ใช้ติดต่อกับระบบ ซึ่งอาจเป็นเพียงการป้อนข้อมูลเข้าสู่ระบบหรือการส่งข้อมูลออกจากระบบหรืออาจเป็นทั้งสองอย่าง อาจมองได้เป็นแอกเตอร์หลัก หมายถึง แอกเตอร์ที่มีความสำคัญโดยตรงต่อความสามารถหลักของระบบซึ่งถูกแสดงด้วยยูสเคสผู้ใช้งานระบบจะให้ความสำคัญกับงานที่แอกเตอร์หลักจะต้องกระทำมากที่สุด แอกเตอร์รอง หมายถึง แอกเตอร์ที่มีหน้าที่สำคัญรองลงไปจาก แอกเตอร์หลัก โดยการเขียนแอกเตอร์จะใช้สัญลักษณ์รูปคน ดังภาพที่ 7 แสดงแอกเตอร์ผู้ดูแลระบบ



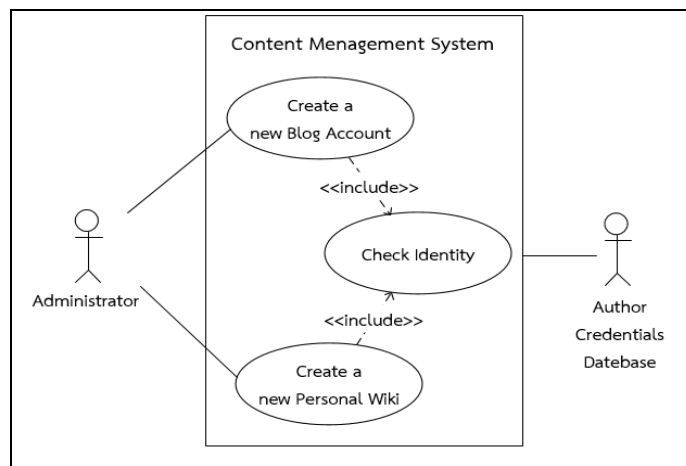
ภาพที่ 7 ตัวอย่างแอกเตอร์
ที่มา : อีรพล ด่านวิริยะกุล. 2549 : 24

3) เส้นความสัมพันธ์ระหว่างยูสเคส (Relationship) คือ เส้นเพื่อแสดงความสัมพันธ์ระหว่างแอกเตอร์ กับ แอกเตอร์ หรือ ยูสเคส กับ ยูสเคส ซึ่งมีอยู่สองชนิด ได้แก่ ความสัมพันธ์แบบขยาย (Extend Relationship) ยูสเคสหนึ่งอาจถูกช่วยเหลือโดยการทำงาน ยูสเคสอื่น สัญลักษณ์ใน UML คือ ลูกศรเส้นประที่ชี้จากยูสเคสแรกไปยังยูสเคสที่ถูกช่วยเหลือ หรือถูกขยาย โดยมีคำว่า “extend” อยู่ในเครื่องหมายสเตอริโอไทป์ (Stereotype) <<extend>> อยู่กึ่งกลางลูกศร ดังแผนภาพที่ 3 แสดงถึงยูสเคสด้านซ้ายได้รับฟังก์ชันการทำงาน จากยูสเคสด้านขวา



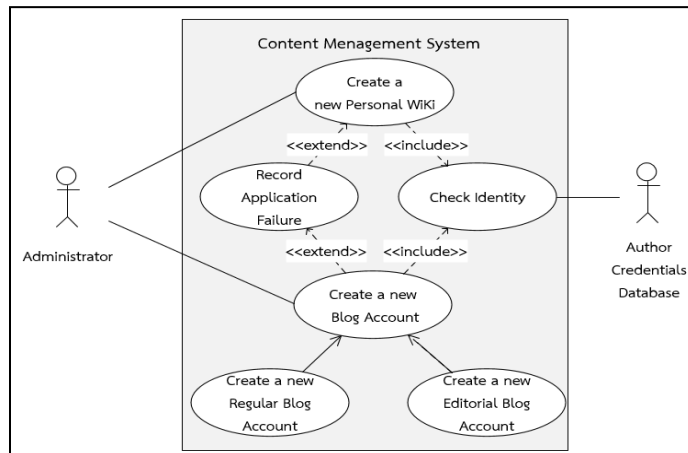
แผนภาพที่ 3 ความสัมพันธ์แบบขยาย
ที่มา : ชีรพล ด้านวิริยะกุล. 2549 : 25

ความสัมพันธ์แบบรวม (Include Relationship) ยูสเคสหนึ่ง ๆ อาจจำเป็นต้องอาศัยการทำงานของยูสเคสอื่น ๆ สำหรับยูสเคสที่ถูกเรียกใช้โดยยูสเคสอื่น สัญลักษณ์ใน UML ของความสัมพันธ์ดังกล่าวคือ ลูกศรเส้นประที่ชี้จากยูสเคสที่ถูกเรียกใช้ โดยมีคำว่า “include” อยู่ในเครื่องหมายสเตริโอไทป์ <<include>> อยู่ที่กึ่งกลางลูกศร ดังแผนภาพที่ 4 ในการสร้างบล็อก (Blog) ใหม่และสร้างข้อมูลส่วนตัวในเว็บวิกิพีเดียจำเป็นต้องผ่านการตรวจสอบ (Check Identity) ทุกครั้ง



แผนภาพที่ 4 ความสัมพันธ์แบบรวม
ที่มา : ชีรพล ด้านวิริยะกุล. 2549 : 25

ตัวอย่างการเขียนยูสเคสไดอะแกรมของระบบบริหารจัดการเว็บไซต์ ดังแผนภาพที่ 5 เมื่อผู้ดูแลระบบต้องการสร้างบล็อก (Blog) จำเป็นต้องสร้างข้อมูลส่วนตัวและสร้างบัญชีผู้ใช้ใหม่ โดยในการสร้างแต่ละครั้งจะทำการเก็บบล็อก (Log) ถ้าการสมัครเกิดความผิดพลาดเมื่อการลงทะเบียนสมบูรณ์ในการสร้างบล็อกและบัญชีจะถูกตรวจสอบโดยผู้ดูแลฐานข้อมูล



แผนภาพที่ 5 ตัวอย่างการเขียนยูสเคสไดอะแกรม
ที่มา : อีรพล ดำนวนวิริยะกุล. 2549 : 26

5.1 คลาสไดอะแกรม (Class Diagram) แสดงโครงสร้างของส่วนที่ไม่เปลี่ยนแปลงของระบบในมุมมองของผู้พัฒนาระบบ ซึ่งสามารถแสดงความสัมพันธ์ได้หลายวิธี ได้แก่ การเชื่อมต่อระหว่างกัน (Association) การพึ่งพาเรียกใช้คลาสอื่น (Dependent) ความเป็นลักษณะเฉพาะของคลาสอื่น (Specialized) รวมกันเป็นหน่วย (Package) ความสัมพันธ์ระหว่างคลาสต่าง ๆ เหล่านี้จะถูกแสดงโดยคลาสไดอะแกรม โดยรวมเข้าเป็นโครงสร้างภายในของคลาสเป็นกลุ่มแอตทริบิวต์ (Attribute) และกลุ่มโอเปอเรชัน (Operation) ในระบบหนึ่งสามารถประกอบด้วยหลายคลาสไดอะแกรม

5.1.1 คลาส (Class) คือ กลุ่มของออบเจกต์ที่มีคุณสมบัติ (Attributes) และพฤติกรรม (Behavior) ร่วมกันรายละเอียดของสัญลักษณ์คลาส ชื่อของคลาสจะขึ้นต้นด้วยตัวใหญ่แบบหนา และเอียง หากเป็น Abstract Class แอททริบิวต์ประกอบด้วยชนิดของการเข้าถึง (Visibility) ของแอตทริบิวต์ ได้แก่ Public ซึ่งถูกแสดงด้วยเครื่องหมาย (+) Private ซึ่งถูกแสดงด้วยเครื่องหมายลบ (-) และโปรเทกต์แสดงด้วยเครื่องหมาย (#) ชื่อของแอตทริบิวต์ ประเภทของแอตทริบิวต์ ซึ่งจะอยู่ต่อจากเครื่องหมายโคลอน (:). โดยอาจเป็น Primitive Data Type ของแต่ละภาษาโปรแกรมมิ่งซึ่งมักจะคล้ายคลึงกัน เช่น Integer, Boolean, Real เป็นต้น ค่าเริ่มต้นของแอตทริบิวต์ คือ Public จะถูกแสดงด้วยเครื่องหมายเท่ากับ

5.1.2 โอเปอเรชันมีชนิดและสัญลักษณ์การเข้าถึงเช่นเดียวกับแอตทริบิวต์ มีชื่อโอเปอเรชัน พารามิเตอร์ (Parameters) ประเภทของค่าที่ส่งคืน (Return Type)

5.1.3 ความสัมพันธ์ระหว่างคลาส (Relationships) สามารถแบ่งออกได้เป็นความสัมพันธ์แบบพึ่งพิง (Dependent) การเปลี่ยนแปลงที่เกิดขึ้นกับคลาสที่ถูกพึ่งพิง (Independent Class) จะส่งผลต่อคลาสที่พึ่งพิง (Dependent Class) การโมเดลความสัมพันธ์แบบนี้สามารถทำได้โดยวาดเส้นตรงแบบมีหัวลูกศรเป็นเส้นไปร่งชี้จากซับคลาสที่พึ่งพิงไปยัง

คลาสที่ถูกพึ่งพิงความสัมพันธ์แบบทั่วไป (Generalization) คือ ความสัมพันธ์ระหว่าง Super Class และ Sub Class การโมเดลความสัมพันธ์แบบนี้วาดเส้นตรงหัวทึบที่มีหัวลูกศรเป็นรูปสามเหลี่ยมโปร่งชี้จาก คลาสไปยัง Super Class ความสัมพันธ์แบบมีความสัมพันธ์กัน (Association) สามารถแบ่งได้เป็น

ความสัมพันธ์แบบปกติ (Normal Association) มักใช้ในระบบโมเดลที่ซับซ้อน โดยเฉพาะระบบสารสนเทศ ปกติจะเป็นความสัมพันธ์แบบสองทิศทาง จะวาดด้วยเส้นตรงทึบเชื่อมระหว่างสองคลาสและมีชื่อความสัมพันธ์กำกับอยู่ โดยชื่อนี้มักเป็นคำกริยาเป็นส่วนใหญ่ นอกจากนี้ยังมีการกำหนดปริมาณของคลาสหรือออบเจกต์ที่สัมพันธ์กันอยู่ เรียกว่า Multiplicity

1 หมายถึง จะมีออบเจกต์ในคลาสใดอะแกรมได้หนึ่งออบเจกต์เท่านั้น

0..1 หมายถึง จะมีออบเจกต์ในคลาสใดอะแกรมได้หนึ่งหรืออาจจะไม่มีก็ได้

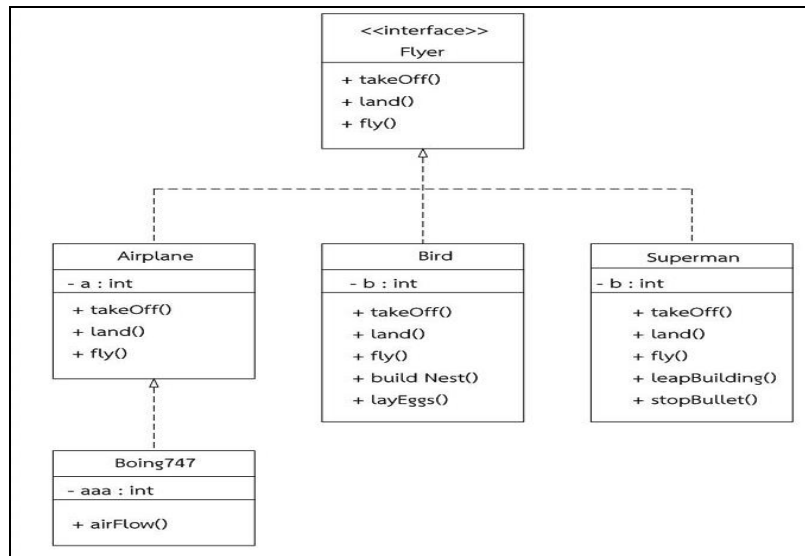
M..N หมายถึง จะมีออบเจกต์ในคลาสใดอะแกรมได้ตั้งแต่ M ถึง N (เมื่อ M, N เป็นจำนวนเต็มบวก)

* หมายถึง จะมีออบเจกต์ในคลาสใดอะแกรมได้ตั้งแต่ศูนย์ขึ้นไป

0..* หมายถึง จะมีออบเจกต์ในคลาสใดอะแกรมได้ตั้งแต่ศูนย์ขึ้นไป

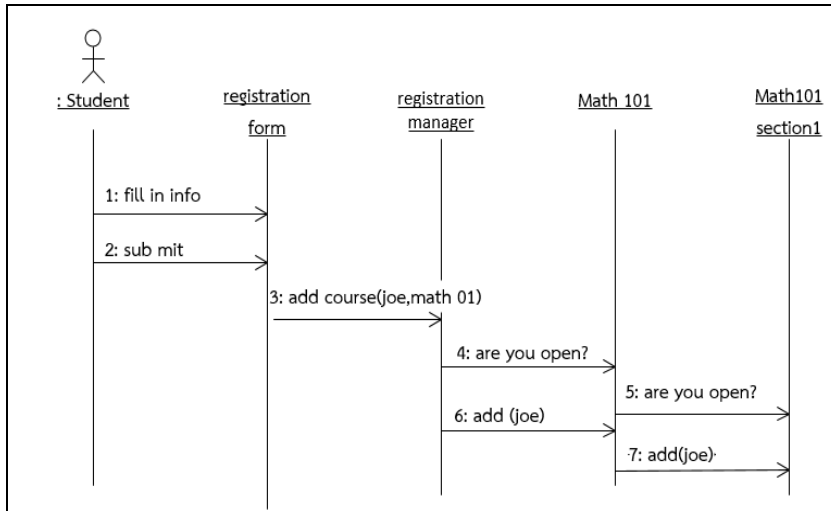
1..* หมายถึง จะมีออบเจกต์ในคลาสใดอะแกรมได้ตั้งแต่หนึ่งขึ้นไป

การรวมกัน (Aggregation) เป็นความสัมพันธ์ระหว่างคลาสหรือออบเจกต์ในแง่ของการรวมกันแสดงด้วยเส้นทึบโยงระหว่างคลาสโดยมีสัญลักษณ์หัวแหลมตัดติดอยู่ระหว่างปลายเส้น ความสัมพันธ์กับคลาสที่หมายถึงสิ่งที่ใหญ่กว่า และส่วนประกอบ (Composition) คล้ายคลึงกับความสัมพันธ์แบบ Normal Aggregation แต่คลาสที่เป็นองค์ประกอบจะเป็นส่วนหนึ่งของคลาสที่ใหญ่กว่าและเมื่อคลาสที่ใหญ่กว่าถูกทำลายคลาสที่เป็นองค์ประกอบจะถูกทำลายด้วยเส้นที่ใช้แสดงการส่งข้อมูลมีอยู่ 4 ชนิด ได้แก่ เส้นทั่วไป เป็นเส้นที่ใช้ส่งเมสเสจแบบทั่วไปไม่เฉพาะเจาะจงจะถูกแสดงเป็นหัวลูกศรธรรมดา คำอธิบายประกอบเป็นคำอธิบายทั่วไป เส้นชิงโครนัส เป็นเส้นที่ส่งข้อมูลไปแล้วจำเป็นต้องรอผลการตอบกลับเหมาะสำหรับงานแบบเรียลไทม์ (Real Time) ที่หลาย ๆ งานอย่างน้อยต้องทำพร้อมกันลักษณะเป็นหัวเส้นตรงโปร่งครึ่งซีก และเส้นตรงส่งกลับจากการเรียกใช้ฟังก์ชัน ลักษณะเป็นเส้นตรงประหัวลูกศรหัวโปร่งชี้จากขวามาซ้ายเป็นการ Return From Method Call มักใช้คู่กับเส้นที่ 1 เมื่อเมธอดที่ถูกเรียกใช้มีค่าบางอย่างที่ต้องการส่งกลับมาตัวอย่างการเขียนคลาสไดอะแกรม ดังแผนภาพที่ 6 แสดงถึงกลุ่มของคลาสการบินที่มีฟังก์ชันบินได้ลงจอด และขึ้นสู่อากาศสามารถแยกย่อยออกเป็นได้ 3 แบบ ได้แก่ เครื่องบิน นัก ยอดมนุษย์ ซึ่งแต่ละคลาสมีความสามารถที่แตกต่างกันโดยยังคงคุณสมบัติของคลาสการบินอยู่ จากภาพจะเห็นคลาสเครื่องบินสามารถแยกออกมาเป็น เครื่องบินโบอิง (Boeing 747) มีความสามารถพิเศษในการใช้เทคโนโลยีไอพ่น เป็นต้น



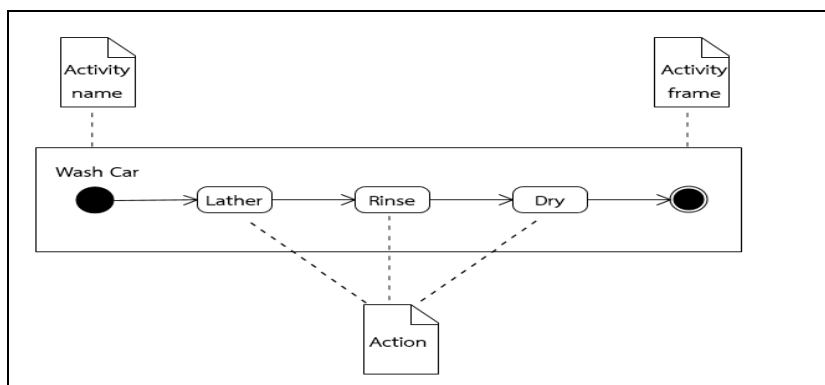
แผนภาพที่ 6 ตัวอย่างการเขียนคลาสไดอะแกรม
ที่มา : ชีรพล ตำนววิริยะกุล. 2549 : 28

6.1 ซีควเอนซ์ไดอะแกรม (Sequence Diagram) จะบอกว่าในยูสเคส นั้นวัตถุแต่ละตัวจะติดต่อสื่อสารกันอย่างไร มีขั้นตอนการทำงานอย่างไร โดยจะเน้นไปที่แกน เวลาเป็นสำคัญถ้าเวลาเปลี่ยนขั้นตอนการทำงานจะเปลี่ยนโดยมีแอกเตอร์เป็นผู้เริ่มกระทำเริ่มต้น ซีควเอนซ์ไดอะแกรมใน UML จะมีแกนสมมุติ 2 แกนคือแกนตั้ง และแกนนอน แกนนอนจะแสดง ขั้นตอนการทำงานหรือการส่งเมสเสจระหว่างวัตถุ โดยแต่ละวัตถุจะส่งข้อมูลถึงกันว่าต้องทำอะไร เมื่อใดส่วนแกนตั้งเป็นแกนเวลา แกนนอนและแกนตั้งต้องสัมพันธ์กันส่วนวัตถุหรือคลาสแทนด้วย รูปสี่เหลี่ยมเรียงกันตามแนวนอน ภายในบรรจุชื่อออบเจกต์ตามด้วยเครื่องหมาย โคลอน และ ชื่อคลาส เส้นประที่อยู่ใต้วงเวลาซึ่งแสดงถึงชีวิตวัตถุ สี่เหลี่ยมแนวตั้งที่อยู่ในตำแหน่ง เดียวกับวัตถุหรือคลาสเรียกว่า Activation ซึ่งใช้แสดงช่วงเวลาที่วัตถุกำลังปฏิบัติงานและส่ง ข้อมูลระหว่างวัตถุรวมถึงแสดงการสิ้นสุดลงของออบเจกต์หรือการถูกทำลายด้วยเครื่องหมาย กากบาทไว้ที่ปลายเส้นชีวิตของออบเจกต์ ตัวอย่าง การเขียนซีควเอนซ์ไดอะแกรม สามารถแสดงได้ ดังแผนภาพที่ 7 อธิบายได้ว่านักเรียนสามารถกรอกข้อมูลลงทะเบียนผ่านฟอร์มเมื่อลงทะเบียน เสร็จระบบจะทำการเพิ่มชื่อพร้อมกับรายวิชาที่ลงทะเบียนให้นายทะเบียนตรวจสอบว่ารายวิชา นั้นเปิดสอนหรือไม่ถ้าเปิดสอนทำการเพิ่มชื่อนักเรียนเข้าเรียน และทำการตรวจสอบช่วงเวลา เรียนที่ว่างถ้าว่างอยู่ระบบจะทำการเพิ่มชื่อ



แผนภาพที่ 7 ตัวอย่างการเขียนซีควเอนซ์ไดอะแกรม
 ที่มา : อีรพล ด้านวิริยะกุล. 2549 : 29

7.1 แอคทิวิตี้ไดอะแกรม (Activity Diagram) แสดงลำดับการไหลของกิจกรรมต่าง ๆ โดยจะอธิบายกิจกรรมในลักษณะของการกระทำจะมีเงื่อนไขและการตัดสินใจ กำหนดไว้เพื่อควบคุมการไหลของกิจกรรมรวมถึงแมสเชสที่รับส่งระหว่างแต่ละกิจกรรมแสดงด้วยสี่เหลี่ยมเหมือนแคปซูลเชื่อมโยงกันด้วยลูกศรเพื่อแสดงลำดับการทำแอคทิวิตี้ (Activity) ถัดไปได้ โดยจะมีเส้นลูกศรชี้เข้ามารวมที่จุดเดียว (เส้นตรงแนวนอน) นั่นคือ แอคทิวิตี้ที่ชี้เข้ามาที่เส้นที่บดงกล่าว เสร็จแล้วก่อน จึงทำให้แอคทิวิตี้ถัดไปได้ การแบ่งเป็นสวิมเลนส์ (Swimlanes) เหมือนสระว่ายน้ำโดยแบ่งช่องในแนวดิ่งและกำหนดแต่ละช่องด้วยชื่อของออบเจกต์ไว้แถวบนสุด ตัวอย่างการเขียนแอคทิวิตี้ไดอะแกรม ดังแผนภาพที่ 8 แสดงตัวอย่างการเขียนแอคทิวิตี้ไดอะแกรมของการล้างรถเริ่มจากล้างด้วยแชมพู ทำการล้างแชมพู เป่าลมให้แห้ง



แผนภาพที่ 8 ตัวอย่างการเขียนแอคทิวิตี้ไดอะแกรม
 ที่มา : อีรพล ด้านวิริยะกุล. 2549 : 30

ซอฟต์แวร์ที่ใช้ในการพัฒนา

1. โปรแกรม Android Studio

Android Studio เป็นเครื่องมือที่สนับสนุนสภาพแวดล้อมอย่างพร้อมสรรพสำหรับใช้ในการพัฒนาซอฟต์แวร์ โดยเฉพาะสำหรับภาษาจาวาและเนื่องจาก เป็นโอเพนซอร์ซที่พัฒนาขึ้นเพื่อใช้โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนา เป็นไปอย่างต่อเนื่องและรวดเร็วสัญลักษณ์ไอคอน มีองค์ประกอบหลักที่เรียกว่า Android Platform ซึ่งทำให้บริการพื้นฐานหลักสำหรับรวบรวมเครื่องมือต่างๆ จากภายนอกให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกันและมีองค์ประกอบที่เรียกว่า Plug-in Development Environment (PDE) ซึ่งใช้ในการเพิ่มความสามารถในการพัฒนาซอฟต์แวร์มากขึ้น เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียกว่า Studio Plug-in ดังนั้นหากต้องการให้ Android ทำงานได้เพิ่มเติมก็เพียงแค่พัฒนาปลั๊กอินสำหรับงานนั้นขึ้นมาและนำปลั๊กอินนั้นมาติดตั้งเพิ่มเติมให้กับ Android ที่มีอยู่เท่านั้น

2. ฐานข้อมูล Mysql

MySQL จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS: Relational Database Management System) ตัวหนึ่ง ซึ่งเป็นที่นิยมกันมากในปัจจุบัน โดยเฉพาะอย่างยิ่งในโลกของอินเทอร์เน็ต สาเหตุเพราะว่า MySQL เป็นฟรีแวร์ทางด้านฐานข้อมูลที่มีประสิทธิภาพสูง เป็นทางเลือกใหม่จากผลิตภัณฑ์ระบบจัดการฐานข้อมูลในปัจจุบัน ที่มักจะเป็นการผูกขาดของผลิตภัณฑ์เพียงไม่กี่ตัว นักพัฒนาระบบ

ฐานข้อมูลที่เคยใช้ MySQL ในความสามารถความเร็ว การรองรับจำนวนผู้ใช้ และขนาดของข้อมูลจำนวนมหาศาล ทั้งยังสนับสนุนการใช้งานบนระบบปฏิบัติการมากมาย ไม่ว่าจะเป็น Unix, OS/2, Mac OS หรือ Windows ก็ตาม นอกจากนี้ MySQL ยังสามารถใช้งานร่วมกับ Web Development Platform ทั้งหลาย ไม่ว่าจะเป็น C, C++, Java, Perl, PHP, Python, Tel หรือ ASP ก็ตามที่ตั้งใจไม่เป็นที่น่าแปลกใจเลยว่า ทำไม MySQL จึงได้รับความนิยมอย่างมากในปัจจุบันและมีแนวโน้มสูงยิ่งขึ้นๆ ไปในอนาคต MySQL จัดเป็นซอฟต์แวร์ประเภท Open Source Software สามารถดาวน์โหลด Source Code ต้นฉบับได้จากอินเทอร์เน็ตโดยไม่เสียค่าใช้จ่ายใดๆ การแก้ไขก็สามารถกระทำได้ตามความต้องการ

MySQL ได้รับการยอมรับและทดสอบเรื่องของความรวดเร็วในการใช้งาน โดยจะมีการทดสอบและเปรียบเทียบกับผลิตภัณฑ์ทางด้านฐานข้อมูลอื่นอยู่เสมอ มีการพัฒนาอย่างต่อเนื่อง โดยเริ่มตั้งแต่เวอร์ชันแรกๆ ที่ยังไม่ค่อยมีความสามารถมากนัก มาจนถึงทุกวันนี้ MySQL ได้รับการพัฒนาให้มีความสามารถมากยิ่งขึ้น รองรับข้อมูลจำนวนมหาศาล สามารถใช้งานหลายผู้ใช้ได้พร้อมๆ กัน (Multi-user) มีการออกแบบให้สามารถแต่งงานออกเพื่อช่วยการทำงานให้เร็วยิ่งขึ้น รองรับข้อมูลจำนวนมหาศาลเพื่อช่วยการทำงานเร็วยิ่งขึ้น (Multi-threaded) วิธีและการเชื่อมต่อที่ดีขึ้น การกำหนดสิทธิและการรักษาความปลอดภัยของข้อมูลมีความรัดกุม

น่าเชื่อถือยิ่งขึ้นเครื่องมือหรือโปรแกรมสนับสนุนทั้งของตัวเองและของผู้พัฒนาอื่นๆ มีมากยิ่งขึ้น นอกจากนี้สิ่งหนึ่งที่สำคัญคือ “MySQL ได้รับการพัฒนาไปในแนวทางตามข้อกำหนดมาตรฐาน SQL ดังนั้น เราสามารถใช้คำสั่ง SQL ในการทำงาน MySQL” นักพัฒนาที่ใช้ SQL มาตรฐานอยู่แล้วไม่ต้องศึกษาคำสั่งเพิ่มเติม แต่อาจจะต้องเรียนรู้ถึงรูปแบบและข้อจำกัดบางอย่างโดยเฉพาะ ทั้งนี้ทั้งนั้น ทางทีมงานผู้พัฒนา MySQL มีเป้าหมายอย่างชัดเจนที่จะพัฒนาให้ MySQL มีความสามารถสนับสนุนตามข้อกำหนด SQL92 มากที่สุดและจะพัฒนาให้เป็นไปตามข้อกำหนด SQL99 ต่อไป

2.1 ลักษณะเด่นของMySQL

2.1.1 MySQLเป็นระบบจัดการฐานข้อมูล MySQL เป็นโปรแกรมสำหรับใช้ในการสร้างฐานข้อมูล โดยมีคุณลักษณะของระบบจัดการฐานข้อมูลที่ไม่ยิ่งหย่อนไปกว่าระบบจัดการฐานข้อมูลชนิดอื่น ๆ รวมทั้งสามารถสร้างและจัดการฐานข้อมูลขนาดใหญ่ได้อย่างรวดเร็วอีกด้วย โดยที่ MySQL มีระบบสืบค้นข้อมูลที่รวดเร็วและแม่นยำ สามารถใช้งานได้กับคอมพิวเตอร์ระบบ Stand-Alone และ Network รวมทั้งทำงานร่วมกับ Application

2.1.2 MySQLเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ MySQL มีความสามารถในการเชื่อมโยงข้อมูลระหว่างตาราง จัดเก็บข้อมูลจำนวนมาก สะดวก และค้นหาง่าย ซึ่งเป็นคุณลักษณะปกติของโปรแกรมที่พัฒนาขึ้นมาจาก SQL แต่การสร้างฐานข้อมูลเชิงสัมพันธ์ของ MySQL ให้ทางเลือกในการออกแบบ และพัฒนาฐานข้อมูลใช้มากกว่าโปรแกรมจัดการฐานข้อมูลชนิดอื่น

2.1.3 MySQLเป็นซอฟต์แวร์แบบฟรีแวร์ และเป็น Open Source ผู้ใช้ MySQL สามารถพัฒนาโปรแกรมต่อเองได้อย่างอิสระ และทุกคนมีสิทธิ์ที่จะ Download ระบบจัดการฐานข้อมูลผ่านทาง Internet หรือทำสำเนาได้ แต่โปรแกรม MySQL ที่การจดลิขสิทธิ์ ดังนั้นสิทธิ์ทางประการ เช่น การจัดจำหน่ายซอฟต์แวร์ ซึ่งพัฒนามาจาก MySQL หรือการจำหน่ายซอฟต์แวร์เสริมทางทำงานของ MySQL จะถูกสงวนไว้โดยบริษัทผู้ผลิต

3. โปรแกรม NetBeans

NetBeans คือ เครื่องมือสำหรับโปรแกรมเมอร์ที่จะใช้พัฒนา Application ด้วยภาษา Java NetBeans นั้นเป็นโปรแกรมประเภท OpenSource software โดยผู้ใช้งานไม่จำเป็นต้องเสียเงิน เพื่อซื้อมาใช้งาน และยังเปิดเผย Source code ให้ผู้สนใจและนักพัฒนานำไปดัดแปลง แก้ไข ตามกฎของ Opensource โดยมี Sun Micro System เป็นผู้สนับสนุน โครงการปัจจุบัน NetBeans ได้รับความนิยมมากยิ่งขึ้น และได้รับการพัฒนาให้มีความสามารถสูงยิ่งขึ้นเรื่อยๆ จนถึงเวอร์ชันล่าสุด คือ นอกจากจะใช้ในการพัฒนา Application ด้วยภาษา Java แล้ว ยังสามารถพัฒนาอื่นๆได้อีกหลากหลายโดยติดตั้งโปรแกรมเสริม(Add-on)ได้จาก เว็บไซต์หรือผ่านตัวอัปเดตเซนเตอร์ (Update Center) ของ NetBeans เช่น ภาษาซี/ซีพลัสพลัส (C/C++), Ruby, UML, SOA, Web Application, Java EE, Mobility(Java ME), Java FX, Java Script, PHP เป็นต้น ในเวอร์ชัน 6.0 เป็นต้นไปมีการรวมโปรแกรมเสริมต่างๆที่สำคัญเข้า

ในตัวติดตั้งของ NetBeans โดยสามารถเลือกติดตั้งได้ภายหลังข้อดีของโปรแกรมนี้ก็คือ โปรแกรม NetBeans นั้นทำงานแยกส่วนต่างๆ ออกจากกันเป็น Module จึงทำให้สามารถนำ Module ต่างๆที่มีผู้ที่ได้พัฒนาต่อเติมมาติดตั้งเพิ่มเติมในภายหลังได้ ใช้งานได้กระบบปฏิบัติการ Windows , Linux, Mac OS X and Solaris

4. ภาษา JAVA

Java หรือ Java programming language คือภาษาโปรแกรมเชิงวัตถุ พัฒนาโดย เจมส์ กอสลิง และวิศวกรคนอื่นๆ ที่บริษัท ซันไมโครซิสเต็มส์ ภาษานี้มีจุดประสงค์เพื่อใช้แทน ภาษาซีพลัสพลัส C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษาอ็อบเจกต์ทีฟซี (Objective-C) แต่เติมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊กใกล้ที่ทำงานของ เจมส์ กอสลิง แล้ว ภายหลังจึงเปลี่ยนไปใช้ชื่อ “จาวา” ซึ่งเป็นชื่อกาแฟแทน จุดเด่นของภาษา Java อยู่ที่ผู้เขียน โปรแกรมสามารถใช้หลักการของ Object-Oriented Programming มาพัฒนาโปรแกรมของตน ด้วย Java ได้

ภาษา Java เป็นภาษาสำหรับเขียนโปรแกรมที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุ (OOP : Object-Oriented Programming) โปรแกรมที่เขียนขึ้นถูกสร้างภายในคลาส ดังนั้น คลาสคือที่เก็บเมทอด (Method) หรือพฤติกรรม (Behavior) ซึ่งมีสถานะ (State) และรูปพรรณ (Identity) ประจำพฤติกรรม (Behavior)

4.1 คุณลักษณะเด่นของภาษา Java

4.1.1 ภาษา Java เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบ สมบูรณ์

4.1.2 โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรม บนเครื่อง Sun โปรแกรมนั้นก็สามารถูก compile และ run บนเครื่องพีซีได้

4.1.3 เมื่อเปรียบเทียบ code ของโปรแกรมที่เขียนขึ้นโดยภาษา Java กับ C++ พบว่า โปรแกรมที่เขียนโดยภาษา Java จะมีจำนวน code น้อยกว่าโปรแกรมที่เขียน โดยภาษา C++ ถึง 4 เท่า และใช้เวลาในการเขียนโปรแกรม น้อยกว่าประมาณ 2 เท่า

4.1.4 Java มี security ทั้ง low level และ high level ได้แก่ electronic signature, public and private key management, access control และ certificatesของภาษาจาวา

4.2 ข้อดีของ ภาษา Java

4.2.1 ภาษา Java เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบ สมบูรณ์ ซึ่งเหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน การพัฒนาโปรแกรมแบบวัตถุจะช่วยให้ เราสามารถใช้คำหรือชื่อ ต่าง ๆ ที่มีอยู่ในระบบงานนั้นมาใช้ในการออกแบบโปรแกรมได้ ทำให้ เข้าใจได้ง่ายขึ้น

4.2.2 โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่ต่างกันได้ ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่อง Sun โปรแกรมนั้นก็สามารรถถูก compile และ run บนเครื่องพีซีธรรมดาได้

4.2.3 ภาษาจาวามีการตรวจสอบข้อผิดพลาดทั้งตอน compile time และ runtime ทำให้ลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม และช่วยให้ debug โปรแกรมได้ง่าย

4.2.4 ภาษาจาวามีความซับซ้อนน้อยกว่าภาษา C++ เมื่อเปรียบเทียบ code ของโปรแกรมที่เขียนขึ้นโดยภาษา Java กับ C++ พบว่า โปรแกรมที่เขียนโดยภาษา Java จะมีจำนวน code น้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ทำให้ใช้งานได้ง่ายกว่าและลดความผิดพลาดได้มากขึ้น

4.2.5 ภาษาจาวาถูกออกแบบมาให้มีความปลอดภัยสูงตั้งแต่แรก ทำให้โปรแกรมที่เขียนขึ้นด้วยจาวามีความปลอดภัยมากกว่าโปรแกรมที่เขียนขึ้น ด้วยภาษาอื่น เพราะ Java มี security ทั้ง low level และ high level ได้แก่ electronic signature, public and private key management, access control และ certificates ของ

4.2.6 มี IDE, Application server, และ Library ต่าง ๆ มากมาย สำหรับจาวาที่เราสามารถใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย ทำให้เราสามารถลดค่าใช้จ่ายที่ต้องเสียไปกับการซื้อ tool และ s/w ต่าง ๆ

5. JSON

JSON ย่อมาจาก JavaScript Object Notation เป็นโครงสร้างในการเก็บข้อมูลชนิดหนึ่ง นอกเหนือจาก XML ด้วยเหตุที่ว่ามีความรวดเร็ว และเข้าใจง่ายกว่า XML เป็นอย่างมาก ส่วนใหญ่นำมาใช้ในการทำ AJAX เพื่อทำให้สามารถรับส่งค่า หรือสั่งการเข้า Server พร้อมกับชุดข้อมูลขนาดใหญ่ได้ ภาษาโปรแกรมมิ่งที่ปัจจุบันยังนิยมอยู่เช่น Java และ PHP (พอดีว่ายังไม่ได้จับภาษาอื่นจริงๆเลยจะเน้น สอง ภาษานี้เป็นหลัก แต่ส่วนใหญ่คงเป็น PHP เพราะมีความเข้าใจง่ายกว่า และคนรู้จัก และเข้าใจเยอะ) มีตัวช่วยให้ใช้หลากหลาย เช่น ของ Java มี GSON ที่ทำให้ JSON กลายเป็นเรื่องง่ายไปเลย หรือของ PHP ก็มีตัว encode_json เป็นของตัวเองอยู่แล้ว JSON ยังมีชนิดไฟล์เป็นของตัวเอง (เหมือน XML) คือ .json

JSON มันดีกว่า XML ยังไง

- 1) JSON เป็นอะไรที่เข้าใจง่ายๆ คนที่ไม่รู้ก็ยังไม่พอเข้าใจได้ ไม่เหมือน XML ที่มันดูซับซ้อน
- 2) JSON ไม่จำเป็นต้องใช้ tag เปิดปิดให้ยุ่งยาก
- 3) JSON มีการเก็บข้อมูลแบบลำดับชั้น (Hierarchical)
- 4) JSON สามารถส่งค่าผ่านทาง JavaScript ได้
- 5) JSON สั้น กระชับกว่า และสามารถถูกอ่านหรือเขียนได้เร็วกว่า
- 6) JSON ไม่มีจำกัดคำเฉพาะ (เช่นพวก public, private, etc.)

7) JSON สามารถเก็บค่าแบบ Array ได้

Syntax

JSON มีลักษณะการเก็บคล้ายๆ Array ที่มี key คือ ใช้ String เป็นชื่อตัวแปร และขึ้นด้วย ":" เพื่อใส่ ข้อมูล สำหรับการแยกค่าจะใช้ "," ในการแบ่งออกไป ส่วน "{}"เป็นการแสดงถึง object ในแต่ละชั้น

`{"id":1,"name":"Cail","lastname":"Potato"},{"id":2,"color":"green","fruit":"Tomato"}` ซึ่งจะเห็นว่า ถ้ามี double quotes หรือจะใช้ single quote ก็ได้ (แต่ควรใช้ double quotes)ครอบจะเป็น String อยู่แล้ว แต่ถ้าหากว่าเป็น พวก Integer หรือ ตัวเลข จะไม่จำเป็นต้องมี

6. ภาษา XML (Extensible Markup Language)

ศุภชัย สมพานิช ได้กล่าวถึงเหตุที่ก่อให้เกิดภาษา XML ขึ้นมาว่า “การใช้โปรแกรมภาษาหลายตัวที่แตกต่างกัน แต่ละภาษาล้วนมีข้อดีข้อเสียที่แตกต่างกันไป จะทำให้เกิดการแลกเปลี่ยนข้อมูลร่วมกันได้อย่างไร” ได้เกิดความตื่นตัวจากวงการคอมพิวเตอร์เป็นอย่างมาก เมื่อ Microsoft ได้ทำการประกาศ การใช้เทคโนโลยี .NET ซึ่งมี XML เป็นกลยุทธ์ในการแลกเปลี่ยนข้อมูลกัน เพราะแต่เดิมหากเราจะแลกเปลี่ยนข้อมูลให้กันได้ จะต้องเป็นการเขียนโปรแกรมที่มาจากตระกูลภาษาเดียวกันเท่านั้น อีกทั้งภาษาของ Sun จำพวกตระกูล Java ก็ได้สร้างสรรค์แนวคิดของการ Programming แบบ non Plat form อยู่แล้ว เรียกได้ว่าเป็นการวิ่งไล่ตาม Sun ของ Microsoft แต่แค่ XML ยังไม่พอที่จะทำให้ Microsoft ไล่ตาม Sun ในเรื่องของ Vendor Technology ได้ Microsoft จึงได้นำเสนอแนวคิดของ Web Service โดยพยายามปิดบังจุดอ่อนตัวเองที่มีอยู่แล้วในเรื่องโปรแกรมภาษาของ Microsoft ที่ต้องอิงระบบปฏิบัติการ Windows เท่านั้น แนวคิดของ “ความเกี่ยวพันระหว่าง Software Service และ Solution” จึงเกิดขึ้น โดยนิยามความคิดของผู้ใช้คอมพิวเตอร์ใหม่ให้ตัดความสนใจจากคำว่า Software ให้เป็น Service เสีย เพราะตราบใดที่เรามีความต้องการ Software เราก็ยังคงติดอยู่กับเรื่องของระบบปฏิบัติการอย่างไม่รู้จักจบสิ้น โดยให้เปลี่ยนจากคำว่า Software ให้เป็น Service และให้ Web Site ต่างๆ เป็นผู้ให้ Service ต่างๆ เมื่อคิดอย่างนี้ระบบปฏิบัติการก็จะเปลี่ยนจาก Window UNIX LINUX SOLALIS ฯลฯ มาเป็นระบบปฏิบัติการ Internet จะได้ไม่มีปัญหาในเรื่องของระบบปฏิบัติการอีกต่อไป โดยสามารถที่จะเข้าไปใช้บริการใน 2 สถานะ นั่นคือ ผู้ให้บริการ หรือผู้ใช้บริการ นั่นเอง ถึงอย่างไรก็ตามหากพูดถึงความจำเป็นในเรื่องการแลกเปลี่ยนข้อมูลแบบ Non Platform บริษัท Microsoft ก็ยังคงต้องใช้ XML เป็นหลัก และในตอนนี้ XML ก็ยังไม่ได้ถูกตอบรับมาตรฐานจาก W3C อย่างเป็นทางการ เพราะ W3C อ้างว่าหากเราจะใช้ XML อย่างมีมาตรฐานต้องมีคุณสมบัติอีกหลายข้อ ทางด้าน Sun เอง ก็ได้ตื่นตัวกับเรื่องนี้โดยในส่วนของ Sun จะใช้ XML ในความหมายที่แตกต่างออกไป โดยใช้ XML ในบทบาทของ Descriptor Language คือ ภาษาที่ทำหน้าที่บรรยายโครงสร้างของ Program ที่สร้างขึ้นใหม่ และ Sun ก็ยังตอบสนองในเรื่อง XML ด้วยการพัฒนาชุดพัฒนา JWSDP (Java Web Service Develop Pack) เพื่อสนองความต้องการของ Web Service เช่นเดียวกับ Microsoft

แต่ Sun จะมีข้อได้เปรียบในเรื่องของการใช้ API ที่มีอยู่แล้ว ในการเพิ่มศักยภาพของ Web Service เพราะแต่เดิมภาพแบบพื้นฐานของ Sun คือการสร้างโปรแกรมที่สามารถทำงานได้ในต่าง Platform อยู่แล้ว สำหรับบริษัทอื่น ๆ อย่าง เช่น Apache ก็ได้ตอบสนอง Solution ทางด้าน Web Service เช่นเดียวกันโดยการคลอด SOAP-AXIS ซึ่งเป็นชุดพัฒนา Web Service ของ Apache เอง และทั้งหมดทั้งมวลนี้ก็เป็ประโยชน์ของ XML ที่ทำให้ทุกๆ เทคโนโลยีคุยกัน ได้เป้าหมายที่สำคัญของ XML จึงเน้นที่จะนำไปใช้งานในอินเทอร์เน็ต เป้าหมายหลักมีดังนี้

1. XML เป็นงานที่จะนำไปประยุกต์บนอินเทอร์เน็ต โดยใช้เอกสาร XML ได้ง่าย สะดวก และได้ผลดีเหมือน HTML
2. XML ออกแบบอย่างพิถีพิถันเน้นความจำเป็น กะทัดรัด เข้าใจง่าย และได้ประโยชน์กว้างขวาง
3. XML สนับสนุนประยุกต์เข้ากับงานต่างๆ และสนับสนุนโปรแกรมประยุกต์ต่างๆ
4. XML เน้นเรื่องการประมวลผลเอกสาร จึงเหมาะกับการงานทางด้านการวิเคราะห์เอกสาร การผลิตเอกสาร การแลกเปลี่ยนและการแสดงผล
5. การเขียนด้วยภาษา XML ทำได้ง่าย
6. คุณสมบัติของ XML ต้องอยู่ในระดับต่ำสุด เพื่อให้ผู้ใช้อื่นร่วมใช้ได้
7. XML ควรอ่านได้ด้วยมนุษย์ โดยไม่ต้องอาศัยโปรแกรมหรือเครื่องมือช่วยแปล เพราะบาง ครั้งควรอ่านและเข้าใจได้ด้วย Text ธรรมดา
8. การเขียน XML ทำได้ตั้งแต่การใช้ Text editor ทั่วๆ ไปและไม่ต้องการเครื่องมือที่ซับซ้อน - ซ้อน อย่างไรก็ตามต้องมีผู้เขียน XML editor ให้ใช้งานได้ง่ายขึ้น
9. XML เป็นมาตรฐานที่กำหนดแล้วใช้งานได้ทันที โดยที่ Browser และอุปกรณ์ต่าง ๆ พร้อมใช้งานร่วมกัน
10. ภาพแบบการเขียนโครงสร้างข้อกำหนดของ XML ต้อง เป็นไปตามหลักการของวิศวกรรมคอมพิวเตอร์ คือ เมื่อเขียนแล้วต้องสามารถใช้โปรแกรมแปลภาษาได้ง่าย โดยทั่วไปเขียนในภาพแบบ BNF ได้ (Bach's Normal Form)

เป้าหมายที่สำคัญของ XML อีกอย่างหนึ่ง คือ ใช้เป็นตัวควบคุมข้อมูล (Meta data) ดังนั้นจึงเป็นแนวทางในการขนส่งข้อมูล และสร้างการเชื่อมโยงระหว่าง Application ได้ง่าย ด้วยเหตุผลของการใช้งานบนเครือข่ายที่มีพัฒนาการการจัดการเอกสารจำนวนมาก การสร้าง Digital Library การแลกเปลี่ยนข้อมูลข่าวสารระหว่างกัน การประยุกต์ XML จึงทำได้กว้างขวาง เช่น XML สนับสนุน UNICODE ทำให้ใช้ได้หลากหลายภาษา และผสมกันได้หลากหลายภาษา การพัฒนา XML Processor ทำให้สามารถดึงเอกสาร XML มาใช้งานได้ง่าย และใช้ร่วมกับโปรแกรมประยุกต์อื่นได้ง่าย เช่น โปรแกรม DB2, Oracle, SAP เป็นต้น XML ช่วยทำให้เกิดการรับส่งข้อมูลแบบ EDI โดยทำให้แนวทางการเชื่อมโยงและสร้างความเป็นเอกสารหรือมาตรฐานระหว่างองค์กร XML มีสภาพช่วยในการขนส่งข้อมูลไปยังปลายทางเพื่อให้แปลความหมายและใช้

งานได้อย่างเต็มประสิทธิภาพ มีการสร้างการประยุกต์ และนำเสนอผลลัพธ์ไปใช้งานจาก XML ได้มาก การประยุกต์การดำเนินกิจกรรมบนเครือข่ายมีมาก เช่น eBusiness EDI eCommerce การจัดการ Supply chain, Demand chain management การดำเนินการแบบ intranet และ web base application

เปรียบเทียบความแตกต่างของภาษา HTML กับภาษา XML

ภาษา XML ไม่ใช่ทั้งเวอร์ชันใหม่และส่วนขยายของ HTML รวมทั้งไม่ใช่ภาษา XHTML ด้วย แต่ XML เป็นภาษาที่ถูกออกแบบมาเพื่อช่วยเพิ่มความสามารถให้กับ HTML ภาษา XML มีจุดประสงค์ในการใช้งานแตกต่างกับภาษา HTML โดยที่ HTML ถูกออกแบบมาให้เน้นไปทางด้านการแสดงผลข้อมูลเพียงอย่างเดียว (โดยไม่รู้ว่าจะข้อมูลที่แสดงคืออะไร) แต่สำหรับ XML แล้ว จะแตกต่างกันอย่างสิ้นเชิง เพราะว่า XML สามารถล่วงรู้หรืออธิบายได้ว่าข้อมูลส่วนนี้คืออะไร ส่วนเรื่องการตกแต่งหรือแสดงผลข้อมูลดังกล่าวออกจะมอบภาระให้ภาษาอื่น ๆ ทำหน้าที่ตกแต่งให้สวยงามแทน เช่น ภาษา XSL

Well-Formed XML

David Hunter ได้กล่าวถึงเรื่อง Well-Formed XML หรือ XML ที่ลงตัวที่พร้อมจะนำไปใช้งานได้จริงไว้ว่า “XML ที่ลงตัว คือ XML ที่เป็นไปตามกฎเกณฑ์ของ XML 1.0”

2.1 ภาพแบบของ Tag ใน XML Tag ที่สร้างใน XML จะต้องประกอบด้วย Tag เปิดและ Tag ปิด ซึ่งแตกต่างจาก HTML ที่ในบางครั้งก็ยังมี Tag เดี่ยวอยู่ เช่น XML <name> Arikato </name> แต่ HTML บางครั้งก็มี Tag เดี่ยว เช่น
 ซึ่งใน XML จะกระทำการเช่นนี้ไม่ได้ โดยทุก Tag จะต้องมีจุดเริ่มต้นและจุดสิ้นสุดเสมอ

2.2 Tag ต่างๆ ของ XML คาบเกี่ยวกันไม่ได้ เช่น <name> ศิริพร <surname> </name> คำเชื่อมแก้ว </surname> หากทำเช่นนี้ในภาษา HTML จะสามารถทำได้แต่ในภาษา XML นั้นเข้มงวดมาก

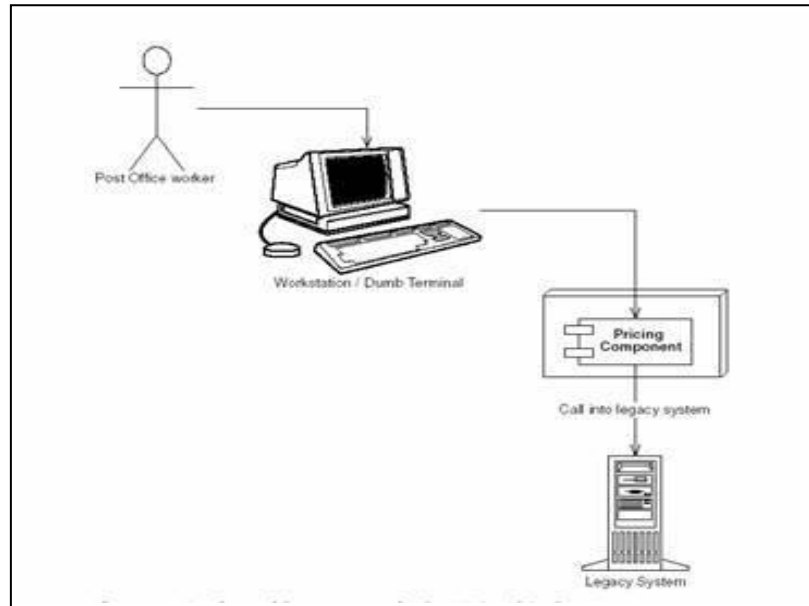
2.3 ในเอกสารหนึ่งต้องมี root Element เพียงหนึ่งเท่านั้นยกตัวอย่าง

```
<name> Siripron </name>
```

```
<name> lang </name>
```

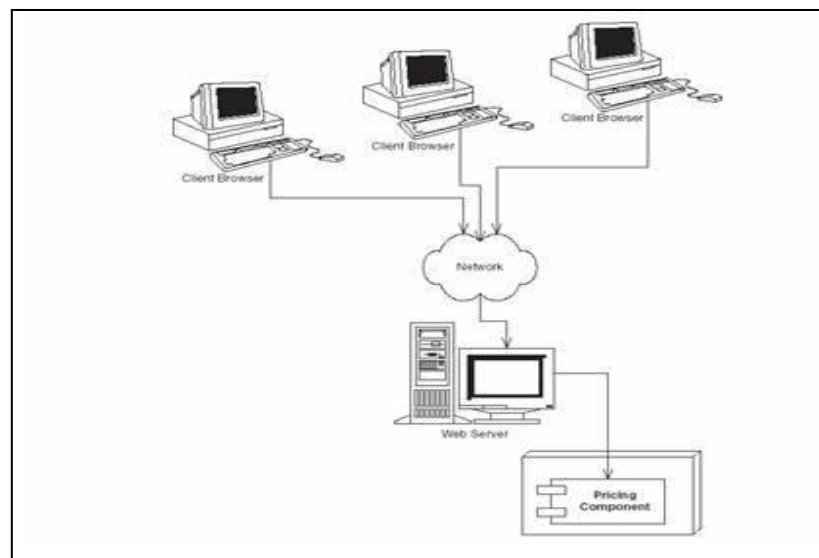
7. Java 2 Enterprise Edition (J2EE)

Software Component เป็นการมอง ซอฟต์แวร์เป็นชุดๆ ส่วนๆ ที่สามารถทำงานบางสิ่งบางอย่างได้ในตัวของมันเอง ซึ่งจะทำให้สามารถนำ Software Component มาประกอบกันเพื่อทำเป็น Application ใช้ในองค์กร เช่น Software Component ด้านการกำหนดราคา Pricing Component (รูปภาพทั้งหมดนำมาจากหนังสือ Mastering EJB Second Edition ของ Ed Roman) ดังภาพที่ 8



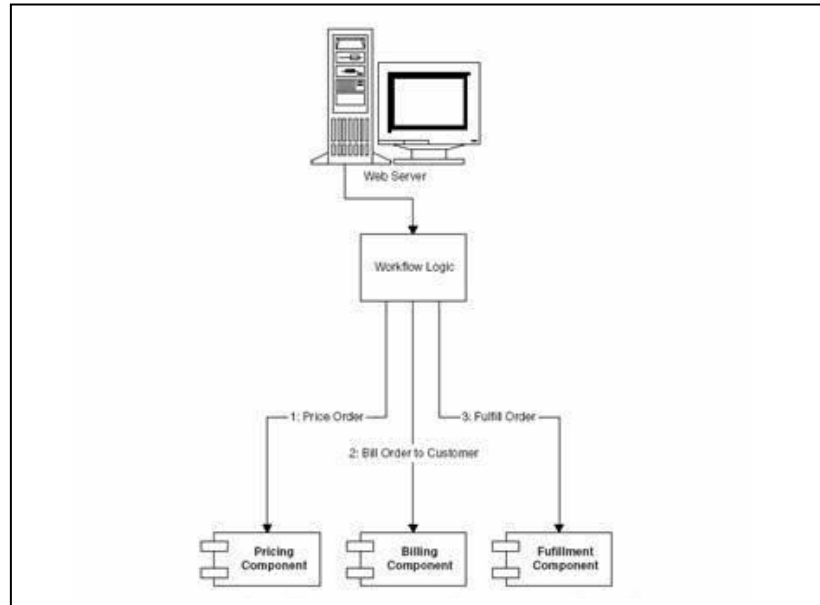
ภาพที่ 8 การใช้งาน Pricing Component ติดต่อกับ Legacy System

จากภาพที่ 8 แสดงให้เห็นถึงการใช้งาน Pricing Component ติดต่อกับ Legacy System ซึ่งเป็นระบบเก่า โดย User ติดต่อผ่าน Dumb Terminal หรือ Work Station



ภาพที่ 9 การทำงานในแบบ Web Application

จากภาพที่ 9 เป็นการแสดงให้เห็นการทำงานในแบบ Web Application ที่ใช้ Pricing Component ในการทำงาน โดยมี Client เป็น Web Browser

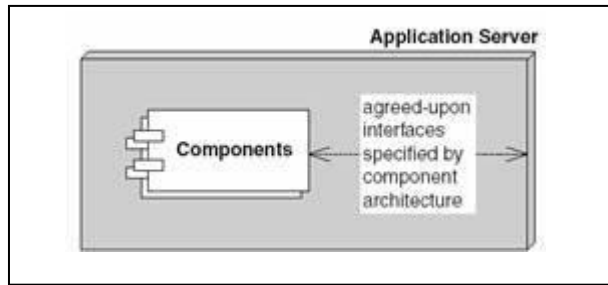


ภาพที่ 10 การพัฒนา Application

จากภาพที่ 10 แสดงให้เห็นการพัฒนา Application ที่นำ Software Component 3 Component มารวมกัน คือ Pricing Component, Billing Component และ Fulfillment Component มาประกอบกัน

จากตัวอย่างข้างต้น จะพบว่า การพัฒนาซอฟต์แวร์เป็น Component มีข้อดี คือ มีโอกาสที่จะนำ Component มาใช้ใหม่ได้สูง (Reusability) ซึ่งจะทำให้ใช้เวลาพัฒนาระบบงานโดยรวมน้อยลง และในตัว Component เอง ก็จะนำ Business Logic ที่ดีที่นิยมใช้ในวงการนั้นๆ มาใช้ (Best Practice) ซึ่งทำให้องค์กรได้รับประโยชน์สูงสุด คือได้ใช้ Business Logic ที่ดี โดยไม่ต้องคิดตั้งแต่เริ่มต้น และในขณะเดียวกัน การพัฒนาเปลี่ยนแปลง Business Logic ให้ดีขึ้นเรื่อยๆ ก็สามารถทำได้ โดยไม่กระทบการใช้งาน ของ Client เลย

เนื่องจาก บริษัทที่ ผลิต Component ก็จะพัฒนา Application Server ขึ้นเองด้วย ปัญหาที่คือผู้ใช้งาน เมื่อซื้อ Software Component มาใช้งานแล้ว ถ้าต้องการเปลี่ยน Application Server ก็ไม่สามารถทำได้ เพราะ Software Component นั้น ไม่สามารถ Deploy และ Run บน Application Server ตัวใหม่ได้ ฉะนั้น ความต้องการ มาตรฐานของ Software Component จึงเกิดขึ้นเพื่อที่จะทำให้ Software Component สามารถที่จะ Deploy และ Run ได้กับทุก Application Server จึงเกิดการสร้าง Component Architecture ขึ้นมา ดังภาพที่ 11



ภาพที่ 11 การกำหนดให้ Software Component

จากภาพที่ 11 เป็นการกำหนดให้ Software Component ให้ปฏิบัติตามข้อตกลงที่อยู่ใน Component Architecture โดยต้องมี Interface ที่กำหนดไว้ ผลก็คือ เมื่อนำ Component ไป Run บน Application Server ที่ยึดถือตาม Component Architecture เดียวกัน ก็จะสามารถ Run ได้อย่างไม่มีปัญหา

EJB ก็คือ Component Architecture หรือข้อตกลง (Agreements) การทำงานของร่วมกันระหว่าง Software Component กับ Application Server (ในที่นี้ก็คือ J2EE Server) โดย EJB หรือ Enterprise Bean จะเป็น Software Component ที่ Run ในฝั่งของ Server (Server - Side) ในลักษณะ Distributed Component โดย EJB จะต้องเขียนด้วย ภาษา Java เท่านั้น

ในลักษณะเช่นนี้ ทางด้าน Microsoft ก็มีเช่นเดียวกัน เรียกว่า .NET Managed Component ซึ่งจะ Run บน Application Server (MTS/COM+) ของ Microsoft เท่านั้น แล้วต้องเขียนด้วยภาษาที่ใช้เทคโนโลยี .NET เช่น VB.NET หรือ C# เป็นต้น และ อีกหนึ่ง Component Architecture เป็นของหน่วยงาน OMG (Object Management Group) ได้จัดทำข้อกำหนดไว้คือ CORBA (Common Object Request Broker Architecture) ในส่วนนี้สามารถใช้ภาษาใดๆ ก็ได้ ที่ Support มาตรฐาน CORBA ในตรงจุดนี้ EJB Servers ส่วนใหญ่ก็สามารถใช้งานหรือติดต่อกับ Software Component ของ CORBA ได้เช่นกัน

แทรกเพิ่มเติม!(Web Services) จากตรงจุดนี้เอง ที่ทำให้เกิดมาตรฐาน Web Services ขึ้นมา เพื่อให้สามารถเรียกใช้งาน Software Component ได้โดยไม่ขึ้นกับ Platform โดยใช้เทคโนโลยี XML (Extensible Markup Language) เป็นตัวกำหนด Interface ของ Component และมาตรฐานส่วนอื่นๆโดยแบ่งออกเป็น 3 ส่วน คือ SOAP (Simple Object Access Protocol), WSDL (Web Service Definition Language) และ UDDI (Universal Description and Discovery Integration) ซึ่งทั้งหมดนี้จะใช้งานบน HTTP/HTTPS ซึ่งเป็น Protocol ที่นิยมใช้ในปัจจุบัน ซึ่งทำให้สามารถ Component สามารถให้บริการผ่าน Web ได้ และเพื่อให้มีการใช้กันได้อย่างแพร่หลาย

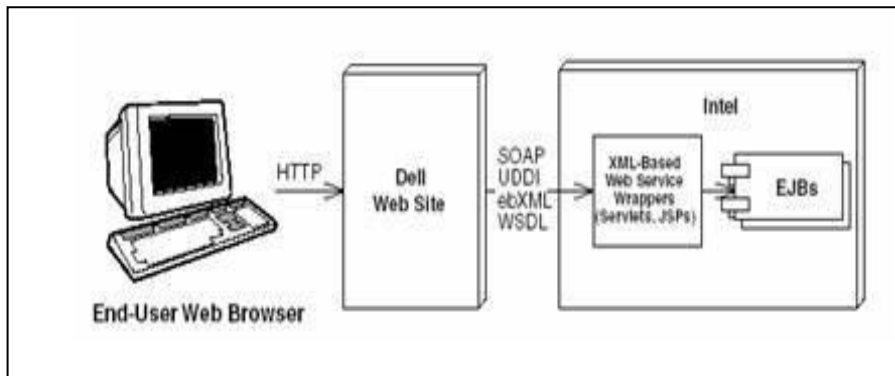
ในส่วนของ Web Service WSDL จะเป็นตัวกำหนด Interface ของ Component ส่วน SOAP จะเป็นรูปแบบการเรียกใช้งาน Component แบบข้ามเครื่อง ข้ามเครือข่าย และ UDDI จะเป็นเหมือนฐานข้อมูลที่เก็บที่อยู่ และบริการของ Software

Component ซึ่งสามารถเข้าไปค้นหา Component ที่ต้องการได้ โดย UDDI จะเก็บ WSDL ของแต่ละ Component ไว้ เมื่อทราบ Interface ก็สามารถใช้ SOAP เพื่อเรียกใช้งาน Component ได้ต่อไป

ตารางที่ 1 ตารางแสดงการเปรียบเทียบเทคโนโลยี Software Component แบบ Distributed

Specification Technology	Interface Definition	Remote Procedure Call	Resource Allocation
Java	EJB	RMI	JNDI
Microsoft	COM	DCOM	ADSI
OMG - CORBA	IDL	IOP	CORBA Naming Service (COS Naming)
Web Services	WSDL	SOAP	UDDI

จากตาราง แสดงให้เห็นว่า ในส่วน Component Architecture นั้นมี 2 ค่ายยักษ์ใหญ่ คือ Sun กับ Microsoft โดยมี OMG เป็นตัวเชื่อมทั้งสองค่ายเข้าด้วย แต่ในขณะเดียวกัน Web Services ซึ่งเป็นมาตรฐานใหม่ เริ่มได้รับความนิยม เพราะมันไม่ขึ้นกับ Platform โดยอาศัยเทคโนโลยี XML และใช้งานผ่าน HTTP/HTTPS ที่นิยมแพร่หลายอยู่ในปัจจุบัน



ภาพที่ 12 สถานการณ์การทำธุรกิจลักษณะ B2B

จากภาพที่ 12 เป็นตัวอย่างแสดงสถานการณ์การทำธุรกิจลักษณะ B2B โดยมี EJB เป็น Software Component ของระบบงานหนึ่งใน บริษัท Intel แล้วมี บริษัท Dell เป็น ผู้ใช้บริการ โดยใช้ SOAP , UDDI , ebXML และ WSDL

การใช้งาน EJB

1. ทำ Business Logic เช่น การคำนวณภาษี การคิดราคาสินค้า หรือการส่งจดหมายอิเล็กทรอนิกส์โดยใช้ API ที่ชื่อว่า Java Mail เป็นต้น

2. ทำ Database Logic โดยใช้ API ที่ชื่อว่า JDBC (Java Database Connectivity)
3. ติดต่อกับ Legacy System หรือระบบอื่นๆ โดยใช้ API ที่ชื่อว่า JCA (JavaBean Connector Architecture)

ความแตกต่างระหว่าง GUI กับ EJB

มองในแง่การใช้งาน GUI จะใช้งานทางฝั่ง Client เป็นหลัก (Client Side Operation) ส่วน EJB จะเป็นใช้งานทางฝั่ง Server เป็นหลัก (Server Side Operation)

ตัวอย่างหน้าที่ของ GUI (Graphic User Interface)

1. Thick Client เช่น Applet หรือ Java Application
2. Dynamic Web Page เช่น Applets หรือ JSP
3. XML-base Web Service ซึ่งในส่วนนี้บางที่อาจจะไม่ต้องการ GUI เลย

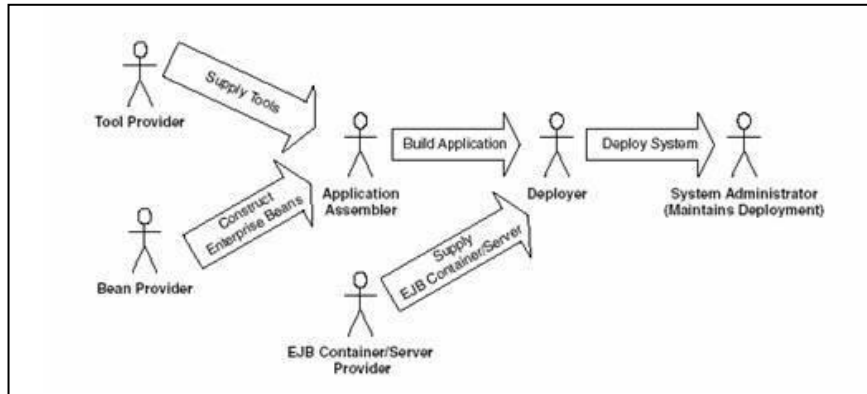
ก็ได้

ตัวอย่างหน้าที่ของ EJB

1. Executing complex algorithms ทำหน้าที่ประมวลผลต่างๆ
2. ทำ Transaction จำนวนมาก
3. รองการทำงาน 24 x 7 , fault-tolerant , transactional and security environment ซึ่งทั้งหมดนี้ Application Server จะสร้างสภาวะการทำงานให้ EJB ทำงาน (ทำให้ EJB มุ่งที่ Business Logic อย่างเดียว บริการอื่นๆ Application Server จะจัดเตรียมไว้ให้)

EJB Ecosystem

EJB Specification ได้ให้คำแนะนำในการแบ่งกลุ่มของการทำงานต่างๆ ของสภาวะแวดล้อมในการพัฒนา EJB ไว้ 6 กลุ่ม เพื่อให้เกิดประสิทธิภาพ ซึ่งในความเป็นจริง ไม่จำเป็นต้องปฏิบัติตามก็ได้ กลุ่มต่างๆ แสดงได้ ดังภาพ



ภาพที่ 13 การแบ่งกลุ่มของการทำงานต่างๆ ของสภาวะแวดล้อมในการพัฒนา EJB

1. Bean Provider เป็นผู้ผลิต EJB ซึ่งจะประกอบด้วย Business Logic Support ให้กับธุรกิจประเภทต่างๆ
2. EJB Containment/Server Provider เป็นผลิต Application Server ตัวอย่าง Web Sphere ของ IBM หรือ JBOSS เป็นต้น
3. Application Assembler จะเป็นผู้ที่นำ Component EJB มาประกอบกันเป็น Application ตามที่ต้องการ
4. Tool Provider จะเป็นผู้ผลิต เครื่องมือต่างๆ เพื่อลดระยะเวลาในการพัฒนา เพิ่มความสะดวกให้กับผู้พัฒนา
5. Deployer เป็น จัดทำให้ Component EJB Deploy และ Run บน Application Server ได้
6. System Administrator หรือ Maintain Deployment เป็นผู้ดูแลระบบงานหลังจาก Deploy แล้ว

จากองค์ประกอบ ทั้ง 6 กลุ่ม ทำให้เห็นผู้ที่ต้องเข้ามาเกี่ยวข้องกับการพัฒนา EJB และได้เห็นขั้นตอนการทำงานโดยรวมของการพัฒนา Application

ความแตกต่างระหว่าง Java Bean และ EJB

Java Bean คือ การพัฒนาพัฒนา Java Class File ที่มี Method Set/Get มีความสามารถในเรื่อง ของ Event Handling , Propoties และ Methods (จะคล้ายกับ ActiveX Component ของ Microsoft) ซึ่ง Java Bean สามารถนำมาประกอบกัน ให้เป็น Java Appliation ได้

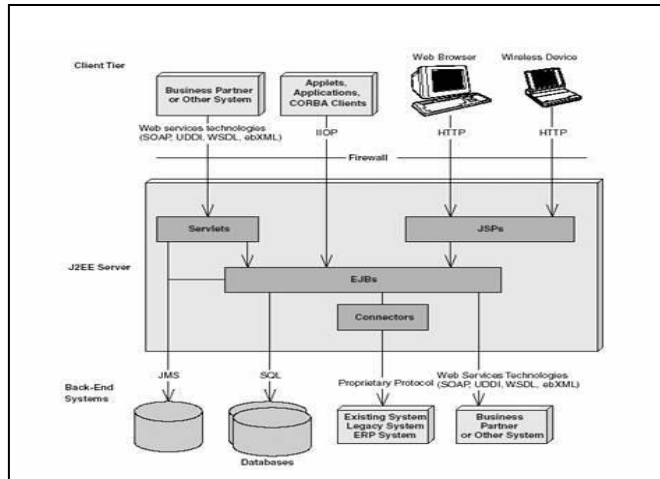
ในส่วนของ Java Bean จะมีขนาดเล็ก กว่า EJB มาก Java Bean จะมีลักษณะ Development Component แต่ส่วน EJB จะมีลักษณะ Deployment Component ทำให้ Java Bean ไม่ต้องการ Application Server เป็น Runtime Environment ที่จะบริการ Service ต่างๆ ให้ เช่น การ Instanciate Class ซึ่งใน Java Bean ตัว Application เองจะเป็นผู้ควบคุมการ Instanciate Class Java Bean แทน และสามารถใช้ EJB ใน Application ที่มีขนาดใหญ่ ได้ดีกว่า Java Bean

J2EE Technology

ในการนำ J2EE มาใช้งาน จะต้องใช้พื้นฐาน จาก J2SE มาใช้งานด้วย ซึ่งจะพบว่า การทำงาน ใน J2EE เป็นการทำงานที่มีขนาดใหญ่มากพอสมควร ดังแสดงในรูปที่ 7 ใน J2EE จะประกอบด้วย API ต่างๆ ดังนี้

1. EJB Specification ซึ่งใน J2EE Version 1.3 จะ Support EJB 2.0
2. Java Remote Method Invocation (RMI) และ RMI/IIOP RMI เป็นการเรียกใช้งาน Component ข้ามเครื่อง (Distributed Component) ส่วน RMI/IIOP นั้นเป็นส่วนที่มีความสอดคล้องกับ CORBA
3. Java Naming and Directory Interface (JNDI) เป็นส่วนที่ติดต่อกับการใช้ Resource ต่างๆ เช่น การติดต่อใช้ EJB ก็ต้องใช้ JNDI ในการติดต่อ
4. Java Database Connectivity (JDBC) ใช้ในการติดต่อฐานข้อมูลเชิงสัมพันธ์ต่างๆ (Relational Database)
5. Java Transaction API (JTA) และ Java Transaction Service (JTS) ใช้การทำ Transaction ของระบบงาน ถ้าจะทำการเขียนเอง ก็ใช้ JTA แต่จะใช้บริการ Application Server ก็จะใช้ JTS
6. Java Message Service (JMS) เป็นการให้การติดต่อโดยใช้ Message แทน ซึ่งเป็นแนวคิดใหม่ที่จะใช้กับ RMI/IIOP โดยจะส่ง Message ไปยัง Message Oriented Middleware (MOM) เพื่อเข้าคิว แล้วตัว MOM จะส่ง Message ไปให้แทน จะใช้ในกรณีที่ไม่ต้องการทำงานแบบ Transaction หรือ Realtime มีข้อดีคือ จะรับประกันการส่ง Message ว่าไม่มีการสูญหาย ส่งถึงผู้รับแน่นอน
7. Java Servlet เป็น Server - Side Component ชนิดหนึ่ง ที่ใช้หลัก Request/Response ซึ่งต่างจาก EJB ที่ต้อง Run บน Application Server
8. JavaServer Page เป็น ภาษา Script ที่ทำหน้าที่คล้ายกับ Servlet มาก เพราะจะต้องแปลง JSP ไปเป็น Servlet เหมาะสำหรับการ Presentation ของ Web Page
9. Java IDL (Interface Definition Language) เป็นการนำ Specification ของ CORBA มาใช้ใน Java
10. Java Mail เป็น API ที่ใช้ในการเขียน โปรแกรมรับส่ง E-Mail
11. JavaBean Connector Architecture เป็นการเขียนโปรแกรมเพื่อเชื่อมต่อรับ Legacy System หรือ ระบบงาน อื่นๆ เช่น ระบบงาน ของ Business Partner เป็นต้น
12. Java API for XML Parsing/Processing เป็นชุด API ที่ใช้เขียนโปรแกรม ร่วมกับ เทคโนโลยี XML
13. Java Authentication and Authorization Service (JAAS) เป็น API ที่ดูแลเกี่ยวกับระบบรักษาความปลอดภัย

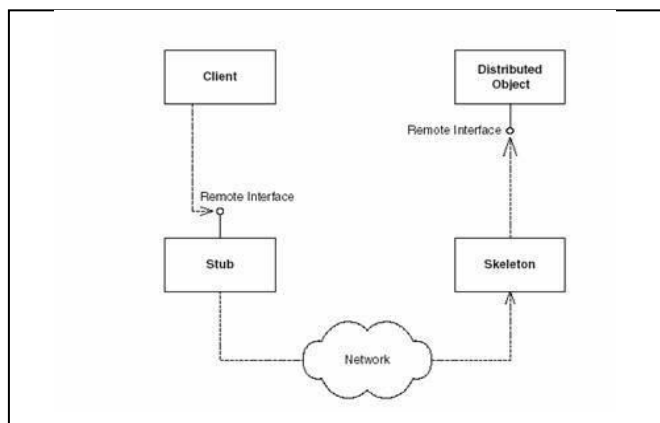
จากรายการ API ข้างต้นทั้งหมด จะพบว่า J2EE ได้จัดเตรียม API ชุดใหญ่ ไว้ อย่างเพียงพอที่จะให้นักพัฒนาที่ต้องการพัฒนาระบบงานในระดับหน่วยงานขนาดใหญ่ได้เป็น อย่างดี ซึ่งทั้งนี้จะต้องรวมกับชุด API ในส่วนของ J2SE ด้วย เช่น AWT, IO หรือ NET เป็นต้น



ภาพที่ 14 การแสดงการนำ J2EE มาใช้งาน

จากภาพที่ 14 เป็นการแสดงการนำ J2EE มาใช้งาน จะพบว่า แบ่งเป็น 3 ส่วน คือ Client Tier จะประกอบไปด้วย Web Browser , Wireless Device, Business Partner โดยใช้ Web Services และ CORBA Client และมีในส่วนของ Application Server ที่ยึดตาม Specification ของ J2EE เราอาจเรียกว่า J2EE Server จะเป็นแบ่งได้เป็น Servlet/JSP , EJB หรือ Connector ต่างๆ ในส่วนที่ 3 คือ ส่วนที่เป็น Back End ก็คือส่วนที่เป็น Database หรือ อาจจะเป็น Legacy System หรือ จะเป็นระบบงานของ Business Partner ก็ได้

หลัก Distributed Object



ภาพที่ 15 แสดงการใช้งาน ของ Distributed Object

จากภาพที่ 15 แสดงการใช้งาน ของ Distributed Object จะพบว่า ในส่วนของ Client นั้นจะมี Stub และในส่วนของ Distributed Object จะมี Skeleton ซึ่งในส่วนของ

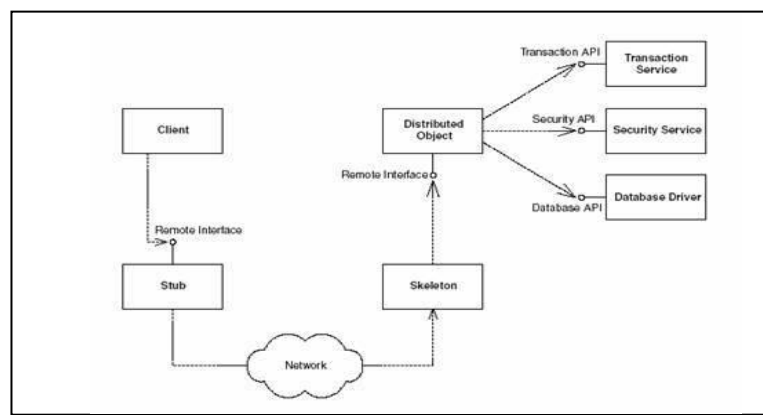
Stub จะมี Remote Interfaces ที่เหมือนกับทางด้าน Distributed Object ด้วยวิธีนี้ ทำให้ Client ไม่รับรู้ว่าได้เรียกใช้ Object ที่ต่างเครื่องกันอยู่ ซึ่งสามารถทำได้โดยใช้มาตรฐานต่างๆ ได้มากมาย เช่น RMI/IIOP, CORBA หรือ DCOM

Distributed Object and Middleware

แบ่ง Middleware ออกเป็น 2 ประเภท คือ

1. Explicit Middleware
2. Implicit Middleware

Explicit Middleware



ภาพที่ 16 การทำงานของ Middleware แบบ Explicit

จากภาพที่ 16 แสดงให้เห็นถึงการทำงานของ Middleware แบบ Explicit ซึ่งในตัวของ Distributed Object จะเป็นผู้เรียกใช้ API เอง ดังตัวอย่าง Pseudo-code ดังภาพที่ 17

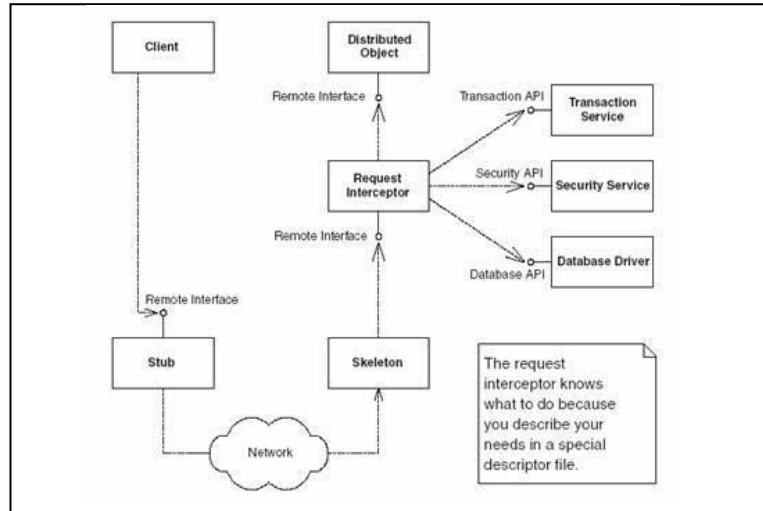
```
transfer(Account account1, Account account2, long amount) {
// 1: Call middleware API to perform a security check
// 2: Call middleware API to start a transaction
// 3: Call middleware API to load rows from the database
// 4: Subtract the balance from one account, add to the other.
// 5: Call middleware API to store rows in the database.
// 6: Call middleware API to end the transaction
}
```

ภาพที่ 17 การเรียกใช้ API ในด้านต่างๆ

จากภาพที่ 17 จะพบว่า ใน Pseudo-code นั้นจะมีการเรียกใช้ API ในด้านต่างๆ เช่น Security Check , Transaction และ JDBC และมีส่วนที่เป็น Business Logic ในบรรทัดที่ 4

ปัญหาในการเขียน Object แบบนี้ คือ ต้องเรียนรู้การเรียกใช้ API ต่างๆ มากมาย ทำให้เขียนยาก และเมื่อมีการเปลี่ยนแปลง API ก็ต้องเข้าไปเปลี่ยนแปลงใน Code ใน Object ด้วย ทำให้การ Maintenance ยาก

Implicit Middleware



ภาพที่ 18 Implicit Middleware

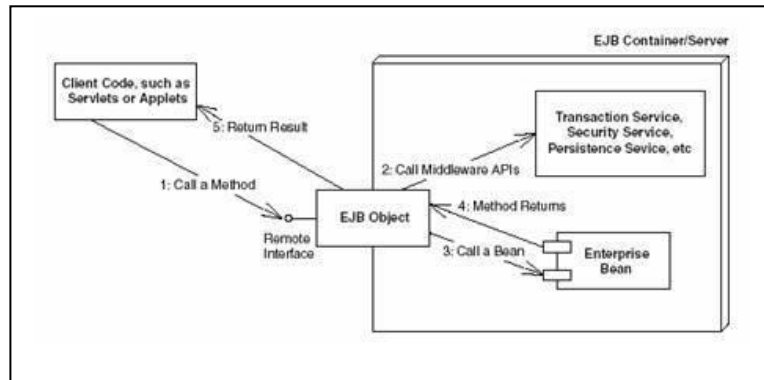
จากภาพที่ 18 จะพบว่า จะมี Request Interceptor เป็นตัวคั่นกลางระหว่าง Skeleton ที่จะเรียกใช้งาน Distributed Object และ ตัว Request Interceptor จะเป็นผู้ที่ทำหน้าที่ใช้ในการเรียกใช้งาน API ต่าง ๆ แทน ทำให้ Code ที่เขียนใน Distributed Object เหลือเพียงแต่ Business Logic เพียงอย่างเดียว ดังตัวอย่าง Pseudo-code ในภาพที่ 19

```
transfer(Account account1, Account account2, long amount) {
    // 1: subtract the balance from one account, add to the other
}
```

ภาพที่ 19 ตัวอย่าง Pseudo-code

จากภาพที่ 19 จะเห็นว่าในส่วนของ Distributed Object จะเหลือ Code ในส่วนที่เป็น Business Logic เท่านั้น โดยในการติดต่อการใช้ API อื่น ๆ จะเป็นหน้าที่ของ Request Interceptor แทน ซึ่งเป็นการทำให้ Distributed Object รับหน้าที่ในการทำ Business Logic เพียงอย่างเดียว ทำให้สะดวกในการพัฒนาเป็นอย่างมาก และ Loose Coupling กับ API ต่างๆ คือ ถ้า เปลี่ยนแปลง API ก็ไม่กระทบกับ Business Logic และ การเปลี่ยนแปลง Business Logic ก็ไม่กระทบกับ API ด้วยเช่นกัน ทำให้ง่ายกับการ Maintenance เป็นอย่างมาก

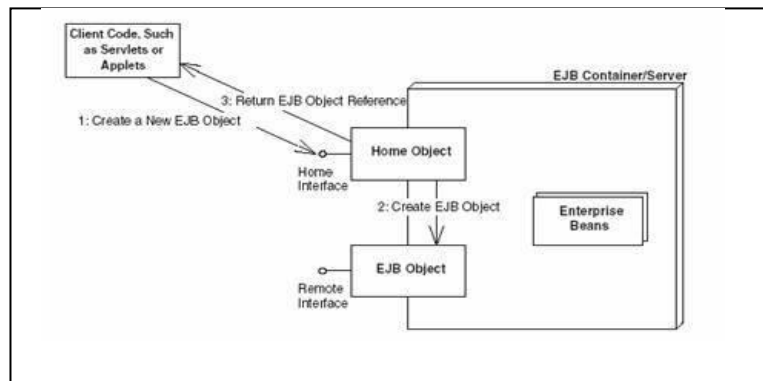
การนำ Implicit Middleware มาใช้ใน J2EE



ภาพที่ 20 การนำ Implicit Middleware มาใช้ใน J2EE

จากภาพที่ 20 พบว่า EJB Object จะทำหน้าที่เป็น Request Interceptor คือ เป็นตัวจัดการเรียกใช้งาน EJB จาก Client ไม่ว่าจะเป็ Servlet หรือ JSP ก็ตาม โดย EJB Object จะทำหน้าที่ในการติดต่อกับ EJB แทน และยังเรียกใช้งาน API แทนด้วย โดยทั้งหมดทำงานภายใต้ EJB Container/Server

The Home Interface



ภาพที่ 21 การสร้าง EJB Object

จากภาพที่ 21 เมื่อ Client ต้องการสร้าง EJB Object จะสร้างโดยผ่าน Home Object ซึ่ง Home Object จะสร้างโดย Client โดยใช้ Method ที่บรรจุอยู่ใน Home Interface เมื่อสร้างแล้ว Home Object จะส่งค่า EJB Object Reference กลับไปให้ Client เพื่อใช้เรียก Business Logic ใน EJB โดย EJB Object ก็จะทำกา Intercept Request จาก Client แล้วส่งต่อไปให้ EJB ทำงานต่อไป

The Local Interface

เป็น Specification ที่เป็นเพิ่มเข้าใน EJB 2.0 ซึ่งเป็นการตัดทอนการทำงานในส่วนของ Stub และ Skeleton ซึ่งเป็นส่วนที่ใช้เวลานาน (Overhead) Local Interface จะใช้

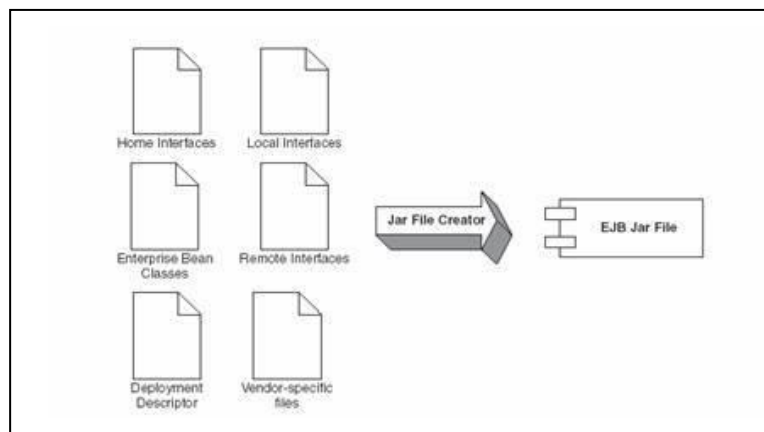
ในกรณีที่ การเรียกใช้กับกับ EJB อยู่ใน Process เดียวกัน ซึ่งจะช่วยลดเวลาการทำงานได้ดีขึ้นอย่างมาก

Local Interface จะทำหน้าที่ Remote Interface และ Local Home Interface จะทำหน้าที่แทน Home Interface โดยมี Local Object แทน EJB Object และ Local Home Object แทน Home Object

Deployment Descriptor

เป็นไฟล์ในรูปแบบ XML ที่แสดงการใช้งาน ทำให้ Middleware ทราบว่า จะต้องใช้บริการต่างๆ อย่างไร กับ EJB ที่ถูกเรียกใช้งาน เช่น ให้ทำหรือไม่ต้องทำ Transaction , Security และอื่นๆ เป็นต้น

EJB - jar File



ภาพที่ 22 การรวบรวมไฟล์ต่างๆ ที่จะใช้ Deploy ลงใน Application Server

จากภาพที่ 22 เป็นการรวบรวมไฟล์ต่างๆ ที่จะใช้ Deploy ลงใน Application Server ให้อยู่ในรูปแบบที่เรียกว่าไฟล์ EJB-jar ซึ่งจะขึ้นอยู่ Application Server ว่าจะต้องทำอย่างไร

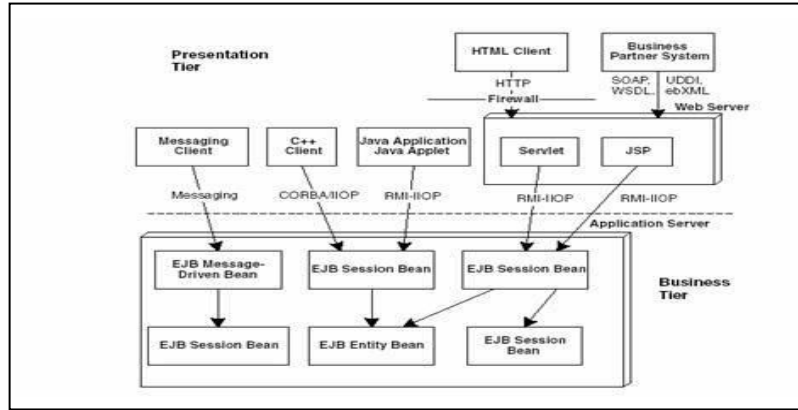
ประเภทของ EJB

EJB แบ่งออกได้เป็น 3 ประเภท ดังนี้

1. Session Bean เป็น EJB ที่เน้นถึงการกระทำ ของ Business Logic แบ่งเป็น 2 ชนิด ย่อย ๆ คือ Stateless Session Bean และ Stateful Session Bean โดยที่ Stateless Session Bean จะไม่การเก็บสถานะ แต่ Stateful Session Bean จะมีการเก็บสถานะการติดต่อไว้ได้ เพื่อใช้ประโยชน์ในการทำ Transaction

2. Entity Bean เป็น EJB ที่เน้นถึงสิ่งที่ต้องการเก็บข้อมูล แบ่งย่อยๆ ตามวิธีการ Persistent ได้ 2 อย่าง คือ แบบ Container Managed Persistent (CMP) ซึ่งจะให้ Container เป็นผู้จัดการทำให้แทน กับ แบบ Bean Managed Persistent (BMP) โดยภายในตัว EJB จะมี Logic ที่จัดการตัวเอง ซึ่งตรงนี้สามารถกำหนดได้ใน Deployment Descriptor นั้นเอง

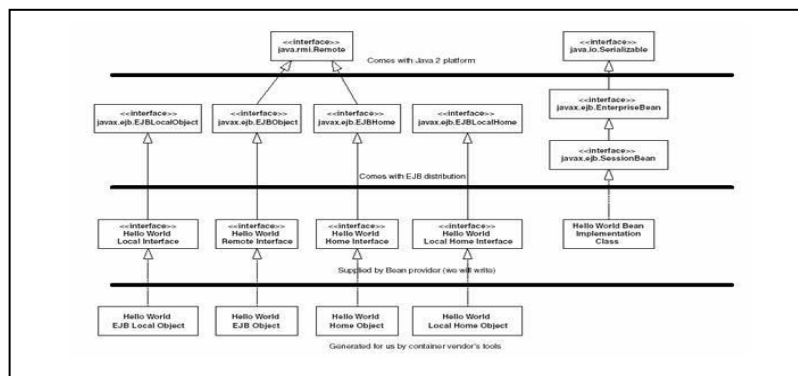
3. Message-Driven Bean เป็น EJB ที่ใช้ในการทำงานที่เกี่ยวกับ Message เป็นหลัก ซึ่งจะบรรจุ Message-Oriented Logic เพื่อให้ Client เรียกใช้งาน



ภาพที่ 23 แสดงการใช้งาน EJB

จากภาพที่ 23 แสดงการใช้งาน EJB ประเภทต่างๆ จาก Client ต่างๆ จะพบว่า EJB สามารถที่จะเรียกกันและกันได้ด้วย จากรูปจะพบว่า จะใช้ Session Bean เป็นตัวควบคุมการทำงานของ Entity Bean หรือ ไปควบคุม Session Bean ตัวอื่นๆ จากรูปยังแสดงให้เห็นในกรณี ที่ Web Server แยกกับ Application Server ซึ่ง ทั้ง Servlet และ JSP จะใช้ RMI/IIOP ในการเรียกใช้ EJB แต่ถ้าเป็น C++ Client จะใช้เป็น CORBA/IIOP แทน ซึ่งในส่วน of Client ถ้า เป็น Browser ก็ใช้ HTTP แต่ถ้าเป็น Business Partner ก็ใช้ Protocol ต่างๆ ที่อยู่ใน Web Services แทน

ตัวอย่างการพัฒนา EJB Component (Session Bean)



ภาพที่ 24 แสดง Component Diagram

จากภาพที่ 24 แสดง Component Diagram ก็พบว่า สิ่งที่จะทำการเขียน ก็คือ 1. Local Interface 2. Remote Interface 3. Home Interface 4. Local Home

Interface 5. Bean File โปรแกรมจะเป็นการทำงานง่ายๆ คือ พิมพ์คำว่า Hello World ออกมา และสุดท้าย ก็เขียนโปรแกรม Client เพื่อทดสอบการเรียกใช้งาน EJB

หลังจากที่จัดทำ ไฟล์ต่างๆ ทั้งหมดเสร็จแล้ว จะต้อง จัดทำไฟล์ Deployment Descriptor เพื่อนำไปรวมกับไฟล์ Bean ต่างๆ 5 ไฟล์ แล้วจัดทำเป็น EJB-jar ไฟล์ แล้วนำไป Deploy บน Application Server เพื่อให้ Client ต่างๆ เรียกใช้งานได้ต่อไป

หน้าต่อๆ ไป จะเป็นการแสดงตัวอย่าง Code ในการพัฒนา Session Bean แบบ Stateless ไล่เรียงกันไปดังนี้

1. Remote Interface (Hello.java)
2. Local Interface (HelloLocal.java)
3. Home Interface (HelloHome.java)
4. Local Home Interface (HelloLocalHome.java)
5. Bean Class (HelloBean.java)
6. Deployment Descriptor (ejb-jar.xml)
7. Client Class (HelloClient.java)

อ้างอิง <https://www.gotoknow.org/posts/69571>

8. โปรแกรม Adobe Photoshop cs6

(Adobe Photoshop) มักเรียกสั้นๆ ว่า โฟโตชอป เป็นโปรแกรมประยุกต์ที่มีความสามารถในการจัดการแก้ไขและตกแต่งรูปภาพ (photo editing and retouching) แบบแรสเตอร์ ผลิตโดยบริษัทอะโดบีซิสเต็มส์ (Adobe System) ซึ่งผลิตโปรแกรมด้านการพิมพ์อีกหลายตัวที่ได้รับความนิยม เช่น Illustrator และ InDesign ปัจจุบันโปรแกรม Photoshop พัฒนามาถึงรุ่น CS6 (Creative Suite 6) Adobe Photoshop CS6 เป็นโปรแกรมทางด้านกราฟิก ที่คิดค้นโดยบริษัท Adobe ที่ใช้กันในอุตสาหกรรมกราฟิกระดับโลก และเป็นที่ยูจิกกันดีในกลุ่มนักออกแบบสื่อสิ่งพิมพ์ทั่วไป และสามารถแลกเปลี่ยนไฟล์ต่างๆ และนำไปใช้งานร่วมกับโปรแกรมอื่นๆ ได้ โปรแกรม Adobe Photoshop เริ่มออกมาเวอร์ชันแรกคือ Photoshop 2 และมีการพัฒนามาเป็นเวอร์ชัน เวอร์ชัน 8 หรือ Photoshop cs และล่าสุดได้พัฒนามาเป็น Photoshop cs6 โดยได้พัฒนาขีดความสามารถในการใช้งานในลักษณะต่างๆ ให้มีประสิทธิภาพมากขึ้น

9. โปรแกรม Brackets

โปรแกรม Brackets (โปรแกรม Brackets แก้ไข เขียนโค้ดโปรแกรมง่ายๆ) : โปรแกรมนี้มีชื่อว่า โปรแกรม Brackets มันเป็น โปรแกรมแก้ไข เขียนโค้ด แบบโอเพ่นซอร์สแจกฟรี (Open-Source) เหมาะสำหรับ โปรแกรมเมอร์ หรือ ผู้พัฒนาโปรแกรม เขียนโปรแกรม ในภาษาต่างๆ ใช้งานง่าย โดยมันจะเน้นไปทางการทำเว็บไซต์ สนับสนุนภาษาเฮชทีเอ็มแอล (HTML)

ซีเอสเอส (CSS) จาวาสคริป (Javascript) ต่างๆ นอกจากนี้มันยังเป็น โพรเจคโอเพ่นซอร์ส ซึ่ง แจกฟรี และนอกจากนี้ ผู้พัฒนาสามารถดาวน์โหลดไปพัฒนามันต่อยอดได้อีกด้วย

ด้านการใช้งาน อารมณ์มันจะประมาณคุณเปิดเว็บไซต์เว็บหนึ่งขึ้นมา ซึ่งฝั่งซ้ายจะเป็นรายการไฟล์ที่กำลังเปิดแก้ไข ใช้งานอยู่ (Working Files) ส่วนฝั่งขวา จะเป็นส่วนของโค้ด ที่ให้คุณได้สามารถแก้ไขได้ง่ายๆ พร้อมมีการแยกสีทำไฮไลต์ ระหว่างโค้ด กับข้อมูลด้านในจริงๆ นอกจากนี้ยังมีระบบการควบคุม การเปิด ปิดแท็ก วงเล็บ ต่างๆ ไม่ให้คุณได้ปิด หรือ ลืมปิด มันอีกด้วย ซึ่งถือได้ว่าจุดนี้ สำคัญมากๆ สำหรับ โปรแกรมเมอร์ ที่มักจะผิดพลาดกันตรงนี้บ่อยครั้ง

นอกจากนี้ยังมีระบบรีวิวกันให้ดูแบบสุดๆ ว่า เขียนโค้ดไปแบบนี้ หน้าเว็บของจริงจะออกมาเป็นรูปแบบยังไง หากไม่พอใจก็กลับไปแก้ไขใหม่ นอกจากนี้แล้วยังมีออปชั่นพิเศษต่างๆ มากมาย อาทิเช่นเครื่องมือในการเลือกสี (Color Picker) หรือจะเป็นการไล่สี (Gradient)

10. Genymotion

Genymotion เป็น Android Emulator ที่มาพร้อมกับ Android System Image และ AVD สำเร็จรูปมากกว่า 10 ตัว ช่วยให้นักพัฒนาไม่ต้องกังวลเรื่องการตั้งค่า AVD อีกต่อไป ที่สำคัญ Genymotion ทำงานเร็วกว่า Emulator ที่มากับ Android SDK มากๆ Genymotion มีเครื่องมือช่วยนักพัฒนาหลายอย่างทั้งการเชื่อมต่อกับ Webcam, GPS, Screencast และอีกมากมาย Genymotion สามารถดาวน์โหลดมาใช้งานได้

แนวคิดและทฤษฎีที่เกี่ยวข้องกับเว็บเซอร์วิส (Web Service) แบบ REST

1. เทคโนโลยีเว็บเซอร์วิส

เว็บเซอร์วิส (Web Service) เป็นโปรแกรมประยุกต์ หรือโปรแกรมที่ทำงานในลักษณะให้บริการโดยจะถูกเรียกใช้งานจากโปรแกรมประยุกต์อื่นๆ ในรูปแบบ RPC (Remote Procedure Call) ซึ่งการให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ โดยภาษาที่ถูกใช้เป็นตัวกลางในการแลกเปลี่ยนคือ XML ทำให้เราสามารถเรียกใช้องค์ประกอบ (Component) ต่างๆ ได้ในระบบหรือ Platform ใดก็ได้บนโปรโตคอล HTTP ซึ่งเป็นโปรโตคอลสำหรับ World Wide Web หรืออินเทอร์เน็ต อันเป็นช่องทางที่ได้รับการยอมรับทั่วโลกในการติดต่อสื่อสารกันระหว่างโปรแกรมประยุกต์กับโปรแกรมประยุกต์ในปัจจุบัน สราวุธ อ้อยศรีสกุล (2544, หน้า 36)

แนวคิดของเว็บเซอร์วิส คือ ระบบอินเทอร์เน็ตเป็นระบบปฏิบัติการ และเว็บไซต์ต่างๆ ที่สร้างด้วยเทคโนโลยีเว็บเซอร์วิสเป็นซอฟต์แวร์ที่ทำงานอยู่ภายใต้ระบบปฏิบัติการ อินเทอร์เน็ตและการเรียกใช้งานเว็บไซต์ คือ เซอร์วิส (Service) หรือเว็บเซอร์วิส เมื่อการใช้งานบริการต่างๆ จากโฮมเพจ (Home Page) เป็นเว็บเซอร์วิส สิ่งที่ย่อยให้บริการต่างๆ ในเว็บที่เข้าไปใช้บริการจะถูกเรียกว่าโซลูชัน (Solutions) หากเปรียบเทียบในยุคปัจจุบัน กล่าวได้ว่าโฮมเพจที่สร้างขึ้นมามีโซลูชันหลากหลาย เช่น E-mail, Free Counter, Free GuestBook เป็นต้น

1.1 หลักการพื้นฐานของเทคโนโลยีที่พัฒนามาสู่เว็บเซอร์วิสซึ่ง ยืน ภู่วรรณ (2546. หน้า 42) กล่าวไว้ดังนี้

1.1.1 การพัฒนาโปรแกรมแบบซอฟต์แวร์คอมโพเนนท์ ตามแนวคิดของการออกแบบและพัฒนาโปรแกรมเชิงวัตถุ (Object-Oriented Concept)

1.1.2 การออกแบบระบบแบบกระจายจากศูนย์กลาง (Distributed Computing) ซึ่งเป็นเป้าหมายสำคัญของการพัฒนาระบบตามสถาปัตยกรรมแบบไคลเอนต์เซิร์ฟเวอร์

1.1.3 การทำอีดีไอ หรือ อิเล็กทรอนิกส์ดาต้าอินเทอร์เช้นต์ (Electronic Data Interchange) ซึ่งสร้างขึ้นโดยกำหนดรูปแบบและมาตรฐานของข้อมูลสำหรับการทำธุรกิจ

1.1.4 การบูรณาการของซอฟต์แวร์ต่างระบบอีเอไอ (EAI) หรือ เอนเทอร์ไพรส แอปพลิเคชันอินทิเกรชัน (Enterprise Application Integration) ที่อยู่บนพื้นฐานของความต้องการใช้ข้อมูลร่วมกัน รวมทั้งการแลกเปลี่ยนข้อมูลระหว่างแอปพลิเคชันให้สามารถทำงานได้อย่างถูกต้องเหมาะสม

1.1.5 รูปแบบการให้บริการซอฟต์แวร์แบบเอเอสพี (ASP) หรือ แอปพลิเคชันเซอร์วิสโพรไวเดอร์ (Application Service Provider)

1.1.6 แนวคิดการทำเหมืองข้อมูล (Data Mining) ซึ่งต้องการนำข้อมูลที่ถูกจัดเก็บไว้ในรูปแบบที่แตกต่างกันตามแหล่งต่างๆ มาใช้งานร่วมกันโดยที่เว็บเซอร์วิสได้นำหลักการต่างๆ เหล่านี้มาดำเนินการปรับปรุง แก้ไขข้อจำกัดต่างๆ โดยการคิดค้นเทคโนโลยีที่เป็นมาตรฐานเปิด (Opened Standard) และไม่มีใครเป็นเจ้าของ

1.2 เว็บเซอร์วิสมีคุณลักษณะ ดังต่อไปนี้ วีรพงศ์ วรไพจิตร (2545. หน้า 191 - 193)

1.2.1 รายละเอียดในการสร้างและพัฒนาเว็บเซอร์วิสจะถูกซ่อนไว้ เพื่อไม่ให้มองเห็นได้จากภายนอก ผู้เรียกใช้เว็บเซอร์วิสจะรู้จักเพียงลักษณะรูปแบบการติดต่อที่ผู้ให้บริการประกาศเอาไว้เท่านั้น กล่าวคือเว็บเซอร์วิสจะเป็นประตูกั้นระบบงานภายในกับภายนอกด้วยกรรมวิธีทางออบเจกต์

1.2.2 ซอฟต์แวร์ที่ทำงานบนระบบเว็บเซอร์วิส สามารถนำมาแก้ไขรายละเอียดภายในได้โดยไม่ส่งผลกระทบต่อผู้ใช้ ทำให้การออกแบบเป็นไปได้ง่าย และผู้ใช้ที่ปลายทางก็ไม่จำเป็นต้องโหลดซอฟต์แวร์ติดตั้งมากเกินไปจนความจำเป็น

1.2.3 โปรแกรมที่เรียกใช้เว็บเซอร์วิสจะรับรู้ได้เอง ว่าเซอร์วิสที่กำลังจะเรียกใช้นั้นมีลักษณะและข้อกำหนดของอินพุตและเอาพุตอย่างไร

1.2.4 ความเป็นโพรโตคอลมาตรฐานนับเป็นคุณสมบัติที่สำคัญที่สุดของเว็บเซอร์วิสเนื่องจากมีมาตรฐานอยู่บนโซฟและเอชทีทีพี

1.2.5 เว็บเซอร์วิสมีคำอธิบายอยู่ในตัวเอง ซึ่งถูกเรียกใช้ขณะที่กำลังจะรัน เท่านั้นเว็บเซอร์วิสสนับสนุนการค้นหาและเรียกใช้แบบไดนามิก แอปพลิเคชันสามารถค้นหาและเรียกใช้เว็บเซอร์วิสได้ในขณะรันไทม์ เพิ่มความยืดหยุ่นให้กับการพัฒนาซอฟต์แวร์

1.3 โครงสร้างเว็บเซอร์วิส

การพัฒนาเว็บเซอร์วิสใช้สถาปัตยกรรมบริการในลักษณะสถาปัตยกรรมของแนวคิดทางด้านบริการที่เรียกว่า เอสโอเอ (SOA) หรือ (Service-Oriented Architecture) เป็นแนวคิดเบื้องต้น แอปพลิเคชันส่วนใหญ่ในโลกของธุรกิจที่ใช้งานในปัจจุบันเป็น แอปพลิเคชันและระบบย่อยที่ถูกสร้างขึ้น มีการทำงานที่ต้องสัมพันธ์กันอย่างไม่เป็นอิสระจากกัน การเปลี่ยนแปลงการทำงานในระบบย่อยหรือแอปพลิเคชันหนึ่ง อาจจะมีผลกระทบต่ออีกแอปพลิเคชันหนึ่ง หรือบางครั้งอาจจะกระทบทั้งระบบ ทำให้การบำรุงรักษานั้นทำได้ยากและมีต้นทุนที่สูงขึ้น รวมทั้งยังเป็นข้อจำกัดในการเชื่อมต่อกับระบบของคู่ค้าอื่นๆ เอสโอเอไม่ใช่แนวคิดใหม่ แต่ได้เกิดขึ้นมานานแล้ว ซึ่งอยู่ในส่วนหนึ่งของแนวคิดการออกแบบระบบแบบกระจายจากศูนย์กลาง (Distributed Computing Concepts) เอสโอเอนั้น เป็นแนวคิดที่ได้รับการยอมรับและประสบความสำเร็จใน

1.4 ซิมเปลอบเจกต์แอกเซสโพรโตคอล

เนื่องจากจุดประสงค์หลักของการใช้งานเว็บเซอร์วิส ต้องการให้แอปพลิเคชัน มีการทำงานกับแอปพลิเคชันที่ทำงานอยู่บนเครื่องอื่น โดยผ่านทางเครือข่าย ซึ่งเทคโนโลยีที่มีอยู่ปัจจุบันที่ใช้มีการสื่อสารระหว่างออบเจกต์ในระยะไกลหรืออาร์พีซี (Remote Procedure Calls: RPC) เช่น ดีคอม (DCOM) อีเจบี (EJB) หรือคอบร้า (COBRA) นั้นไม่ได้ถูกออกแบบมาให้ใช้สำหรับโพรโตคอลเอชทีทีพีเทคนิคอาร์พีซีของเทคโนโลยีที่กล่าวข้างต้น ต่างก็มีปัญหาในด้านการนำมาใช้งานในแง่ของความเข้ากันได้ของการเรียกใช้งานข้ามเทคโนโลยี เนื่องจากเป็นเทคโนโลยีเฉพาะของแต่ละค่าย (ยกเว้น คอบร้า) ผู้พัฒนาระบบจะต้องพัฒนาโปรแกรมที่มีความซับซ้อน และยังมีปัญหาในส่วนของไฟร์วอลล์ (Firewall) และพร็อกซีเซิร์ฟเวอร์ (Proxy Server) ด้วยเนื่องจากโดยปกติเซิร์ฟเวอร์จะปิดการสื่อสารที่ไม่ใช่โพรโตคอลเอชทีทีพีออกไป เพื่อความปลอดภัยของระบบที่มีการติดต่อสื่อสารกับภายนอก วิศิษฐ์ วงศ์วีไล (2545: เว็บไซด์)

ดังนั้นทางเลือกของการสื่อสารที่จะนำมาใช้ในการให้บริการเว็บเซอร์วิส คือ ให้ทำงานอยู่บนโพรโตคอลเอชทีทีพี ซึ่งซอฟต์แวร์จะทำงานบนโพรโตคอลเอชทีทีพีแล้ว ยังเป็นมาตรฐานเปิดที่จะทำให้สามารถติดต่อสื่อสารกับคอมพิวเตอร์ที่มีความแตกต่างทางด้านระบบปฏิบัติการและเทคโนโลยีรวมถึงภาษาที่ใช้ในการพัฒนาด้วยซอฟต์แวร์เป็นโพรโตคอลที่ใช้ เอ็กซ์เอ็มแอลเป็นพื้นฐาน เพื่อให้ซอฟต์แวร์คอมโพเนนท์ และแอปพลิเคชันสามารถติดต่อกันผ่านเอชทีทีพีซึ่งเป็นมาตรฐานอินเทอร์เน็ตโพรโตคอลได้ ขั้นตอนการทำงานของซอฟต์แวร์ดังกล่าวของ สุชาติรัตน์บำรุงศิลป์ (2545: หน้า 91 - 95) ซึ่งอธิบายไว้ดังนี้

1.4.1 แอปพลิเคชันของผู้ขอใช้บริการสร้าง SOAP message เพื่อเรียกใช้ บริการของเว็บเซอร์วิส

1.4.2 เว็บเซอร์วิส ของผู้ให้บริการ ได้รับ SOAP message จากผู้ร้องขอซึ่ง อยู่ในรูปแบบ XML

1.4.3 เว็บเซอร์วิส ทำการประมวลผลคอมโพเนนต์ที่ให้บริการหลังจากนั้น ทำการส่งผลลัพธ์กลับมาในรูปของ SOAP message ให้กับผู้ร้องขอบริการ

1.4.4 แอปพลิเคชันของผู้ขอใช้บริการรับ SOAP message กลับมาแล้วทำ การแปลงให้อยู่ในรูปที่ต้องการเพื่อนำไปประมวลผลต่อ

1.5 ส่วนประกอบของโซฟ

โซฟใช้ไวยากรณ์ของเอ็กซ์เอ็มแอลในการสร้าง ประกอบด้วย 3 ส่วน ซึ่ง วิชาศรัฎ วงศรัไล (2545. หน้า 30) ได้อธิบายไว้ดังนี้คือ

1.5.1 โซฟเอนวิลอป (SOAP Envelop) จะเป็นส่วนสำหรับใช้ในการระบุสิ่ง ที่อยู่ในเอกสารว่าจะต้องจัดการอย่างไร และบอกถึงความจำเป็นในการใช้งาน

1.5.2 โซฟเอนโค้ดดิ้งรูล (SOAP Encoding Rule) จะเป็นส่วนสำหรับ กำหนดกลไกที่ใช้ในการแลกเปลี่ยนข้อมูล ว่ามีข้อตกลงอย่างไรในการแลกเปลี่ยนข้อมูล

1.5.3 โซฟอาร์พีซีรีพีเร็นเทชัน (SOAP RPC Representation) จะเป็น ส่วนสำหรับนิยามรูปแบบรีโมทโพรซีเยอร์คอล (Remote Procedure Call) และการตอบสนอง

2. การทำงานของ Web service

2.1 Document type : รูปแบบเอกสาร โดยนิยามข้อมูลในรูปแบบ Element ที่แสดงข้อมูลที่มีอยู่จริง ในระบบระหว่างผู้ขอใช้บริการ (Service request) กับผู้ให้บริการ (Service Provider)

2.2 Semantics : ความหมายในการสื่อสารของแต่ละ Element ซึ่งต้องสามารถ สื่อสารได้ถูกต้อง ระหว่างผู้ขอใช้บริการ (Service request) กับผู้ให้บริการ (Service Provider)

2.3 Transport Binding : รูปแบบในการส่งข้อมูลระหว่างผู้ขอใช้บริการ (Service request) กับผู้ให้บริการ (Service Provider) โดยจัดส่งข้อมูลในรูปแบบ message

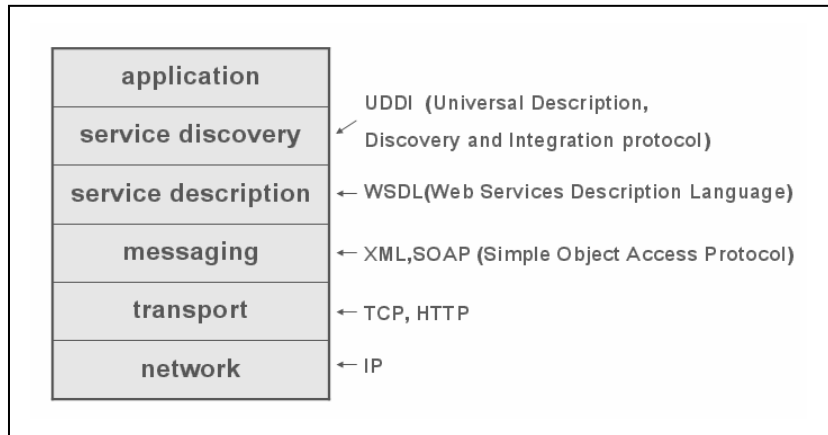
2.4 Exchange sequence definition : ในการแลกเปลี่ยนข่าวสารผู้ขอใช้ บริการ (Service request) กับผู้ให้บริการ (Service Provider) จะมีการจัดลำดับการ แลกเปลี่ยนข้อมูลผ่านทางโปรโตคอลโดยใช้ message ในการแลกเปลี่ยนข้อมูลข่าวสาร ด้วยกา เพิ่มความน่าเชื่อถือในการส่งข้อมูลในแต่ละครั้งควรรู้ Time out และเทคนิคอื่นๆ ในการส่ง ข้อมูล

2.5 Process definition : การดำเนินการพื้นฐานบน message ที่ทำการ เปลี่ยนข้อมูลระหว่างผู้ขอใช้บริการ (Service request) กับผู้ให้บริการ (Service Provider)

2.6 Security : การเพิ่มความปลอดภัยให้กับ message ที่ใช้ในการแลกเปลี่ยนข้อมูลโดยใช้หลักการ Encrypton

2.7 Syntax : เอกสารที่เป็นตัวแทนในการแลกเปลี่ยนข้อมูล ใช้โครงสร้างภาษา XML

2.8 Trading partner Specific Configuration : องค์การทางธุรกิจที่มีส่วนร่วมในการทำงานของ Web service



ภาพที่ 25 โพรโตคอลที่ใช้งานบนเว็บเซอร์วิส

ที่มา : ศิริศักดิ์ เสนาราช. (2553 : 27)

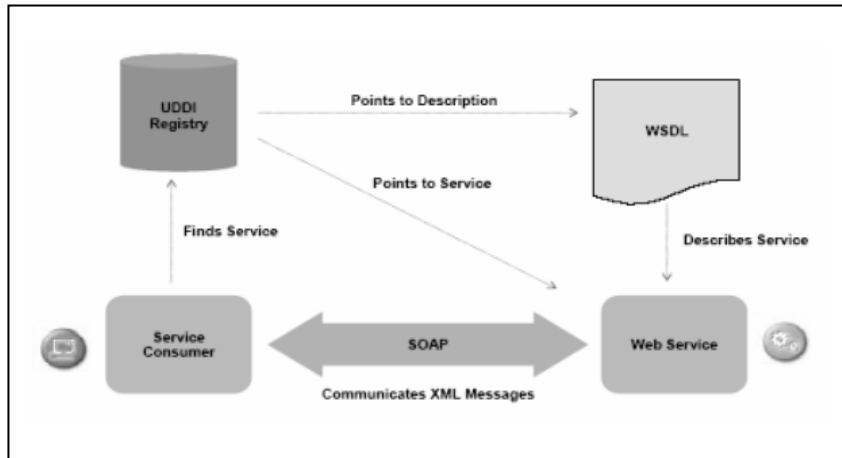
จากภาพที่ 25 มีรายละเอียดดังต่อไปนี้

1. Application โปรแกรมที่เรียกใช้เว็บเซอร์วิส
2. Service Discovery บริการค้นหาเว็บเซอร์วิสที่เปิดให้บริการ (UDDI)
3. Service Description เอกสารที่ระบุการทำงานของเว็บเซอร์วิส (WSDL)
4. Massaging การส่งข้อมูลระหว่างเว็บเซอร์วิส (SOAP) โดยใช้ไวยากรณ์ทางภาษา

XML

5. Transport วิธีการส่งข่าวสารผ่านทางโปรโตคอล TCP และ HTTP

6. Network เครือข่ายที่ใช้ในการรับส่งข้อมูลระหว่างเว็บเซอร์วิส ได้แก่ เครือข่ายอินเทอร์เน็ต โดยเครื่องที่ทำงานในระบบจะมีการกำหนดหมายเลข IP (Internet Protocol) ประจำเครื่องการ



ภาพที่ 26 การทำงานของเว็บเซอร์วิส

ที่มา : ศิริศักดิ์ เสนาราช. (2553 : 27)

เปรียบเทียบเทคโนโลยีระหว่างเว็บแอปพลิเคชันและเว็บเซอร์วิส การทำงานของเว็บแอปพลิเคชันและเว็บเซอร์วิส จะเห็นว่าเครื่องมือทั้งสองต่างใช้ HTTP โพรโทคอล หรือ อินเทอร์เน็ต เป็นช่องทางในการสื่อสารเหมือนกัน แต่มีวัตถุประสงค์ต่างกัน โดยเว็บแอปพลิเคชันใช้เพื่อการแลกเปลี่ยนไฟล์ HTML ระหว่างเว็บเซิร์ฟเวอร์ แต่เว็บเซอร์วิสเป็นการแลกเปลี่ยน “บริการ” (Software Components) ระหว่างระบบสารสนเทศผ่านเว็บเซิร์ฟเวอร์ ความสามารถโดยส่วนใหญ่จะใช้เว็บแอปพลิเคชันในการติดต่อกับผู้ใช้ผ่านทางเว็บเบราว์เซอร์ (Web Browser) เพื่อนำเสนอข้อมูลและการทำธุรกรรมต่างๆ ส่วนเซอร์วิสจะทำหน้าที่ในการติดต่อกับเว็บเซิร์ฟเวอร์เพื่อแลกเปลี่ยนข้อมูลและการทำงานหรือใช้บริการข้ามระบบกันโดยใช้เว็บแอปพลิเคชัน หรือแอปพลิเคชันอินเทอร์เฟซ (Application Interface) ในการติดต่อกับผู้ใช้ นอกจากนี้เว็บเซอร์วิสยังสามารถทำงานกับระบบต่างๆ ได้มากกว่า 1 ระบบ ในขณะที่เว็บแอปพลิเคชันไม่สามารถทำได้โดยตรง ซึ่งการเปรียบเทียบการทำงานของเว็บแอปพลิเคชันและเว็บเซอร์วิสสามารถสรุปได้ดังตารางต่อไปนี้

ตารางที่ 2 ตารางเปรียบเทียบความแตกต่างระหว่างเว็บแอปพลิเคชันและเว็บเซอร์วิส

หัวข้อเปรียบเทียบ	Web Service	Web Application
การเชื่อมต่อ	Program-Program	Human-Program
ภาษาที่ใช้	XML	HTML
รายชื่อการให้บริการ	ค้นหาผ่าน UDDI	ค้นหาผ่าน Search Engine
ขอบเขตการงาน	Business to Business(B2B)	Business to Customer(B2C)
โปรโตคอล(Protocol)	SOAP+HTTP	HTTP

ที่มา : จันทร์ศิริ จันทะเน (2553:28)

จากตารางที่ 2 เมื่อพิจารณาจากหัวข้อเปรียบเทียบ ได้แก่ การเชื่อมต่อเว็บเซอร์วิสทำการติดต่อแลกเปลี่ยนข้อมูลระหว่างโปรแกรมกับโปรแกรมผู้ใช้สามารถใช้บริการจากแหล่งอื่นได้ ในขณะที่ เว็บแอปพลิเคชันเชื่อมต่อกับผู้ใช้ผ่านทางบราวเซอร์ผู้ใช้สามารถใช้บริการได้จากแหล่งข้อมูลภายในเว็บแอปพลิเคชันที่เข้าใช้งานเท่านั้น ภาษาที่ใช้เว็บเซอร์วิสใช้ภาษาเอ็กซ์เอ็มแอลซึ่งข้อมูลมากกว่าเว็บแอปพลิเคชันที่ใช้ภาษาแสดงผลอย่าง HTML รายชื่อการบริการเว็บเซอร์วิสสามารถสืบค้นบริการผ่าน UDDI ในขณะที่เว็บแอปพลิเคชันค้นหาข้อมูลผ่าน Search Engine ขอบเขตการใช้งานเว็บเซอร์วิสจะกว้างกว่าโดยใช้งานในเชิงพาณิชย์ แลกเปลี่ยนข้อมูลระหว่างองค์กรกับองค์กรซึ่งเป็นลักษณะ Business-to-Business มากกว่าเว็บแอปพลิเคชันที่ให้บริการในลักษณะเฉพาะองค์กรกับลูกค้า Business-to-Customer โพรโตคอลที่ใช้งานเว็บเซอร์วิสจึงมีความซับซ้อนกว่าโดยมีการใช้โปรโตคอล SOAP บนโปรโตคอล HTTP ที่อยู่ชั้นบน ในขณะที่เว็บแอปพลิเคชันมีการส่งด้วยโปรโตคอล HTTP อย่างเดียว

3. ความรู้เรื่อง REST (Representational State Transfer)

Representational State Transfer (REST) นั้นถูกพูดถึงครั้งแรกในปี 2000 โดย Roy Thomas Fielding ซึ่ง Representational State Transfer (REST) นั้นเป็น Architecture (สถาปัตยกรรมการสื่อสารข้อมูล) รูปแบบหนึ่งที่ใช้แพร่กระจายสื่อ เป็นแนวทางใหม่ในการสร้าง Web Service แบบเรียบง่าย โดยเรียกใช้ผ่านทาง HTTP Method GET / POST / PUT / DELETE และส่งข้อมูลออกมาในรูปของ XML ทำให้ปริมาณข้อมูลที่รับส่ง น้อยกว่าการใช้ Protocol SOAP อยู่มาก ซึ่งข้อดีข้อนี้ของ REST ทำให้ Developer หลายคนมาสนใจการเขียนโปรแกรมแบบใช้ RESTful Web Service กันมากขึ้น เพราะมีผลกับเรื่อง Performance ของการใช้งานโปรแกรมเป็นอย่างมาก แต่เนื่องจากเรื่อง REST นี้เพิ่งเกิดขึ้นมาเมื่อปี 2000 ทำให้ยังไม่มีมาตรฐานที่กำหนดให้บังคับใช้งานเหมือน Protocol SOAP เดิม ซึ่งจะสังเกตได้ว่าจะไม่มี REST Specification อยู่บน W3C และไม่มี REST Developer Toolkit ถ้าหากอยากใช้ RESTful ก็ต้องกำหนดเงื่อนไขจากภาษาที่เขียนเอาเอง โดยภาษาในยุคปัจจุบันนี้มีการรองรับ RESTful Web Service หรืออีกชื่อหนึ่งคือ RESTful Web API กันเป็นส่วนใหญ่แล้ว

3.1 พื้นฐานการทำงานของ REST

การใช้งาน resource ต่างๆ ผ่าน HTTP protocol ผ่าน URI ที่สื่อสารได้ชัดเจน ผ่าน HTTP method ที่เหมาะสม เช่น

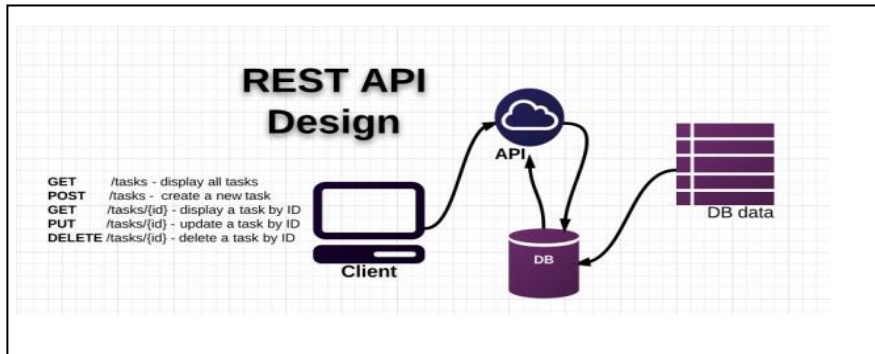
HTTP GET สำหรับการดึงข้อมูล

HTTP POST สำหรับการสร้างข้อมูล

HTTP PUT สำหรับการแก้ไขข้อมูล

HTTP DELETE สำหรับลบข้อมูล

รูปแบบของข้อมูลที่ใช้สื่อสารกันอยู่ในรูปแบบ XML, JSON, Atom และอื่นๆ อีกมากมาย แต่หนึ่งในรูปแบบที่ได้รับความนิยมคือ JSON



ภาพที่ 27 การใช้งาน resource ต่างๆ

การประเมินเครื่องมือในการศึกษา

1. การหาคุณภาพเครื่องมือความเที่ยงตรงตามเนื้อหา

พิสุทธา อารีราษฎร์ (2551 : 119-121) ได้กล่าวว่า ความเที่ยงตรงตามเนื้อหา (Content Validity) หมายถึง การที่ผู้สอนออกแบบทดสอบได้ตรงกับเนื้อหาที่สอน ในการทดสอบความเที่ยงตรงตามเนื้อหา สามารถดำเนินการได้โดยใช้ผู้เชี่ยวชาญในด้านเนื้อหา พิจารณาถึงความสอดคล้องระหว่างวัตถุประสงค์กับแบบทดสอบ โดยพิจารณาเป็นรายชื่อวิธีการพิจารณาแบบนี้เรียกว่า การหาค่าสัมประสิทธิ์ความสอดคล้อง (Index of Item – Congruence : IOC) หาค่าความเที่ยงตรงเชิงเนื้อหาของแบบทดสอบวัดผลสัมฤทธิ์ทางการเรียน และแบบสอบวัดเจตคติต่อการเรียน

ไพศาล วรคำ (2554 : 260-262) ได้กล่าวว่า เครื่องมือในการเก็บรวบรวมข้อมูลสำหรับการวิจัยทุกประเภทไม่ว่าจะเป็นแบบทดสอบ แบบสอบถาม แบบสังเกต แบบสัมภาษณ์ สามารถหาความเที่ยงตรงเชิงเนื้อหาได้เช่นเดียวกัน การตรวจสอบความเที่ยงตรงเชิงเนื้อหานี้จะต้องดำเนินการก่อนไปทดลองใช้ โดยการนำนิยามเชิงทฤษฎี นิยามเชิงปฏิบัติการ โครงสร้างของข้อคำถาม (รวมทั้งคำตอบ สำหรับกรณีที่กำหนดคำตอบให้ผู้ตอบเลือก) ให้ผู้เชี่ยวชาญพิจารณาความสอดคล้องระหว่างข้อคำถามกับประเด็นที่ต้องการวัด วัตถุประสงค์หรือนิยามศัพท์ในแบบฟอร์มรายงานผลการตรวจสอบเครื่องมือ

สำหรับผู้เชี่ยวชาญที่จะให้ทำการตรวจสอบความสอดคล้องควรมีตั้งแต่ 3 คนขึ้นไป เพื่อหลีกเลี่ยงความคิดเห็นที่แบ่งเป็น 2 ด้าน ดังนั้นควรใช้จำนวนผู้เชี่ยวชาญเป็นจำนวนคี่ เช่น 3 คน 5 คน 7 คน เป็นต้น ส่วนคุณสมบัติของผู้เชี่ยวชาญจะต้องสอดคล้องกับสาขาวิชาของเครื่องมือที่ต้องการตรวจสอบ เช่น ถ้าเป็นการวัดตัวแปรทางจิตวิทยาก็ควรมีผู้เชี่ยวชาญทางด้านจิตวิทยา ผู้เชี่ยวชาญด้านการวัดและประเมินผล และผู้เชี่ยวชาญทางด้านภาษา เป็นต้น ซึ่งจะสามารถประเมินความสอดคล้องและให้ข้อเสนอแนะเกี่ยวกับเครื่องมือครบทั้งในส่วน of เนื้อหา ลักษณะข้อคำถามและความเหมาะสมของภาษาที่ใช้ นำผลการตรวจสอบของผู้เชี่ยวชาญมาคำนวณหาดัชนีที่บ่งบอกถึงความเที่ยงตรงเชิงเนื้อหา ซึ่งคำนวณได้จากความ

สอดคล้องระหว่างประเด็นที่ต้องการวัดกับข้อความที่สร้างขึ้น ดัชนีนี้เรียกว่า ดัชนีความสอดคล้องระหว่างข้อความกับวัตถุประสงค์ (Index of Item –Congruence : IOC)

สรุป วิธีการหาสัมประสิทธิ์ความสอดคล้อง (Index of Item –Congruence : IOC) สามารถใช้ได้กับเครื่องมือในการเก็บรวบรวมข้อมูลสำหรับการวิจัยทุกประเภท ไม่ว่าจะเป็นแบบทดสอบ แบบสอบถาม แบบสังเกต แบบสัมภาษณ์ ในการศึกษาครั้งนี้ ผู้ศึกษาได้นำวิธีการหาสัมประสิทธิ์ความสอดคล้อง (Index of Item –Congruence : IOC) ในการหาดัชนีความสอดคล้องระหว่างเนื้อหา กับแบบประเมินและแบบสอบถามที่พัฒนาขึ้น โดยให้ผู้เชี่ยวชาญจำนวน 3 คน ประเมินดัชนีความสอดคล้อง (IOC) เพื่อตรวจสอบหาคุณภาพเครื่องมือที่ใช้ในการประเมิน ก่อนทำการจัดทำเป็นเครื่องมือฉบับสมบูรณ์เพื่อใช้ในการเก็บรวบรวมข้อมูลในการวิจัยต่อไป

2. การประเมินผลบทเรียนคอมพิวเตอร์ตามแนวทางวิศวกรรมซอฟต์แวร์

มนต์ชัย เทียนทอง (2554 : 283) ได้กล่าวว่า การประเมินผลบทเรียนคอมพิวเตอร์ตามแนวทางวิศวกรรมซอฟต์แวร์ มีวิธีการทดสอบ ดังต่อไปนี้

- 2.1 White box Testing
- 2.2 Black box Testing
- 2.3 Integration Testing
- 2.4 Performance Testing
- 2.5 Usability Testing

Black box Testing แปลว่า การทดสอบแบบกล่องดำ เปรียบเสมือนการทดสอบภายนอกกล่อง โดยพิจารณาเฉพาะส่วนของการนำเข้าและส่วนของการแสดงผลไม่พิจารณาภายในกล่องอันได้แก่ โครงสร้างของโปรแกรมและรหัสของโปรแกรม ซึ่งเป็นการประเมินผลตรงกันข้ามกับ White box Testing เมื่อนำไปประเมินผลบทเรียนคอมพิวเตอร์ จึงมุ่งเน้นเฉพาะผลลัพธ์ที่ได้จากบทเรียนและส่วนของการนำเข้าเท่านั้น ไม่ได้ให้ความสำคัญกับส่วนประกอบภายในตัวบทเรียนแต่อย่างใด วิธี Black box Testing จึงใช้พิจารณาด้านการทำงาน (Functionality) ตลอดจนคุณสมบัติของบทเรียน (Behavioral) เป็นหลัก ประเด็นทั้งสองนี้จะเกี่ยวข้องกับผู้ออกแบบบทเรียน ผู้เชี่ยวชาญ ผู้สอน และผู้ใช้บทเรียนทั่วไป ซึ่งจะเป็นผู้ประเมินผลบทเรียนหลังจากได้ศึกษาบทเรียนแล้ว ไม่จำเป็นต้องใช้โปรแกรมเมอร์ให้เป็นผู้ประเมินแต่อย่างใด ในการประเมินแบบ White box Testing และ Black box Testing จะใช้แบบสอบถามแบบมาตราส่วนประเมินค่า เพื่อสอบถามความคิดเห็นในประเด็นต่างๆตามที่กำหนดไว้

สรุปในการศึกษานี้ ผู้ศึกษาได้ใช้วิธีการประเมินแบบ Black box Testing หรือการทำแบบทดสอบแบบกล่องดำ โดยนำมาใช้พิจารณาทางด้านหน้าที่การทำงาน (Functionality) ตลอดจนคุณสมบัติของบทเรียน (Behavioral) เป็นหลัก ประเด็นที่ใช้ในการประเมินประกอบด้วย การประเมินฟังก์ชันการทำงานของระบบ (Functional Test) ด้านการใช้งานของโปรแกรม

(Usability Test) ด้านผลลัพธ์ที่ได้จากโปรแกรม (Result Test)ด้านความปลอดภัย (Security Test) และด้านคู่มือการใช้งานระบบ (Documentation)

3. การประเมินความพึงพอใจ

3.1 ความหมายของความพึงพอใจ

ความพึงพอใจ หมายถึง พอใจ ชอบใจ (ราชบัณฑิตยสถาน. 2542 : 775) นอกจากนี้ นักการศึกษาได้ให้ความหมายของความพึงพอใจไว้ดังนี้

ศุภสิริ โสมาเกต (2544 : 9) ได้กล่าวว่า ความพึงพอใจ หมายถึง ความรู้สึกนึกคิด หรือเจตคติของบุคคลที่มีต่อการทำงาน หรือการปฏิบัติกิจกรรมในเชิงบวก ศิริพรรณ ชุตันตานนท์ (2545 : 32) ได้กล่าวว่า ความพึงพอใจ หมายถึง การแสดงออกถึง ความรู้สึกชอบ มีความสุข ที่เกิดขึ้นเมื่อบุคคลได้รับการสนองความต้องการหรือได้รับผลสำเร็จตามความมุ่งหมายที่ตนเองได้ตั้งไว้

พิสุทธา อารีราษฎร์ (2551 : 174) ได้กล่าวว่า ความพึงพอใจ (Satisfaction)

หมายถึง ความรู้สึกของบุคคลที่มีต่อสิ่งหนึ่งสิ่งใดโดยเฉพาะ ความรู้สึกนั้นทำให้บุคคลเอาใจใส่และอาจกระทำการบรรลุถึงความมุ่งหมายที่บุคคลมีต่อสิ่งนั้น ซึ่งพอสรุปได้ว่า ความพึงพอใจ เป็นเรื่องของความรู้สึก ทักษะคิด หรือระดับความพึงพอใจที่มีต่อสิ่งนั้น สามารถตอบสนองความต้องการหรือทำให้บรรลุจุดมุ่งหมายนั้น ๆ ได้

มนต์ชัย เทียนทอง (2554 : 306) ได้กล่าวว่า ความพึงพอใจ หมายถึง ความชื่นชม ความนิยม หรือความรู้สึกยอมรับในสิ่งที่ได้เห็นหรือได้สัมผัส สำหรับความพึงพอใจของผู้เรียน จะเป็นการสอบถามความคิดเห็นของผู้เรียนหลังจากการทดลองใช้บทเรียน ซึ่งเป็นการประเมินผลในภาพรวมว่าผู้เรียนพึงพอใจหรือไม่

สรุป ความหมายของความพึงพอใจ หมายถึง สภาวะของอารมณ์ความรู้สึกของคนที่มีความพอใจที่ได้พบเห็น หรือได้รับสิ่งที่ต้องการให้เกิดขึ้น เช่นชอบหรือไม่ชอบ ซึ่งเกิดมาจากความสนใจและเจตคติของแต่ละบุคคลที่อาจไม่เท่ากัน ขึ้นอยู่กับการได้รับการตอบสนองในสิ่งที่สงสัยอยากรู้ หรือบรรลุในจุดมุ่งหมายที่ตนตั้งเอาไว้หรือไม่ มากน้อยเพียงใด

3.2 การวัดความพึงพอใจ

ซึ่งนักวิชาการหลายคนได้กล่าวถึง การวัดความพึงพอใจไว้สรุปได้ดังนี้

โยธิน คันสนนุทธ (2530 : 77-86) ได้กล่าวถึงเครื่องมือวัดความพึงพอใจว่าการจะค้นหาว่าบุคคลมีความพึงพอใจหรือไม่ วิธีที่ง่ายที่สุดก็คือการถาม ซึ่งการศึกษาในระยะหลัง ๆ ที่ต้องมีผู้บอกข้อมูลจำนวนมาก ๆ มักใช้แบบสอบถามที่ใช้มาตราส่วนประมาณค่าตามแบบของลิเคิร์ต (Likert) ประกอบด้วยชุดของคำถาม และมีตัวเลือก 5 ตัว สำหรับเลือกตอบคือมากที่สุด มากปานกลาง น้อย น้อยที่สุด และคะแนนความพึงพอใจนั้นสามารถนำมาวิเคราะห์ได้ว่า บุคคลมีความพึงพอใจในด้านใดสูง และด้านใดต่ำโดยใช้วิธีการทางสถิติ

พิสุทธา อารีราษฎร์ (2550 : 178) ได้กล่าวว่า การวัด หรือประเมิน ประสิทธิภาพบทเรียนคอมพิวเตอร์ช่วยสอน ด้านความพึงพอใจในการใช้บทเรียนคอมพิวเตอร์ช่วยสอน โดยอาจจะเป็นผู้สอนหรือนักเรียน ถือเป็นวิธีการหนึ่งในการวัดประสิทธิภาพของบทเรียนคอมพิวเตอร์ช่วยสอน ถ้าผู้ใช้มีความพึงพอใจต่อบทเรียนคอมพิวเตอร์ช่วยสอน จะส่งผลให้นักเรียนยอมรับและตอบสนองการเรียนด้วยความเต็มใจ โดยการสนใจในการเรียน หรือ การเข้าร่วมกิจกรรม ซึ่งมีผลทำให้นักเรียนมีผลการเรียนที่ดีขึ้น ในการวัดความพึงพอใจ จะใช้แบบสอบถามวัดทัศนคติ ตามมาตรวัดของลิเคิร์ต (Likert Scales) ซึ่งแบ่งความรู้สึกเป็น 5 ช่วง หรือ 5 ระดับ ดังนี้

- ระดับ 5 หมายถึง มีความพึงพอใจมากที่สุด
- ระดับ 4 หมายถึง มีความพึงพอใจมาก
- ระดับ 3 หมายถึง มีความพึงพอใจปานกลาง
- ระดับ 2 หมายถึง มีความพึงพอใจน้อย
- ระดับ 1 หมายถึง มีความพึงพอใจน้อยที่สุด

สรุปได้ว่า การวัดความพึงพอใจ เป็นการตรวจสอบความชอบหรือไม่ชอบของบุคคลที่มีต่อสิ่งหนึ่งสิ่งใดตามลำดับขั้นของความพึงพอใจ ซึ่งขึ้นอยู่กับว่าสิ่งนั้นมีสำคัญอยู่ในระดับใด เช่น มาก ปานกลาง หรือน้อยเพียงใด

งานวิจัยที่เกี่ยวข้อง

พิทวรรณ หล่อสุวรรณรัตน์ (2546, น.16) ได้ให้ความหมายไว้เป็น 2 นัย คือ เทคโนโลยีสารสนเทศในความหมายแบบแคบ หมายถึง อุปกรณ์หรือเครื่องมือที่เกี่ยวข้องกับการรวบรวม ประมวลผล เก็บรักษา และเผยแพร่ข้อมูลสารสนเทศ โดยรวมถึงฮาร์ดแวร์ (Hardware) ซอฟต์แวร์ (Software) ฐานข้อมูล และการสื่อสาร โทรคมนาคม ส่วนเทคโนโลยีสารสนเทศในความหมายแบบกว้างหมายถึง ข้อมูลที่ได้ผ่านการประมวลผลหรือจัดระบบแล้ว เพื่อให้มีความหมายและคุณค่าสำหรับผู้ใช้งาน ซึ่งโดยทั่วไปจะนิยมใช้ความหมายแบบแคบโดยพิจารณาว่าเทคโนโลยีสารสนเทศเป็นส่วนประกอบหนึ่งของระบบสารสนเทศ

สุชาติ กิระนันท์ (2543, น.23 – 24) ข้อมูลและสารสนเทศ (Data and Information) หมายถึง ข้อมูลที่เกี่ยวข้องกับงานจะถูกเก็บรวบรวมมาประมวลผล เพื่อให้ได้สารสนเทศที่เป็นประโยชน์ต่อผู้ใช้ซึ่งจะนำเอาระบบคอมพิวเตอร์มาเป็นเครื่องมือที่ใช้แปลงข้อมูลเป็นสารสนเทศได้อย่างมีประสิทธิภาพสารสนเทศเป็นสิ่งที่ช่วยในการตัดสินใจของผู้บริหาร โดยสารสนเทศที่มีประโยชน์จะมีคุณสมบัติคือ 1) มีความสัมพันธ์กัน (Relevant) สามารถนำมาประยุกต์ใช้ได้อย่างเหมาะสมกับสถานการณ์ปัจจุบัน 2) มีความทันสมัย (Timely) และพร้อมที่จะใช้งานได้ทันทีเมื่อต้องการ 3) ถูกต้องแม่นยำ (Accurate) เมื่อป้อนข้อมูลเข้าสู่คอมพิวเตอร์ผลลัพธ์ที่ได้ต้องถูกต้องในทุกส่วน 4) กระชับรัดกุม (Concise) มีความยาวที่เหมาะสม 5) มีความสมบูรณ์ครบถ้วนในตัวเอง (Complete)

เคนเนท เลาดอน และ จินส์ เลาดอน (2545, น.6) ระบบสารสนเทศ (Information System) หมายถึงกลุ่มของระบบงานที่ประกอบด้วยฮาร์ดแวร์และซอฟต์แวร์ ที่ทำหน้าที่รวบรวม ประมวลผล จัดเก็บและแจกจ่ายข้อมูลข่าวสาร เพื่อสนับสนุนการตัดสินใจและการควบคุมภายใน นอกจากนี้ยังช่วยบุคลากรในองค์กรในการประสานงาน การวิเคราะห์ปัญหา และการสร้างแบบจำลองวัตถุที่มีความซับซ้อนกระบวนการทำงานของระบบสารสนเทศประกอบด้วย 3 กิจกรรม คือการนำข้อมูลเข้าสู่ระบบ (Input) การประมวลผล (Processing) และการนำเสนอผลลัพธ์ (Output) โดยการนำข้อมูลเข้าสู่ระบบเป็นการจัดการรวบรวมข้อมูลจากส่วนต่าง ๆ ขององค์กรหรือจากสิ่งแวดล้อมภายนอกองค์กรเข้าสู่ระบบ ส่วนการประมวลผลทำหน้าที่ปรับเปลี่ยนข้อมูลที่นำเข้ามาให้อยู่ในรูปแบบที่มีความหมายต่อองค์กร ซึ่งสามารถนำไปใช้งานได้ และการนำเสนอผลลัพธ์เป็นการจัดการนำสารสนเทศหรือข้อมูลที่ผ่านการประมวลผลแล้วไปเสนอให้แก่ผู้ใช้ตามความเหมาะสมหรือนำไปส่งต่อให้กับส่วนอื่นที่ต้องนำสารสนเทศนี้ไปใช้งานต่อไป ระบบสารสนเทศบางระบบต้องการการตอบสนอง (Feedback) ซึ่งก็คือส่วนหนึ่งของข้อมูลที่ผ่านการประมวลผลแล้ว แต่ถูกส่งกลับไปยังส่วนการนำข้อมูลเข้าอีกครั้งหนึ่งเพื่อการตรวจสอบคุณภาพหรือปรับปรุงระบบให้มีความสอดคล้องกับสภาวะที่ต้องการ

ศดานันท์ ศรีชัยปัญหา (2558)ทำการวิจัย เรื่อง การพัฒนาระบบจ่ายเงินค่ามาปนกิจสงเคราะห์ผ่านธนาคารโดยใช้เทคโนโลยีเว็บเซอร์วิส กรณีศึกษาสมาคมมาปนกิจสงเคราะห์ธนาคารเพื่อการเกษตรและสหกรณ์การเกษตร อำเภอโกสุมพิสัย จังหวัดมหาสารคาม การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อ 1) การพัฒนาระบบจ่ายเงินค่ามาปนกิจสงเคราะห์ผ่านธนาคารโดยใช้เทคโนโลยีเว็บเซอร์วิส กรณีศึกษาสมาคมมาปนกิจสงเคราะห์ธนาคารเพื่อการเกษตรและสหกรณ์การเกษตร อำเภอโกสุมพิสัย จังหวัดมหาสารคาม 2) เพื่อประเมินคุณภาพของการพัฒนาระบบจ่ายเงินค่ามาปนกิจสงเคราะห์ผ่านธนาคารโดยใช้เทคโนโลยีเว็บเซอร์วิส กรณีศึกษาสมาคมมาปนกิจสงเคราะห์ธนาคารเพื่อการเกษตรและสหกรณ์การเกษตร อำเภอโกสุมพิสัย จังหวัดมหาสารคาม กลุ่มเป้าหมายคือ ผู้เชี่ยวชาญด้านคอมพิวเตอร์และเทคโนโลยีสารสนเทศ เครื่องมือในการศึกษาได้แก่ ระบบจ่ายเงินค่ามาปนกิจสงเคราะห์ผ่านธนาคารโดยใช้เทคโนโลยีเว็บเซอร์วิส กรณีศึกษาสมาคมมาปนกิจสงเคราะห์ธนาคารเพื่อการเกษตรและสหกรณ์การเกษตร อำเภอโกสุมพิสัย จังหวัดมหาสารคาม แบบประเมินคุณภาพระบบจ่ายค่ามาปนกิจสงเคราะห์ผ่านธนาคารโดยใช้เทคโนโลยีเว็บเซอร์วิส กรณีศึกษาสมาคมมาปนกิจสงเคราะห์ธนาคารเพื่อการเกษตรและสหกรณ์การเกษตร อำเภอโกสุมพิสัย จังหวัดมหาสารคาม สถิติที่ใช้ในการวิเคราะห์ข้อมูล คือ ค่าเฉลี่ยเลขคณิต \bar{X} และ ส่วนเบี่ยงเบนมาตรฐาน S.D.

วัชระ ไสธฤทธิ (2558) ทำการวิจัย เรื่อง การพัฒนาระบบอนุมัติการทำประกันชีวิตโดยใช้เทคโนโลยีเว็บเซอร์วิส การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อ 1) เพื่อพัฒนาระบบอนุมัติการทำประกันชีวิตโดยใช้เทคโนโลยีเว็บเซอร์วิส 2) เพื่อหาคุณภาพของระบบอนุมัติการทำประกันชีวิตโดยใช้เทคโนโลยีเว็บเซอร์วิส กลุ่มเป้าหมาย คือ กลุ่มผู้เชี่ยวชาญเพื่อประเมินคุณภาพของระบบอนุมัติการทำประกันชีวิตโดยใช้เทคโนโลยีเว็บเซอร์วิส เครื่องมือในการศึกษาได้แก่ ระบบอนุมัติการทำประกันชีวิต โดยใช้เทคโนโลยีเว็บเซอร์วิส และแบบประเมินคุณภาพระบบอนุมัติการทำ

ประกันชีวิต โดยใช้เทคโนโลยีเว็บเซอร์วิส สถิติที่ใช้ในการวิเคราะห์ข้อมูล คือ ค่าเฉลี่ยเลขคณิต \bar{x} และส่วนเบี่ยงเบนมาตรฐาน S.D. ซึ่งผลการวิจัยพบว่า 1) ระบบอนุมัติการทำประกันชีวิต โดยใช้เทคโนโลยีเว็บเซอร์วิส สามารถใช้งานได้อย่างมีคุณภาพ 2) ผลการประเมินคุณภาพจากผู้เชี่ยวชาญ 3 ท่าน มีผลการประเมินอยู่ในระดับดีมาก ($\bar{x} = 4.76$, S.D. = 0.15) สามารถสรุปผลการศึกษารพัฒนาระบบอนุมัติการทำประกันชีวิต โดยใช้เทคโนโลยีเว็บเซอร์วิสมีคุณภาพสามารถทำงานได้ตามขอบเขตระบบงานที่ได้ทำ การวิเคราะห์มาบรรลุดตามจุดประสงค์ของการศึกษาที่ตั้งไว้

อรรรถพล จันดา (2558) ทำการวิจัย เรื่อง การพัฒนาระบบการอนุมัติบัตรเครดิต ธนาคารพาณิชย์ โดยใช้เทคโนโลยีเว็บเซอร์วิส (Web Service) การศึกษาครั้งนี้มีวัตถุประสงค์เพื่อ 1) เพื่อพัฒนาระบบการอนุมัติบัตรเครดิต ธนาคารพาณิชย์ โดยใช้เทคโนโลยีเว็บเซอร์วิส (Web Service) 2) เพื่อหาคุณภาพของการพัฒนาระบบการอนุมัติบัตรเครดิต ธนาคารพาณิชย์ โดยใช้เทคโนโลยีเว็บเซอร์วิส (Web Service) ที่พัฒนาขึ้น กลุ่มเป้าหมาย คือ กลุ่มผู้เชี่ยวชาญเพื่อประเมินคุณภาพของระบบอนุมัติการทำประกันชีวิตโดยใช้เทคโนโลยีเว็บเซอร์วิสได้แก่ อาจารย์คณะเทคโนโลยีสารสนเทศ ที่มีความเชี่ยวชาญทางด้านคอมพิวเตอร์ จำนวน 3 คน เครื่องมือในการศึกษาได้แก่ ระบบการอนุมัติบัตรเครดิต ธนาคารพาณิชย์ โดยใช้เทคโนโลยีเว็บเซอร์วิส (Web Service) แบบประเมินคุณภาพระบบการอนุมัติบัตรเครดิต ธนาคารพาณิชย์ โดยใช้เทคโนโลยีเว็บเซอร์วิส (Web Service) สถิติที่ใช้ในการวิเคราะห์ข้อมูล คือ ค่าเฉลี่ยเลขคณิต \bar{x} และส่วนเบี่ยงเบนมาตรฐาน S.D.