บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในโครงงาน

บทนำ

โครงงานนี้เป็นการสร้างโปรแกรมระบบโมบายล์คอมเมิร์ซสำหรับ Pocket PC โดยอาศัยระบบโลเคชันเบสด้วยเทคโนโลยีด็อทเน็ทคอมแพคเฟรมเวิร์ก ซึ่งเป็นการพัฒนาApplication บน Pocket Pc ติดต่อกับ Web Services โดยที่ตัวPocket PC ต้อง Support GPS แล้วใช้ GPS Receiver ตรวจสอบพิกัดละติจูดและ ลองจิจูด โดยใช้โปรโตคอล NMEA แล้วส่งค่าพิกัดดังกล่าวไปให้เว็บเซอร์วิสแล้วให้เว็บเซอร์วิสดึงพิกัด มา Map กับแผนที่จาก Thaimapguide และเรียกข้อมูลบริการจากเว็บเซอร์วิส มาแสดงผลบนApplication บน Pocket Pc และแสดงผลบนเว็บ Application ผ่าน Browser โดยใช้เทคโนโลยีASP.NET

2.1 ทฤษฎี .NET เทคโนโลยี

.NET (อ่านว่า ด็อทเน็ท) คือแนวคิดหนึ่งที่ไมโครซอฟท์ภูมิใจนำเสนอ โดย .NET ตัวนี้ไม่ได้เกี่ยวข้องกับโดเมนเนมของเว็บใดๆทั้งสิ้น แต่ .NET ตัวนี้ หมายถึง การนำเอาอุปกรณ์ทุกอย่างบนโลกมาเชื่อมโยงต่อกันเหมือนตาข่าย (net = ตาข่าย) ซึ่งหากว่าทำสำเร็จแล้วไม่ต้องนึกเลยว่าไมโครซอฟท์จะได้เป็นเจ้าแห่งเทคโนโลยีอย่างไม่ต้องสงสัย แต่เรื่องมันไม่ง่ายอย่างที่คิด เพราะอุปกรณ์ต่างๆเหล่านั้นล้วนถูกออกแบบมาต่างๆกัน การที่มันจะติดต่อสื่อสารกันรู้เรื่องนั้น ย่อมเป็นเรื่องที่เป็นไปได้ยาก ไมโครซอฟท์เล็งเห็นจุดนี้ จึงได้พยายามที่จะคิดค้นสิ่ง ที่เป็นมาตรฐานขึ้น เพื่อให้อุปกรณ์ทุกๆชนิดทั่วโลกติดต่อสื่อสารกันได้อย่างรู้เรื่อง จึงไม่ใช่เรื่องแปลกถ้าในอนาคตเราเปิดเว็ปไซต์เล่นอินเตอร์เน็ตด้วยอุปกรณ์อื่นๆนอกเหนือจากคอมพิวเตอร์

อย่างที่ได้กล่าวไปแล้วในตอนต้นว่าไมโครซอฟท์ต้องการที่จะสร้างอะไรที่เป็นมาตรฐานขึ้นมา เพื่อให้ทุกสิ่งทุกอย่าง เพื่อให้ทุกสิ่งทุกอย่างสามารถติดต่อสื่อสารกันได้หมด โดยคิดค้นระบบซึ่งหมายมั่นปั้นมือว่าจะให้เป็นระบบมาตรฐาน ระบบนี้คือ .NET Framework

2.1.1 ด็อทเน็ทเฟรมเวิร์ก (.NET Framework)

.Net Framework คือ โครงร่างการพัฒนาโปรแกรมคอมพิวเตอร์ ที่ถูกออกแบบมาเพื่อให้อำนวยความสะดวกในการพัฒนาโปรแกรมสมัยใหม่ ที่ใช้งานในระบบเครือข่าย (Internet,

Intranet, Mobile Devices, ฯลฯ) Bill Gates และ Steve Ballmer ได้บรรยายสรุปวิสัยทัศน์ ที่เกี่ยวกับ .Net เอาไว้ 3 ข้อหลัก ๆ ได้แก่

1. การพัฒนาโปรแกรมในรูปแบบของ Web Service จะเป็นหัวใจหลักในการขับเคลื่อนโปรแกรมต่าง ๆ ที่ใช้งานบน Internet. Web Service จะช่วยให้การติดต่อสื่อสารระหว่าง application บน Internet นั้นง่ายขึ้น และเป็นระบบมากยิ่งขึ้น

2. Web Service ขั้นพื้นฐานเช่น การตรวจสอบ user ที่ log in เข้าสู่ระบบ จะถูกพัฒนาให้เป็นมาตรฐาน และสามารถนำไปใช้ได้ทั่วไปบน Internet

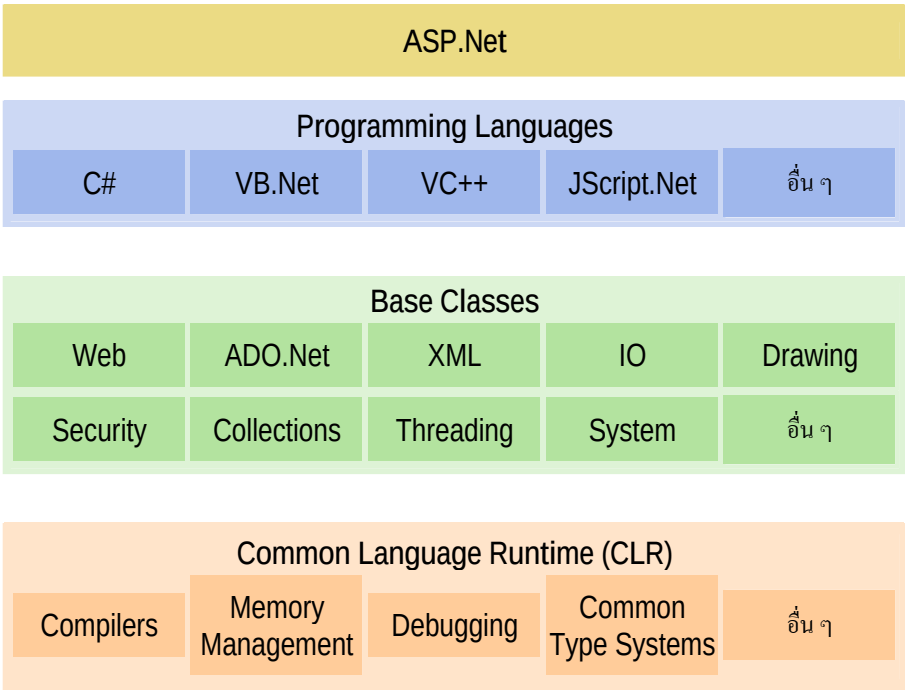
3. PC (desktop, notebook) และ Mobile Device ที่ต่อเชื่อมกับ Internet ได้ เช่น PDA และโทรศัพท์มือถือ จะมีบทบาท และประโยชน์มากขึ้นไปอีก เมื่อสามารถติดต่อใช้งานโปรแกรมต่าง ๆ บน Internet ได้

Microsoft จึงได้พัฒนารูปแบบการพัฒนาระบบคอมพิวเตอร์ขึ้นมา ซึ่งเรียกว่า .Net Framework นั่นเอง (อันที่จริงแล้ว Microsoft ไม่ได้เป็นผู้คิดค้นเรื่องพวกนี้ขึ้นมาแต่เพียงผู้เดียว อย่าเข้าใจผิด สิ่งต่าง ๆ เหล่านี้มีผู้คิดค้นพัฒนาจากหลายบริษัท หลายหน่วยงาน ยกตัวอย่างเช่น Sun Microsystems, IBM, ฯลฯ หากแต่ว่า Microsoft นำแนวคิดเหล่านั้นมาออกแบบให้อยู่ในรูปแบบที่สามารถต่อเชื่อมกันได้ง่ายขึ้น เป็นระบบมากขึ้น) เพราะฉะนั้น ถ้าจะให้นิยามคำว่า .Net คงไม่สามารถชี้เฉพาะได้ว่า .Net คืออะไร เพราะจริง ๆ แล้ว .Net ประกอบไปด้วยส่วนประกอบต่าง ๆ หลายส่วนด้วยกัน ส่วนประกอบเหล่านี้ถูกออกแบบมาเพื่อให้ทำงานได้เข้าขากันได้ดีขึ้น

ระบบนี้ไม่ใช่ระบบปฏิบัติการ (OS) แต่เปรียบเสมือนโปรแกรมหนึ่งที่จะสามารถสร้างสภาวะแวดล้อมหนึ่ง ซึ่งสามารถทำงานในระบบ .NET นี้ได้

รูปที่ 2.1 แสดงโครงสร้างของ .Net Fremework

ส่วนประกอบหลัก ๆ ของ .Net Framework แบ่งเป็นชั้น ๆ ได้ดัง diagram ต่อไปนี้



รูปที่ 2.2 แสดง diagram ของ .Net Fremework

1. Common Language Runtime (CLR) เป็นส่วนพื้นฐานที่ติดต่อกับระบบปฏิบัติการ Windows ทำหน้าที่เป็น run-time environment ให้กับโปรแกรมที่เขียนขึ้นสำหรับใช้บน .Net นับเป็นสิ่งสำคัญแทบจะที่สุดของระบบ .NET นี้ก็ว่าได้ เพราะ CLR ที่ว่านี้มีหน้าที่ทำให้โปรแกรมที่เขียนขึ้นมาด้วยภาษาต่างๆกัน กลายเป็นภาษารูปแบบมาตรฐานเดียวกัน ทั้งหมด เราเรียกภาษาที่ว่านี้ว่า Intermediate language (IL) ซึ่งเมื่อต้องการที่จะรันโปรแกรมใด CLR ที่ว่านี้จะตรวจสอบเครื่องที่รันว่ามีสภาวะแวดล้อมการทำงานเช่นใดหลังจากนั้นก็จะคอมไพล์เป็นโปรแกรมที่เหมาะสมต่อการทำงานของเครื่องนั้น ทำให้เราสามารถใช้งานโปรแกรม ต่างๆได้อย่างมีประสิทธิภาพสูงสุดในแต่ละเครื่องCLR มีส่วนของ compiler ทั้งที่เป็นแบบปกติ (compile ก่อนที่จะนำโปรแกรมไปใช้) และแบบ Just-In-Time (compile เมื่อจะใช้โปรแกรมนั้น ๆ) มีส่วนของ Memory Management ที่เอาไว้สำหรับจัดสรรหน่วยความจำของเครื่องให้กับโปรแกรม รวมไปถึงการทำ Garbage Collection (การเรียกคืนหน่วยความจำที่ไม่ได้ใช้อีกต่อไป) ส่วนของ Common Type Systems (CTS) ทำให้ภาษาต่าง ๆ ที่เขียนขึ้นบน .Net สามารถทำงานร่วมกันได้ เพราะขนาด และรูปแบบของข้อมูลที่เก็บไว้นั้นเป็นรูปแบบเดียวกัน

2. Base Classes เป็น class library พื้นฐาน ที่โปรแกรมต่าง ๆ ไม่ว่าจะเขียนด้วยภาษาใดบน .Net ก็สามารถใช้ร่วมกันได้ เช่น การติดต่อระบบฐานข้อมูล (ADO.Net), การติดต่อกับ file

system ของ server (IO), ฯลฯLibrary นั้นเปรียบเสมือนชุดคำสั่งสำเร็จรูปย่อยๆที่เพิ่มเข้ามา ซึ่งส่วนใหญ่จะเป็นชุดคำสั่งที่ต้องใช้งานอยู่เป็นประจำ ดังนั้นจึงมีผู้คิดค้นเพื่ออำนวยความสะดวกในการเขียนโปรแกรม ซึ่ง Library ในภาษาต่างๆส่วนใหญ่จะอยู่ในรูปแบบไฟล์ incould แต่ถ้าเป็น ASP สิ่งที่เป็น library ก็คือ คอมโพเนนท์ต่างๆนั่นเอง ซึ่งภายในระบบ .NET จะสร้างสิ่งที่เรียกว่าเป็น Library พื้นฐานขึ้น ทำให้ไม่ว่าจะใช้ภาษาใดในการพัฒนาโปรแกรมก็สามารถที่จะเรียกใช้ Libraryที่ตัวเดียวกันหมด

3. Programming Languages เป็นเซ็ตของ ภาษาคอมพิวเตอร์ ที่ถูกออกแบบมาเพื่อการเขียนโปรแกรมบน .Net Framework โดยที่ทางไมโครซอฟท์ได้เปิดตัวภาษาหลักๆที่จะใช้พัฒนาบน .NET นี้ 3ภาษา

- C# เป็นภาษาใหม่ที่ไมโครซอฟท์พัฒนามาจาก C++ กับ JAVA เป็นหลัก
- VB.NET เป็นภาษาที่พัฒนามาจาก Visual Basic ในเวอร์ชั่น 6.0
- JScript.net เป็นภาษาที่พัฒนามาจาก JScript ซึ่งเป็น JavaScript ในเวอร์ชั่นของ

ไมโครซอฟท์ไมโครซอฟท์นั้นเน้นไปที่ 3-4 ภาษาหลัก ๆ ได้แก่ VB.Net ซึ่งเป็นตัวที่พัฒนาต่อมาจาก VB, C# ซึ่งเป็นภาษาใหม่ที่มี syntax ใกล้เคียงกับ Java และ C++, Visual C++, และ JScript.Net ส่วนภาษาอื่น ๆ นั้น มีบริษัท หรือหน่วยงานอื่น ๆ เป็นผู้พัฒนาขึ้น ซึ่งคาดว่าจะมีเป็นสิบ ๆ ภาษา สำหรับ .Net Framework นั้นไม่ว่าจะเขียนโปรแกรมด้วยภาษาใดก็ตาม Compiler ใน CLR ก็จะ compile โปรแกรมนั้นให้อยู่ในรูปของ Intermediate Langauge (IL) ซึ่งจะถูกนำไปแปลเป็นภาษาเครื่อง (Native Code) อีกทีเมื่อตอนที่นำไปใช้

4. ASP.Net เป็นภาษา script ที่พัฒนาต่อมาจาก ASP ตัวเก่า เพื่อให้โปรแกรมเมอร์สามารถพัฒนา web application ให้ใช้ .Net ได้สะดวกขึ้น ASP.Net นี้ถึงแม้จะอ้างอิงมาจาก ASP ตัวเก่า แต่ก็มี syntax หลายส่วนที่เปลี่ยนแปลงไป อย่างไรก็ดี ผู้ที่เคยเขียน ASP มาก่อนไม่น่าจะมีปัญหาใรการอ่าน และเขียน ASP มากนัก

หมายเหตุ: สำหรับผู้อ่านที่คุ้นเคยกับ Java ของ Sun มาก่อน จะเห็นได้ว่า .Net Framework กับ Sun's J2EE นั้นคล้ายกันมากทีเดียว สรุปโดยคร่าว ๆ ได้ดังนี้ CLR = JVM (Java Virtual Machine), IL = Java Bytecode, .Net base classes = Java Class Library, .Net Programming Languages = Java Langauge, ASP.Net = Java Server Page (JSP). ข้อแตกต่างในด้าน Architecture หลัก ๆ ก็คือ IL ของ .Net นั้นต้อง run บน Windows เท่านั้น (ไมโครซอฟท์บอกว่า run บน OS ไหนก็ได้ แต่อย่างที่ทราบกันดี คงยากที่จะได้เห็นในอนาคตอันใกล้ หรือไม่ก็คงไม่ดี

เท่ากับ run บน Windows) แต่ Java Bytecode นั้นสามารถ run บน OS ไหนก็ได้ที่มี JVM. ส่วนในด้าน performance นั้น ไมโครซอฟท์ ได้ทำการเปรียบเทียบโดยพัฒนาโปรแกรม Pet Shop ด้วย .Net โปรแกรม Pet Shop นี้เป็น reference application ที่พัฒนาบน J2EE โดย Sun ไมโครซอฟท์แสดงให้เห็นว่า .Net นั้นทำให้ code สั้นลงหลายเท่าตัว ทำให้โปรแกรมทำงานเร็วขึ้นหลายเท่าเป็นต้น ในส่วนนี้ก็คงต้องรอดูกันต่อไปว่า Sun จะพัฒนา Java อย่างไรเพื่อแก้เกมส์ในจุดนี้

ประโยชน์และข้อดีของ .NET Framework นั้นพอจะสรุปออกมาได้เป็นข้อๆดังนี้

1. เป็นระบบที่มีไลบรารีที่เป็นมาตรฐานเดียวกัน : เนื่องจากมีไลบรารีที่เป็นมาตรฐานเดียวกันทั้งหมดทำให้เราไม่ต้องกังวล ว่าภาษาที่ใช้เขียนนั้นมีไลบรารีตัวนั้นตัวนี้หรือไม่ รวมทั้งไม่ต้องคอยกังวลว่าถ้าใช้ไลบรารีของภาษาหนึ่งแล้วอีกภาษาหนึ่งจะไม่มีไลบรารีตัวนั้น

2. ไม่ขึ้นกับระบบประฏิบัติการ (OS) : เนื่องจากระบบปฏิบัติการ ที่แต่ละบุคคลหรือองค์กรใช้นั้นย่อมไม่เหมือนกัน แต่ภายใน .NET Framework จะไม่มีปัญหานี้ของเพียงแค่มีระบบ .NET Framework ก็จะทำให้สามารถใช้งานโปรแกรมต่างๆได้ ซึ่งเป็นข้อดีตรงที่เราจะสามารถใช้โปรแกรมต่างๆได้ทุกระบบปฏิบัติการ

3. ใช้ในการพัฒนาได้ทุกภาษา :ทำให้เราไม่ต้องคอยมาศึกษาภาษาใหม่ๆเมื่อต้องการสร้างโปรแกรมในแต่ละครั้ง นอกจากนั้นเรายังสามารถเลือก ใช้ภาษาที่เราถนัดที่สุดในการพัฒนาโปรแกรมต่างๆได้ด้วย

4. มีการควบคุมสิ่งแวดล้อมในการทำงานเป็นอย่างดี :เนื่องจากเป็นระบบที่เป็นมาตรฐานทำให้การควบคุมจัดสรรระบบต่างๆ ทำได้ง่ายขึ้น ไม่ว่าจะเป็นการจัดสรรหน่วยความจำ ด้านการใช้งานเครื่องก็มีความรวดเร็วมากขึ้น ลดโอกาศที่เครื่องจะแฮงก์ได้เป็นอย่างดี

5. ความปลอดภัยที่มีมากขึ้น :.NET Framework สามรถกำหนดสิทธิ์การใช้งานหรือ permission ของผู้ใช้งานได้มากขึ้นทำให้สมารถกำหนดว่า จะให้โปรแกรมในส่วนใดใช้งานได้หรือไม่ได้ แล้วแต่เฉพาะบุคคล

2.1.2 เริ่มต้นรู้จักกับ ASP.NET

ASP.NET หรืออีกชื่อหนึ่งว่า ASP+ ซึ่งเป็นชื่อที่ไมโครซอฟท์ใช้เรียกในตอนแรก ถือว่าเป็น ASP เวอร์ชั่นล่าสุดต่อจาก ASP 3.0 แต่คงไม่สามารถพูดได้เต็มปากว่า ASP.NET พัฒนามาจาก ASP เพราะรูปแบบ และไวยากรณ์ต่างๆ และภาษาที่นำมาใช้งานนั้นต่างจากเดิมแทบทั้งสิ้น แทบจะเรียกได้ว่ายกเครื่องใหม่เลยทีเดียว น่าจะพูดได้ว่า ASP.NET เป็นอีก Generation หนึ่งของ ASP มากกว่า เรามาลองดูกันว่าใน ASP.NET นั้นมีอะไรที่แตกต่างจาก ASP รุ่นก่อน ๆบ้าง

1) ใช้ภาษาใดๆในการเขียนสคริปต์ก็ได้ : จากเดิมที่เราสามรถใช้ได้เฉพาะภาษาที่เป็นสคริปต์ของ VBScript และ JScript แต่ใน ASP.NET เราสามารถที่จะใช้ภาษาที่มีรูปแบบของภาษาเต็มๆ ซึ่ง ในเบื้องต้น มี3 ภาษาคือ C#, VB.NET และ JScript.Net ที่ออกมาเป็นมาตรฐาน แต่ในอนาคตไมโครซอฟท์มีแผนที่จะเพิ่มตัวแปลภาษาให้ครบทุกภาษา

2) มีความยืดหยุ่นในการเขียนโปรแกรมมากขึ้น : โดยที่เราสามารถใช้ภาษาในการเขียน ASP.NET ได้มากกว่า 1 ภาษาภายในไฟล์เดียวกัน ทำให้สามารถเลือกรูปแบบของภาษาที่ง่ายที่สุดต่อการเขียน ในแต่ละส่วนได้

3) ลักษณะการแปลภาษาและนามสกุลไฟล์เปลี่ยนไป : ใน ASP เวอร์ชั่นก่อนๆ มีลักษณะการแปลภาษาเป็นแบบอินเตอร์พรีเตอร์ (Interpreter) คือการจะทำคำสั่งใดค่อยแปลคำสั่งนั้น แต่ในเวอร์ชั่น .NET นี้จะมี ลักษณะเป็นคอมไพเลอร์ (Compiler) คือการแปลคำสั่งรวมทั้งโปรแกรม นอกจากนี้นามสกุลของไฟล์ก็มีการเปลี่ยนแปลงจากเดิมที่ใช้นามสกุลไฟล์เป็น " *.asp " เป็น " *.aspx "

4) รูปแบบและการใช้งานคอมโพเนนต์ที่ง่ายขึ้น : รูปแบบของคอมโพเนนต์จะเน้นไปที่ XML มากที่สุด และที่สำคัญคือการใช้งานคอมโพเนนต์ใน ASP.NET นั้นเราสามารถอัพโหลดไฟล์ไปไว้ในไดเร็คตอรีที่ผู้ดูแลเซิร์ฟเวอร์ (Admin) กำหนดหลังจากนั้นคอมโพเนนต์จะติดตั้งตัวเองโดยอัตโนมัติ ลดปัญหาที่เกิดจาก ASP เวอร์ชั่นก่อนๆได้เป็นอย่างดี เนื่องจากใน ASP เวอร์ชั่นก่อนนั้นการติดตั้งคอมโพเนนต์กระทำได้เพียงผู้ดูแลเซิร์ฟเวอร์เพียงคดเดียวเท่านั้น ทำให้เวลาต้องการใช้คอมโพเนนตีต่างๆที่เซิร์ฟเวอร์ไม่มี จึงเป็นเรื่องที่ลำบาก

5) มีไลบรารีให้เลือกใช้ได้มากขึ้น : ใน ASP เวอร์ชั่นก่อนๆนั้นแอพพลิเคชั่นบางอย่างสร้างได้ไม่สะดวกนัก ต้องอาศัยคอมโพเนนต์ต่างๆมากมาย แต่ใน ASP.NET นั้นได้เพิ่มไลบรารีในส่วนเหล่านี้ให้กลายเป็นพื้นฐานของการใช้งาน

6) มีคอนโทรลทำให้การใช้งานในบางสิ่งง่ายขึ้น : เป็นส่วนพิเศษที่เพิ่มเติมมาจาก ASP รุ่นก่อนๆที่ไม่มีส่วนที่เรียกว่า คอนโทรล ซึ่งคอนโทรลนี้จะช่วยให้เราสามารถสร้างเว็บไซท์ได้อย่างง่ายและมีประสิทธิภาพมากขึ้น จึงไม่ต้องกังวลว่าบราวเซอร์รุ่นนั้นรุ่นนี้จะรองรับกับภาษาที่เราเขียนหรือไม่

7) สามารถเรียกขอข้อมูลจากเซิร์ฟเวอร์ได้ : ใน ASP เวอร์ชั่นก่อนๆเซิร์ฟเวอร์สามารถเรียกขอข้อมูลได้จากเครื่องผู้ใช้เท่านั้นแต่ใน ASP.NET เครื่องเซิร์ฟเวอร์สามารถเรียกขอข้อมูลจากเครื่องเซอร์เวอร์ด้วยกันได้

8) ไม่ต้องต่อ Hardware : เนื่องจากเป็นระบบใน .NET Framework ดังนั้นจึงมีคุณสมบัติของ Common Languare Runtime (CLR) ทำให้มีการคอมไพล์โปรแกรมเป็นภาษามาตรฐานที่เรียกว่า IL ก่อน ดังนั้นไม่ว่าคุณจะเล่นเครื่องปาล์มหรือโน้ตบุ๊ค PDA ก็ไม่เกิดปัญหา

9) ง่ายต่อการหาจุดผิดพลาดในการเขียนโปรแกรม : หากเป็น ASP รุ่นก่อนเวลาเกิดความผิดพลาด (error) เครื่องจะบอกแค่ว่าเป็นความผิดพลาดชนิดใดบรรทัดไหน แต่ใน ASP.NET นี้เครื่องจะแสดงรายละเอียดที่มากขึ้น พร้อมแนวทางแก้ไข

10) มีการตรวจสอบเหตุการณ์ต่างๆได้ภายในเว็บเพจ : มีการตรวจสอบเหตุการณ์ต่างๆตั้งแต่โหลดหน้าเว็บเพจไปจนถึงปิดหน้าเว็บเพจลง ทำให้เราสามารถเขียนโปรแกรมกำหนดเหตุการณ์ต่างๆได้ง่ายขึ้น

11) แยกส่วนที่เป็น HTML กับ ASP ออกมาอย่างชัดเจน : ในเวอร์ชั่นก่อนๆส่วนที่เป็น HTML กับ ASP จะเขียนปนกันไปมา แต่ในเวอร์ชั่นนี้จะแยกส่วนกันอย่างชัดเจนว่าส่วนไหนเป็น HTML และส่วนไหนเป็น ASP

2.1.3 MicrosoftVisual Studio.NET

MicrosoftVisual Studio.NET คือเครื่องมือในการพัฒนาซอฟต์แวร์สมบูรณ์แบบ ที่สนับสนุนการใช้ภาษาโปรแกรมหลายภาษา สำหรับการสร้าง และประสาน Web services และแอพพลิเคชั้นที่ใช้ XML เข้าด้วยกันช่วยเพิ่มศักยภาพให้กับการทำงานของนักพัฒนา และก่อให้เกิดโอกาสใหม่ๆ ทางธุรกิจ โดยได้รับการออกแบบให้รวมเข้ากับมาตรฐาน และโปรโตคอล

ของอินเตอร์เน็ต เช่น XML และ SOAP อย่างแนบแน่น Visual Studio.NETจึงทำให้วงจร ของการพัฒนาแอพพลิเคชั่นง่ายขึ้นอย่างมาก

เป็นชุดเครื่องมือพัฒนาที่ครอบคลุมสำหรับการสร้างแอพพลิเคชั่นทุกรูปแบบ ซึ่งมีส่วนติดต่อกับผู้ใช้ (User Interface) ที่หลากหลายเช่น สำหรับ rich client , บราวเซอร์และอุปกรณ์ไร้สาย ซึ่งสิ่งสำคัญต่อนักพัฒนาบนแพล็ตฟอร์ม .NET ก็คือ นักพัฒนาสามารถจะใช้งานได้โดยเรียนรู้อีกเพียงเล็กน้อย หรืออาจไม่ต้องเรียนรู้เพิ่มเติมเลยใช้ทักษะที่มีอยู่ให้เป็นประโยชน์และเลือกใช้ภาษาโปรแกรมจำนวนมากที่สนับสนุน .NET เพื่อให้เกิดประสิทธิผลสูงสุด คุณสามารถนำแอพพลิเคชั่นที่ถูกเขียนด้วยภาษาต่างๆมาใช้งานร่วมกัน หรือนำมาใช้ใหม่ ทำให้มั่นใจได้ว่านักพัฒนาสามารถนำชุดคำสั่งเดิมที่มีอยู่มาใช้ได้ และสร้างแอพพลิเคชั่นได้เร็วขึ้น นอกจากนี้ ยังช่วยลดจำนวนชุดคำสั่งที่ต้องเขียนได้ ด้วยการลากและวางองค์ประกอบต่างๆ อีกด้วย ซึ่ง .NET Framework จะเตรียมการสำหรับ XML Web services ให้โดยอัตโนมัติ คุณจึงมุ่งไปที่การเขียนชุดคำสั่งที่เกี่ยวกับการทำงานในธุรกิจได้อย่างเต็มที่ แทนที่จะเสียเวลากับการเขียนชุดคำสั่งด้านโครงสร้างพื้นฐาน ด้วยรูปแบบการเขียนโปรแกรมที่สอดคล้องกันเช่นนี้จึงทำให้การพัฒนาง่ายขึ้น

ทีมพัฒนาสามารถใช้สถาปัตยกรรมแอพพลิเคชั่นร่วมกัน เริ่มการพัฒนาได้อย่างรวดเร็วและก่อให้เกิดการปฏิบัติงานที่ดีที่สุด Visual Studio .NET ได้รวมกระบวนการพัฒนาทั้งหมดเข้าไว้ด้วยกัน ซึ่งได้แก่ การออกแบบ ทดสอบ นำแอพพลิเคชั่นไปใช้งาน รวมทั้งการประสานงานระหว่างทีมงานต่างๆ จึงเหมาะสำหรับการพัฒนาในองค์กรขนาดใหญ่

ครอบคลุมวงจรการพัฒนาทั้งหมด Visual Studio .NET เป็นแพล็ตฟอร์มทรงพลังสำหรับระบบพัฒนาของผู้ผลิตอื่นๆ โดยได้เตรียมการเรียกใช้เครื่องมือในการพัฒนาจำนวนมากภายใต้สภาพแวดล้อมที่เป็นหนึ่งเดียว

2.2 ทฤษฎี Web services

Web Services คือ application หรือ program ที่ทำงานอย่างใดอย่างหนึ่ง ในลักษณะให้บริการ โดยจะถูกเรียกใช้งานจาก application อื่นๆ ในรูปแบบ RPC(Remote Procedure Call) ซึ่งการให้บริการจะมีเอกสารที่อธิบายคุณสมบัติของบริการกำกับไว้ โดยภาษาที่ถูกใช้เป็นสื่อในการแลกเปลี่ยนคือ XML ทำให้เราสามารถเรียกใช้ component ใด ๆ ก็ได้ ใน ระบบ หรือ platform ใด ๆ ก็ได้ บน protocol HTTP ซึ่งเป็น protocol สำหรับ World Wide Web หรืออินเทอร์เน็ต อันเป็นช่องทางที่ได้รับการยอมรับทั่วโลกในการติดต่อสื่อสารกันระหว่าง application กับ application ในปัจจุบัน

การทำงานของ Web Services ประกอบไปด้วย มาตรฐานหลัก 4 อย่าง ซึ่งสามารถอธิบายอย่างง่ายๆ ได้ดังนี้

1. XML (Extensible Markup Language) เป็นภาษามาตรฐานที่ทุกระบบสนับสนุน ทำให้ข้อมูลที่มีโครงสร้างของภาษา XML จะถูกนำไปประมวลผลต่ออย่างอัตโนมัติได้อย่างง่ายดาย ภาษา XML จึงถูกนำมาใช้เป็นภาษามาตรฐานในการแลกเปลี่ยนข้อมูลของ Web Services

2. SOAP (Simple Object Access Protocol) หรือโซพ เป็นมาตรฐานของเทคโนโลยี Distributed Objects แบบหนึ่ง โดยทำหน้าที่ส่งข้อมูลผ่านอินเทอร์เน็ต ในรูปแบบของ XML ทำให้เรียกใช้งานโปรแกรมข้ามระบบผ่านทางอินเทอร์เน็ตได้ ซึ่งเป็น protocol ที่ใช้เรียก component ต่าง ๆ ใน platform ใด ๆ ก็ได้ เนื่องจากใช้ XML เป็นรูปแบบของข้อมูลที่ส่ง โดย protocol นี้จะทำงานอยู่บน HTTP ซึ่งเป็น protocol มาตรฐานสำหรับ web อยู่แล้ว ซึ่งนั้นหมายความว่า การเรียกใช้ component โดยอาศัย SOAP ก็คือการเรียกผ่าน web นั้นเอง

อย่างไรก็ดี SOAP ก็ยังมีข้อเสียอยู่บ้าง ได้แก่

1. เนื่องจาก SOAP message นั้นเก็บอยู่ในรูปแบบ XML ทำให้เสียเวลาในการแปลเอกสาร XML กลับมาเป็นรูปแบบที่โปรแกรมเข้าใจ

2. SOAP ทำงานอยู่กับ HTTP protocol ซึ่งมีความเร็วในการรับ และส่งข้อมูล ค่อนข้างต่ำ ด้วยเหตุนี้ทำให้ความเร็วของ SOAP อยู่ในระดับเดียวกับ HTTP ในขณะที่ protocol อื่นเช่น FTP มีความเร็วสูงกว่ามาก

3. WSDL (Web Services Description Language) เป็นภาษามาตรฐานที่ใช้สำหรับอธิบายการใช้งานโปรแกรมที่เปิดให้บริการ ซึ่งเขียนขึ้นตามแบบมาตรฐาน XML ดังนั้น WSDL จึงเป็นเสมือนคู่มือให้กับระบบ เพื่อเรียนรู้วิธีการเรียกใช้งาน Web Services

WSDL เกิดจากความร่วมมือระหว่าง IBM และ Microsoft WSDL เป็นภาษาที่ใช้อธิบายคุณลักษณะของ web service และวิธีการติดต่อกับ web service นั้น ๆ โดยใช้ไวยากรณ์ของภาษา XML ซึ่ง WSDL อยู่ในความดูแลของ W3C version ล่าสุด คือ WSDL 1.1 ซึ่งหากผู้อ่านสนใจในรายละเอียด สามารถหาอ่านเพิ่มเติมได้จากเว็บไซต์ของ W3C ที่ http://www.w3.org/TR/wsdl ส่วนในทางปฏิบัติ หากเราต้องการ สร้าง web service ขึ้นมาเป็นของตนเอง ก็สามารถสร้าง

เอกสาร WSDL ได้โดยอัตโนมัติ เราจึงไม่ต้องไปกังวลในรายละเอียด ในข้อกำหนดใน WSDL มากนัก

4. UDDI (Universal Description, Discovery, and Integration) เป็นมาตรฐานที่จัดตั้งขึ้นโดยบริษัท IBM, Microsoft และบริษัทยักษ์ใหญ่ทางธุรกิจ B2B (Business-to-Business) อื่น ๆ UDDI ถูกสร้างขึ้นมาเป็นระบบมาตรฐานในการอธิบายและค้นหา Web Services โดยเป็นตัวกลางให้ Provider มาลงทะเบียนไว้ โดยใช้ไฟล์WSDL บอกรายละเอียดของบริษัทและบริการที่มีให้ ทำให้ Requestor สามารถค้นหาและทราบว่าบริษัทมีผลิตภัณฑ์และบริการอะไรบ้าง สามารถติดต่อขอดำเนินธุรกิจการค้ากับบริษัทได้โดยอัตโนมัติผ่านทาง Web Services สำหรับคู่ค้าทางธุรกิจ ซึ่งเปรียบได้กับฐานข้อมูล ขนาดใหญ่ ซึ่งมีข้อมูลของ web service ที่เปิดให้บริการ โดยที่ web site สำหรับค้นหา web service มีอยู่หลายที่อย่างเช่น

http://uddi.microsoft.com/search.aspx

http://www-3.ibm.com/services/uddi/testregistry/find

http://uddi.org

จากมาตรฐานทั้ง 4 อย่างที่กล่าวข้างต้นสามารถสรุปลำดับขั้นของการทำงานของ Web Services ได้ดังนี้

1. Provider จัดทำระบบหรือบริการที่เป็น Web Services ขึ้นมา
2. ทำการลงทะเบียน Web Services กับหน่วยงานที่ให้บริการระบบ UDDI (หรือ Registry)
3. นำ WSDL ไฟล์ไปไว้ในระบบ UDDI ที่ได้ลงทะเบียนไว้
4. Requestor ทำการค้นหาระบบหรือบริการที่ต้องการจากระบบ UDDI
5. เมื่อ Requestor ได้พบระบบหรือบริการที่ต้องการจะนำไฟล์ WSDL ไปเรียนรู้วิธีการเรียกใช้ผ่านระบบของตน
6. Requestor ทำการติดต่อและเรียกใช้ระบบหรือบริการจาก Provider ได้โดยตรงผ่าน SOAP ในระบบของตน

ถ้าพิจารณาลักษณะการทำงานและความสามารถของ Web Services แล้ว เราสามารถสรุปได้ว่า Web Services ก็คือ วิวัฒนาการอีกก้าวหนึ่งของ Web Application นั่นเอง

เปรียบเทียบระหว่าง Web Application และ Web Services

เมื่อเราเข้าใจความหมายและการทำงาน Web Application และ Web Services แล้ว จะเห็นว่าเครื่องมือทั้งสองต่างใช้ HTTP Protocol หรือ อินเทอร์เน็ต เป็นช่องทางในการสื่อสารเหมือนกัน แต่มีวัตถุประสงค์ต่างกัน โดย Web Application ใช้เพื่อการแลกไฟล์ HTML ระหว่างเว็บเซิร์ฟเวอร์ แต่ Web Services เป็นการแลก “บริการ” (ก้อนโปรแกรม Software Components) ระหว่างระบบสารสนเทศ ผ่านเว็บเซิร์ฟเวอร์

ในเรื่องของความสามารถโดยส่วนใหญ่จะใช้ Web Application ในการติดต่อกับผู้ใช้ผ่านทาง Internet browser เพื่อนำเสนอข้อมูลและการทำธุรกรรมต่างๆ ส่วน Web Services จะทำหน้าที่ในการติดต่อกับ เว็บเซิร์ฟเวอร์ เพื่อแลกเปลี่ยนข้อมูลและการทำงานหรือใช้บริการข้ามระบบกันโดยใช้ Web Application หรือ Application Interface ในการติดต่อกับผู้ใช้ นอกจากนี้ Web Services ยังสามารถทำงานกับระบบต่างๆ ได้มากกว่า 1 ระบบ ในขณะที่ Web Application ไม่สามารถทำได้โดยตรง ซึ่งสามารถสรุปการเปรียบเทียบได้ดังตาราง

ตารางที่ 2.1 การเปรียบเทียบระหว่าง Web Services VS Web Application

หัวข้อ	**Web Services**	**Web Applications**
การเชื่อมต่อ	program-program	human-program
ภาษาที่ใช้	XML	HTML
รายชื่อการให้บริการ	ค้นหาผ่าน UDDI	ค้นหาผ่าน search engine
ขอบเขตการใช้งาน	Business-to-Business (B2B)	Business-to-Customer (B2C)
โปรโตคอล(Protocol)	SOAP+HTTP	HTTP

หลังจากที่ได้ทำความเข้าใจความหมายและข้อเปรียบเทียบระหว่าง Web Application และ Web Services แล้ว ความแตกต่างระหว่างเครื่องมือทั้งสองจะเห็นได้ชัดในเรื่องของความสามารถ แต่ในการนำไปใช้งานจำเป็นจะต้องประยุกต์ใช้เครื่องมือทั้งสองชนิดร่วมกันเพื่อให้เกิดประสิทธิภาพมากที่สุด

ในเรื่องของความปลอดภัย เนื่องจาก Web Service ทำงานอยู่บน Internet ซึ่งปัจจุบันมีเทคโนโลยีในการรักษาความปลอดภัยมากมายรองรับอยู่แล้ว และ Web Services สามารถผ่านระบบรักษาความปลอดภัย (Firewall) ได้เนื่องจาก SOAP ถูกส่งโดยผ่านโปรโตคอล HTTP นอกจากนี้ Web Services ยังมีระบบรักษาความปลอดภัยตามมาตรฐานของ PKI (Public Key Infrastructure) เช่น MD5 (Message Digest),

SSL (Secure Socket Layer) และ PGP (Pretty Good Privacy) ทำให้มั่นใจได้ว่าการใช้ Web Services เป็นเครื่องมือธุรกิจมีมาตรฐานในการรักษาความปลอดภัยรองรับ

ส่วนเครื่องมือที่ใช้ในการพัฒนา Web Services ก็มีอยู่มากมาย เนื่องจาก Web Services เป็นมาตรฐานที่พัฒนาได้ง่าย เพราะอยู่ในรูปแบบของข้อมูลตัวอักษรหรือ ภาษา XML ทำให้มีชุดเครื่องมือที่สนับสนุนการพัฒนา Web Services มากมาย ซึ่งสามารถแบ่งออกเป็น 4 กลุ่มใหญ่ได้ดังนี้

1. ชุดเครื่องมือที่สนับสนุนโดย Microsoft ตาม Platform ของ Microsoft .NET Framework
2. ชุดเครื่องมือที่สนับสนุนโดย Sun Microsystem ตาม Platform ของ Sun ONE (Sun Open Net Environment)
3. ชุดเครื่องมือที่สนับสนุนโดย IBM เช่น Web Services Toolkit
4. ชุดเครื่องมืออื่นๆที่สนับสนุน SOAP, XML ทั้งที่เป็น Commercial Product และ Open Source

2.3 ทฤษฎี GPS โปรโตคอล NMEA

ทฤษฎีการกำหนดตำแหน่งด้วยระบบGPS ระบบดังกล่าวจะบอกตำแหน่งละติจูดลองจิจูดของทุกจุดบนพื้นโลก โดยรับสัญญาณจากดาวเทียมบอกตำแหน่งซึ่งทำการโคจรอยู่รอบโลกอย่างน้อย 4 ดวง จากทั้งหมด 24 ดวง ระบบGPS ใช้ดาวเทียมนาฟสตาร์(NAVSTAR) โดยการส่งคลื่นวิทยุจากดาวเทียมในอวกาศมายังภาคพื้นดิน และใช้ความต่างของเวลาในการรับส่งสัญญาณระหว่างดาวเทียมกับตัวรับสัญญาณ โดยคำนวณหาตำแหน่ง, ความเร็ว และเวลาให้กับผู้ใช้ตามปรกติระบบGPS จะมีการใส่รหัสเพื่อให้เกิดความผิดพลาดเล็กน้อยทั้งนี้ เนื่องจากระบบนี้ถูกพัฒนาขึ้นเพื่อให้กระทรวงกลาโหมของสหรัฐอเมริกานำระบบนี้มาใช้ในทางการทหาร แต่สัญญาณดาวเทียมที่ถูกส่งออกมานั้นสามารถถูกรับได้โดยผู้ใช้ทั่วไป ดังนั้นจึงมีการใส่รหัสเพื่อให้ผู้ใช้ที่ได้รับอนุญาตเท่านั้นใช้งานได้ ในการใช้งานระบบGPS ในโหมดมาตรฐานจะมีความเที่ยงตรงโดยเฉลี่ย 100 เมตร ในแนวนอนและนอนตั้ง 156 เมตร ในงานที่ต้องการความเที่ยงตรงมากๆจึงต้องใช้ DGPS (Difference Global Position System) โดยรูปแบบของDGPS จะประกอบด้วยเครื่องรับทีเรียกว่า “เครื่องรับอ้างอิง” ซึ่งทราบตำแหน่งที่ถูกต้องอย่างแท้จริงอยู่แล้ว ส่วนเครื่องรับGPS อื่นๆ จะเป็นของผู้ใช้งาน โดยเครื่องรับอ้างอิงหรือสถานีอ้างอิงนี้จะนำสัญญาณดาวเทียมGPS มาคำนวณพิสัยของดาวเทียมแต่ละดวง

มาตรฐานในการสื่อสาร National Marine Electronics Assciation (NMEA) หน่วยงานที่หน้าที่ในการกำหนดมาตรฐานในการสื่อสารข้อมูลGPS คือหน่วยงาน NMEA ได้กำหนดโปรโตคอล NMEA-0183 เป็นโปรโตคอลมาตรฐานในการสื่อสารข้อมูลจากเครื่องรับGPSไปสู่อุปกรณ์เชื่อมต่อภายนอก ซึ่งรูปแบบเหล่านี้จะอยู่ในรูปของรหัสแอสกี้ (ASCII Codes) และสามารถเชื่อมต่อกับเครื่องคอมพิวเตอร์ได้ โดยผ่านพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ (RS-232) โดยกำหนดอัตราความเร็วในการส่งข้อมูลอยู่ที่ 4800 บิตต่อวินาที โดยใช้การส่งข้อมูล 8 บิต แบบไม่มีพาริตี้ แต่มีบิตเริ่มต้น (Start bit) และบิตสิ้นสุด (Stop bit)

การสื่อสารระหว่างตัวรับ GPS ได้ถูกระบุไว้ใน Specification นี้เช่นกัน โปรแกรมคอมพิวเตอร์ส่วนใหญ่ถูกใช้งานหาตำแหน่งในลักษณะ real time โดยการใช้รูปแบบ format ของ NMEA ข้อมูลจะถูกจัดอยู่ในรูป PVT (position , velocity , time) เป็นโซลูชั่นที่ใช้ในตัวรับ GPS โดยแนวคิดของ NMEA จะส่งข้อมูลที่มีลักษณะเป็นประโยค ที่มีรูปแบบเฉพาะ โดยประโยคมาตรฐานของแต่ละชนิดของอุปกรณ์จะถูกระบุเป็นรูปแบบของตัวเองถูกใช้งานในแต่ระบริษัท โดยประโยคมาตรฐานทั้งหมดจะมีคำนำหน้า 2 ตัวเพื่อระบบตัวอุปกรณ์และรูปแบบของประโยค (สำหรับตัวรับ GPS จำใช้คำนำหน้าว่า GP) และตามด้วยตัวอักษรอีก 3 ตัวเพื่อกำหนดถึงข้อมูลที่ใช้อยู่ในประโยคนี้ NMEA ยอมให้ผู้ผลิตฮาร์ดแวร์สามารถระบุเป็นรูปแบบประโยคของตัวเองได้ โดยจะต้องขึ้นต้นด้วยตัวอักษร P และตามด้วยตัวอักษร 3 ตัวที่แสดงถึงผู้ผลิต ยกตัวอย่าง ประโยคของ Garmin จะขึ้นต้นด้วย PGRM และ Magellan จะขึ้นต้นด้วย PMGN

ในแต่ระประโยคจะขึ้นต้นด้วย $ และจบด้วย carriage return/line feed ตามลำดับ และจำนวนตัวอักษรไม่เกิน 80 ตัวอักษรที่สามารถมองเห็นได้ รวมทั้งตัวอักษรขึ้นบรรทัดใหม่ data จะมีเพียงบรรทัดเดียว และใช้ comma เป็นตัวขั้นกลาง data ที่ใช้เป็นเพียง ascii text ธรรมดา และสามารถเพิ่มขยายโดยการนำหลายๆประโยคมารวมกัน แต่โดยปกติแล้วจะใช้เพียงแค่ประโยคเดียว ข้อมูลอาจจะมีการเปลี่ยนแปลงจำนวนของประโยค ยกตัวอย่าง ข้อมูลเวลาอาจจะถูกระบบเป็น เลขทศนิยมที่มีหน่วยของวินาที หรือในบางประโยคอาจจะแสดงข้อมูล 3 หลักหรือ 4 หลักหลังจุดทศนิยม โปรแกรมจะอ่านข้อมูลโดยใช้ comma เป็นตัวขั้น โดนไม่สนใจจำนวน column และปิดท้ายด้วย checksum ในทุกๆประโยค ซึ่งอาจจะใช้ในการเช็คหรือไม่ใช้ ใน checksum จะประกอบด้วย * และเลขฐาน 16 2 หลัก ซึ่งเป็นค่าที่คำนวณจาก Exclusive Or ของทุกตัวอักษรรวมกัน แต่ไม่รวมตัว $ กับ * checksum จะใช้สำหรับบางประโยค

2.3.1 การเชื่อมต่อกับ Hardware

การเชื่อมต่อกับอุปกรณ์ GPS ถูกออกแบบให้ใช้โปรโตคอล NMEA ซึ่งสามารถรองรับการทำงานของคอมพิวเตอร์ส่วนใหญ่ผ่านทาง serial port โดยใช้ โปรโตคอล RS232 อย่างไรก็

ตามถ้าพูดถึงมาตรฐาน NMEA จะไม่จำเป็นจะต้องเป็น RS232 แต่จะนิยมใช้ EIA-422 ซึ่งเป็น Interface ที่สามารถปรับความเร็วได้บนโมเดลบางโมเดล โดยที่ NMEA มาตรฐานจะใช้ความเร็ว 4800 baud , 8 bit of data , no parity , one stop bit อุปกรณ์ทุกตัวที่สนับสนุน NMEA จะต้องสนับสนุนความเร็วตามที่กล่าวมา ข้อควรจำ ที่ความเร็ว baud rate 4800 คุณสามารถส่งข้อมูลได้มากกว่าที่ต้องการในช่วงระยะเวลา 1 วินาที ด้วยเหตุผลนี้ อุปกรณ์บางตัวส่งข้อมูลอัพเดททุกๆ 2 วิหรืออาจะส่งข้อมูลในทุกวินาที หรือในอุปกรณ์บางตัวส่งข้อมูล 2 ครั้งใน 1 วินาที โดยทั่วไประยะเวลาในการส่งข้อมูลบางส่วนจะสามารถทำเสร็จได้ภายใน 1 วินาที

2.3.2 ชนิดของประโยค

NMEA ประกอบขึ้นจากประโยค โดยที่คำแรกจะแสดงถึงชนิดของข้อมูลเรียกว่า data type ซึ่งจะเป็นตัวระบุรูปแบบของคำที่เหลือที่อยู่ในประโยค ซึ่งแต่ละ data type จะมีการตีความหมายของประโยคแตกต่างกันซึ่งถูกกำหนดโดยมาตรฐาน NMEA

ประโยค GGA ที่แสดงดังตัวอย่างด้านล่าง จะมีรูปแบบข้อมูลที่คงที่ ซึ่งข้อมูลจะถูกส่งมาซ้ำๆในรูปแบบเดิม แต่ตัวข้อมูล data จะเปลี่ยนไป เมื่ออุปกรณ์หรือโปรแกรมทำการอ่านข้อมูลหรือประโยคที่ได้รับ เป็นที่น่าสนใจว่าโปรแกรมสามารถที่จะเลือกรับเฉพาะข้อมูลรูปแบบที่ต้องการได้ และละทิ้งประโยคอื่นๆที่ไม่สนใจ ในมาตรฐาน NMEA จะไม่มีส่วนของคำสั่งที่ระบุว่า gps จะต้องทำงานอย่างไร อุปกรณ์รับสัญญาณ gps บางตัวจะมีคำสั่งภายใน ที่สามารถสั่งให้ส่งข้อมูลมาเฉพาะประโยคที่ต้องการได้

ประโยคของมาตรฐาน NMEA มีหลายชนิดขึ้นอยู่กับชนิดของตัวอุปกรณ์ สำหรับประโยคที่ใช้กับงานทางทะเล ที่ใช้อยู่ในตัว gps มีดังนี้คือ

ทุกประโยคจะขึ้นต้นด้วย GP

- AAM - Waypoint Arrival Alarm
- ALM - Almanac data
- APA - Auto Pilot A sentence
- APB - Auto Pilot B sentence
- BOD - Bearing Origin to Destination
- BWC - Bearing using Great Circle route
- DTM - Datum being used.
- GGA - Fix information
- GLL - Lat/Lon data
- GSA - Overall Satellite data
- GSV - Detailed Satellite data
- MSK - send control for a beacon receiver
- MSS - Beacon receiver status information.
- RMA - recommended Loran data
- RMB - recommended navigation data for gps
- RMC - recommended minimum data for gps
- RTE - route message
- VTG - Vector track an Speed over the Ground
- WCV - Waypoint closure velocity (Velocity Made Good)
- WPL - Waypoint information

- XTC - cross track error
- XTE - measured cross track error
- ZTG - Zulu (UTC) time and time to go (to destination)
- ZDA - Date and Time

ในเครื่องรับ gps บางตัวจะมีคำสั่งพิเศษคือ

- HCHDG - Compass output
- PSLIB - Remote Control for a DGPS receiver

เวอร์ชันล่าสุดของ NMEA คือ เวอร์ชัน 2.3 ที่ได้เพิ่มการแสดงสถานะโหมดการทำงานเข้าไปในประโยคหลายประโยค

ตารางที่ 2.2 NMEA 2.0

Name	Garmin	Magellan	Lowrance	Notes:
GPAPB	N	Y	Y	Auto Pilot B
GPBOD	Y	N	N	bearing, origin to destination - earlier G-12's do not transmit this
GPGGA	Y	Y	Y	fix data
GPGLL	Y	Y	Y	Lat/Lon data - earlier G-12's do not transmit this
GPGSA	Y	Y	Y	overall satellite reception data, missing on some Garmin models
GPGSV	Y	Y	Y	detailed satellite data, missing on some Garmin models
GPRMB	Y	Y	Y	minimum recommended data when following a route
GPRMC	Y	Y	Y	minimum recommended data
GPRTE	Y	U	U	route data, only when there is an active route. (this is sometimes bidirectional)
GPWPL	Y	Y	U	waypoint data, only when there is an active route (this is sometimes bidirectional)

ตารางที่ 2.3 NMEA 1.5 อุปกรณ์บางตัวจะไม่รองรับกับ NMEA เวอร์ชั่น 1.5

Name	Garmin	Magellan	Notes:
GPAPA	N	Y	Automatic Pilot A
GPBOD	Y	N	bearing origin to destination - earlier G-12's do not send this
GPBWC	Y	Y	bearing to waypoint using great circle route.
GPGLL	Y	Y	lat/lon - earlier G-12's do not send this
GPRMC	Y	N	minimum recommend data
GPRMB	Y	N	minimum recommended data when following a route
GPVTG	Y	Y	vector track and speed over ground
GPWPL	Y	N	waypoint data (only when active goto)
GPXTE	Y	Y	cross track error

รูปแบบของประโยค NMEA 1083 ที่ใช้ในตัว GPS

RMB

$GPRMB,A,x.x,a,c--c,d--d,llll.ll,e,yyyyy.yy,f,g.g,h.h,i.i,j*kk

RMB = Recommended Minimum Navigation Information

1 = Data Status (V=navigation receiver warning)

2 = Crosstrack error in nautical miles

3 = Direction to steer (L or R) to correct error

4 = Origin waypoint ID#

5 = Destination waypoint ID#

6 = Destination waypoint latitude

7 = N or S

8 = Destination waypoint longitude

9 = E or W

10 = Range to destination in nautical miles

11 = Bearing to destination, degrees True

12 = Destination closing velocity in knots

13 = Arrival status; (A=entered or perpendicular passed)

14 = Checksum

RMC

$GPRMC,hhmmss.ss,A,llll.ll,a,yyyyy.yy,a,x.x,x.x,ddmmyy,x.x,a*hh

RMC = Recommended Minimum Specific GPS/TRANSIT Data

1 = UTC of position fix

2 = Data status (V=navigation receiver warning)

3 = Latitude of fix

4 = N or S

5 = Longitude of fix

6 = E or W

7 = Speed over ground in knots

8 = Track made good in degrees True

9 = UT date

10 = Magnetic variation degrees (Easterly var. subtracts from true course)

11 = E or W

12 = Checksum

GGA

$GPGGA,hhmmss.ss,llll.ll,a,yyyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx*hh

GGA = Global Positioning System Fix Data

1 = UTC of Position

2 = Latitude

3 = N or S

4 = Longitude

5 = E or W

6 = GPS quality indicator (0=invalid; 1=GPS fix; 2=Diff. GPS fix)

7 = Number of satellites in use [not those in view]

8 = Horizontal dilution of position

9 = Antenna altitude above/below mean sea level (geoid)

10 = Meters (Antenna height unit)

11 = Geoidal separation (Diff. between WGS-84 earth ellipsoid and mean sea level. -=geoid is below WGS-84 ellipsoid)

12 = Meters (Units of geoidal separation)

13 = Age in seconds since last update from diff. reference station

14 = Diff. reference station ID#

15 = Checksum

VTG

$GPVTG,t,T,,,s.ss,N,s.ss,K*hh

VTG = Actual track made good and speed over ground

1 = Track made good

2 = Fixed text 'T' indicates that track made good is relative to true north

3 = not used

4 = not used

5 = Speed over ground in knots

6 = Fixed text 'N' indicates that speed over ground in in knots

7 = Speed over ground in kilometers/hour

8 = Fixed text 'K' indicates that speed over ground is in kilometers/hour

9 = Checksum

RMA

$GPRMA,A,llll.ll,N,lllll.ll,W,,,ss.s,ccc,vv.v,W*hh

RMA = Navigation data from present position

1 = Data status

2 = Latitude

3 = N/S

4 = longitude

5 = W/E

6 = not used

7 = not used

8 = Speed over ground in knots

9 = Course over ground

10 = Variation

11 = Direction of variation E/W

12 = Checksum

GSA

$GPGSA,A,3,19,28,14,18,27,22,31,39,,,,,1.7,1.0,1.3*35

GSA = GPS receiver operating mode, SVs used for navigation, and DOP values.

1 = Mode:

M=Manual, forced to operate in 2D or 3D

A=Automatic, 3D/2D

2 = Mode:

1=Fix not available

2=2D

3=3D

3-14 = IDs of SVs used in position fix (null for unused fields)

15 = PDOP

16 = HDOP

17 = VDOP

GSV

$GPGSV,4,1,13,02,02,213,,03,-3,000,,11,00,121,,14,13,172,05*67

GSV = Number of SVs in view, PRN numbers, elevation, azimuth & SNR values.

1 = Total number of messages of this type in this cycle

2 = Message number

3 = Total number of SVs in view

4 = SV PRN number

5 = Elevation in degrees, 90 maximum

6 = Azimuth, degrees from true north, 000 to 359

7 = SNR, 00-99 dB (null when not tracking)

8-11 = Information about second SV, same as field 4-7

12-15= Information about third SV, same as field 4-7

16-19= Information about fourth SV, same as field 4-7

ตัวอย่างข้อมูลที่ได้รับ

$GPRMC,183729,A,3907.356,N,12102.482,W,000.0,360.0,080301,015.5,E*6F

$GPRMB,A,,,,,,,,,,,V*71

$GPGGA,183730,3907.356,N,12102.482,W,1,05,1.6,646.4,M,-24.1,M,,*75

$GPGSA,A,3,02,,,07,,09,24,26,,,,,1.6,1.6,1.0*3D

$GPGSV,2,1,08,02,43,088,38,04,42,145,00,05,11,291,00,07,60,043,35*71

$GPGSV,2,2,08,08,02,145,00,09,46,303,47,24,16,178,32,26,18,231,43*77

$PGRME,22.0,M,52.9,M,51.0,M*14

$GPGLL,3907.360,N,12102.481,W,183730,A*33

$PGRMZ,2062,f,3*2D

$PGRMM,WGS 84*06

$GPBOD,,T,,M,,*47

$GPRTE,1,1,c,0*07

$GPRMC,183731,A,3907.482,N,12102.436,W,000.0,360.0,080301,015.5,E*67

$GPRMB,A,,,,,,,,,,,V*71

ตัวอย่างการถอดรหัสข้อมูล

GGA - essential fix data which provide 3D location and accuracy data.

$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47

Where:

GGA	Global Positioning System Fix Data
123519	Fix taken at 12:35:19 UTC
4807.038,N	Latitude 48 deg 07.038' N
01131.000,E	Longitude 11 deg 31.000' E
1	Fix quality: 0 = invalid
	1 = GPS fix (SPS)
	2 = DGPS fix
	3 = PPS fix
	4 = Real Time Kinematic

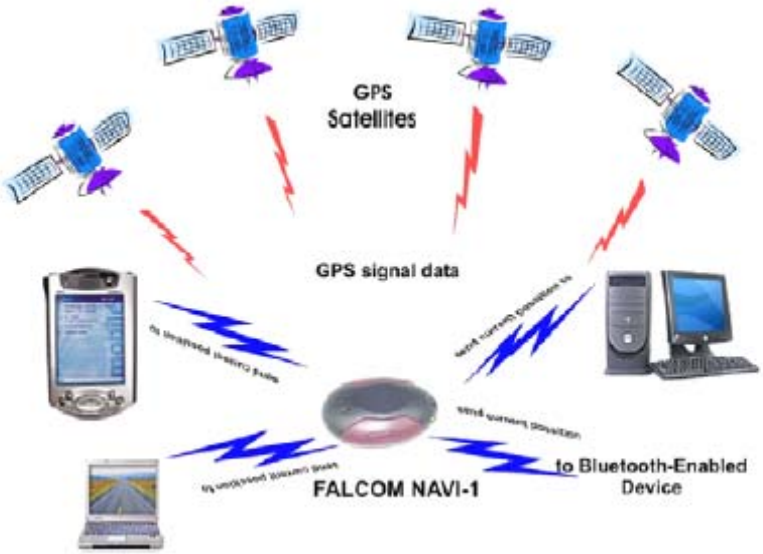
	5 = Float RTK
	6 = estimated (dead reckoning) (2.3 feature)
	7 = Manual input mode
	8 = Simulation mode
08	Number of satellites being tracked
0.9	Horizontal dilution of position
545.4,M	Altitude, Meters, above mean sea level
46.9,M	Height of geoid (mean sea level) above WGS84 ellipsoid
(empty field)	time in seconds since last DGPS update
(empty field)	DGPS station ID number
*47	the checksum data, always begins with *

อุปกรณ์ที่ใช้ในการทดลอง

FALCOM NAVI-1

(Bluetooth GPS Receiver)

รูปที่ 2.3 ตัวอุปกรณ์GPS



รูปที่ 2.4 แสดงการเชื่อมต่อกับตัวอุปกรณ์

รูปแบบการเชื่อมต่อกับอุปกรณ์อื่นๆ

ติดต่อผ่านสัญญาณ Bluetooth โดยใช้การจำลองการทำงานแบบ Serial Profile

Software Interface

อุปกรณ์รับ GPS NAVI-1 จะแสดงข้อมูลในมาตรฐาน NMEA-0183 ที่กำหนดโดย National Marine Electronics Association (NMEA)

NMEA output messages

ตารางที่ 2.4 NMEA output messages

Option	Description
GGA	Time, position and fix type data.
GLL	Latitude, longitude, UTC time of position fix and status.
GSA	GPS receiver operating mode, satellites used in the position solution and DOP values.
GSV	The number of GPS satellites in view satellite ID numbers, elevation, azimuth and SNR values.
RMC	Time, date, position, course and speed data.
VTG	Course and speed information relative to the ground.