รายงานการวิจัย

การพัฒนาระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สาย
(Development of Automatic Ambulance Location System)

ชื่อผู้วิจัย

ผศ.ดร. วนิดา แก่นอากาศ

อาจารย์ สหชัย แก่นอากาศ

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยอุบลราชธานี

โครงการวิจัยนี้ได้รับทุนอุดหนุนการวิจัยจากสำนักงบประมาณแผ่นดิน

ประจำปีงบประมาณ 2547

กิตติกรรมประกาศ

ทางคณะผู้วิจัยขอขอบพระคุณนายแพทย์ไพบูลย์ เพ็ญสุวรรณ หัวหน้าศูนย์อุบัติเหตุ ประจำโรงพยาบาลจังหวัดร้อยเอ็ดที่เอื้อเฟื้อข้อมูล ทีมงาน และความสะดวกในทดลองงานวิจัยในโรงพยาบาล และเจ้าหน้าที่เกี่ยวข้องที่ให้ความสนใจและกล้าที่จะทดลองใช้เทคโนโลยีใหม่ซึ่งดูเหมือนจะเป็นการปรับตัวที่ค่อนข้างยาก แต่ก็ให้ผลที่คุ้มค่าต่อการใช้งาน นอกจากนี้แล้วงานวิจัยนี้ยังได้รับแรงสนับสนุนด้านกำลังใจ และ แนวคิดในการแก้ปัญหาท้องถิ่น ซึ่งเป็นแนวทางหลักของการนำเอาความรู้ทางวิชาการไปการพัฒนางานเพื่อสังคม จากท่านคณบดีคณะวิศวกรรมศาสตร์ และอธิการบดี ศ. ดร. ประกอบ วิโรจนกูฏ

คณะผู้วิจัย

บทคัดย่อ

งานวิจัยนี้ได้นำเสนอระบบและรูปแบบการพัฒนาระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สายซึ่งมีวัตถุประสงค์หลักคือเพิ่มประสิทธิภาพในการรักษาพยาบาลผู้ป่วย โดยจัดเก็บตำแหน่งการเคลื่อนที่ของรถพยาบาลให้เป็นข้อมูลในการบริหารจัดการ และช่วยให้ข้อมูลเส้นทางที่ดีที่สุดในการเดินทางให้กับผู้ขับ ระบบนี้รับข้อมูลพิกัดตำแหน่งโดยใช้เทคโนโลยีจีพีเอส และ วิเคราะห์ตำแหน่งที่ได้โดยระบบในการนำเสนอข้อมูลการเคลื่อนที่ของรถพยาบาล ซึ่งได้ประสบความสำเร็จในทดลอบการใช้งานบนท้องถนน

ABSTRACT

This research presents alternative system architecture, and the prototype implementation of automatic ambulance location system. The objective of the automatic ambulance location system is to enhance performance on medical redemption by keeping track of the ambulance location for a hospital management, and perhaps to provide the driver with optimal directions to his destination. The system receives position data from the GPS receiver and then analyses them to determine the vehicle positions; it also displays the positions of ambulance as a trace of the moving vehicle on the map. The system has been tested, with success on road.

สารบัญ

สารบัญรูปภาพ

บทที่ 1
บทนำ

1. บทนำ

ในการช่วยเหลือชีวิตผู้ป่วย การให้บริการการรักษาในสภาวะฉุกเฉิน เจ้าหน้าที่รักษาพยาบาลฉุกเฉิน (paramedics) มีความจำเป็นที่จะต้องสื่อสารกับโรงพยาบาลอย่างมีประสิทธิภาพ และ สามารถเข้าถึงข้อมูลการรักษา ข้อมูลผู้ป่วย และข้อมูลโรงพยาบาลที่ต้องการได้อย่างทันท่วงที ซึ่งเดิมทีการติดต่อสื่อสารระหว่างเจ้าหน้าที่พยาบาลฉุกเฉิน และโรงพยาบาล ต้องเสียเวลาในการสื่อสารค่อนข้างมากจากการสื่อสารผ่านคลื่นวิทยุ [4] และให้ข้อมูลไม่เพียงพอ และสะดวกนักในการติดต่อ ซึ่งเหตุการณ์ฉุกเฉินเช่นนี้จำเป็นอย่างยิ่งที่จะต้องมีการสื่อสารที่ทันสมัยทันเวลา และข้อมูลที่ใช้ในการสื่อสารเป็นข้อมูลที่ทันสมัย และเป็นปัจจุบัน [11]

นอกจากนี้แล้วในการช่วยชีวิตผู้ป่วย โดยใช้ระบบส่งต่อผู้ป่วยด้วยรถพยาบาลเพื่อให้สามารถนำผู้ป่วยไปรักษายังสถานพยาบาลที่มีเครื่องมือ แพทย์ ยารักษาโรคที่ดีกว่า หรือเหตุการณ์ที่รถพยาบาลรับผู้ป่วยฉุกเฉิน เพื่อต้องการจะนำส่งโรงพยาบาล เป็นเหตุการณ์ที่ต้องการใช้ข้อมูลสื่อสารที่ทันเวลา ในการติดต่อสื่อสารระหว่างรถพยาบาลและโรงพยาบาล ซึ่งปัจจุบันอาศัยระบบการติดต่อผ่านคลื่นวิทยุนั้นมีความจำเป็นอย่างยิ่งในการทราบถึงตำแหน่งการติดตามรถพยาบาล ในการนำส่งผู้ป่วยปัจจุบันนั้นใช้วิทยุติดตามในการติดต่อสื่อสารกับโรงพยาบาลต้นสังกัดและโรงพยาบาลปลายทาง และ ระหว่างรถพยาบาล เพื่อรายงาน สภาพการปฏิบัติงาน การดูแลผู้ป่วย และการส่งต่อผู้ป่วย

ข้อจำกัด และปัญหาดังกล่าว ก่อให้เกิดแนวคิดในการพัฒนา ปรับปรุง แก้ไขปัญหาที่เกิดขึ้นโดยใช้เทคโนโลยีที่เหมาะสม ราคาไม่แพง และสามารถพัฒนาใช้ได้เอง เสริมสร้างศักยภาพในการใช้เทคโนโลยี โดยลดการนำเข้าเทคโนโลยี โดยการพัฒนาระบบการติดตามผ่าน Global Positioning System (GPS) [2] ที่มีการสื่อสารโดยอ้างตำแหน่ง และมีการส่งผ่านข้อมูลผ่านเครือข่ายสารสนเทศทำให้เมื่อมีเหตุฉุกเฉินเจ้าหน้าที่พยาบาลฉุกเฉินสามารถเข้าถึงข้อมูลที่ต้องการใช้ในการพยาบาลเข้าถึงข้อมูลผู้ป่วยจากฐานข้อมูล และสามารถรักษาอาการจาก สถานะการเจ็บป่วยของคนไข้ได้ทำให้ลดปัญหาเรื่องการเข้าถึงข้อมูล [3] และขณะเดียวกันการรายงานตำแหน่งของรถพยาบาลทำให้โรงพยาบาล ยังทำให้สามารถติดตาม สภาพการเดินทางของรถ ทำให้สามารถเตรียมความพร้อมได้ทันท่วงที

2. วัตถุประสงค์ของโครงการ

1. เพื่อลดปัญหารูปแบบการติดต่อสื่อสารของรถพยาบาลและโรงพยาบาลโดยพัฒนาระบบการติดตามรถพยาบาล แบบอ้างอิงอัตโนมัติ
2. เพื่อสร้างระบบข้อมูลการทำงานของรถพยาบาลในการบริการ รักษาผู้ป่วย
3. เพื่อพัฒนาและปรับปรุงรูปแบบการช่วยชีวิตผู้ป่วย ฉุกเฉินด้วยเทคโนโลยีสมัยใหม่ที่ราคาไม่แพง
4. เพื่อเพิ่มประสิทธิภาพในการรักษาพยาบาล ช่วยชีวิตผู้ป่วย จากการเข้าถึงข้อมูล การรับการรักษาพยาบาลที่ถูกต้อง ทันท่วงที
5. เพื่อเสริมสร้างคุณภาพชีวิตให้สูงขึ้น โดยใช้เทคโนโลยีที่เหมาะสม ในการเพิ่มประสิทธิภาพในการรักษาพยาบาล

3. ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถลดปัญหารูปแบบการติดต่อสื่อสารของรถพยาบาลและโรงพยาบาลโดยพัฒนาระบบการติดตามรถพยาบาล แบบอ้างอิงอัตโนมัติ
2. เพิ่มประสิทธิภาพในการรักษาพยาบาล และช่วยชีวิตผู้ป่วยจากเหตุการณ์ อุบัติเหตุและฉุกเฉิน
3. เสริมสร้างศักยภาพในการพัฒนาและใช้เทคโนโลยีที่เหมาะสม
4. เสริมสร้างคุณภาพชีวิตให้สูงขึ้น โดยใช้เทคโนโลยีที่เหมาะสม ในการเพิ่มประสิทธิภาพในการรักษาพยาบาล

4. ขอบเขตของการวิจัย

ข้อมูลที่ใช้ในการพัฒนาระบบติดตาม คัดเลือกจากโรงพยาบาลร้อยเอ็ดและเครือข่ายภายใต้กำกับการทำงานของโรงพยาบาลร้อยเอ็ด เนื่องจากมีความร่วมมือทางวิชาการในการพัฒนาระบบสารสนเทศ และโรงพยาบาลร้อยเอ็ดเป็นหน่วยงานที่มีความพร้อมทางด้านบุคลากร และความรู้ ในการประสานงานวิจัย ซึ่งผลของงานวิจัยสามารถนำไปใช้ได้กับหน่วยงานสาธารณสุขอื่นๆได้ต่อไป

5. การนำเสนอผลการวิจัย

ในรายงานวิจัยนี้ได้กล่าวนำให้เห็นที่มาของแนวคิดในการพัฒนางานวิจัยในบทนี้ และ ในส่วนต่อไปในรายงาน เป็นรายละเอียดของการทำวิจัย โดยได้แบ่งโครงสร้างของเนื้อหาและการนำเสนอตามลำดับดังนี้ บทที่ 2 นำเสนอความสำคัญของปัญหา และ งานวิจัยที่เกี่ยวข้อง บทที่ 3 นำเสนอระเบียบวิธีการทำวิจัย และ แผนการดำเนินงานวิจัย บทที่ 4 นำเสนอผลการทดลองของงานวิจัย ที่ได้ทำใน 1 ปีตามแผนงานวิจัย คือช่วงเดือนตุลาคม 2546 ถึงเดือนกันยายน 2547 และ

บทสุดท้ายคือ บทที่ 5 นำเสนอ บทสรุป ข้ออภิปรายงานวิจัย และ ข้อเสนอแนะ ในการดำเนินงานวิจัยต่อไป

บทที่ 2
ความสำคัญของงานวิจัย และ งานวิจัยที่เกี่ยวข้อง

2.1 ความสำคัญของงานวิจัย

เป็นที่ทราบกันดีว่าการรักษาพยาบาลฉุกเฉินจะไม่เกิดผลดีหากมีความล่าช้า ผู้เจ็บป่วยฉุกเฉินจะเสียโอกาสในการอยู่รอดทุกนาทีที่ผ่านไป และเป็นที่ทราบกันดีอีกว่าการลำเลียงขนย้ายผู้ป่วยที่ไม่เหมาะสมทำอันตรายซ้ำเติมให้แก่ผู้บาดเจ็บ และยังมีหลักฐานแน่ชัดว่าการนำส่งโรงพยาบาลที่ไม่เหมาะสม ทำให้เกิดผลเสียแก่ผู้เจ็บป่วยฉุกเฉินได้อย่างมาก อีกด้วย ความพยายามในการจัดระบบบริการที่เหมาะสมจึงได้เกิดขึ้นเรื่อยมาในอดีตเพื่อแก้ไขข้อบกพร่องดังกล่าว

ระบบบริการการแพทย์ฉุกเฉินมีความหมายถึงการจัดให้มีการระดมทรัพยากรในพื้นที่หนึ่ง ๆ ให้สามารถช่วยเหลือผู้อยู่อาศัยในพื้นที่ได้มีโอกาสขอความช่วยเหลือในกรณีเจ็บป่วยฉุกเฉิน ทั้งในภาวะปกติและในภาวะภัยพิบัติได้ โดยจัดให้มีระบบการรับแจ้งเหตุ ระบบการเข้าช่วยเหลือผู้เจ็บป่วยฉุกเฉินณ จุดที่เกิดเหตุ ระบบการลำเลียงขนย้าย และการส่งผู้เจ็บป่วยฉุกเฉินให้แก่โรงพยาบาลที่เหมาะสม ได้อย่างมีคุณภาพและรวดเร็วตลอด 24 ชม. ระบบดังกล่าวนี้ควรเป็นการรับผิดชอบและดำเนินการโดยหน่วยงานที่รับผิดชอบดูแลท้องถิ่นนั้น ๆ ร่วมกับหน่วยงานที่เกี่ยวข้องต่างๆ และประชาชนในพื้นที่ เป็นระบบที่ต้องมีการดูแลรับผิดชอบโดยแพทย์หรือระบบทางการแพทย์ และควรเป็นระบบที่ไม่มีผลประโยชน์เป็นที่ตั้งหรือแอบแฝง

ในประเทศที่พัฒนาแล้ว ได้มีระบบการลำเลียงขนย้ายผู้ป่วยด้วยยานพาหนะ ที่เรียกว่ารถพยาบาลฉุกเฉิน หรือแอมบูลานซ์ มานานกว่าหนึ่งร้อยปีมาแล้ว เช่นในประเทศสหรัฐอเมริกา ออสเตรเลีย อังกฤษและประเทศในยุโรปอีกจำนวนมาก แต่การจัดให้เกิดเป็นระบบการช่วยเหลือฉุกเฉินจริง ๆ นั้น เริ่มต้นในสหรัฐอเมริกาเมื่อมี ค.ศ.1966 และได้มีการพัฒนาปรับปรุงเรื่อยมาจนกระทั่งปัจจุบัน ในขณะที่ประเทศอื่น ๆ ก็ได้มีการจัดตั้งและพัฒนาในลักษณะเดียวกันแต่จะมีโครงสร้างและการใช้ทรัพยากรแตกต่างกันพอสมควรโดยมีเป้าหมายใหญ่เหมือนกัน คือการทำให้มีการรักษาพยาบาลฉุกเฉินที่รวดเร็วมีคุณภาพอันจะส่งผลให้อัตราการเสียชีวิต พิการ หรือปัญหาในการรักษาพยาบาลลดลง

ในประเทศไทย ได้มีการช่วยเหลือในลักษณะสังคมสงเคราะห์และการกู้ภัย โดยควบคู่กับการเก็บศพผู้เสียชีวิตในกรณีต่าง ๆ ดำเนินการโดยมูลนิธิป่อเต็กตึ๊งมาตั้งแต่ พ.ศ.2480 และมูลนิธิร่วมกตัญญูตั้งแต่ พ.ศ.2513 ซึ่งได้ให้ความช่วยเหลือผู้ป่วยขั้นต้นและลำเลียงนำส่งโรงพยาบาล โดยที่บุคลากรและไม่มีความรู้ความสามารถและไม่มีอุปกรณ์ที่เหมาะสมและถูกวิพากษ์วิจารณ์จากวงการแพทย์ว่าทำให้เกิดความพิการและสูญเสียมากกว่า ได้มีความพยายามเริ่มต้นระบบบริการการ

แพทย์ฉุกเฉินมาเมื่อประมาณ 20 กว่าปีที่ผ่านมาโดยได้มีการประชุมปรึกษาหารือกันหลายครั้ง เพื่อจัดระบบการช่วยเหลือผู้บาดเจ็บ ที่เป็นเครือข่ายของโรงพยาบาลต่าง ๆ ต่อมาได้จัดทำแผนร่วมมือกันระหว่างโรงพยาบาลต่าง ๆ ในกรุงเทพมหานครกับศูนย์ส่งกลับของกรมตำรวจโดยพัฒนาเครือข่ายวิทยุสื่อสารร่วมระหว่างโรงพยาบาลซึ่งมีสังกัดต่างกัน มีระบบรถพยาบาลฉุกเฉินที่ใช้ของศูนย์ส่งกลับเป็นหน่วยงานหลัก ความร่วมมือดังกล่าวมีอุปสรรคตามมาค่อนข้างมากเนื่องจากขาดความร่วมมือของโรงพยาบาลต่าง ๆ ด้วยกันเอง

ต่อมาภายหลังจากมีการปฏิวัติภายใต้การนำของ พล.เอก อาทิตย์ กำลังเอก ได้พัฒนากองกำลังรักษาพระนคร และจัดให้มีโทรศัพท์สายด่วนหมายเลข 123 เพื่อบริการเหตุด่วนแก่ประชาชน ได้จัดให้มีหน่วยรถพยาบาลฉุกเฉินขึ้น มีจำนวน ประมาณ 40 คัน ให้บริการประชาชนในพื้นที่กรุงเทพมหานคร แต่ได้ให้บริการไปไม่นานก็ยุติลงด้วยเหตุผลทางอำนาจและการเมือง

กระทรวงสาธารณสุข[1] โดย กรมการแพทย์ รับงบประมาณสนับสนุนให้จัดทำระบบบริการการแพทย์ฉุกเฉินที่โรงพยาบาลราชวิถีตั้งแต่ปีงบประมาณ พ.ศ.2532 จำนวน 150 ล้านบาท ได้ทำการก่อสร้างอาคาร EMS แล้วเสร็จและเปิดดำเนินการบางส่วนในปี พ.ศ.2536 ได้บรรจุแผนการพัฒนาระบบบริการการแพทย์ฉุกเฉินนี้ไว้ในแผนพัฒนาเศรษฐกิจและสังคมแห่งชาติฉบับที่ 7 (พ.ศ. 2535 – 2539) ได้เริ่มมีการจัดตั้งโครงการศูนย์อุบัติเหตุที่โรงพยาบาลศูนย์ขอนแก่นเมื่อ พ.ศ.2536 ซึ่งมีความครอบคลุมถึงการให้การรักษาพยาบาล ณ จุดที่เกิดเหตุ ต่อมากรุงเทพมหานครโดยวชิรพยาบาล ได้เปิดหน่วยแพทย์กู้ชีวิตขึ้นเป็นทางการเมื่อเดือนธันวาคม 2537 ให้บริการแก่ผู้บาดเจ็บโดยเน้นอุบัติเหตุจราจรและอุบัติภัยต่าง ๆ กรมการแพทย์ ได้เปิดศูนย์กู้ชีพ "นเรนทร" อย่างเป็นทางการเมื่อวันที่ 10 มีนาคมพ.ศ.2538 ให้บริการรักษาพยาบาลฉุกเฉินและขนย้ายทั้งผู้บาดเจ็บและผู้เจ็บป่วยฉุกเฉินในพื้นที่ระยะเวลาไม่เกิน 15 นาทีโดยรอบโรงพยาบาลราชวิถี และต่อมากรมการแพทย์ได้ขยายพื้นที่บริการโดยจัดตั้งศูนย์กู้ชีพเลิดสินและศูนย์กู้ชีพนพรัตนราชธานีขึ้นในปีต่อมา และได้พัฒนาความร่วมมือระหว่างกรมการแพทย์และกรุงเทพมหานครให้มีการแบ่งพื้นที่ในการให้บริการออกเป็น 7 พื้นที่ และมีหมายเลขแจ้งเหตุ 2 หมายเลข คือ 1669 ในส่วนของกรมการแพทย์ และ1554 ในส่วนของพื้นที่กรุงเทพมหานครแต่การให้บริการยังไม่ทั่วถึง ยังขาดงบประมาณที่เหมาะสมในการดำเนินการ โรงพยาบาลหลายแห่งต้องระดมเงินจากมูลนิธิของโรงพยาบาลและขอความช่วยเหลือจากองค์กรภายนอกระบบราชการ

ในช่วงแผนพัฒนาเศรษฐกิจและสังคมแห่งชาติฉบับที่ 8 (พ.ศ.2540 – 2544) ได้มีการบรรจุแผนงานอุบัติเหตุและสาธารณภัยให้มีการจัดตั้งและพัฒนาระบบบริการการแพทย์ฉุกเฉินในทุกจังหวัดโดยเน้นถึงความสามารถในการจัดหน่วยบริการมากกว่าการจัดระบบบริการ เมื่อสิ้นแผนฯ พบว่าโรงพยาบาลศูนย์และโรงพยาบาลทั่วไปจำนวน กว่า 90 แห่งได้จัดให้มีหน่วยปฏิบัติการการ

แพทย์ฉุกเฉินได้ แต่มีข้อจำกัดในการให้บริการเนื่องจากยังไม่มี "ระบบ" อย่างเป็นทางการที่มีกฎหมายและระบบการเงินการคลังรองรับ

ในช่วงแผนพัฒนาเศรษฐกิจและสังคมแห่งชาติฉบับที่ 9 (พ.ศ.2545 – 2549) กระทรวงสาธารณสุขได้กำหนดแผนพัฒนาระบบบริการการแพทย์ฉุกเฉินให้ลงไปสู่ระดับชุมชน โดยเน้นให้ชุมชนมีส่วนร่วมและมีความครอบคลุมพื้นที่ทั่วประเทศ โดยจัดให้มีระบบการเงินการคลังที่เหมาะสมรองรับในปีงบประมาณ พ.ศ.2545 กระทรวงสาธารณสุข ได้ประกาศให้การพัฒนาระบบบริการการแพทย์ฉุกเฉินเป็น นโยบาย 1 ใน 4 ประการของกระทรวงสาธารณสุข จัดตั้งสำนักงานระบบบริการการแพทย์ฉุกเฉิน (ศูนย์นเรนทร กระทรวงสาธารณสุข) เป็นหน่วยงานรับผิดชอบในการพัฒนา จัดงบประมาณในส่วนงบลงทุนจากกองทุนระบบประกันสุขภาพถ้วนหน้าให้จำนวน 10 บาทต่อหัวประชากรที่จดทะเบียน (คาดว่ามีประมาณ 42 ล้านคน) จำนวนเงินประมาณ 420 ล้านบาทเพื่อให้เริ่มดำเนินงานในบางพื้นที่และให้แล้วเสร็จขั้นตอนในการพัฒนาในระยะเวลา 3 ปี หลังจากนั้นแล้วจะจัดให้มีระบบงบประมาณในการบริหารจัดการและดำเนินการระบบโดยมีงบประมาณส่วนหนึ่งจากระบบประกันสุขภาพแห่งชาติ และจากแหล่งเงินทุนต่าง ๆ ตามความเหมาะสม ในอัตรา 18 บาท ต่อหัวประชากรทั้งประเทศ ซึ่งในแต่ละปีจะต้องมีเงินงบประมาณในการสนับสนุนระบบนี้ปีละ 1,200 ล้านบาท(อัตรา 18 บาทต่อหัวประชากร ได้มาจากการศึกษาของคณะวิจัยในสถาบันวิจัยระบบสาธารณสุข ในปีพ.ศ.2543)

ลักษณะการทำงานของระบบบริการการแพทย์ฉุกเฉินโดยทั่วไป แบ่งออกเป็นระยะ ได้ดังนี้

1. การเจ็บป่วยฉุกเฉินและการพบเหตุ (Detection) การเจ็บป่วยฉุกเฉินเป็นเหตุที่เกิดขึ้นเกินอย่างไม่สามารถคาดการณ์ไว้ล่วงหน้าได้ แม้ว่าจะสามารถเตรียมการป้องกันได้ก็ตาม การจัดให้มีผู้ที่มีความรู้ในการตัดสินใจแจ้งเหตุเมื่อพบเหตุ ซึ่งผู้นั้นอาจเป็นผู้เจ็บป่วยเองหรือคนข้างเคียงเป็นเรื่องที่จำเป็นมาก เพราะว่าจะสามารถทำให้กระบวนการช่วยเหลือมาถึงได้รวดเร็ว ตรงกันข้ามหากล่าช้า นาทีที่สำคัญต่อชีวิตของผู้เจ็บป่วยจะหมดไปเรื่อย ๆ จนกระทั่งสายเกินแก้ไขได้

2. การแจ้งเหตุขอความช่วยเหลือ (Reporting) การแจ้งเหตุที่รวดเร็วโดยระบบการสื่อสารที่มีประสิทธิภาพและมีหมายเลขที่จำได้ง่ายเป็นเรื่องที่จำเป็นมากเช่นกัน เพราะว่าเป็นประตูเข้าไปสู่การช่วยเหลือที่เป็นระบบ แต่ผู้แจ้งเหตุอาจจะต้องมีความรู้ความสามารถในการให้ข้อมูลที่ถูกต้องรวมทั้งมีความสามารถในการให้การดูแลขั้นต้นตามความเหมาะสมอีกด้วย

3. การออกปฏิบัติการของหน่วยการแพทย์ฉุกเฉิน (Response) หน่วยปฏิบัติการซึ่งโดยทั่วไปจะแบ่งเป็น 2 ระดับ คือระดับ Advanced Life Support กับระดับ Basic Life Support จะต้องมีความพร้อมเสมอที่จะออกปฏิบัติการตามคำสั่งและจะต้องมีมาตรฐานกำหนดระยะเวลาใน

การออกตัว ระยะเวลาเดินทาง โดยศูนย์รับแจ้งเหตุจะต้องคัดแยกระดับความรุนแรงหรือความต้องการของเหตุและสั่งการณ์ให้หน่วยปฏิบัติการที่เหมาะสมออกปฏิบัติการ

4. การรักษาพยาบาลฉุกเฉิน ณ จุดเกิดเหตุ (On scene care) หน่วยปฏิบัติการจะทำการประเมินสภาพแวดล้อมเพื่อความปลอดภัยของตนและคณะ ประเมินสภาพผู้เจ็บป่วยเพื่อให้การดูแลรักษาตามความเหมาะสม และให้การรักษาพยาบาลฉุกเฉินตามที่ได้รับมอบหมายจากแพทย์ผู้ควบคุมระบบ โดยมีหลักในการดูแลรักษาว่าจะไม่เสียเวลา ณ จุดที่เกิดเหตุ นานจนเป็นผลเสียต่อผู้ป่วย กล่าวคือ ในผู้ป่วยบาดเจ็บจากอุบัติเหตุจะเน้นความรวดเร็วในการนำส่งมากกว่าผู้ป่วยฉุกเฉินทางอายุรกรรม

5. การลำเลียงขนย้ายและการดูแลระหว่างนำส่ง (Care in transit) หลักที่สำคัญยิ่งในการลำเลียงขนย้ายผู้เจ็บป่วยคือการไม่ทำให้เกิดการบาดเจ็บซ้ำเติมต่อผู้เจ็บป่วย ผู้ลำเลียงขนย้ายจะต้องผ่านการฝึกอบรมเทคนิควิธีมาเป็นอย่างดี ในขณะขนย้ายจะต้องมีการประเมินสภาพผู้เจ็บป่วยเป็นระยะ ๆ ปฏิบัติการบางอย่างอาจอาจกระทำบนรถในขณะลำเลียงนำส่งได้ เช่นการให้สารช้า การตามส่วนที่มีความสำคัญลำดับรองลงมา เป็นต้น

6. การนำส่งสถานพยาบาล (Transfer to definitive care) การนำส่งไปยังสถานที่ใดเป็นการชี้ชะตาชีวิตและมีผลต่อผู้เจ็บป่วยได้เป็นอย่างมาก การนำส่งจะต้องใช้ดุลยวินิจว่าโรงพยาบาลที่จะนำส่งสามารถรักษาผู้เจ็บป่วยรายนั้น ๆ ได้เหมาะสมดีหรือไม่ มิฉะนั้นแล้ว เวลาที่เสียไป กับความสามารถที่ไม่ถึงและความไม่พร้อมของสถานพยาบาลนั้น ๆจะทำให้เกิดการเสียชีวิตพิการหรือปัญหาในการรักษาพยาบาลอย่างไม่ควรจะเกิดขึ้นการจัดระบบบริการการแพทย์ฉุกเฉินในแต่ละพื้นที่ควรจะต้องพิจารณาองค์ประกอบหลักเหล่านี้ ได้แก่

1. **ระบบการแจ้งเหตุ** คือการจัดให้มีระบบบริการแจ้งเหตุที่ง่ายต่อการจำ ง่ายต่อการเรียก ง่ายต่อการถ่ายทอดข้อมูล ง่ายต่อการได้รับการช่วยเหลือที่เหมาะสมซึ่งอาจเป็นเพียงคำแนะนำ การจัดหน่วยบริการการแพทย์ฉุกเฉินไปดูแล หรือการจัดหายานพาหนะเพื่อการลำเลียงนำส่งอย่างเดียว ดังนั้นในแต่ละพื้นที่ควรมีศูนย์รับแจ้งเหตุ ซึ่งสามารถรับแจ้งเหตุจากประชาชนได้ด้วยหมายเลขที่จำง่าย เช่น 191 หรือ 1669 เป็นต้น โดยผู้แจ้งสามารถใช้ระบบโทรศัพท์ระบบใดก็ได้ในการแจ้ง เมื่อแจ้งเหตุในพื้นที่หนึ่งควรตรงไปที่ศูนย์รับ แจ้งเหตุของพื้นที่นั้น หากมีข้อผิดพลาดในการแจ้งจะต้องมีระบบเชื่อมโยงต่อให้ศูนย์ที่รับผิดชอบของพื้นที่ได้รับทราบโดยเร็วที่สุด ศูนย์นี้จะต้องทำงาน 24 ชั่วโมง มีเจ้าหน้าที่ซึ่งมีความรู้ในระดับให้คำแนะนำด้านการรักษาพยาบาลขั้นต้นได้ ประจำการ และมีผู้ตัดสินใจสั่งการและรับผิดชอบทางการแพทย์(แพทย์ผู้ควบคุมระบบประจำการหรือติดต่อได้ทันที) ตลอดเวลา

2. **ระบบการสื่อสาร** ได้แก่การจัดให้มีการสื่อสารระหว่างผู้ปฏิบัติงาน ระหว่างผู้ให้บริการ ระหว่างผู้ให้บริการและระบบควบคุมทางการแพทย์ และโรงพยาบาลที่จะนำส่ง ควรมีความสามารถในการส่งผ่านข้อมูลได้ทันทีและมีช่องทางเลือกที่ใช้สำรองในกรณีที่ช่องสัญญาณหลักมีผู้ใช้งานอยู่ ระบบนี้ควรครอบคลุมในพื้นที่ปฏิบัติงานอย่างเต็มที่ ไม่ว่าอยู่ในหุบเขา ในอาคารใหญ่หรือในเมืองที่มีอาคารสูงจำนวนมาก ในปัจจุบันใช้ระบบการสื่อสารชนิด VHF ซึ่งในศูนย์รับแจ้งเหตุจะทำหน้าที่เป็นสถานีแม่ข่ายไปในตัว ระบบนี้เป็นการสื่อสารชนิดเปิดที่ผู้อื่นในเครือข่ายสามารถรับฟังได้ตลอดเวลา ร่วมกับระบบโทรศัพท์เซลลูล่าร์ซึ่งสามารถสื่อสารในรายละเอียดของผู้ป่วยแต่ละรายได้ดี

3. **บุคลากรและการอบรม** ในการออกแบบระบบควรคำนึงถึงบุคลากรว่าจะกำหนดให้ใครทำหน้าที่อะไร ควรคำนึงถึงบุคลากรที่มีอยู่เดิมเป็นหลักและมองไปในอนาคตเพื่อหาความเหมาะสมที่ดีกว่า ในประเทศไทยในระยะเริ่มแรก ได้มีการพิจารณาบุคลากรที่เกี่ยวข้องในระบบ ดังนี้

ก. แพทย์ ทำหน้าที่ควบคุมระบบ เพื่อให้การรักษาพยาบาลที่เกิดขึ้นมีสถานะเหมือนกับที่แพทย์ได้เป็นผู้ให้เอง นอกจากนั้นยังมีบทบาทในการฝึกอบรม การจัดมาตรฐานระบบและการประเมินผล บทบาทนี้เป็นบทบาทที่คล้ายคลึงกันในระบบทั่วโลก

ข. พยาบาล ทำหน้าที่เป็นผู้ให้บริการในระดับสูง (ALS) เป็นผู้ช่วยในระบบควบคุมทางการแพทย์ เป็นผู้สอน และพัฒนาหลักสูตรเจ้าหน้าที่ในระดับต่าง ๆ รวมทั้งประชาชน เป็นผู้บริหารหน่วยปฏิบัติการที่เหมาะสมมาก พยาบาลที่จะทำหน้าที่นี้ควรได้รับการอบรมเพิ่มเติมในหลักสูตรประมาณ 10 ว่าด้วยระบบบริการการแพทย์ฉุกเฉินและ ACLS

ค. เวชกรฉุกเฉิน ในประเทศไทยขณะนี้มีเวชกรฉุกฉินอยู่ 2 ระดับ คือ เวชกรฉุกเฉินขั้นพื้นฐาน (EMT-basic) และเวชกรฉุกเฉินขั้นกลาง (EMT-intermediate)หลักสูตรในการผลิตเวชกรฉุกเฉินขั้นพื้นฐานเป็นแนวทางที่กรมการแพทย์ได้ทำการทดลองในโรงพยาบาล 3 แห่ง รวม 6 รุ่น มีผู้ผ่านการอบรมไปแล้ว 120 คน จากทั้งในส่วนกลางและส่วนภูมิภาค หลักสูตรนี้พัฒนามาจากหลักสูตร EMT-basic ของสหรัฐอเมริกา หลักสูตรเวชกรฉุกเฉินขั้นกลางหรือเรียกว่าเจ้าพนักงานกู้ชีพเป็นหลักสูตรเทียบเท่า EMT-intermediate ของสหรัฐอเมริกา แต่ปรับให้เข้ากับระบบการศึกษาของประเทศไทย ทำเป็นหลักสูตร 2 ปีโดยเริ่มต้นที่ วิทยาลัยสาธารณสุขสิรินธร จังหวัดขอนแก่น (วสส.ขอนแก่น) ขณะนี้ กำลังผลิตรุ่นละ 60 คน ใน วสส.

และวิทยาลัยพยาบาลหลายแห่ง บุคลากร 2 ระดับนี้สามารถให้การรักษาพยาบาลขั้นพื้นฐานได้ และมีบทบาทสำคัญในการช่วยในหน่วยปฏิบัติการระดับสูง ในอนาคตจะมีการพัฒนาเพื่อให้เกิดขั้นบันไดในสายวิชาชีพนี้ให้มีการเรียนการสอนระดับมหาวิทยาลัยที่เรียกว่าเวชการฉุกเฉินขั้นสูงหรือ EMT-paramedic ที่เทียบเท่าปริญญาตรี และมีใบประกอบโรคศิลป์ได้ สามารถให้การรักษาพยาบาลฉุกเฉินในระดับ ALS ได้

ง. ชุดปฏิบัติการปฐมพยาบาล (First responder) โดยทั่วไปหมายถึงเจ้าหน้าที่หน่วยกู้ภัย อาสาสมัคร เจ้าหน้าที่ตำรวจ เจ้าหน้าที่ดับเพลิงหรือกลุ่มบุคคลที่ที่แสดงตนว่า พร้อมที่จะให้การช่วยเหลือและบริการประชาชน มักจะเป็นเจ้าหน้าที่ชุดแรกที่ไปถึงที่เกิดเหตุ ควรมีความรู้พื้นฐานหลักสูตรการอบรม 20 ชม.เป็นขั้นต่ำ (หลักสูตรปฐมพยาบาลสำหรับเจ้าหน้าที่และอาสาสมัคร ของกรมการแพทย์) สามารถให้การประเมินสภาพผู้ป่วยที่บอกได้ว่า ผู้ป่วยต้องการการรักษาพยาบาลในระดับใด หากแน่ใจว่ามีความรุนแรงน้อยสามารถดำเนินการลำเลียงขนย้ายเองแต่หากพบว่ามีความรุนแรงสูงหรือไม่แน่ใจให้เรียกหน่วยบริการการแพทย์ฉุกเฉินมาสนับสนุน

จ. ประชาชนทั่วไป ควรมีความรู้ความสามารถในการบอกได้ว่าผู้เจ็บป่วยที่พบเห็นเป็นผู้ที่ต้องการความช่วยเหลือหรือไม่ รู้จักวิธีป้องกันตนไม่ให้ได้รับอันตรายจากการเข้าช่วยเหลือผู้อื่น รู้จักการแจ้งเหตุและการให้ข้อมูลที่เพียงพอ รู้จักการช่วยเหลือขั้นต้นตามพื้นฐานของตนเพื่อให้การดูแลผู้เจ็บป่วยไปพลางก่อน หลักสูตรในการอบรมประชาชนทั่วไปนี้ควรมีต่ำกว่า 1 วัน

4. **กฎและระเบียบ** ควรมีกฎและระเบียบรองรับการปฏิบัติงานของเจ้าหน้าที่ระดับต่าง ๆ และการคุ้มครองสิทธิของผู้ป่วย ซึ่งทั้งหมดนี้เป็นเรื่องที่จะต้องทำให้เกิดขึ้นเพื่อให้การรักษาพยาบาลฉุกเฉินนี้สามารถเรียกได้ว่าเป็น "ระบบบริการการแพทย์ฉุกเฉิน" กฎและระเบียบข้อบังคับต่าง ๆ อาจอยู่ภายใต้บทบาทและหน้าที่ของกระทรวงสาธารณสุข แต่ในระยะยาวควรมีพระราชบัญญัติรองรับ

5. **การเงินการคลัง** การจัดระบบบริการการแพทย์ฉุกเฉินเป็นสิ่งที่จำเป็นต้องใช้งบประมาณในการจัดตั้งและดำเนินการ ซึ่งแหล่งของงบประมาณอาจมองได้ 2 มุมมอง คือ ส่วนกลางและส่วนท้องถิ่น งบประมาณส่วนกลางอันได้มากจากภาษีอากรของประเทศ ระบบประกันสุขภาพต่าง ๆ ระบบประกันภัยและภาษีอากรในส่วนที่เกี่ยวข้องกับสุขภาพ เช่นภาษีเหล้า ภาษีบุหรี่ ภาษีทะเบียนรถเป็นต้น ควรมีส่วนในการสนับสนุนการสร้างระบบ

ในแต่ละท้องถิ่นและให้งบประมาณสนับสนุนการดำเนินการในลักษณะการซื้อบริการ แก่ท้องถิ่นตามลักษณะและปริมาณงาน งบประมาณส่วนท้องถิ่นอันได้มาจากภาษีท้องถิ่นและงบประมาณสนับสนุนองค์กรท้องถิ่น ควรมีบทบาทในการลงทุนในส่วนใหญ่ของระบบในแต่ละท้องถิ่นทั้งในด้านครุภัณฑ์ บุคลากรและระบบ

6. **การประชาสัมพันธ์** มีความสำคัญในการทำให้ประชาชนที่จะเรียกใช้บริการสามารถเรียกใช้บริการได้อย่างถูกต้องตามความจำเป็นและสมคุณค่า ไม่ทำให้เกิดการใช้งานในด้านฟุ่มเฟือยเกินกว่าเหตุ มีความเข้าใจในระบบงานและเป้าหมายของการทำงาน รวมทั้งมีความรู้สึกเป็นเจ้าของ

7. **การมีส่วนร่วมของชุมชน** ระบบบริการการแพทย์ฉุกเฉินเป็นระบบที่จัดทำเพื่อชุมชน โดยโครงสร้างขององค์กรทุกภาคี ชุมชนควรมีส่วนร่วมที่จะจัดให้มีการทำความเข้าใจของสมาชิกในชุมชนถึงประโยชน์ที่จะได้รับ การจัดกลุ่มอาสมัครภายในชุมชน การส่งเสริมความรู้ การเตรียมความพร้อมและการซ้อมแผนปฏิบัติในกรณีฉุกเฉินต่าง ๆ ที่อาจเกิดกับชุมชนเอง เป็นต้น ตัวแทนของชุมชนควรมีส่วนร่วมในคณะกรรมการระบบการแพทย์ฉุกเฉินของท้องถิ่น

8. **มาตรฐานและโครงสร้างที่เหมาะสม** ระบบบริการการแพทย์ฉุกเฉินในแต่ละพื้นที่ไม่จำเป็นจะต้องมีรูปร่าง มาตรฐานและโครงสร้างที่เหมือนกันหมด แต่ควรมีหลักการใหญ่หรือเกณฑ์มาตรฐานขั้นต่ำเป็นอันเดียวกันโดยเฉพาะอย่างยิ่งในเชิงผลลัพธ์ การกำหนดมาตรฐานกลางควรจะต้องมีขึ้นในคณะกรรมการที่หน่วยงานและองค์กรต่าง ๆ มีส่วนร่วม ในขณะเดียวกันคณะกรรมการของท้องถิ่น ในแต่ละพื้นที่ควรมีบทบาทในการปรับปรุงรายละเอียดของแต่ละท้องถิ่นเอง เพื่อให้เกิดความเหมาะสมกับสภาพภูมิศาสตร์ สังคมเศรษฐกิจและวัฒนธรรมของท้องถิ่น โดยให้มีประสิทธิภาพและผลลัพธ์ในการดำเนินงานที่เทียบเท่าเกณฑ์มาตรฐานกลาง

9. **ระบบข้อมูล** ระบบบริการการแพทย์ฉุกเฉินทั่วประเทศควรมีระบบข้อมูลเป็นอันหนึ่งอันเดียวกัน มีการกำหนดตัวแปรขั้นต่ำร่วมกัน สามารถที่จะเชื่อมโยงกันได้อย่างเป็นปัจจุบันยกเว้นในบางพื้นที่ซึ่งไม่อาจสื่อสารกับพื้นที่อื่นได้เนื่องจากการขาดแคลนระบบสื่อสารที่จำเป็น

10. **การเตรียมพร้อมและการจัดหมวดหมู่ของสถานพยาบาล** ในแต่ละพื้นที่ควรมีการกำหนดโรงพยาบาลสำหรับนำส่งผู้เจ็บป่วยในกรณีสภาพต่าง ๆ เพื่อการตัดสินใจที่ทันการณ์ และเกิดความเป็นธรรมระหว่างสถานพยาบาลกับหน่วยปฏิบัติการ และสะดวกต่อระบบควบคุมทางการแพทย์ ที่ดูแลพื้นที่ป้องกันไม่ให้เกิดความผิดพลาด ที่เกิดจากการ

นำส่งผู้เจ็บป่วย ไปยังโรงพยาบาลที่ไม่เหมาะสม และอาจทำให้เกิดการเสียชีวิต พิการหรือปัญหาในการรักษาพยาบาลได้

11. **การรับผิดชอบโดยระบบควบคุมทางการแพทย์** ระบบบริการการแพทย์ฉุกเฉิน เป็นระบบที่ใช้บุคลากรที่ไม่ใช่แพทย์ออกไปทำหน้าที่ในการรักษาพยาบาลแทนแพทย์ จำเป็นจะต้องมีแพทย์เป็นผู้รับผิดชอบ การรับผิดชอบดังกล่าว อาจทำได้โดยตรงคือการควบคุมสั่งการโดยตรงผ่านวิทยุสื่อสารหรือโทรศัพท์ (Online or Direct) หรือ ทางอ้อม (Offline or Indirect) โดยการผ่านเอกสารมอบหมายที่เรียกว่า Protocol และ Standing order ระบบควบคุมทางการแพทย์ดังกล่าวอาจกระทำโดยแพทย์ที่ได้รับมอบหมายที่เรียกว่า Medical Director หรือโดยคณะกรรมการที่มีแพทย์เป็นผู้รับผิดชอบ

12. **การประเมินผล** การประเมินผลเป็นกิจกรรมที่สำคัญมากเนื่องจากเกี่ยวข้องโดยตรงกับคุณภาพการรักษาพยาบาลและสวัสดิภาพของผู้ป่วย ระบบนี้ไม่สามารถให้บริการโดยไม่มีการเฝ้าดูจากภายนอกและจากประชาชนในพื้นที่ได้เนื่องจากอาจทำให้เกิดการใช้ทรัพยากรผิด และมีผลประโยชน์ส่วนบุคคลเกิดขึ้นได้

2.2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.2.1 ทฤษฎีสารสนเทศภูมิศาสตร์ (Geographic Information Systems: GIS)

ระบบสารสนเทศภูมิศาสตร์หรือ GIS เป็นระบบของโปรแกรมคอมพิวเตอร์ที่มีการนำเข้าข้อมูลการจัดหรือใช้งานเก็บข้อมูลและการเรียกข้อมูลเพื่อปรับปรุงแก้ไขที่มีประสิทธิภาพในการเก็บรวบรวมข้อมูลเชิงพื้นที่และเชื่อมโยงผสมผสานข้อมูลทั้งข้อมูลเชิงพื้นที่และข้อมูลเชิงบรรยายที่เก็บไว้ในฐานข้อมูลสามารถดัดแปลงแก้ไขและวิเคราะห์และนำเสนอข้อมูลเพื่อให้เห็นมิติและความสัมพันธ์ด้านพื้นที่ของข้อมูลซึ่งมีส่วนช่วยให้เกิดความเข้าใจปัญหาและประกอบการตัดสินใจในการแก้ปัญหาเกี่ยวกับการวางแผนใช้ทรัพยากรเชิงพื้นที่ซึ่งมีโปรแกรมสำเร็จรูปหลายรูปแบบตามประเภทของการใช้งาน

I.) องค์ประกอบของสารสนเทศภูมิศาสตร์

เนื่องจากลักษณะข้อมูลของระบบสารสนเทศภูมิศาสตร์มีความซับซ้อนโดยตัวของตัวเองการประมวลผลข้อมูลของระบบสารสนเทศภูมิศาสตร์จึงมักนิยมใช้เครื่องสมองกลที่มีความสามารถสูงมาใช้เป็นหลักทำให้สามารถจำแนกองค์ประกอบของระบบสารสนเทศออกได้เป็น 5 ระบบใหญ่ๆ ดังนี้คือ

1. ระบบฮาร์ดแวร์ (Hardware) ได้แก่ระบบสมองกลและอุปกรณ์ช่วยอาทิ หน่วยประมวลผลกลางหน่วยสำรองข้อมูลหน่วยป้อนข้อมูลและหน่วยแสดงผลเป็นต้น

2. ระบบซอฟท์แวร์ (Software) ได้แก่กลุ่มโปรแกรมที่จำเป็นต้องได้รับการติดตั้งบนระบบฮาร์ดแวร์เพื่อให้ระบบสารสนเทศภูมิศาสตร์สามารถทำงานได้ตามที่ได้รับการออกแบบไว้ โปรแกรมหลักที่จำเป็นได้แก่โปรแกรมระบบเช่นโปรแกรม WINDOW, UNIX เป็นต้น โปรแกรมระบบสารสนเทศภูมิศาสตร์ เช่น โปรแกรม ARC/INFO โปรแกรม INTERGRAPH นอกจากนั้น ยังอาจมีโปรแกรมช่วยงานต่างๆ(Utilities) เช่นโปรแกรมช่วยจัดการหน่วยความจำ โปรแกรมเอดิเตอร์(Editor) อีกด้วย

3. ระบบข้อมูล (Data) แหล่งข้อมูลของระบบสารสนเทศภูมิศาสตร์ที่สำคัญได้แก่แผนที่ภูมิประเทศมาตราส่วน 1:50,000 รูปถ่ายทางอากาศ(Aerial Photographs) หรือภาพถ่ายดาวเทียม (Satellite Imagery) นอกเหนือจากข้อมูลเชิงพื้นที่แล้วระบบสารสนเทศยังต้องการข้อมูลเชิงบรรยายซึ่งขยายความด้านรายละเอียดของข้อมูลเชิงพื้นที่ตัวอย่างของข้อมูลเชิงบรรยายได้แก่ชื่อของหมู่บ้านจำนวนครัวเรือนจำนวนประชากรชาย-หญิงเป็นต้นแหล่งที่มาของข้อมูลเชิงบรรยายอาจได้มาจากหน่วยงานที่เกี่ยวข้องหรือได้มาจากการสำรวจข้อมูลภาคสนาม(Field Data Collection) ก็ได้ข้อมูลเชิงบรรยายจะถูกบันทึกเก็บในลักษณะของบันทึก(Record) โดยแต่ละบันทึกจะถูกแบ่งย่อยออกเป็นช่องสนาม (Field) ช่องสนามแต่ละช่องอาจถูกกำหนดให้บันทึกข้อมูลที่เป็นตัวอักษร (Alphabetic)หรือข้อมูลที่เป็นตัวเลข(Numeric)ก็แล้วแต่ความเหมาะสม

4. บุคลากร (Peopleware) ได้แก่บุคคลที่มีความรู้พื้นฐานทางด้านคอมพิวเตอร์และทางด้านภูมิศาสตร์มาอย่างดีสามารถวิเคราะห์และออกแบบแผนที่และแผนภูมิที่เป็นผลลัพธ์ของการวิเคราะห์เพื่อแสดงผลได้อย่างถูกต้องตามมาตรฐานว่าด้วยวิชาการออกแบบแผนที่ (Cartography) บุคลากรสำหรับงานสารสนเทศภูมิศาสตร์ยังสามารถจำแนกตามภารกิจของการปฏิบัติงานและโดยลักษณะของงานเช่นพนักงานภาคสนามพนักงานเตรียมข้อมูลและต้นร่างพนักงานป้อนข้อมูลพนักงานวิเคราะห์ข้อมูลและพนักงานออกแบบแผนที่เป็นต้น

5. วิธีการการใช้งานสารสนเทศภูมิศาสตร์ที่ประสบความสำเร็จขึ้นอยู่กับแผนงานออกแบบการกำหนดขั้นตอนการปฏิบัติงานเพื่อให้งานเป็นไปตามขั้นตอนมีความเชื่อถือได้และกฎทางธุรกิจที่ดีซึ่งรูปแบบและการปฏิบัติจะแตกต่างไปตามความเหมาะสมของงานแต่ละอย่างจากองค์ประกอบทั้ง 5 ที่ได้กล่าวมาข้างต้นนี้เป็นการยากที่จะระบุว่าองค์ประกอบใดเป็นส่วนที่สำคัญที่สุดเพราะระบบสารสนเทศภูมิศาสตร์ที่ประสบความสำเร็จและมีประสิทธิภาพจะต้องประกอบด้วยองค์ประกอบทั้ง 5 จึงจะเป็นระบบสารสนเทศภูมิศาสตร์ที่สมบูรณ์ภาระกิจที่นำเอาระบบสารสนเทศภูมิศาสตร์มาประยุกต์ใช้จึงจะประสบกับความสำเร็จสมตามเจตนารมณ์ที่ตั้งไว้

2.) การนำเข้าข้อมูลแบ่งได้เป็น 2 ประเภทคือ

1. ข้อมูลแบบอธิบาย (Attribute Data) เป็นข้อมูลทั้งตัวเลขและตัวอักษรที่จัดเก็บได้ในรูปของฐานมูลธรรมดาเช่น MS Access, MS SQL Server, MySQL, Oracle

2. ข้อมูลตำแหน่ง (Position Data) เป็นข้อมูลแผนที่ที่สามารถแสดงเป็นรูปหลายเหลี่ยมหลายมุม (Polygons) เส้น (Lines) หรือ จุด (Points) การนำเข้าต้องใช้เทคนิคและอุปกรณ์ต่างๆเครื่องมือทางด้านคอมพิวเตอร์กราฟฟิกเครื่องสแกนเนอร์ (Scanner) พลอตเตอร์ (Plotter) และโปรแกรมสำเร็จรูปเฉพาะด้านทั้งข้อมูลแบบอธิบายและข้อมูลตำแหน่งจะถูกเก็บอยู่ในรูปของฐานข้อมูลที่เรียกว่า Spatial Data

3.) การจัดเก็บข้อมูลของสารสนเทศภูมิศาสตร์แบ่งได้เป็น 2 ประเภทคือ

1. ระบบสารสนเทศแบบเชิงเส้น (Vector GIS) ระบบนี้จะแสดงตำแหน่งข้อมูลใน 3 ลักษณะคือ จุด (Point) เส้น (Line) และ เส้นขอบเขต (Regions, Polygon) ข้อมูลเชิงพื้นที่ถูกจัดเก็บในลักษณะของเชิงเส้นที่มีโครงสร้างในการกำกับก่อนหลัง,ซ้าย-ขวาโดยการใช้เส้นและจุดเป็นองค์ประกอบพื้นฐานในการจัดเก็บเชิงพื้นที่โครงสร้างของแฟ้มข้อมูลสารสนเทศภูมิศาสตร์ในระบบนี้จะประกอบด้วยเครื่องหมายประจำตัว (ID) ตำแหน่งพิกัด X,Y และตัวชี้ลำดับก่อน-หลังหรือซ้าย-ขวาของข้อมูลข้างเคียงโครงสร้างของข้อมูลระบบนี้จะใช้เนื้อที่ในการจัดเก็บน้อยแต่การปรับปรุงแก้ไขจะทำได้ยากและไม่สะดวกเท่าที่ควรตัวอย่างของสารสนเทศภูมิศาสตร์ระบบนี้ได้แก่ โปรแกรม PC Arc/Info เป็นต้น

2. ระบบสารสนเทศเชิงตารางกริด (Raster GIS) ระบบสารสนเทศที่จัดเก็บข้อมูลในลักษณะตารางกริดนี้จะแบ่งพื้นที่ออกเป็นตารางกริดที่มีรูปเป็นสี่เหลี่ยมจัตุรัสเล็กๆจำนวนมากโดยในรูปสี่เหลี่ยมจัตุรัสเล็กๆเหล่านี้มีศัพท์เรียกเฉพาะว่าหน่วยภาพย่อย (Picture Element) หรือนิยมเรียกสั้นๆว่า Pixel โดยที่แต่ละ Pixel จะเป็นหน่วยที่เล็กที่สุดของข้อมูลถ้าข้อมูลที่มีความละเอียดสูงขนาดของ Pixel ก็จะมีขนาดเล็กแต่ถ้าข้อมูลที่ใช้ในงานสารสนเทศค่อนข้างหยาบขนาดของ Pixel จะมีขนาดใหญ่ข้อดีของระบบข้อมูลแบบ Raster นี้ก็คือภายหลังจากการจัดเก็บแล้วสามารถแก้ไขข้อมูลได้ง่ายสะดวกรวดเร็วและมีประสิทธิภาพแต่ข้อเสียของข้อมูลระบบนี้ก็คือต้องการแฟ้มข้อมูลขนาดใหญ่เพื่อการจัดเก็บหน่วยภาพย่อยทั้งหมดในพื้นที่ตัวอย่างของข้อมูลในระบบ Raster ได้แก่ข้อมูลจากภาพถ่ายดาวเทียมและตัวอย่างของสารสนเทศภูมิศาสตร์ที่ใช้ระบบนี้ในการจัดเก็บข้อมูลได้แก่โปรแกรม SPANS, INTERGRAPH เป็นต้น

4.) ประโยชน์ของระบบสารสนเทศภูมิศาสตร์

เนื่องจากชีวิตประจำวันของคนส่วนใหญ่โดยทั่วไปจะมีความเกี่ยวข้องกับภูมิศาสตร์ไม่มากก็น้อยการตัดสินใจใดๆก็ตามมักจะมีส่วนเกี่ยวข้องทางด้านภูมิศาสตร์เสมอ ดังนั้น เทคโนโลยีสารสนเทศ

ภูมิศาสตร์สามารถช่วยในการจัดการและบริหารข้อมูลเชิงพื้นที่พร้อมทั้งทำให้สามารถเข้าใจในความสัมพันธ์ของสิ่งต่างๆในเชิงพื้นที่ได้เป็นอย่างดีซึ่งเป็นรากฐานที่ดีในการตัดสินใจอย่างฉลาด

การนำระบบคอมพิวเตอร์มาใช้ในเทคโนโลยีสารสนเทศภูมิศาสตร์ทำให้ผู้ใช้สามารถลดเวลาที่ต้องเสียไปในการวิเคราะห์ข้อมูลได้มากเช่นเดียวกับการที่สำนักพิมพ์นำเสนอข่าวสารต่างๆผ่านทางมวลชนได้อย่างรวดเร็วและในราคาถูกเทคโนโลยีสารสนเทศภูมิศาสตร์ก็จะสามารถทำให้ข้อมูลเชิงพื้นที่เป็นที่แพร่หลายและแพร่กระจายไปสู่ผู้ใช้ต่างๆได้ในขณะเดียวกันก็ช่วยลดต้นทุนของการผลิตการปรับปรุงและการเผยแพร่ข้อมูล

นอกจากนี้เทคโนโลยีสารสนเทศภูมิศาสตร์ยังสามารถเปลี่ยนรูปแบบของการวิเคราะห์ข้อมูลเชิงพื้นที่โดยเปลี่ยนวิธีการนำเสนอและการใช้ประโยชน์ข้อมูลเชิงพื้นที่เหล่านั้นข้อมูลเชิงพื้นที่นับว่าเป็นข้อมูลที่สามารถดัดแปลงให้มีความเหมาะสมกับความต้องการด้านต่างๆได้ง่ายโดยการนำเสนอเทคโนโลยีสารสนเทศภูมิศาสตร์เข้ามาช่วยเมื่อเปรียบกับการใช้แผนที่กระดาษเห็นได้ว่าการใช้สารสนเทศภูมิศาสตร์มีข้อได้เปรียบมากกว่าเป็นต้นว่าความสามารถในการปรับปรุงแก้ไขข้อมูลเชิงพื้นที่ให้มีความทันสมัยได้ง่ายกว่าหรือความสามารถในการรวบรวมข้อมูลเชิงพื้นที่ประเภทต่างๆและเก็บไว้ในชุดเดียวกันความสามารถในการปรับข้อมูลเชิงพื้นที่ได้มีการเปลี่ยนแปลงและนำมาผลิตเป็นแผนที่ซึ่งสามารถผลิตฐานข้อมูลเชิงพื้นที่ที่สามารถแสดงขั้นตอนของการเปลี่ยนแปลงได้อย่างต่อเนื่องทำให้สามารถประหยัดค่าใช้จ่ายในการวิเคราะห์และตรวจสอบข้อมูลและทำให้กระบวนการวิเคราะห์ข้อมูลบรรลุผลอย่างรวดเร็วผู้ที่ทำหน้าที่ในการตัดสินใจจะสามารถวางแผนแล้วเปรียบเทียบความเปลี่ยนแปลงที่เกิดขึ้นได้โดยเปลี่ยนรูปแบบของการวิเคราะห์เป็นไปในแบบต่างๆซึ่งผลที่ได้จะสามารถนำเสนอในหลายรูปแบบ

ในทางตรงกันข้ามการวิเคราะห์และการตรวจสอบข้อมูลโดยอาศัยการทำด้วยมือจะทำให้เสียค่าใช้จ่ายสูงเมื่อผู้วิเคราะห์ต้องการนำเสนอผลงานในลักษณะเช่นนี้ปัจจุบันสาขาวิชาการจัดการทรัพยากรธรรมชาติและสิ่งแวดล้อมได้เปลี่ยนรูปแบบได้อย่างรวดเร็วทั้งนี้มีสาเหตุมาจากการเปลี่ยนแปลงทางด้านเทคโนโลยีนำมาใช้รวมทั้งการเปลี่ยนแปลงนโยบายทางการเมืองมีอยู่สอดคล้องกับความต้องการทั้งด้านสังคมและการปกครองในสังคมที่ต้องมีการวิเคราะห์ข้อมูลไม่ว่าจะเป็นทรัพยากรธรรมชาติและสิ่งแวดล้อมที่มีความซับซ้อนและมีการเปลี่ยนอย่างรวดเร็ว

การใช้งานเทคโนโลยีสารสนเทศภูมิศาสตร์ประกอบกับระบบคอมพิวเตอร์ Hardware และ Software ที่มีการพัฒนาอย่างไม่หยุดยั้งทำให้เทคโนโลยีสารสนเทศภูมิศาสตร์เป็นเรื่องกล่าวถึงฐานะที่เป็นเครื่องมือที่มีคุณสมบัติในการบริหารและจัดการทรัพยากรและสิ่งแวดล้อมได้อย่างมีประสิทธิผลเทคโนโลยีสารสนเทศภูมิศาสตร์ไม่ใช่เพียงแฟชั่นที่ผ่านไปแต่เทคโนโลยีสารสนเทศภูมิศาสตร์เป็นเครื่องมือที่ทำให้ทราบถึงข้อมูลเชิงพื้นที่ที่อยู่ระหว่างการเปลี่ยนแปลงหรือที่ได้เปลี่ยนแปลงไปได้ในทุกวันนี้

5.) ระบบแผนที่ (Map System)

แผนที่ที่ใช้ในปัจจุบันแบ่งได้เป็น 2 ชนิดใหญ่ๆคือแผนที่เฉพาะเรื่อง (Thematic map) และแผนที่ภูมิประเทศ (Topographic map) โดยที่แผนที่เฉพาะเรื่องนี้เป็นแผนที่ที่มีองค์ประกอบอื่นๆเข้ามามากส่วนแผนที่ภูมิประเทศจะเป็นแผนที่ที่เน้นแสดงสภาพทางภูมิศาสตร์

แผนที่เฉพาะเรื่องคือแผนที่ที่แสดงรายละเอียดของข้อมูลเชิงพื้นที่ที่ต้องการนำเสนอโดยการแปลงข้อมูลเหล่านั้นให้เป็นเครื่องหมายแผนที่เสียก่อนแล้วนำไปพิมพ์ซ้อนทับลงบนแผนที่ฐานตามตำแหน่งที่ตั้งของข้อมูลนั้นๆซึ่งหมายถึงประกอบไปด้วยข้อมูลเชิงพื้นที่ (Spatial data) และแผนที่ฐาน (Base map) ในการทำแผนที่นี้ เมื่อเตรียมการเสร็จแล้วจะทำการพิมพ์ลงบนกระดาษ (Paper map) สำหรับปัญหาของแผนที่แบบกระดาษคือถ้ามีการเพิ่มเติมหรือไขข้อมูลจะไม่สามารถแก้ไขข้อมูลในเวลาสั้นๆได้จะต้องทำการพิมพ์แผนที่ออกมาใหม่ทั้งหมดทำให้เสียเวลาและค่าใช้จ่ายมาก

ปัจจุบันนี้ได้ทำการดัดแปลงแผนที่เฉพาะเรื่องมาจัดเก็บไว้ในคอมพิวเตอร์โดยนำข้อมูลไปเก็บไว้ในฐานข้อมูลคอมพิวเตอร์ (Database) มีการแสดงผลโดยการวางซ้อนทับฐานข้อมูลการนำคอมพิวเตอร์เข้ามาใช้ในการจัดเก็บข้อมูลนี้จะทำได้ดีกว่าแผนที่กระดาษเพราะว่าสามารถเลือกดูชั้นข้อมูลที่จำเป็นเท่านี้ทำให้เข้าใจง่ายกว่าแผนที่กระดาษ

6.) แผนที่ดิจิตอล

แผนที่ดิจิตอล (Digital map) หรือแผนที่ทางซวเลขเป็นแผนที่ที่ใช้คอมพิวเตอร์ในการประมวลผลและมีการจัดเก็บข้อมูลของแผนที่ให้อยู่ในรูปของข้อมูลคอมพิวเตอร์ซึ่งข้อมูลคอมพิวเตอร์จะทำการจัดเก็บในรูปแบบของฐานข้อมูลคอมพิวเตอร์แผนที่ดิจิตอลแบ่งการจัดเก็บออกเป็น 2 แบบคือแบบราสเตอร์ (raster) และแบบเวคเตอร์ (vector) แผนที่แบบราสเตอร์หมายถึงแผนที่ที่มีการจัดเก็บและแสดงผลในรูปของจุดภาพการสร้างแผนที่แบบนี้ทำได้โดยรับภาพแผนที่จากแผนที่กระดาษผ่านทางเครื่องสแกนภาพ (scanner) ซึ่งวิธีการสแกนภาพเป็นการนำรูปภาพทั้งรูปเข้าไปเก็บในลักษณะของรูปภาพซึ่งการแก้ไขจะทำได้ยากรวมทั้งใช้เนื้อที่ในการจัดเก็บมาก

แผนที่แบบเวคเตอร์หมายถึงแผนที่ที่มีการจัดเก็บและแสดงผลในรูปของลายเส้นและมีทิศทางการสร้างแผนที่แบบนี้ทำได้โดยใช้วิธีการลอกแบบจากเครื่องดิจิไทเซอร์ (digitizer) ซึ่งจะเก็บเฉพาะข้อมูลในส่วนที่ต้องการลอกแบบดังนั้นข้อมูลแบบนี้จึงใช้เนื้อที่ในการจัดเก็บน้อยกว่าสามารถแก้ไขได้ในภาพหลังโดยที่มาตรส่วนไม่ผิดไปจากเดิม

7.) ระบบพิกัดบนแผนที่

ระบบพิกัดบนแผนที่จะมีการอ้างอิงพิกัดที่เหมือนกับระบบพิกัดฉากในทางเลขาคณิตที่ประกอบไปด้วยแกน X และแกน Y โดยจุดกำเนิดหมายถึงจุดระหว่างแกน X และ แกน Y เมื่อแทนด้วย

ระบบพิกัดบนแผนที่แล้วแกน X จะหมายถึงเส้นละติจูดและแกนYจะหมายถึงเส้นลองจิจูดเมื่อพิจารณาระบบพิกัดบนโลกแล้วจะพิจารณาเป็นลักษณะของ 3 มิติคือ X, Y, Z โดย Z จะหมายถึงค่าความสูงระบบนี้จะใช้ในการอ้างอิงในระบบหาพิกัดตำแหน่งด้วยดาวเทียมเป็นหลัก สำหรับงานวิจัยนี้จะพิจารณาเฉพาะเส้นละติจูดและลองจิจูดเป็นหลักเป็นการเปรียบเทียบระหว่างพิกัดในทางเรขาคณิตกับพิกัดบนแผนที่

8.) การคำนวณระยะทางบนแผนที่

เนื่องจากพื้นที่บนแผนที่จะประกอบด้วยตำแหน่งพิกัดมากมายดังนั้นการคำนวณระยะทางจึงหมายถึงระยะห่างระหว่าง 2 ตำแหน่งบนแผนที่ซึ่งเมื่อทราบระบบพิกัดตำแหน่งบนแผนที่แล้วทำให้สามารถหาระยะทางบนแผนที่ได้

ระยะทางระหว่างจุด 2 จุดบนแผนที่ หาได้ตามสูตรต่อไปนี้เมื่อdคือระยะทางระหว่างตำแหน่งทั้งสองและ (x,y) คือพิกัดตำแหน่งใดๆ

$$d = \sqrt{\left((x_2 - x_1)^2 + (y_2 - y_1)^2\right)}$$

ระยะทางระหว่างจุดถึงเส้นตรง หาได้ตามสูตรต่อไปนี้เมื่อ d คือระยะทางระหว่างตำแหน่งทั้งสองและ (x,y) คือพิกัดตำแหน่งใดๆ

$$d = |Ax1 + By1 + C|\sqrt{\left((x_2 - x_1)^2 + (y_2 - y_1)^2\right)}$$

หลักการนี้จะถูกนำไปใช้ในการคำนวณหาพื้นที่บนแผนที่เช่นพื้นที่สี่เหลี่ยมผืนผ้าคำนวณได้จาก d1*d2 เมื่อ d1 คือความกว้าง และ d2 คือความยาวเป็นต้น

2.2.2 ทฤษฎีและหลักการระบบหาพิกัดตำแหน่งด้วยดาวเทียม (Global Positioning System: GPS)

1.) ส่วนประกอบของระบบหาพิกัดตำแหน่งด้วยดาวเทียม

ลักษณะทั่วไปของระบบหาพิกัดตำแหน่งด้วยดาวเทียม ประกอบด้วยส่วนสำคัญ 3 ส่วน ได้แก่

1. ส่วนอวกาศ

ประกอบด้วยดาวเทียมทั้งหมด 24 ดวง โดยจะใช้บอกพิกัด 21 ดวง ส่วนอีก 3 ดวงจะสำรองเอาไว้ ดาวเทียม 24 ดวงนี้ มีวงโคจรอยู่ 6 วงโคจร วงละ 4 ดวง และรัศมีวงโคจรจะสูงจากพื้นโลกประมาณ 20,000 กม. แต่ละวงโคจรจะเอียงทำมุมกับเส้นศูนย์สูตรเป็นมุม 55 องศาในลักษณะสานกันคล้ายลูกตะกร้อ ดาวเทียมแต่ละดวงใช้เวลาโคจรรอบโลก 12 ชั่วโมง ความถี่ที่ใช้ในการบอกตำแหน่งค่าพิกัดของดาวเทียมแต่ละดวงมี 2 ความถี่คือ L1: 1,575.42 MHz และ L2: 1,227.60 MHz

2. สถานีควบคุม

ประกอบด้วย 5 สถานีย่อย(Monitor Station) ตั้งอยู่ ที่เมือง Diego Gacia, Asension Island, Kwajalein, Hawaii และ Colorado Springs ซึ่งที่สุดท้ายทำหน้าที่เป็นสถานีหลัก (Master Control) ซึ่งมีหน้าที่เป็นศูนย์ควบคุมการทำงานของระบบดาวเทียมจี พีเอส เพื่อดูแลความน่าเชื่อถือของระบบ สถานีต่างๆ เหล่านี้ มีหน้าที่คอยติดต่อสื่อสาร (Tracking) กับดาวเทียม ทำการคำนวณผล (Computation) เพื่อบอกตำแหน่งของดาวเทียมแต่ละดวง และส่งข้อมูลที่ได้ไปยังดาวเทียมอยู่ตลอดเวลา

3. ผู้ใช้

ประกอบด้วย 2 ส่วนใหญ่ๆ คื อ ส่ วนที่เกี่ยวข้ องกับพลเรือน (Civilian) และส่วนที่เกี่ยวข้องกับทางทหาร (Military) อุปกรณ์ที่ใช้ในส่วนนี้จะเป็นอุปกรณ์รับสัญญาณระบบหาพิกัดตำแหน่งด้วยดาวเทียม ไปประมวลผลเพื่อแสดงพิกัดตำแหน่งผู้ใช้ทราบ

2.) การหาตำแหน่งของระบบ ระบบหาพิกัดตำแหน่งด้วยดาวเทียม ประกอบไปด้วย 5 ขั้นตอนดังนี้

ขั้นตอนที่ 1 การรับสัญญาณจากดาวเทียมเพื่อให้ได้ตำแหน่ง

สิ่งที่จำเป็นต้องรู้เพื่อใช้ในการคำนวณระยะทางระหว่างดาวเทียมกับเครื่องรับระบบหาพิกัดตำแหน่งด้วยดาวเทียม คือตำแหน่งของดาวเทียมดวงนั้นเพื่อให้ได้ระยะทางที่ถูกต้องเช่นถ้ารู้ระยะห่างของสิ่งที่ต้องการรู้ตำแหน่งกับดาวเทียม 2 ดวง ดังนั้นตำแหน่งที่วัดจะอยู่ที่จุดใดจุดหนึ่งใน 2 จุดที่ดาวเทียมทั้ง 2 ตัดกันเพื่อความถูกต้องแน่นอนยิ่งขึ้นจะใช้ดาวเทียม 3 ดวงในการบอกค่าตำแหน่งแบบ 2 มิติและดาวเทียม 4 ดวงสำหรับบอกตำแหน่งแบบ 3 มิติ

ขั้นตอนที่ 2 การวัดระยะจากดาวเทียม

จากการที่ระบบหาพิกัดตำแหน่งด้วยดาวเทียม ต้องรู้ระยะทางจากเครื่องรับถึงดาวเทียมจึงต้องมีวิธีการหาระยะสามารถหาโดยใช้สมการง่ายๆคือ อัตราเร็วxเวลา ซึ่งคลื่นวิทยุเดินทางด้วยความเร็วแสงคือ 186,000 ไมล์/วินาทีดังนั้นถ้ารู้เวลาแน่นอนในการเริ่มปล่อยและเริ่มรับสัญญาณวิทยุนั้นได้ก็สามารหาเวลาที่สัญญาณเดินทางได้เมื่อเป็นเช่นนี้จำเป็นต้องมีนาฬิกาที่ดีมากเพราะเวลาที่วัดได้จากการเดินทางของแสงจะต้องน้อยมากโดยปกติแล้วถ้าดาวเทียมที่ส่งสัญญาณอยู่เหนือศีรษะพอดีเวลาที่คลื่นวิทยุใช้คือ 0.06 วินาทีเท่านั้น

ผู้ออกแบบเครื่องระบบหาพิกัดตำแหน่งด้วยดาวเทียม ใช้หลักการจำลองแบบสัญญาณที่ส่งจากดาวเทียมและสัญญาณที่อยู่ในเครื่องรับให้เป็นแบบเดียวกันดังนั้นเครื่องทั้งสองจะต้องสร้างรหัสในเวลาที่ตรงกันหรือรหัสสุ่มเทียม (Pseudo Random Code) ซึ่ง

รหัสดังกล่าวจะไม่ซ้ำกันเลยสำหรับดาวเทียมทุกเครื่องดังนั้นสิ่งที่ต้องทำคือการรอรับรหัสที่ดาวเทียมส่งออกมาและมองย้อนว่าเครื่องรับเริ่มสร้างรหัสที่มีลักษณะเหมือนกันแล้วเป็นเวลานานเท่าใดเวลาที่แตกต่างคือเวลาที่คลื่นวิทยุใช้เดินทางข้อดีของการใช้รหัสที่ส่งเป็นชุดคือจะสามารถเปรียบเทียบหาตรงเวลาใดก็ได้ตามต้องการ

รหัสสุ่มเทียมในระบบหาพิกัดตำแหน่งด้วยดาวเทียม ไม่ใช่ตัวเลข ในดาวเทียมและเครื่องรับจะสร้างชุดรหัสเชิงตัวเลขที่ซับซ้อนที่ต้องเป็นตัวเลขที่ซับซ้อนเพื่อให้สามารถนำรหัสทั้งสองมาเปรียบเทียบกันได้โดยง่ายและรหัสซ้ำซ้อนนี้จะมองเห็นเป็นคลื่นวิทยุที่ต่อเนื่องกันยาวๆ

ขั้นตอนที่ 3 การได้เวลาที่ถูกต้อง

แสงเดินทางด้วยความเร็ว 186,000 ไมล์/วินาที ถ้าเครื่องรับนับเวลาพลาดไป 0.01 วินาทีผลคือการวัดจะผิดพลาดไป 1,860 ไมล์สำหรับนาฬิกาในดาวเทียมจะใช้นาฬิกาอะตอมซึ่งจะให้เวลาที่ถูกต้องนาฬิกาดังกล่าวนี้ไม่ได้เดินด้วยพลังงานอะตอมแต่ใช้หลักการวัดจากอนุภาคของสารเฉพาะเหมือนเครื่องเคาะจังหวะอะตอมนี้จะให้เวลาที่แน่นอนและถูกต้องที่สุดเท่าที่มนุษย์ประดิษฐ์มาแต่สำหรับเครื่องรับจะไม่สามารถใช้นาฬิกาอะตอมได้เพราะมีราคาแพงมากจึงต้องใช้วิธีหาเวลาให้ได้ถูกต้องมาใช้งานในเครื่องรับระบบหาพิกัดตำแหน่งด้วยดาวเทียม จากนาฬิกาทีมีความถูกต้องระดับธรรมดาเท่านั้นและวิธีนั้นคือจะต้องทำการวัดระยะจากดาวเทียม (สำหรับกรณีนี้เพิ่มขึ้นอีก 1 ดวงเพื่อใช้ในการปรับแก้เวลาของเครื่องรับที่ไม่สมบูรณ์)

ยกตัวอย่างในกรณีของการวัดแบบ 2 มิติความจริงแล้วถ้าห่างจากดาวเทียมดวงแรก 4 วินาทีและห่างจากดวงที่สอง 6 วินาทีจะได้จุดตัดจากดาวเทียมสองดวงนี้อยู่ 2 จุดคือ A และ B แต่เนื่องจากนาฬิกาเดินช้าไป1วินาทีเครื่องรับจะบอกว่ามีระยะห่างจากดาวเทียม 5 และ 7 วินาทีทำให้ได้จุดตัดเสมือนคือจุด Ax และ Bx ถ้าไม่มีวิธีที่จะรู้ว่านาฬิกาเดินช้าหรือไม่ก็ต้องถือเวลานั้นเป็นเวลาที่ถูกต้องดังนั้นจึงต้องใช้ดาวเทียมอีก1ดวงดังนี้ทำการเพิ่มดาวเทียมดวงที่สามเข้าไปถ้านาฬิกาของเครื่องรับเที่ยงตรงจุดตัดของดาวเทียมทั้งสามดวงจะต้องอยู่ที่จุด A หรือ B แต่เนื่องจากนาฬิกาไม่ตรงจึงตัดกันที่อื่นทำให้เครื่องรับรู้ว่าขณะนี้นาฬิกาเดินไม่ตรงแล้วภายในเครื่องรับระบบหาพิกัดตำแหน่งด้วยดาวเทียม จะมีโปรแกรมที่จะนำเอาผลการวัดที่ไม่ถูกต้องมาคำนวณและหาค่าเวลาที่นาฬิกาเดินคลาดเคลื่อนมาปรับแก้ไขให้ถูกต้องในกรณีของการวัดแบบ3มิติก็เช่นเดียวกันแต่จะเพิ่มเป็นใช้ดาวเทียม 4 ดวง

ขั้นตอนที่ 4 ต้องรู้ตำแหน่งดาวเทียมก่อน

การที่จะให้ทราบตำแหน่งของดาวเทียมซึ่งอยู่ที่ระดับความสูงถึง 11,000 ไมล ซึ่งความสูงระดับนี้ที่ไม่มีคลื่นรหัสจากโลกไปรบกวนได้ ดังนั้นวัตถุที่อยู่สูงขึ้นไปพ้นชั้นบรรยากาศของโลกจะสามารถแสดงวงโคจรได้ด้วยสมการทางคณิตศาสตร์ดาวเทียมระบบหาพิกัดตำแหน่งด้วยดาวเทียม เดินตามวงโคจรตามแนวที่กำหนดไว้แน่นอนกองทัพอากาศสหรัฐมีหน้าที่นำดาวเทียมเข้าสู่วงโคจรในตอนต้นและเนื่องจากในอวกาศไม่มีแรงเสียดทานทำให้ดาวเทียมโคจรอยู่ในวงโคจรที่แน่นอนตามกำหนด

เครื่องรับระบบหาพิกัดตำแหน่งด้วยดาวเทียม สามารถรับตารางดาวเทียม (Almanac) ไว้ในหน่วยความจำได้ตารางดาวเทียมจะบอกว่าในท้องฟ้าจะมีดาวเทียมขึ้น-ลงเวลาใดบ้างและมีการติดตามวงโคจรของดาวเทียมอย่างสม่ำเสมอจากกระทรวงกลาโหมของสหรัฐเพื่อให้ทุกอย่างสมบูรณ์

ข้อที่ 5 การช้าของสัญญาณในการเดินทางผ่านชั้นบรรยากาศ

ในบรรยากาศชั้นไอโอโนสเฟียร์ซึ่งเป็นชั้นของประจุไฟฟ้าอนุภาคเหล่านี้มีผลต่อความเร็วแสงและความเร็วของสัญญาณวิทยุเช่นกันเพื่อสัญญาณวิทยุเดินทางผ่านตัวกลางที่มีความหนาแน่น เช่น ชั้นที่มีประจุไฟฟ้าที่หนาหลายไมล์ย่อมทำให้ความเร็วลดลงบ้างและการที่คลื่นวิทยุเดินทางช้าลงนี้ทำให้ระยะที่ได้ไม่ถูกต้องซึ่งสามารถลดความคลาดเคลื่อนนี้ได้โดยต้องรู้ค่าความเปลี่ยนแปลงเฉลี่ยรายวันตามสภาพบรรยากาศจึงสามารถนำมาเป็นค่าแก้กับทุกค่าที่วัดมาได้

แต่ในความจริงสภาพอากาศจะไม่คงที่ตลอดเวลาดังนั้นการนำค่าเฉลี่ยมาใช้จะไม่ถูกต้องทั้งหมดอีกวิธี คือวัดหาค่าความแปรของสัญญาณวิทยุโดยวัดความเร็วสัมพัทธ์ของสัญญาณสองแบบ (ความถี่ต่างกัน) ที่ส่งจากดาวเทียมเทียบพร้อมกันซึ่งวิธีหลังนี้มักใช้กับเครื่องระบบหาพิกัดตำแหน่งด้วยดาวเทียม ที่มีความละเอียดถูกต้องสูงเรียกว่าเครื่องรับความถี่คู่ (Dual Frequency) หลังจากเดินทางผ่านชั้นไอโอโนสเฟียร์ก็จะถึงชั้นบรรยากาศโลกที่มีละอองน้ำในอากาศซึ่งมีผลต่อความเร็วของสัญญาณเช่นกันแต่ค่าคลาดเคลื่อนดังกล่าวยังไม่มีการปรับแก้ซึ่งจะรวมอยู่ในความคาดเคลื่อนรวมของเครื่องเป็นระยะประมาณ 25 เมตร

3.) รหัสสุ่มเทียม(PseudoRandomCode)

เหตุผลที่สร้างรหัสสุ่มเทียมคือการประหยัดเปรียบเทียบได้จากดาวเทียมทีวีซึ่งกระจาย

เสียงด้วยสัญญาณกำลังแรงดีมากแต่เครื่องรับบนโลกยังต้องใช้จานดาวเทียมรับซึ่งมีขนาดใหญ่ถ้าระบบหาพิกัดตำแหน่งด้วยดาวเทียม ต้องใช้จานรับสัญญาณแบบเดียวกันจะต้องเทอะทะมากและยิ่งกว่านั้นดาวเทียมทีวีหยุดนิ่งแต่ดาวเทียมระบบหาพิกัดตำแหน่งด้วยดาวเทียม เคลื่อนที่ดังนั้นจะยุ่งยากมากขึ้นในการรับสัญญาณเพราะต้องคอยปรับจานตามดาวเทียม

การใช้รหัสสุ่มเทียมช่วยลดความจำเป็นอื่นในการส่งข้อมูลทำให้การส่งสัญญาณระบบหาพิกัดตำแหน่งด้วยดาวเทียม กินไฟน้อยและสัญญาณระบบหาพิกัดตำแหน่งด้วยดาวเทียม อ่อนมากที่ไม่รับเอาสัญญาณวิทยุรบกวนอื่นๆโดยสัญญาณวิทยุรบกวนจะเป็นคลื่นที่ไม่มีรูปแบบซึ่งคล้ายกับรหัสสุ่มเทียมมากแต่มีข้อแตกต่างคือถ้ารู้รูปร่างของรหัสสุ่มเทียมแล้วหากนำไปเปรียบเทียบกับสัญญาณรบกวนการเปรียบเทียบจะแบ่งคลื่นออกเป็นช่วงเวลาจะพบว่าคลื่นรบกวนจะมีโอกาสเหมือนรหัสสุ่ม เทียมประมาณครึ่งหนึ่งจะให้ค่าคลื่นที่เหมือนกันในช่วงเวลานั้นๆเป็น +1 และต่างกันเป็น –1 จะพบว่าหลังจากเปรียบเทียบนานๆจะได้ค่าสุดท้ายเป็น 0 แต่ถ้าเครื่องรับได้รับสัญญาณที่มีรูปแบบเหมือนรหัสสุ่มเทียมจากเครื่องส่งก็จะได้คลื่นที่เข้ากันได้มากขึ้นเรื่อยๆคะแนนก็จะมากขึ้นในช่วงเวลานี้จะส่งกำลังขยายให้แก่สัญญาณดาวเทียมเป็นพันเท่า

รหัสสุ่มเทียมช่วยให้สามารถจับสัญญาณที่อ่อนมากได้ทำให้เครื่องรับระบบหาพิกัดตำแหน่งด้วยดาวเทียม ไม่ต้องใช้ไฟมากและใช้เสาอากาศขนาดเล็กได้เหตุผลอื่นที่ใช้รหัสสุ่มเทียมคือหนึ่งในเวลาสงครามจะสามารถควบคุมไม่ให้ศัตรูใช้ระบบได้รหัสสุ่มเทียมมี 2 แบบคือ รหัสแบบซี/เอ (C/A) และรหัสแบบพี (P) โดยรหัสแบบพีใช้ในราชการทหารเท่านั้นโดยกระทรวงกลาโหมของสหรัฐสามารถลดความถูกต้องของรหัสแบบซี/เอได้โดยใช้มาตรการเลือกผู้ใช้งาน (Select Availability) หรือเอส/เอ (S/A) วิธีเอส/เอที่สำคัญคือการทำให้นาฬิกาดาวเทียมบอกสัญญาณคลาดเคลื่อน ถ้านำเวลานี้ไปใช้ก็จะได้ตำแหน่งที่คลาดเคลื่อนมากเหตุผลอีกข้อคือดาวเทียมทุกดวงสามารถใช้คลื่นความถี่เดียวกันได้โดยไม่รบกวนกันเพราะดาวเทียมแต่ละดวงมีรหัสสุ่มเทียมเป็นของตัวเอง ดังนั้นเวลาเครื่องรับนำรหัสมาใช้ต้องให้ถูกตามหมายเลขดาวเทียมนั้นด้วย

4.) วิธีระบบหาพิกัดตำแหน่งด้วยดาวเทียมผลต่าง (Differential Global Positioning System :DGPS)

การใช้งานระบบหาพิกัดตำแหน่งด้วยดาวเทียม ปกติจะมีความคลาดเคลื่อนอยู่มาก (ประมาณ100 มตรในแนวนอน 156 เมตรในแนวตั้ง) การใช้งานบางอย่างที่ต้องการความถูกต้องสูงจึงต้องใช้ดีระบบหาพิกัดตำแหน่งด้วยดาวเทียม มาช่วยมีหลักการคือจะมีเครื่องรับอ้างอิงซึ่งทราบพิกัดตำแหน่งแน่นอนและมีเครื่องรับระบบหาพิกัดตำแหน่งด้วยดาวเทียม ของผู้ใช้งานอีกเครื่องหนึ่งเครื่องรับอ้างอิงจะนำสัญญาณจากดาวเทียมมาคำนวณหาตำแหน่งของดาวเทียมแต่ละดวงได้เพราะรู้ระยะพิสัยเทียมของเครื่องรับอ้างอิงอยู่แล้วระยะพิสัยเทียมที่คลาดเคลื่อนนี้เรียกว่า"ไบอัส"

จากนั้นเครื่องรับอ้างอิงจะส่งค่าไบอัสซึ่งเรียกว่า ค่าแก้ไขความต่าง (Differential Correction) ไปยังเครื่องรับระบบหาพิกัดตำแหน่งด้วยดาวเทียม ของผู้ใช้งานเพื่อนำไปแก้ไขข้อมูลพิกัดตำแหน่งให้ถูกต้องต่อไป

5.) สัญญาณดาวเทียมระบบหาพิกัดตำแหน่งด้วยดาวเทียม

ดาวเทียมระบบหาพิกัดตำแหน่งด้วยดาวเทียม จะส่งสัญญาณพาหะสองความถี่ (L1,L2) สัญญาณแอล1(L1)มีความถี่ 1575.42เมกกะเฮิร์ซสัญญาณแอล2 (L2)มีความถี่ 1227.6เมกกะเฮิร์ซโดยสัญญาณแอล1 จะถูกผสมกับรหัสรบกวนโดยสุ่ม (Pseudo Random Noise:PRN) 2 รหัส (รวมถึงข้อมูลในก์รนำร่อง) คือรหัสแบบซี/เอ (Coarse/Acquisition code) และรหัสแบบพี (Precision code) สัญญาณแอล2 จะถูกผสมโดยรหัสรหัสรบกวนโดยสุ่ม1รหัส โดยข้อมูลความถี่50 บิต/วินาทีจะถูกรวมเข้ากับรหัสแบบซี/เอ และ พีก่อนที่จะผสมกับสัญญาณแอล1 ซึ่งรวมโดยการมอดดูโลและสำหรับความถี่พาหะแอล2 สามารถผสมกับรหัส P มอดดูโลกับข้อมูลหรือรหัสซี/เอมอดดูโลกับข้อมูลหรือกับรหัสพีอย่างเดียวก็ได้ขึ้นอยู่กับผู้ควบคุมเป็นผู้เลือก

แต่รหัสพีและซี/เอจะไม่เกิดขึ้นพร้อมกันบนความถี่พาหะแอล2 ซึ่งโดยปกติแล้วส่วนควบคุมจะเลือกรหัสพีมอดดูโลกับข้อมูล

6.) ข่าวสารการนำร่อง

ข่าวสารการนำร่องจะบรรจุรายละเอียดเกี่ยวกับฐานเวลาของดาวเทียมวงโคจรของดาวเทียมสถานะของดาวเทียมและข้อมูลการปรับปรุงต่างๆข่าวสารการนำร่องนี้จะถูกส่งที่ความเร็ว 50 บิต/วินาที ข่าวสารมีความยาวทั้งหมด 1500 บิต ใน 1500 บิตจะแบ่งออกเป็น 5เฟรมย่อย (sub frame) เฟรมละ 300 บิตดาวเทียมจะใช้เวลา 30 วินาทีในการส่งข่าวสาร 1ครั้งข้อมูลในแต่ละเฟรมย่อยจะเริ่มต้นด้วยเทเลเมทรีเวิร์ด(Telemetryword: TLM) ซึ่งจะบรรจุรูปแบบการซิงโครไนซ์เวิร์ดที่สองของแต่ละเฟรมย่อยจะเป็นแฮนด์โอเวอร์เวิร์ด (Hand-overword: HOW) ในเฟรมย่อยเฟรมแรกจะบรรจุลำดับจำนวนสัปดาห์ทางระบบหาพิกัดตำแหน่งด้วยดาวเทียม (ระบบหาพิกัดตำแหน่งด้วยดาวเทียม weeknumber)

การทำนายความเที่ยงตรงทางระยะของผู้ใช้งานการเตือนเกี่ยวกับสภาวะของดาวเทียมอายุของข้อมูลการประมาณค่าการหน่วงเวลากลุ่มของสัญญาณและค่าสัมประสิทธิ์ในการจำลองแบบการแก้ไขฐานเวลาของดาวเทียม (Satellite clock correction) ส่วนเฟรมย่อยที่สองและสามจะบรรจุข้อมูลอิฟีเมอริส (Ephemeris Data) ของดาวเทียมส่วนเฟรมย่อยที่สี่และห้าจะแบ่งข้อมูลข่าวสารออกเป็น 25 หน้า โดยส่งครั้งละ 1 หน้าดังนั้นการส่งข้อมูลจะครบ 1 ครั้งต้องใช้เวลา

7.) มาตรฐานเอ็นเอ็มอีเอ(NMEA-National Marine Electronics Association)

เป็นมาตรฐานการเชื่อมต่ออุปกรณ์อิเลคทรอนิกส์สำหรับการสื่อสารและรับส่งข้อมูลระหว่างอุปกรณ์มาตรฐานนี้อนุญาตให้มีตัวส่ง (Talker) ได้ตัวเดียวแต่มีตัวรับ (Listeners) หลายตัวใน1วงจรโดยแนะนำให้ใช้สายชิลด์ทวิสต์แพร์(Shield twisted pair) ในการเชื่อมต่อมาตรฐาน NMEA-0180 และ 0182 เจาะจงว่าข้อมูลที่ออกมาจากตัวส่งต้องเป็น RS-232 หรือจากทีทีแอลบัฟเฟอร์ (TTL buffer) เท่านั้นและห้ามเอาต์พุตโวลต์เตจสูงกว่า +5.7 V ส่วน NMEA-0183 รับในส่วนนี้ได้แต่แนะนำให้ข้อมูลที่ออกจากตัวส่งเป็นอีไอเอ-412 NMEA-0180 และ NMEA-0182 ใช้เฉพาะการติดต่อสื่อสารด้วยโล่แรน-ซีและออโตไพลอตโดยรวมแล้วทั้ง2มาตรฐานนี้มีลักษณะเหมือนกันแต่จะแตกต่างกันที่NMEA-0180ใช้ "รูปแบบประโยคพื้นฐาน (simple format)" ส่วน NMEA-0182 ใช้ "รูปแบบประโยคซับซ้อน (complex format)"

"รูปแบบประโยคพื้นฐาน"เป็นการส่งข้อมูลไบต์เดี่ยวอยู่ในช่วง 0.8-5 วินาทีที่ บอร์ดเรต 1200 โดยที่ใช้พาริตี้คี่บิตที่ 5-0 เป็น Cross-track error บิตที่ 6 เป็น 1 ถ้าข้อมูลถูกต้องและบิตที่ 7 เป็น 0 เพื่อแสดงว่าเป็นข้อมูลชนิดพื้นฐาน

"รูปแบบประโยคซับซ้อน"ประกอบด้วยแอสกี (ASCII) จำนวน 37 ไบต์แสดงละติจูด/ลองติจูดและสถานะข้อมูลจะถูกส่งในช่วง2-8 วินาทีบิตที่ 7 เป็น 1 ในทุกไบต์เพื่อแสดงว่าเป็นข้อมูลชนิดซับซ้อนอนุญาตให้อุปกรณ์ส่งส่งได้ทั้งรูปแบบประโยคพื้นฐานและซับซ้อนและสามารถแทรกข้อมูลแบบพื้นฐานลงในประโยคซับซ้อนได้

NMEA-0183 ทุกอักขระเป็นแอสกีรวมทั้งแคริเอจรีเทอร์นและไลน์ฟีดข้อมูลส่งที่บอร์ดเรต 4800ในรูปแบบประโยคทุกประโยคประกอบด้วย$ตามด้วย2ตัวอักษรที่ระบุตัวส่ง (talker ID) และอีก3ตัวอักษรระบุชนิดประโยค (Sentence ID) (ซึ่งโดยปกติแล้วระบบหาพิกัดตำแหน่งด้วยดาวเทียมที่เลือกใช้จะเป็นตัวกำหนด)ตามด้วยข้อมูลอีกจำนวนหนึ่งทุกส่วนแยกด้วย "," และจบด้วยเช็คซัมหรือเอนเทอร์ประโยคยาวที่สุดได้ 82 ตัวอักษรรวม $ กับ CR/LF

ถ้าไม่มีข้อมูลในช่วงไหนก็จะถูกข้ามไปแต่ตัวลูกน้ำจะยังถูกส่งโดยที่ไม่มีข้อมูลอยู่ภายใน "," เพราะบางครั้งต้องการนับจำนวนข้อมูลโดยใช้ลูกน้ำเป็นตัวนับในประโยคอาจจะมีเช็คซัมหรือไม่ก็ได้ ซึ่งเช็คซัมประกอบด้วย"*"และเลขฐาน 8 อีก 2 หลักที่ได้มาจากการเอ็กครูซีฟออร์(exclusive OR) ทุกอักษรระหว่างลูกน้ำโดยไม่รวม "$" และ "*" เช็คซัมมีความจำเป็น สำหรับบางประโยค เอ็นเอ็มอีเอ-0183 อนุญาตให้กำหนดชนิดประโยคขึ้นได้เองโดยจะต้องขึ้นต้นด้วย "$P" แล้วตามด้วย 3 ตัวอักษรที่เป็นอักษรระบุคนที่ทำชนิดประโยคนั้นๆ (Manufacturer ID) แล้วจึงตามด้วยข้อมูลแสดงตำแหน่งของละติจูดลองติจูดเวลาและค่าต่างๆที่เป็นข้อมูลมาตรฐานของระบบหาพิกัดตำแหน่งด้วยดาวเทียม

2.2.3 งานวิจัยที่เกี่ยวข้อง

จากระบบบริการการแพทย์ฉุกเฉินในขั้นตอนการนำส่งสถานพยาบาล (Transfer to definitive care) ซึ่งเป็นขั้นตอนหลักที่สามารถปรับปรุงคุณภาพโดยใช้เทคโนโลยีคอมพิวเตอร์ที่เหมาะสมมาช่วย อันเป็นเป้าหมายหลักของงานวิจัยนี้เพื่อพัฒนาระบบที่ช่วยในการรักษาพยาบาลผู้ป่วยได้อย่างมีประสิทธิภาพ ทันท่วงทีผ่านระบบเครือข่ายสารสนเทศ ในขณะที่เกิดสภาวะฉุกเฉิน โดยจะพัฒนาระบบติดตามอัตโนมัติใช้กับรถพยาบาล

ปัจจุบันได้มีเทคโนโลยีที่สามารถบอกตำแหน่ง พิกัดของวัตถุ โดยอ้างจากตำแหน่งแผนที่ Global Positioning System (GPS) ซึ่งให้ประโยชน์ในการบอกตำแหน่ง การอ้างตำแหน่งการเคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่ง จากข้อมูลการติดตามการเคลื่อนที่ และ การบอกเวลาที่แม่นยำ [5]

ระบบติดตามอัตโนมัติของรถยนต์ (Automatic Vehicle Location: AVL) โดยใช้เทคโนโลยี GPS-based สามารถแจ้งเหตุฉุกเฉิน รายงานสภาพการเคลื่อนที่ จากตำแหน่งที่ได้อ้างจากแผนที่ ทำให้สามารถแก้ปัญหา [7] สถานการณ์จากข้อมูลตำแหน่งที่ได้ อย่างทันท่วงทีและมีประสิทธิภาพ

การสื่อสารของรถพยาบาลไปยังโรงพยาบาลโดยผ่านระบบติดตามอัตโนมัติ (AVL) [6] จะทำให้โรงพยาบาลสามารถรู้ตำแหน่ง การเคลื่อนที่ของรถพยาบาลได้ นอกจากนี้การรายงานสภาพอาการผู้ป่วยไม่ว่าจะเป็นความดันเลือด ชีพจร อาการป่วย ผ่านระบบติดตามรถพยาบาลไปยังโรงพยาบาล นอกจากตำแหน่งอ้างอิงของรถพยาบาล ทำให้โรงพยาบาลได้ทราบถึงสภาวะอาการป่วยของผู้ป่วย ขณะนำส่งผู้ป่วย

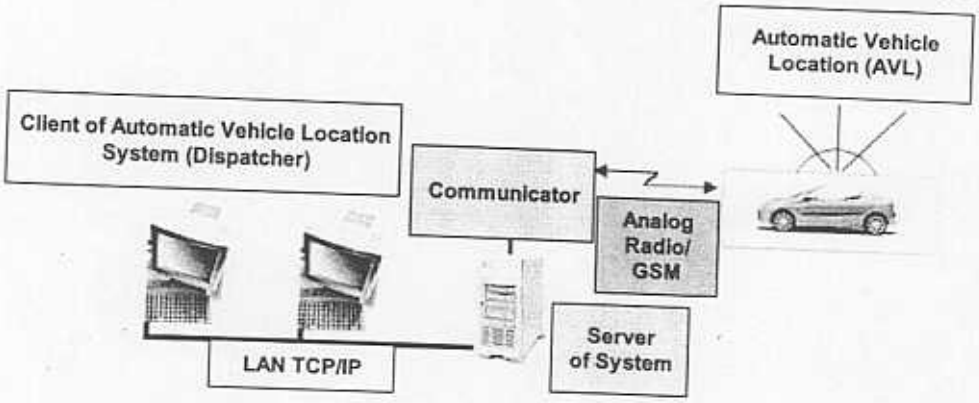
หลักเกณฑ์ในการพัฒนาระบบ

แนวทางหลักในการดำเนินงานวิจัยโครงการนี้ คือ วิจัยรูปแบบและเทคโนโลยีที่เหมาะสมในการพัฒนาระบบ และประเมินประสิทธิภาพของระบบ โดยตัวระบบติดตามรถพยาบาล จะเป็นการพัฒนาในรูปแบบ client-server ซึ่งจะต้องพัฒนาในสองส่วนประกอบหลัก (ดังรูปที่ 2.1) คือ

- ระบบส่งข้อมูลรถพยาบาลไปยังศูนย์ข้อมูล
- ระบบรับข้อมูลรถพยาบาล

ข้อมูลของรถพยาบาลที่จะถูกนำส่งผ่านระบบสื่อสารไร้สายไปยังศูนย์ข้อมูลหลักซึ่งจะจัดเก็บข้อมูลของรถพยาบาลแต่ละคัน โดยข้อมูลนั้นจะประกอบไปด้วย แผนที่ตำแหน่งของรถพยาบาล ซึ่งจะทำให้รถพยาบาลแต่ละคันจะสามารถเข้าถึงตำแหน่งของรถพยาบาลคันอื่นได้ ซึ่งการอ้างตำแหน่งของรถอ้างอิงจากแผนที่ สามารถแสดงผลแบบทันที (Real-time) นอกจากนี้แล้วยังมีข้อมูลประวัติของรถพยาบาล และข้อมูลการควบคุมการทำงาน (Profile and Control) เช่น

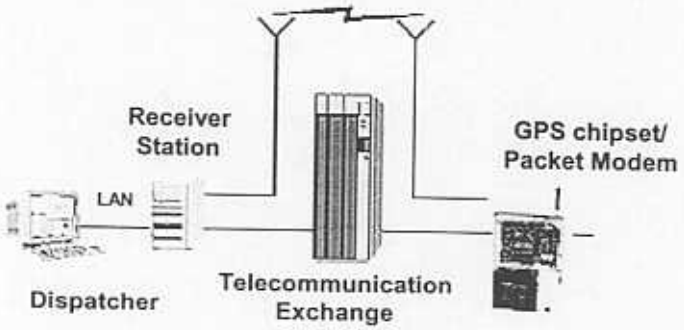
หมายเลขรถ วันเวลาที่ทำการติดต่อ ความเร็วในการเคลื่อนที่ สถานะการทำงานและในการติดต่อสื่อสารโดยสามารถแสดงรายงานสถานะการทำงานของการเชื่อมต่อของสัญญาณวิทยุ [10]



รูปที่ 2.1 ระบบการติดตามรถยนต์อัตโนมัติ

จากรูปที่ 2.1 แสดงการพัฒนาระบบส่ง (client) ซึ่งประกอบไปด้วยอุปกรณ์ GPS chipsets ที่ติดตั้งกับเครื่องคอมพิวเตอร์แบบพกพา ที่ติดตั้งในรถพยาบาล (Automatic Ambulance Location) แล้วทำการส่งข้อมูลตำแหน่งผ่านมือถือใช้เทคโนโลยี Short Message Service (S MS) ไปยังระบบรับคือ ศูนย์ข้อมูลหลัก (Central servers) ในโรงพยาบาล ที่รับข้อมูลที่ได้จากรถพยาบาล แล้วทำการแปลงข้อมูลที่ได้อ้างอิงไปยังแผนที่

นอกจากศูนย์ข้อมูลหลัก (Central servers) จะรับข้อมูลตำแหน่งของรถพยาบาลอ้างอิงกับแผนที่แล้ว ก็นำมาประมวลผลร่วมกับฐานข้อมูลผู้ป่วย ประวัติการรักษา เครื่องมือ และอุปกรณ์รักษา และ จำนวนเจ้าหน้าที่ในโรงพยาบาล ซึ่งสามารถเข้าถึงได้หมอ พยาบาล เจ้าหน้าที่ในโรงพยาบาล และรถพยาบาล [8, 9] ซึ่งข้อมูลที่ประมวลได้จะนำใช้ในการช่วยตัดสินใจในการรักษาผู้ป่วยหรือเพิ่มประสิทธิภาพในระบบส่งต่อ



รูปที่ 2.2 ระบบเครือข่ายสารสนเทศไร้สาย

ในขณะที่ผู้ป่วยเข้าสู่กระบวนการรักษา ระบบข้อมูลที่เชื่อมต่อผ่านระบบเครือข่ายไร้สาย สามารถทำให้ทราบสภาพ อาการของผู้ป่วย และ สถานที่รักษา ซึ่งเป็นประโยชน์ต่อการส่งต่อผู้ป่วย ลดความยุ่งยากในการติดตามข้อมูล และ เกิดประโยชน์ต่อผู้ที่จะเข้าถึงข้อมูลของผู้ป่วยเพื่อใช้ในการรักษาและตัดสินใจ ได้อย่างทันท่วงที

ซึ่งการติดต่อสื่อสารผ่านระบบเครือข่ายไร้สาร (Wireless LANs) รูปที่ 2.2 ทำให้ระบบการทำงานภายใต้โรงพยาบาล และรถพยาบาล สามารถให้บริการผู้ป่วย ได้อย่างรวดเร็ว สามารถเคลื่อนย้ายผู้ป่วยไปรักษาอย่างทันท่วงที ลดข้อผิดพลาด เข้าถึงข้อมูลผู้ป่วยได้อย่างถูกต้อง และรวดเร็ว ทำให้การรักษาพยาบาลมีประสิทธิภาพสูงสุด

บทที่ 3
วิธีการดำเนินการวิจัย

บทนี้นำเสนอระเบียบวิธีการวิจัย ขอบเขตของการวิจัย แผนการดำเนินงานวิจัยตลอดทั้งโครงการ และ แผนที่ได้ดำเนินการจริงในช่วงระหว่างเดือนตุลาคม 2545 ถึงเดือนกันยายน 2547 และเครื่องมือในการทำวิจัย ตลอดจนกระบวนการวิเคราะห์ผลการวิจัย

3.1 ระเบียบวิธีการวิจัย

จากหลักการการพัฒนาที่ได้กล่าวไว้ในบทที่ 2 สามารถนำมาใช้ในการพัฒนาระบบติดตาม โดยมีระเบียบวิธีอันประกอบไปด้วยงานสองส่วนประกอบหลักคือ

(ก) พัฒนาระบบส่งข้อมูล (Client Side)

(ข) พัฒนาระบบรับข้อมูล (Central Server Side)

ดังนี้

1. ทำการศึกษาระบบการรักษาพยาบาลโดยใช้รถพยาบาล ขั้นตอนในการปฏิบัติงาน และเส้นทางในการปฏิบัติงาน
2. จัดเก็บข้อมูลเส้นทางการเดินทางของรถพยาบาล รูปแบบ วิธีการในการรับผู้ป่วย และระบบส่งต่อผู้ป่วย
3. พัฒนาแผนที่เส้นทางเพื่อใช้ในการอ้างตำแหน่งของรถพยาบาลที่ปฏิบัติงานซึ่งใช้ได้
4. พัฒนาระบบฐานข้อมูลข้อมูลประวัติของรถพยาบาล และข้อมูลการควบคุมการทำงาน (Profile and Control) เช่น หมายเลขรถ วันเวลาที่ทำการติดต่อ ความเร็วในการเคลื่อนที่ สถานะการทำงานและในการติดต่อสื่อสาร เส้นทาง และวิธีปฏิบัติงาน
5. พัฒนาระบบส่งข้อมูลที่ได้จากขั้นตอนที่ 2, 3 และ 4 จากรถพยาบาลไปยังศูนย์ข้อมูลหลัก
6. พัฒนาระบบรับข้อมูลจากรถพยาบาล ติดตั้งไว้ที่ศูนย์ข้อมูลหลัก
7. พัฒนาระบบเชื่อมต่อของข้อมูลติดตามอัตโนมัติเข้ากับระบบฐานข้อมูลผู้ป่วย ประวัติการรักษา เครื่องมือ และอุปกรณ์รักษา และ จำนวนเจ้าหน้าที่ในโรงพยาบาล
8. พัฒนาระบบวิเคราะห์ข้อมูลที่ได้จากระบบติดตาม เพื่อประเมินประสิทธิภาพของระบบ
9. ทำการทดลองระบบจากข้อมูลทดสอบ
10. ทำการทดลองใช้จริง เพื่อวิเคราะห์หาข้อผิดพลาดของระบบ
11. ดำเนินการปรับปรุงระบบ เพื่อให้สามารถใช้งานได้อย่างมีประสิทธิภาพ

12. สรุปผลการพัฒนาระบบและวิเคราะห์ประสิทธิภาพของระบบ
13. ดำเนินการถ่ายทอดเทคโนโลยีที่ได้พัฒนาให้ผู้ปฏิบัติงาน ได้สามารถนำเอาระบบไปใช้ได้อย่างมีประสิทธิภาพสูงสุด

3.2 ขอบเขตของการวิจัย

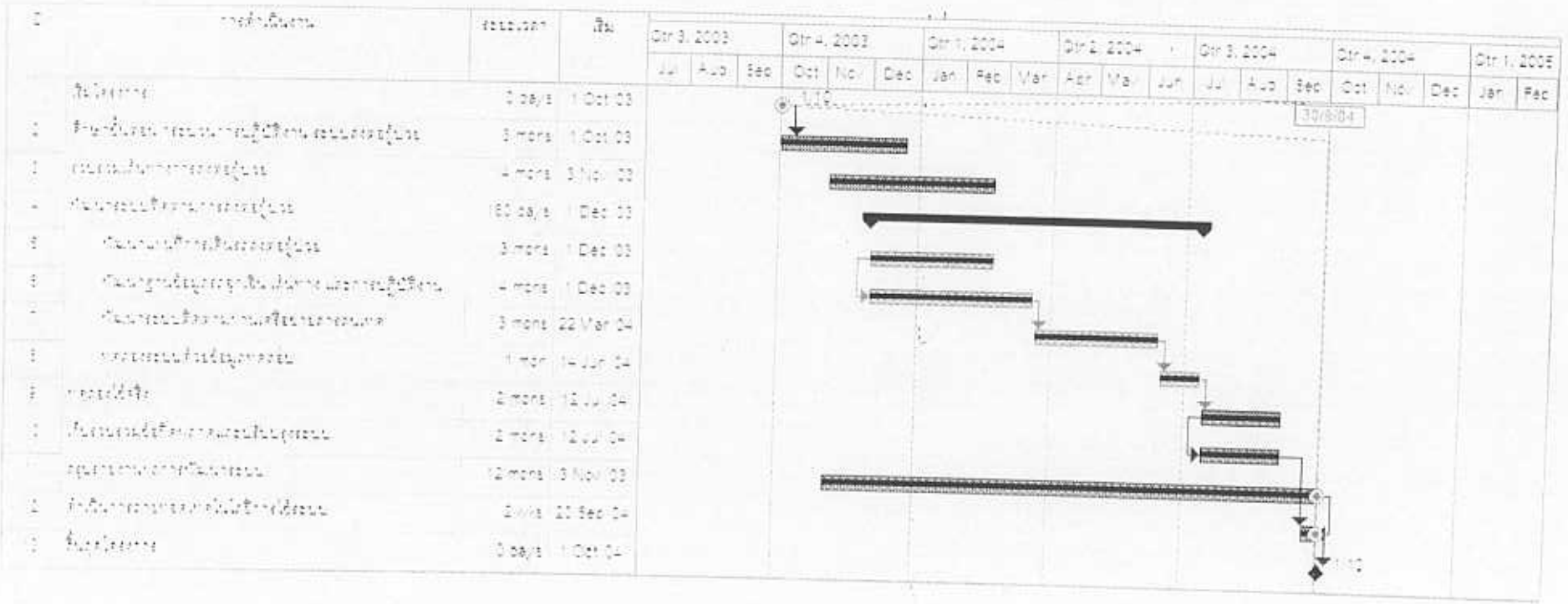
ข้อมูลที่ใช้ในการพัฒนาระบบติดตาม คัดเลือกจากโรงพยาบาลร้อยเอ็ดและเครือข่ายภายใต้กำกับการทำงานของโรงพยาบาลร้อยเอ็ด เนื่องจากมีความร่วมมือทางวิชาการในการพัฒนาระบบสารสนเทศ และโรงพยาบาลร้อยเอ็ดเป็นหน่วยงานที่มีความพร้อมทางด้านบุคลากร และความรู้ ในการประสานงานวิจัย ซึ่งผลของงานวิจัยสามารถนำไปใช้ได้กับหน่วยงานสาธารณสุขอื่นๆได้ต่อไป

3.3 ระยะเวลาดำเนินการวิจัย

งานวิจัยนี้มีกำหนดระยะเวลา 1 ปี โดยเริ่มตั้งแต่ 1 ตุลาคม 2546 ถึง 30 กันยายน 2547 ซึ่งแนวการดำเนินงานวิจัย ได้แสดงในรูปของแผนการดำเนินงานตามขั้นตอน 13 ขั้นตอน และได้นำเสนอผลการดำเนินงานจริง เพื่อเปรียบเทียบกับแผนงานที่วางไว้

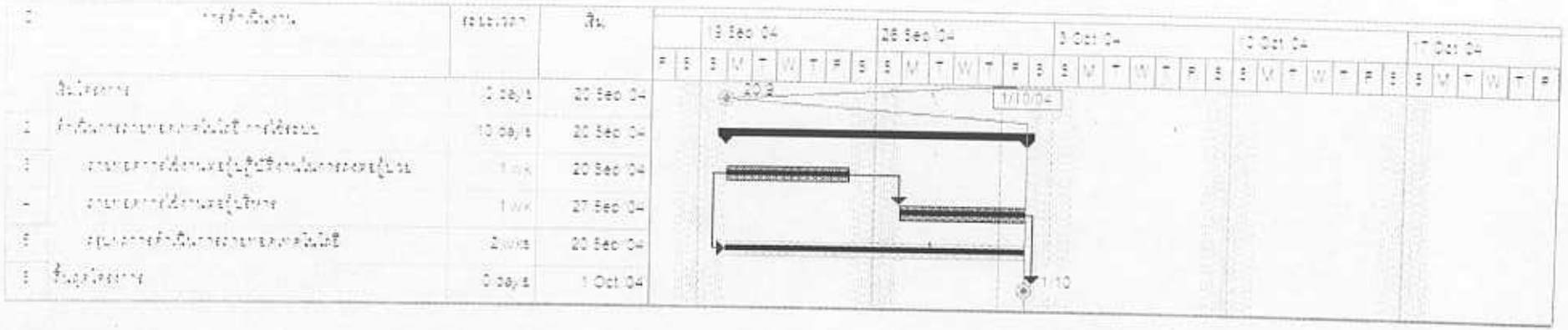
แผนการดำเนินงาน ตลอด 12 เดือน

1. แผนการดำเนินงานในภาพรวม



Ubon Rajathanee University

2. แผนการถ่ายทอดเทคโนโลยีหรือผลการวิจัยสู่กลุ่มเป้าหมาย



หมายเหตุ

◆ 30/9 คือ วันเริ่มต้น และ สิ้นสุดโครงการ (Milestone)

คือ แผนการดำเนินงาน

คือ งานที่ทำสำเร็จ

คือ เส้นแสดงความก้าวหน้าโครงการ ณ วันที่ทำการประเมินผลการทำงาน

Ubon Rajathanee University

3.4 การเก็บรวบรวมข้อมูล

แบ่งเป็น 2 ส่วนคือ

1. ข้อมูลทุติยภูมิ คือการศึกษาเอกสาร งานวิจัยที่เกี่ยวข้องเกี่ยวกับ ระบบติดตามอัตโนมัติของยานพาหนะโดยใช้เทคโนโลยีของ GPS และระบบเครือข่ายสารสนเทศไร้สายในประเทศไทยและต่างประเทศ แหล่งข้อมูลคือ รายงานประจำปี รายงานวิจัย บทความวิชาการ
2. ข้อมูลปฐมภูมิ เป็นการเก็บข้อมูลจากการทดลองใช้งานระบบจริง จากผู้ปฏิบัติงาน และผู้ที่เกี่ยวข้อง

3.5 เครื่องมือในการทำวิจัย

ประกอบไปด้วย 2 ส่วนหลักคือ

I. ระบบส่งข้อมูล ซึ่งใช้อุปกรณ์ในการหาตำแหน่งของรถพยาบาล โดยใช้เทคโนโลยี GPS และการส่งพิกัดตำแหน่งผ่านระบบสื่อสารด้วยเทคโนโลยี Short Message Service(SMS) โดยใช้มือถือในการส่งข้อมูลไปยังระบบรับข้อมูล

2. ระบบรับข้อมูล ใช้อุปกรณ์มือถือในการรับข้อมูลพิกัด แล้วทำการป้อนพิกัดตำแหน่งไปยังระบบฐานข้อมูลแผนที่การส่งต่อของรถพยาบาล ซึ่งพัฒนาด้วยเทคโนโลยีของ JAVA และระบบฐานข้อมูลบนเครือข่ายสารสนเทศคือ MySQL

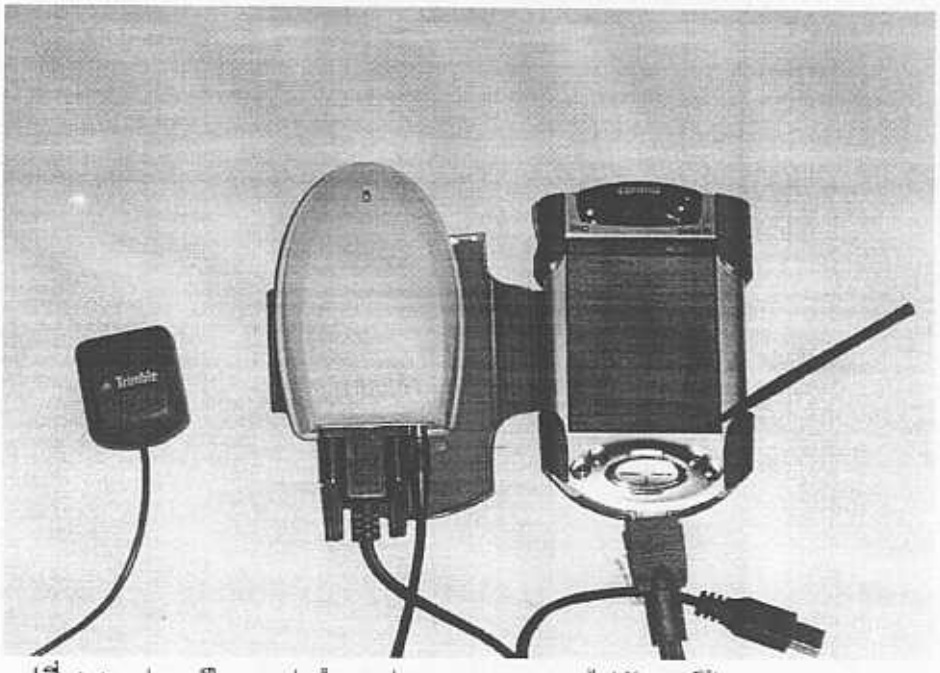
บทที่ 4
ผลการวิจัย

ผลของวิจัยที่นำเสนอในบทนี้ นำเสนอผลของการพัฒนาระบบติดตามพาหนะอัตโนมัติซึ่งใช้หลักการของเทคโนโลยีพิกัดภูมิศาสตร์ ที่ใช้ในการติดตามตำแหน่งของยานพาหนะและเทคโนโลยีเครือข่ายสารสนเทศในการจัดเก็บข้อมูลพิกัด แสดงผลของตำแหน่งเพื่อใช้ข้อมูลนี้ในการบริหารจัดการ ตามประเภทการใช้งานของยานพาหนะ ในงานวิจัยนี้ได้ใช้เทคโนโลยีนี้กับรถพยาบาลฉุกเฉิน เพื่อต้องการทราบตำแหน่งของการเดินทางของรถพยาบาลฉุกเฉิน นำมาใช้ในการบริหารระบบส่งต่อผู้ป่วยที่มีประสิทธิภาพ

ระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สายนั้นแบ่งงานพัฒนาออกเป็นสองส่วนคือ ระบบส่งข้อมูล และ รับข้อมูล

4.1 ระบบส่งข้อมูล

ในรถพยาบาลจะมีอุปกรณ์ GPS ร่วมกับ Pocket PC ดังรูปที่ 4.1 ที่ใช้ในการรับตำแหน่งพิกัดของรถ และมือถือในการส่งพิกัดตำแหน่งผ่านเทคโนโลยี SMS ไปยังศูนย์ข้อมูลรถพยาบาลที่โรงพยาบาลจังหวัดร้อยเอ็ด



รูปที่ 4.1 อุปกรณ์ในการส่งตำแหน่งของรถพยาบาลไปยังศูนย์ข้อมูลรถพยาบาลโรงพยาบาลจังหวัดร้อยเอ็ด

4.1.1 โปรแกรมการส่งตำแหน่งพิกัดโดยส่งแบบข้อความผ่านมือถือด้วยเทคโนโลยีจาวา (SMS Transceiver Program)

อุปกรณ์ GPS ที่ใช้ในการรับพิกัดตำแหน่งของการเคลื่อนที่ของรถ ข้อมูลตำแหน่งจะถูกส่งผ่านไปยังมือถือโดยมีโปรแกรมการส่งข้อความผ่านมือถือที่ถูกพัฒนาด้วยเทคโนโลยีจาวาบนอุปกรณ์ Pocket PC โดยอุปกรณ์ GPS ซึ่งต่อเข้ากับ Pocket PC นั้นจะมีมือถือที่ต่อพ่วงเข้าผ่านพอร์ตอนุกรม (serial line connection) และใช้คำสั่ง AT commands ที่เป็นคำสั่งเดียวกันกับที่ใช้กับโมเด็ม หากแต่ความแตกต่างจะอยู่ตรงที่การเข้ารหัสของข้อมูลที่ส่งผ่านมือถือ นั้นจะต่างกับรูปแบบรหัสข้อมูลที่ส่งผ่านโมเด็ม

คำสั่ง AT commands สำหรับอุปกรณ์มือถือนั้นถูกกำหนดขึ้นโดยมาตรฐาน ETSI standards ซึ่งมีรูปแบบดังนี้

TS 27.005 Data Circuit terminating Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS)
TS 07.07 AT command set for GSM Mobile Equipment (ME)
TS 27.007 AT command set for 3G User Equipment (UE)
TS 23.040 coding of PDUs

โปรแกรมการส่งตำแหน่งพิกัดโดยส่งแบบข้อความผ่านมือถือด้วยเทคโนโลยีจาวา (SMS Transceiver) เป็นโปรแกรมที่ใช้ในการส่งพิกัดตำแหน่งรถพยาบาลด้วยเทคโนโลยี Short Message Service (SMS) เป็นอุปกรณ์พื้นฐานในระบบมือถือ ซึ่งได้รับการติดตั้งเป็นมาตรฐานภายในเครือข่ายโทรศัพท์เคลื่อนที่ทั่วไป การทำงานโดยพื้นฐานเป็นพิเศษ อุปกรณ์ SMS มีรูปแบบการทำงานแบบ "รับและส่งต่อ" (Store and Forward) กล่าวคือทำหน้าที่เก็บรวบรวมข้อมูล Short Message จากแหล่งต่างๆ ไม่ว่าจะมาจากการส่งผ่านเครือข่ายอินเทอร์เน็ต, จากผู้ใช้บริการโทรศัพท์เคลื่อนที่ด้วยกันเอง หรือเป็นข้อมูล ที่ส่งมาจากอุปกรณ์อื่นๆ ภายในระบบเครือข่าย แล้วนำมาส่งต่อไปยังจุดหมายปลายทาง ซึ่งมักจะเป็นเครื่องลูกข่าย โทรศัพท์เคลื่อนที่ หรืออุปกรณ์ควบคุมต่างๆ ภายในระบบเครือข่าย มีรูปแบบการทำงานคล้ายๆ กับหน่วยงานที่ทำหน้าที่ รับส่งไปรษณีย์ คุณสมบัติของ SMS ที่น่าสนใจก็คือ ความสามารถในการเชื่อมต่อกับอุปกรณ์ภายนอกได้โดยใช้โปรโตคอลได้หลายๆ ประเภท ไม่ว่าจะเป็นโปรโตคอลทางด้านโทรคมนาคม เช่น Mobile Application Part (MAP) หรือโปรโตคอลทางด้านระบบเครือข่ายคอมพิวเตอร์ เช่น SMPP ซึ่งสามารถนำไปเชื่อมต่อกับ เครื่องคอมพิวเตอร์เซิร์ฟเวอร์ ซึ่งทำหน้าที่ดูแลบริการพิเศษต่างๆ ได้โดยตรง โดยใช้ javax.comm Service จากบริษัท Sun ในการส่งข้อมูลผ่านพอร์ตอนุกรมซึ่งต่อตรงกับอุปกรณ์มือ Nokia 6210 ซึ่งใช้ในการวิจัย โดยหลักการทำงานนั้นโปรแกรมจะทำการ query อุปกรณ์มือถือเป็นช่วงเวลา ซึ่งในการวิจัยนี้จะกำหนดให้มีการเช็คข้อความทุก 1 นาทีเพื่อเช็คดูข้อความที่ส่งเข้ามา (incoming message) และเพื่อตรวจพบก็จะทำการส่งข้อความนั้นออกไปจัดเก็บไว้ตารางข้อความซึ่งใช้ database (jdbc) link ในการเชื่อมต่อกับตารางข้อความที่ถูกจัดเก็บ

ในรูปของ text หรือ XML file ซึ่งข้อมูลที่จัดเก็บจะถูกจัดเก็บในฐานข้อมูลดังในตารางที่แสดงในรูปที่ 4.8 ดังรายละเอียดโครงสร้างโค๊ดโปรแกรมจัดเก็บในภาคผนวก ค.

4.2 ระบบรับข้อมูล

ศูนย์ข้อมูลรถพยาบาล ในโรงพยาบาลจังหวัดร้อยเอ็ดจะประกอบไปด้วยอุปกรณ์รับข้อมูลพิกัดตำแหน่งของรถพยาบาลผ่านมือถือ ซึ่งทำการเชื่อมต่อข้อมูลเข้ากับตัว Server ซึ่งเป็นเครื่อง PC ที่มีระบบโปรแกรมการส่งตำแหน่งพิกัดโดยส่งแบบข้อความผ่านมือถือด้วยเทคโนโลยีจาวาที่ใช้ในการรับตำแหน่งผ่านมือถือที่เชื่อมต่อกับเครื่อง PC ผ่านพอร์ตอนุกรม และระบบฐานข้อมูลรถพยาบาลซึ่งพัฒนาด้วยเทคโนโลยี JAVA และฐานข้อมูล MySQL ที่ทำการจัดเก็บแผนที่ของจังหวัดร้อยเอ็ด ข้อมูลรถพยาบาล และข้อมูลพิกัดตำแหน่งที่ได้จะถูกนำเสนอในรูปของแผนที่ตำแหน่ง ที่สามารถใช้งานตามหลักแผนที่ทางภูมิศาสตร์คือ zoom, pan, identify และ queries

การพัฒนาระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สายโดยใช้ภาษา JAVA ซึ่งเป็น Object Oriented Programming (OOP) จาวาเป็นภาษาหนึ่งที่อาจเรียกได้ว่าสืบทอดมาจากภาษา C++ อีกทีหนึ่ง เป็นภาษาที่รวมความเป็นพื้นฐานทางภาษาของจาวาอยู่ในรูปของภาษา C++ ทั้งการกำหนดฟังก์ชั่น การสร้างอ๊อปเจ็กต์ต่าง ๆ และได้ตัดความซับซ้อนและข้อเสียบางอย่างออกไป

พื้นฐานการเขียนโปรแกรมภาษาจาวาต้องรู้จักกับการเขียนโปรแกรมแบบ OOP เสียก่อนสักเล็กน้อย เพื่อให้การเขียนโปรแกรมเป็นไปได้ง่ายขึ้น ความสามารถในการทำงานข้ามระบบของจาวาทำให้เป็นข้อดีที่งานวิจัยนี้เลือกที่ใช้เป็นเครื่องมือในการพัฒนาระบบ เพื่อที่ตัวระบบจะไม่ยึดติดกับสภาพ และ รูปแบบการใช้งานคอมพิวเตอร์ที่แตกต่างของผู้ปฏิบัติงาน

ในการสร้างโปรแกรมหรือจาวาแอพเพล็ตต้องสร้างซอร์สโค๊ดภาษาจาวาและทำการคอมไพล์โปรแกรมด้วย Java Compiler โปรแกรมจะอยู่ในรูปของไบต์โค๊ด (Byte Code) และนำไปรันในระบบต่าง ๆ ได้หลายระบบเช่น Unix , Mac หรือ Windows ซึ่งเรียกการทำงานลักษณะนี้ว่า การข้ามระบบหรือ Cross Plateform

คุณสมบัติของภาษาจาวาข้อนี้อาจจะไม่พบในการเขียนโปรแกรมบนวินโดวส์ด้วยภาษาอื่น ๆ เช่น Visual Basic หรือ Visual C++ ซึ่งไม่สามารถนำไปรันบนระบบอื่นได้ ยกเว้นระบบวินโดวส์อย่างเดียวแต่จาวาได้ทำลายกำแพงกั้นระหว่างระบบนี้ลงได้แล้ว

จาวาได้เตรียมไลบรารี่และส่วนประกอบจำเป็นต่าง ๆ ในการเขียนโปรแกรมมาให้ เพื่อให้สามารถพัฒนาโปรแกรมได้ง่าย ๆ และธรรมดาที่สุด เหมือนการกลับไปสู่การพัฒนาโปรแกรมแบบเดิม โดยใช้โปรแกรม Notepad ธรรมดา ๆ หรือโปรแกรม Vi ในยูนิกส์ ก็สามารถสร้างโปรแกรมภาษาจาวาได้ ทำให้เกิดความง่ายและสะดวกต่อผู้ปฏิบัติงานในการแก้ไข เพิ่มเติมระบบได้ในภายหลัง

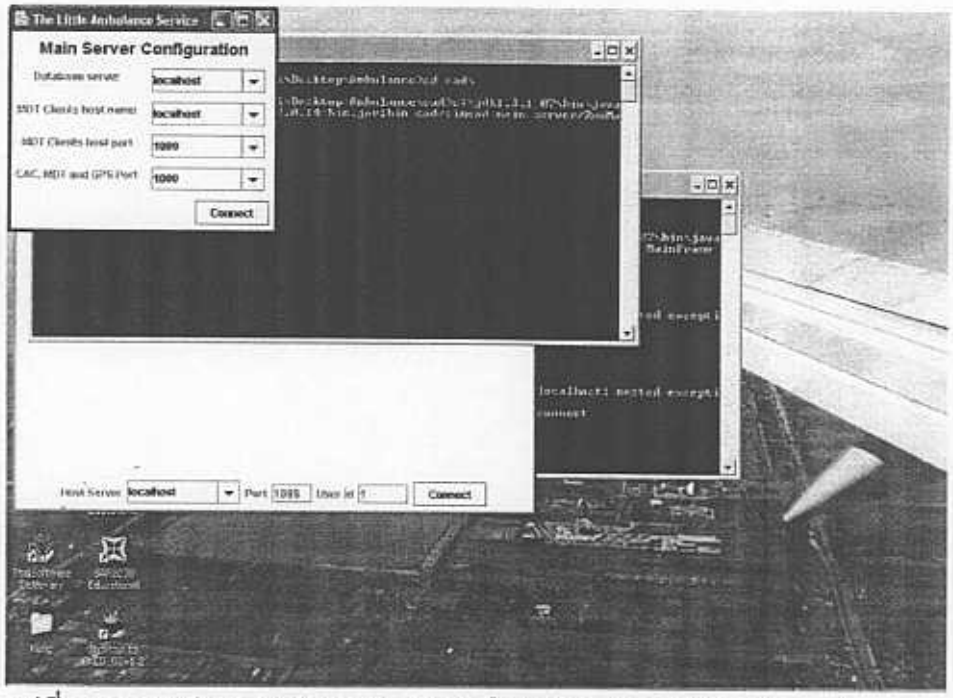
ระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สาย มีส่วนการทำงานดังต่อไปนี้ รูปภาพที่ 4.2 เป็นส่วนเริ่มต้นของการใช้งานโดยผู้ใช้ต้องทำการกำหนดการเข้าใช้งานระบบฐานข้อมูลรถพยาบาล ซึ่งได้ถูกออกแบบให้มีการจัดเก็บตำแหน่งพิกัดที่ได้จากมือถือ ซึ่งจะทำการอ่านข้อมูลที่โปรแกรมการส่งตำแหน่งพิกัดโดยส่งแบบข้อความผ่านมือถือด้วยเทคโนโลยีจาวา ที่จัดเก็บข้อมูลพิกัดตำแหน่งในรูปของ text file ไปจัดเก็บลงไปในระบบฐานข้อมูล database server และอยากออกแบบให้สามารถรับข้อมูลพิกัดตำแหน่ง ด้วยระบบ MDT (mobile data terminal) ซึ่งกำหนดให้จัดเก็บผ่าน TCP Port 1099 นอกจากการรับข้อมูลพิกัดผ่านระบบ SMS เมื่อทำการกำหนดการเข้าใช้งานของระบบฐานข้อมูลเรียบร้อยแล้วระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สายก็จะเริ่มเข้าสู่การใช้งานหลักดังในรูปที่ 4.3 และรูปที่ 4.4 จะเป็นส่วนในการวิเคราะห์ประเภทของผู้ใช้ ซึ่งจะต้องทำการ Authenticate ให้ระบบทราบถึงข้อมูลผู้ใช้ว่าเป็นผู้ดูแลระบบ (Super user) หรือ ผู้ปฏิบัติงาน (user)

โครงสร้างการใช้งานจริงของระบบถูกนำเสนอในรูปที่ 4.6 ซึ่งจะประกอบด้วย 3 ส่วนใช้งานหลักคือ ข้อมูลรถพยาบาล (Call Assistant) ข้อมูลพิกัดตำแหน่งของรถพยาบาล (Tracker) และการจัดสรรการใช้งานระบบโดยผู้ดูแลระบบ (Super user)

ในส่วนที่ 1 ข้อมูลรถพยาบาลจะถูกแบ่งรายละเอียดออกเป็น 3 ส่วนคือ ข้อมูลตำแหน่งพิกัด ข้อมูลระบบส่งต่อผู้ป่วย และ ข้อมูลรถพยาบาล

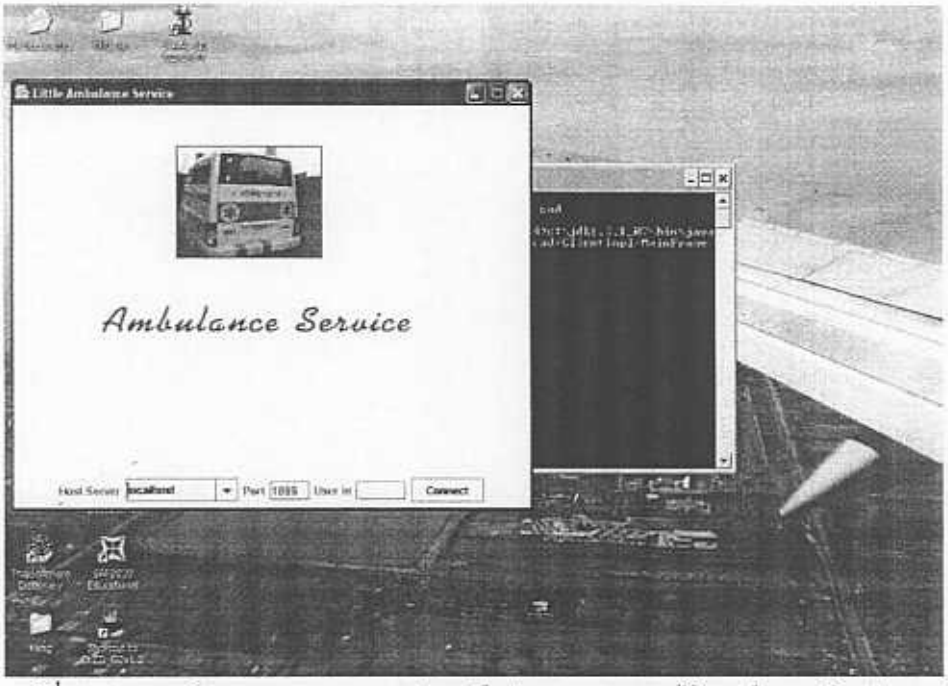
ในส่วนที่ 2 ข้อมูลพิกัดตำแหน่งของรถพยาบาล จะถูกนำเสนอในรูปของพิกัดตำแหน่งของรถในแผนที่ของจังหวัดร้อยเอ็ด (ภาคผนวก ก.) ซึ่งข้อมูลนี้จะทำให้ผู้ใช้สามารถทราบถึงตำแหน่งการเดินทางของรถพยาบาล ซึ่งนำไปใช้ในการบริหารระบบส่งต่อผู้ป่วยต่อไป

ในส่วนที่ 3 การจัดสรรการใช้งานระบบโดยผู้ดูแลระบบ จะเป็นส่วนที่กำหนดประเภทการใช้งานของผู้ใช้ซึ่งต้องดูแบ่งแยกออกตามลำดับความสำคัญ คือ ผู้ปฏิบัติงาน (user) ซึ่งสามารถเรียกดูข้อมูลได้เท่านั้นและ ผู้ดูแลระบบ (Super user) ที่สามารถจัดการ แก้ไข ลบ หรือเพิ่มเติมฐานข้อมูลของรถพยาบาล และข้อมูลการเชื่อมต่อข้อมูลพิกัดระหว่างรถพยาบาลและตัวระบบข้อมูลหลัก

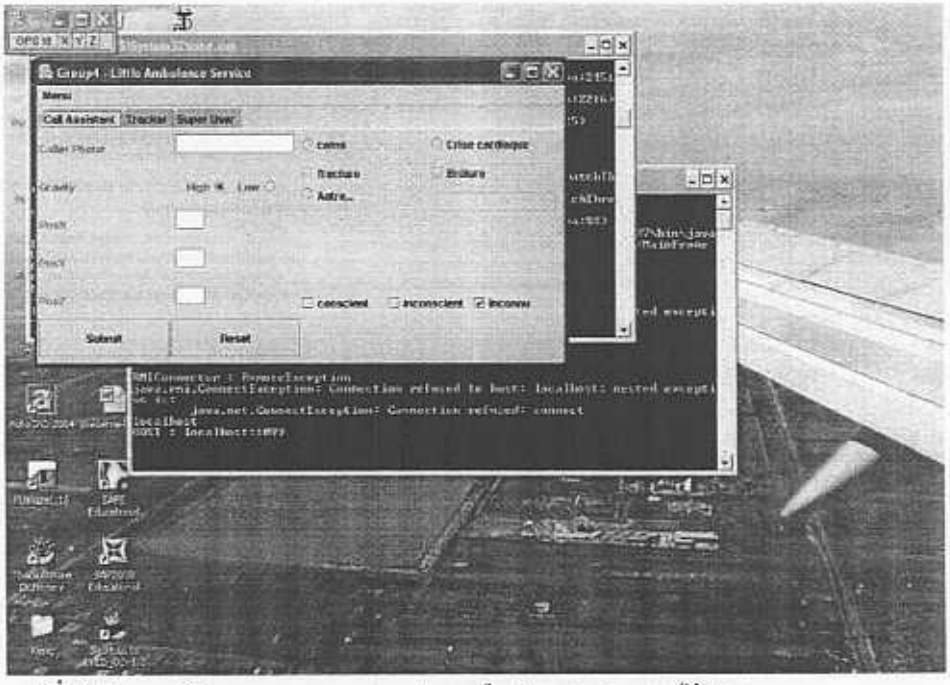


รูปที่ 4.2 ระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สายในการกำหนดการใช้งานระบบฐานข้อมูล

รูปที่ 4.3 ระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สายหน้าจอหลัก

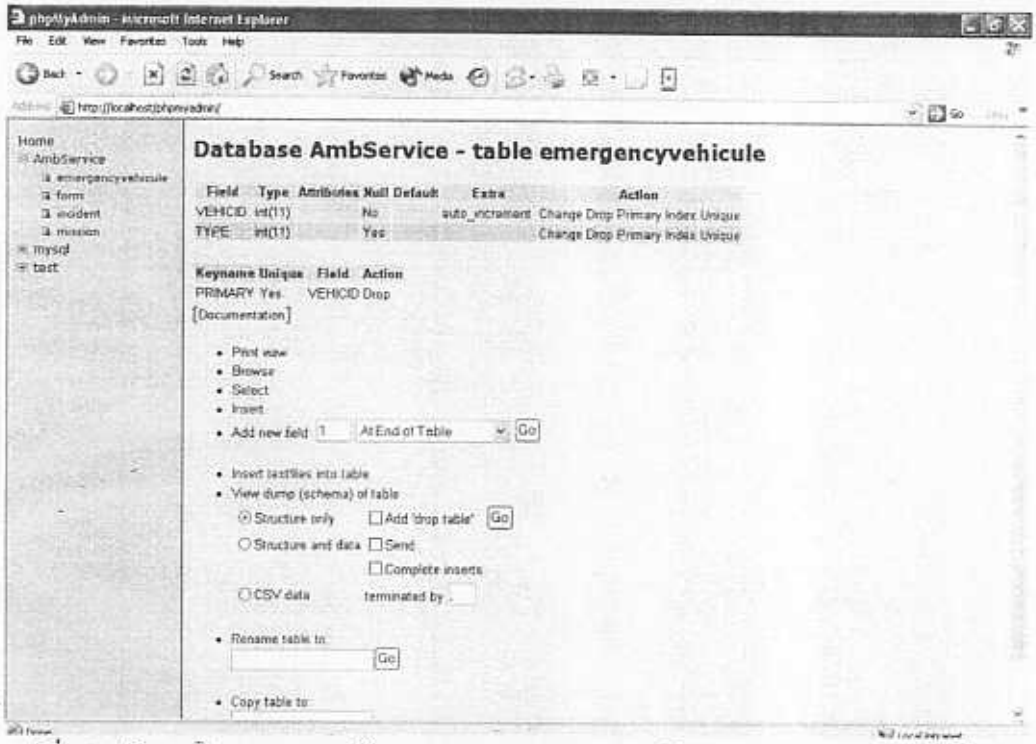


รูปที่ 4.4 ระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สายในการติดต่อฐานข้อมูล

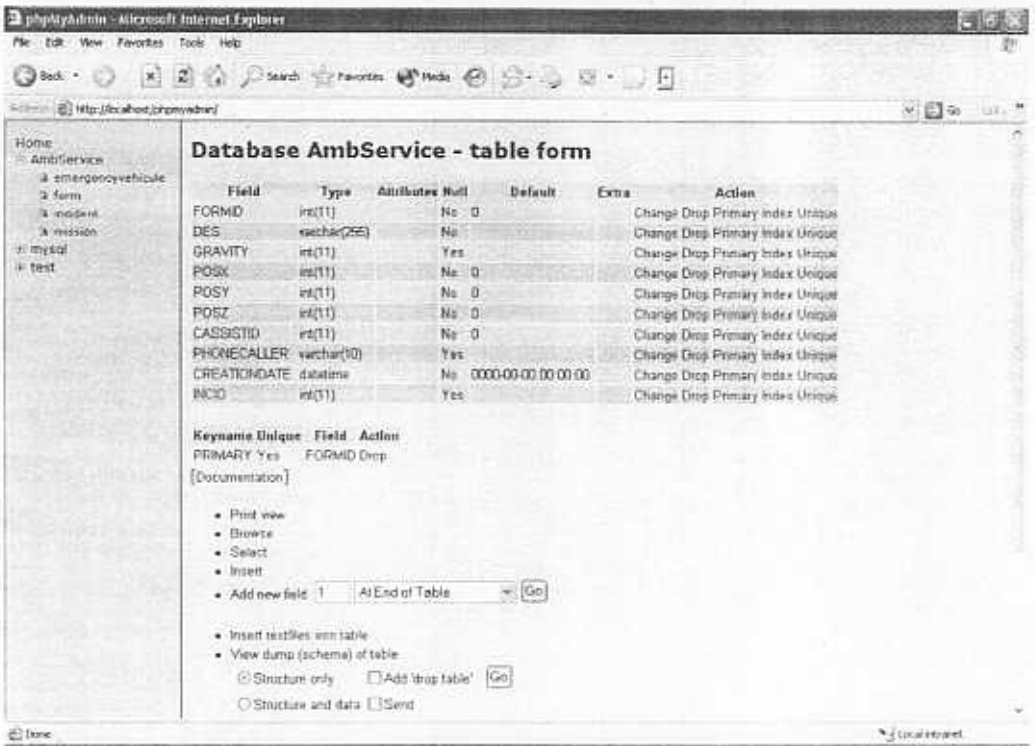


รูปที่ 4.5 ระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สาย

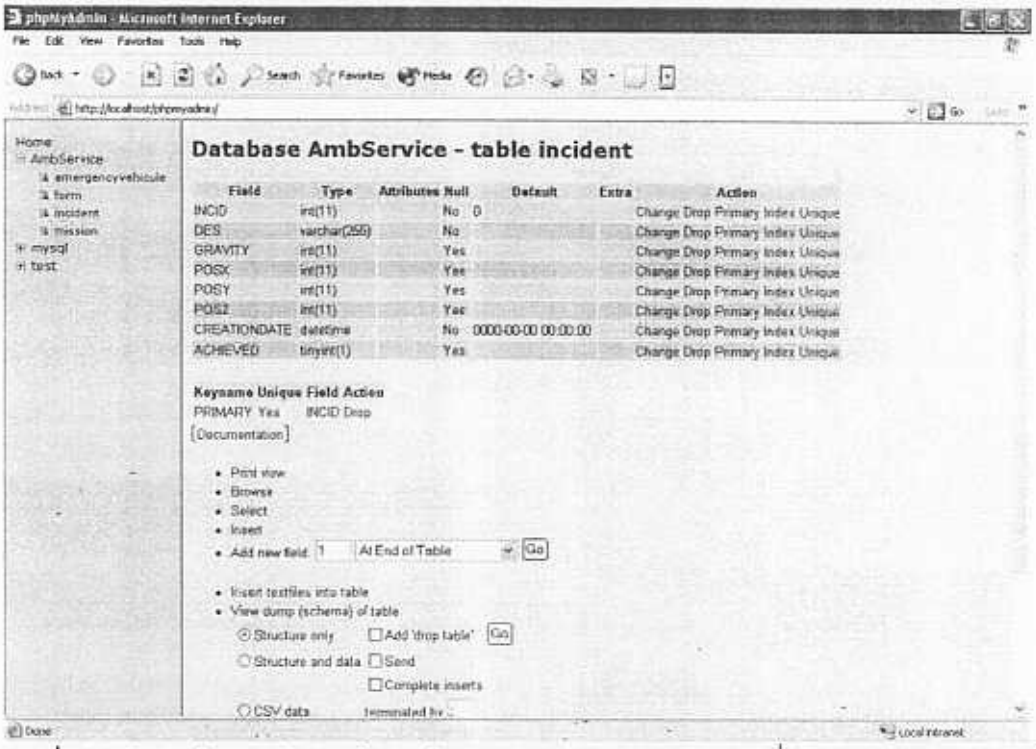
จากรูปที่ 4.6-4.9 จะเป็นรายละเอียดโครงสร้างของฐานข้อมูลที่ใช้ในการพัฒนาระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สาย



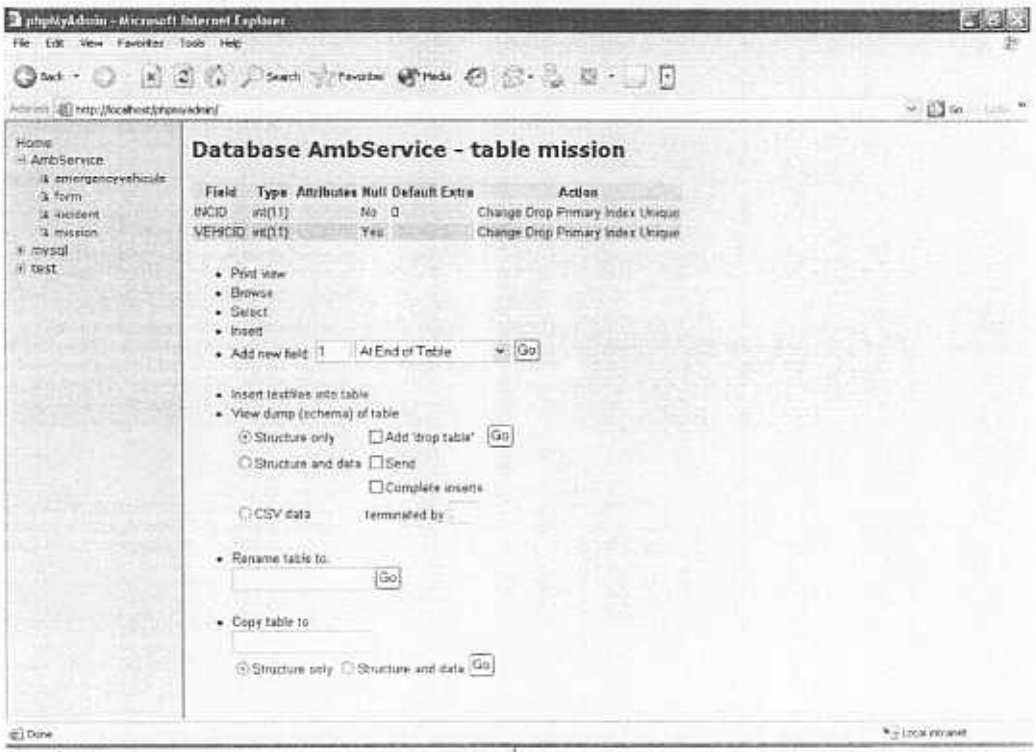
รูปที่ 4.6 โครงสร้างระบบฐานข้อมูลรถพยาบาลของระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สาย



รูปที่ 4.7 โครงสร้างระบบฐานข้อมูลการรับข้อมูลพิกัดตำแหน่งของรถพยาบาลของระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สาย



รูปที่ 4.8 โครงสร้างระบบฐานข้อมูลตำแหน่งรถพยาบาลและสถานที่เกิดอุบัติเหตุของระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สาย



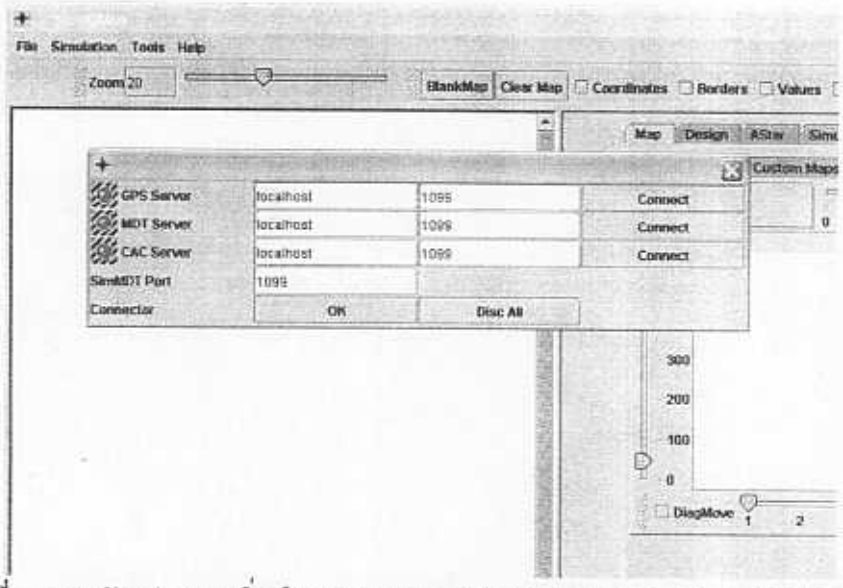
รูปที่ 4.9 โครงสร้างระบบฐานข้อมูลการสั่งงานรถพยาบาลของระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สาย

ในการวิจัยได้ทำการทดลองส่งพิกัดตำแหน่งการเคลื่อนที่ของรถผ่านมือถือมายังศูนย์ข้อมูลหลัก แล้วทำการเรียกโปรแกรมติดตามที่พัฒนาขึ้นเพื่อให้แสดงถึงการเคลื่อนที่ของตำแหน่งรถที่ได้จากค่าข้อมูลพิกัดที่ถูกส่งมาที่ Server โดยระบบโปรแกรมถูกออกแบบหน้าต่างการใช้งานออกเป็นสองส่วน คือ ส่วนที่ 1 คือหน้าต่างด้านซ้ายมือที่เป็นภาพของเคลื่อนที่ของรถพยาบาล ส่วนที่ 2 คือหน้าต่างการกำหนดข้อมูลเพิ่มเติมของแผนที่ ซึ่งในการใช้งานจะมีสถานะ (status bar) โชว์สถานะการทำงานของระบบ ซึ่งจะอยู่ด้านล่างสุดของโปรแกรม

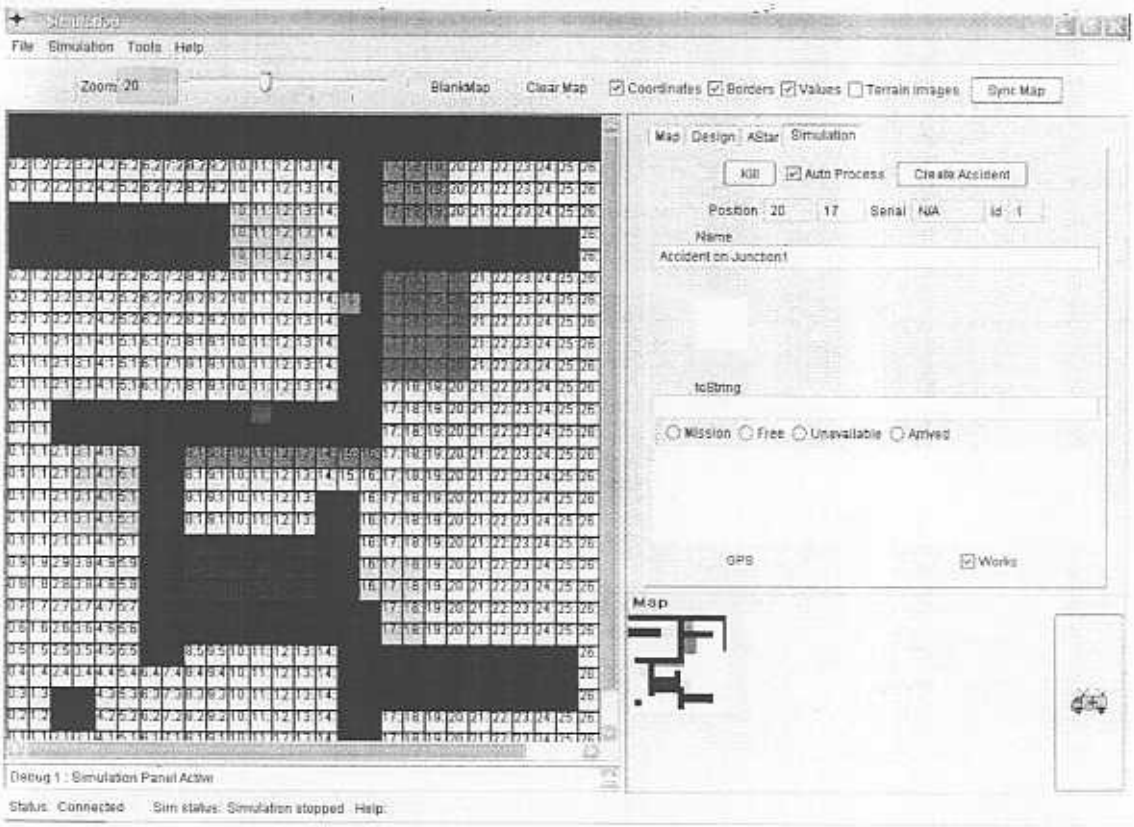
ในการเริ่มต้นใช้งานระบบนี้ ผู้ใช้ต้องทำการเชื่อมต่อระบบเข้ากับฐานข้อมูลพิกัดตำแหน่งซึ่งแสดงในรูปที่ 4.10 โดยต้องเริ่มต้นด้วยการ connect ไปยัง database server ที่จัดเก็บพิกัดตำแหน่ง เมื่อเชื่อมต่อโปรแกรมเข้ากับระบบฐานข้อมูลพิกัดตำแหน่งได้แล้วก็ให้ทำการเปิดแผนที่ จากนั้นปุ่มคำสั่ง Simulation จะเป็นคำสั่งที่ใช้ในการ upload ข้อมูลพิกัดและแสดงตำแหน่งอ้างอิงกับแผนที่ โดยแสดงในรูปที่ 4.11 โดยมีตำแหน่งของรถพยาบาลที่ แสดงเป็นจุดสีเขียวคือ จุดเริ่มต้นของพิกัดตำแหน่ง และจุดสีแดงคือตำแหน่งที่เกิดขึ้นจากข้อมูลตำแหน่งที่มีการส่งมาทุก 1 นาที ซึ่งจุดสีแดงจะมีการเปลี่ยนแปลงตำแหน่งอ้างกับตำแหน่งแผนที่จริง โดยตัวระบบมีฟังก์ชั่นในการย่อ ขยาย แผนที่ตามสภาพจริงได้

ซึ่งนอกจากความสามารถของตัวระบบที่สามารถแสดงพิกัดตำแหน่งการเคลื่อนที่ของรถพยาบาลเทียบกับแผนที่ที่เกิดขึ้นจริงแล้ว ยังสามารถกำหนดข้อมูลในเรื่องของอุบัติเหตุเพิ่มเติมเข้าไปยังระบบฐานข้อมูลด้วยว่า รถพยาบาลที่ส่งข้อมูลการเคลื่อนที่นั้นมีสถานะการทำงานแบบใดเมื่อเทียบกับพิกัดตำแหน่งของการเกิดอุบัติเหตุ ซึ่งมีตัวเลือกอยู่ 4 สถานะคือ อยู่ในระหว่างภาระกิจส่งต่อผู้ป่วย สถานะว่างงาน สถานะไม่สามารถติดต่อได้ และได้ถึงที่จุดเกิดเหตุแล้ว โดยข้อมูลนี้จะเป็นการกำหนดข้อมูลการเกิดอุบัติเหตุ ที่ได้รับแจ้งเข้ามา

และยังมีส่วนในการจำลองการค้นหาเส้นทางที่สั้นที่สุดของจุดอุบัติเหตุที่เกิดขึ้น เทียบกับจุดเริ่มต้นของรถพยาบาล โดยใช้อัลกอริทึม A* ซึ่งเป็นอัลกอริทึมด้านค้นหาเส้นทางที่สั้นที่สุดในระบบอัจริยะ (intelligent system) ซึ่งเป็นเพียงข้อมูลช่วยในการตัดสินใจในการเลือกเส้นทางของการเดินทาง จากจุดเริ่มต้น ไปยังจุดที่เกิดเหตุ ซึ่งทางศูนย์ข้อมูลสามารถติดต่อกลับไปยังรถพยาบาลถึงข้อมูลเส้นทางที่ระบบประมวลผลให้ เพื่อเสนอแนะให้เลือกเส้นทางที่สั้นที่สุดในการเดินทาง



รูปที่ 4.10 ตัวอย่างการเริ่มทำงานของระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สาย



รูปที่ 4.11 ตัวอย่างการทำงานของระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สาย

ผลการพัฒนางานวิจัยระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สาย หลังจากทดสอบการใช้งาน ซึ่งผลการวิจัยสามารถแสดงข้อมูลการเคลื่อนที่ของรถพยาบาลเทียบกับแผนที่ได้ หากแต่ข้อจำกัดในการใช้งาน คือ การได้ข้อมูลพิกัดตำแหน่งที่ขาดเคลื่อนเนื่องจากจีพีเอสทำงานโดย

รับข้อมูลจากดาวเทียม ดังนั้นสภาพอากาศ เช่น ปริมาณเมฆบนท้องฟ้าจึงมีผลกระทบต่อการรับข้อมูลของจีพีเอสค่อนข้างมาก ด้วยเหตุนี้เองทำให้การทำงานของชุดรับสัญญาณ จีพีเอสจะรับสัญญาณได้อย่างแม่นยำในวันที่สภาพอากาศค่อนข้างดีคือ ท้องฟ้าปลอดโปร่งไม่มีเมฆมาก นอกจากนั้นจีพีเอสจะทำงานในที่โล่งแจ้งได้ดีกว่าบริเวณที่มีสิ่งกีดขวางเส้นทางการรับสัญญาณจากดาวเทียมของจีพีเอส และอีกประการที่สำคัญคือการส่งข้อมูลพิกัดมายังศูนย์ข้อมูลที่ใช้เทคโนโลยี SMS นั้นมีข้อจำกัดในเรื่องของสัญญาณของอุปกรณ์มือถือ หากเป็นเส้นทางที่สัญญาณมือถือไม่เอื้ออำนวยนักจะทำให้ไม่สามารถส่งข้อมูลได้ในเวลาที่ต้องการ

ในการกำหนดช่วงระยะเวลาในการส่งข้อมูลตำแหน่งพิกัดที่กำหนดให้ส่ง และอ่านข้อมูลที่ส่งมาทุก 1 นาที ระบบฐานข้อมูลต้องรองรับข้อมูลที่มีขนาดจำนวนมาก ทำให้การประมวลผลข้อมูลของระบบติดตามเป็นไปได้ช้า

บทที่ 5
สรุปผลการวิจัย และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

รูปแบบการพัฒนาระบบซึ่งได้ถูกแบ่งออกเป็นสองส่วนหลักนั้นได้ถูกพัฒนาจนแล้วเสร็จ โดยในส่วนแรกคือ การส่งข้อมูลพิกัดนั้นสามารถรับข้อมูลพิกัดตำแหน่งของรถพยาบาลผ่านอุปกรณ์ GPS ที่เชื่อมต่อกับคอมพิวเตอร์แบบพกพา แล้วส่งข้อมูลนี้ต่อไปยังศูนย์ข้อมูลหลักผ่านมือถือ โดยใช้เทคโนโลยี SMS ซึ่งเป็นเทคโนโลยีที่ได้เลือกใช้ เมื่อพิจารณาในเรื่องความสะดวก ประหยัดและง่ายต่อการใช้งานของผู้ใช้

ระบบรับข้อมูลซึ่งเป็นองค์ประกอบหลักของการใช้งานได้พัฒนาโครงสร้างการใช้งานตามความต้องการของผู้ใช้ โดยการใช้งานระบบได้อธิบายในบทที่ 4 นั้นมีรายละเอียดในส่วนของการนำเอาแผนที่ และ ข้อมูลพิกัดตำแหน่งของรถพยาบาลมาประมวลผล พร้อมทั้งเพิ่มรายละเอียดในการจัดการข้อมูลของรถพยาบาลที่สามารถใช้ในการบริหารระบบส่งต่อผู้ป่วย

ข้อดีของระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สายได้ถูกพัฒนาบนพื้นฐานแนวคิดในการใช้ซอฟแวร์ที่เป็น opensource โดยที่ผู้ใช้สามารถพัฒนา เพิ่มเติม และแก้ไขปรับปรุงได้ง่าย นอกจากนี้แล้วระบบติดตามรถพยาบาลผ่านเครือข่ายสารสนเทศไร้สายที่เสร็จสมบูรณ์จะต้องมีการประเมินเปรียบเทียบประสิทธิภาพการทำงานของระบบติดตามรถพยาบาลกับระบบเดิมที่ไม่ได้ใช้เทคโนโลยีการติดตามอัตโนมัติ

ผลของงานวิจัยของการพัฒนาระบบติดตามรถพยาบาลผ่านระบบเครือข่ายสารสนเทศไร้สายนี้จะเป็นข้อมูลเบื้องต้นที่จะชี้แจ้งให้เห็นรูปแบบ และเทคโนโลยีคอมพิวเตอร์ที่เหมาะสมที่สามารถนำมาปรับปรุงคุณภาพชีวิตมนุษย์ โดยผ่านกระบวนการที่เพิ่มประสิทธิภาพในการตัดสินใจในการรักษาพยาบาล จากข้อมูลที่เป็นปัจจุบัน รวดเร็ว และถูกต้องแม่นยำที่ได้จากระบบติดตามอัตโนมัติ

5.2 ข้อเสนอแนะ

จากแผนงานที่ได้ปฏิบัติ มีอุปสรรคเพียงเล็กน้อยที่เกิดขึ้นในขั้นตอนที่ 5, 6 และ 7 ในการรวบรวมข้อมูลสำหรับระบบติดตามเนื่องจากเป็นขั้นตอนที่สำคัญในการ ออกแบบความต้องการของผู้ใช้งานจริง ให้ครอบคลุม และเพียงพอต่อการใช้งานจริงได้ ซึ่งข้อมูลที่ได้อาจจะสามารถทำการปรับเปลี่ยนได้ เมื่อระบบได้ทำการพัฒนาเสร็จสมบูรณ์และได้ทำการทดลองใช้จริง ก็สามารถทำงานตามที่กำหนดไว้ หากแต่เกิดความล่าช้า เนื่องจากใช้เวลามากกว่าที่วางแผนไว้ แต่ก็ยังสามารถ

ดำเนินการได้ตามแผนที่วางไว้ ซึ่งเมื่อมีการทดลองใช้งานจริง ก็พบอุปสรรค และข้อจำกัดที่เกิดขึ้น ดังแสดงในบทที่ 4 ถึงผลการใช้งานจริง ส่วนหลักคือการทำงานของตัวโปรแกรมใช้เวลาในการประมวลผล เนื่องจาก ไม่ได้มีการออกแบบการทำลายข้อมูลเดิม และการบีบอัดข้อความ ซึ่งแนวทางการพัฒนาต่อไปที่เป็นไปได้ คือ ออกแบบระบบ trigger ในระบบฐานข้อมูลให้มีการ update ข้อมูลตามเงื่อนไขของเวลาที่กำหนด และ การ delete ข้อมูลที่ประมวลผลไปแล้ว ตามเงื่อนไขของเวลาที่กำหนดซึ่งอาจจะช่วยแก้ปัญหาในส่วนของการจัดเก็บข้อมูลขนาดใหญ่

ซึ่งข้อจำกัดของฐานข้อมูลที่ใช้คือ MySQL ไม่มีคำสั่ง trigger ดังนั้นแนวทางการจัดเก็บข้อมูลหากพิจารณาจะปรับปรุงระบบให้สามารถใช้งานได้อย่างมีประสิทธิภาพมากยิ่งขึ้น ก็ต้องเลือกฐานข้อมูลที่รองรับคำสั่งนี้ และอีกแนวทางหนึ่งคือ การกำหนดระยะเวลาในการส่งพิกัดตำแหน่งอาจจะกำหนดให้มีช่วงระยะเวลาที่นานขึ้น เป็นการลดข้อมูล ซึ่งระยะเวลาที่เหมาะสมนี้ก็ต้องทำการทดลองเพิ่มเติม เพื่อศึกษาสภาพที่เหมาะสมของการใช้งาน เนื่องจากงานวิจัยนี้ได้ทดสอบการใช้งานจริง หากแต่ไม่ได้ทดลองกับสภาพการใช้งานของรถพยาบาลที่เกิดขึ้นจริง ซึ่งต้องมีระยะเวลาในการศึกษาเพิ่มเติมถึงปัจจัยเรื่องข้อมูลที่ส่งในช่วงเวลาที่เหมาะสม

การส่งข้อมูลพิกัดอาจจะเลือกส่งโดยใช้การส่งข้อมูลผ่านเครือข่ายสารสนเทศไร้สาย GPRS โดยใช้อุปกรณ์เสริม MDT ซึ่งระบบได้ออกแบบรองรับหากมีการนำรูปแบบการใช้งานนี้มาใช้ เพียงผู้ใช้กำหนดพอร์ตในการรับข้อมูลที่ส่งมา ซึ่งงานวิจัยนี้ไม่ได้ทำการทดลอง หรือ ทดสอบการใช้งานประเภทนี้

นอกจากนี้แล้วปัจจัยที่ส่งผลต่อการใช้งานคือ ค่าใช้จ่ายในการลงทุนในการจัดซื้ออุปกรณ์ ซึ่งต้องลงทุนค่อนข้างสูงสำหรับการจัดซื้ออุปกรณ์รับตำแหน่งพิกัด และอุปกรณ์สื่อสาร ซึ่งค่าใช้จ่ายนี้เมื่อเทียบกับระบบส่งต่อผู้ป่วยเดิมที่ใช้สัญญาณวิทยุนั้น ราคาจะต่างกันค่อนข้างสูง หากแต่สิ่งหนึ่งที่ได้ประโยชน์จากงานวิจัยนี้คือ การที่สามารถทราบข้อมูลที่แท้จริงของการทำงานของรถพยาบาล และซอฟต์แวร์ที่พัฒนาขึ้นสามารถนำไปใช้งานได้ ตามความต้องการของผู้ใช้ โดยพัฒนาขึ้นบนพื้นฐานของเทคโนโลยี opensource และนอกจากนี้แล้วยังสามารถเอาหลักการของระบบนี้ไปประยุกต์ใช้งานกับยานพาหนะประเภทอื่นๆที่ต้องการทราบตำแหน่งเคลื่อนที่การเดินรถได้

บรรณานุกรม

[1] รายงานแนวทางการจัดระบบบริการการแพทย์ฉุกเฉินในพื้นที่กรุงเทพมหานคร พ.ศ. 2545-2546

[2] Bhat, C.R. and Schofer, J.L., "A Conceptual Framework of Individual Activity Program Generation," Transportation Research 27A, 6, 1993, pp. 443-446.

[3] Collura, J. and Hazarvartian, K.E., "Factors Influencing the Use of the Adoption of Small Computers in Transportation Engineering Project Management," Journal of Transportation Engineering, American Society of Civil Engineers,2002.

[4] Collins T. NHS millenium fix goes on critical list. Computer Weekly 18 Dec 1997: 1.

[5] Medical Devices Agency. Medical devices and the year 2000. In: Year 2000 and healthcare computing. Health Informatics J 1997 3(3/4): 173-175.

[6] Ross J. Anderson, "Information technology in medical practice: safety and privacy lessons from the United Kingdom", http://www.cl.cam.ac.uk/users/rja14/austmedjour/austmedjour.html

[7] Report of the inquiry into the London Ambulance Service. London: South West Thames Regional Health Authority, 1993. ftp://cs.ucl.ac.uk/acwf/info/lascase0.9.pdf

[8] Shapiro DZ, Hughes JA, Randall D, Harper R. Visual re-representation of database information: the flight data strip in air traffic control. In: Aspects of visual languages and visual interfaces. London: Elsevier, 1994: 249-376.

[9] Shuldiner, P.W., Ketselidou, Z. and Collura, J., "Expert System Application to Freeway Incident Management," Proceedings of the Fifth National Conference on Microcomputers on Civil Engineering, University of Central Florida, Orlando, FL, November 1987, pp. 141-145.

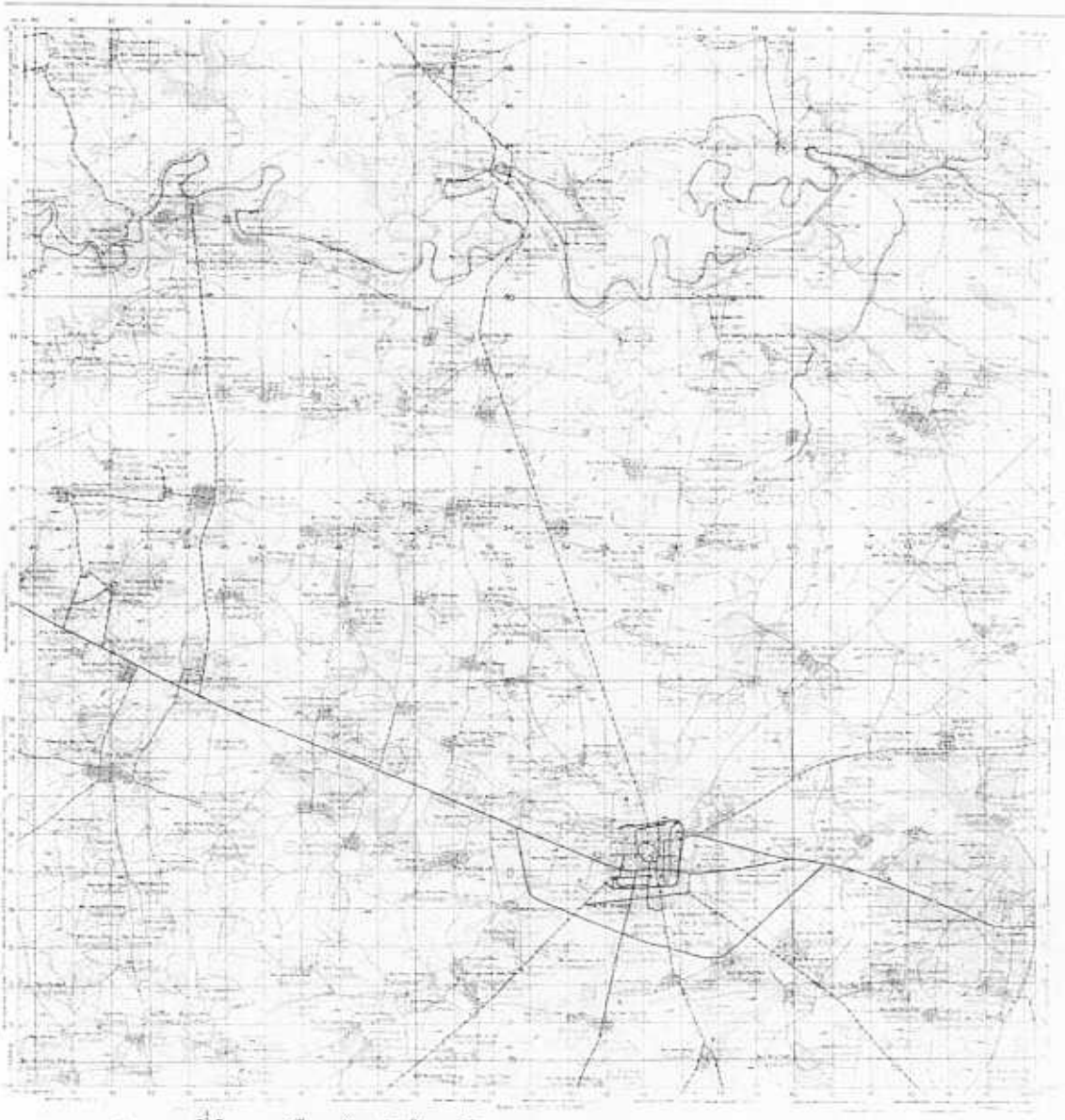
[10] Shuldiner, P.W., Ketselidou, Z. and Collura, J., "Knowledge Based Expert Systems for Post-Incident Freeway Traffic Control," Proceedings of the Third International Conference on Microcomputers on Transportation, American Society of Civil Engineers, New York, NY, 1990.

[11] Vowler J. Patient care at risk from millenium bug. Computer Weekly 8 May 1997: 3.

ภาคผนวก

ภาคผนวก ก.

แผนที่จังหวัดร้อยเอ็ด



ก.1. แผนที่อำเภอเมืองจังหวัดร้อยเอ็ด

ภาคผนวก ข.

รายนามคณะผู้วิจัย

หัวหน้าโครงการวิจัย

ผศ.ดร. วนิดา แก่นอากาศ

นักวิจัย

อาจารย์ สหชัย แก่นอากาศ

นายแพทย์ ไพบูลย์ เพ็ญสุวรรณ

นาง บุษบา บัวผัน

นาง กัลยา กองเงิน

ผู้ช่วยนักวิจัย

นาย ศิโรฒ บานแบ่ง

นางสาว พรจิตร อ่อนท้าว

ที่ปรึกษา

Dr. Neil Davey
Dr. Rod Adams

ภาคผนวก ค.
โครงสร้างโปรแกรม
โปรแกรมรับส่งพิกัดตำแหน่ง

Class CATCommands
java.lang.Object
└ org.jsmsengine.CATCommands

public class CATCommands
extends java.lang.Object
This class keeps the various AT commands used by jSMSEngine.

Field Summary	
protected static java.lang.String	AT_ASCII_MODE
protected static java.lang.String	AT_AT
protected static java.lang.String	AT_BATTERY
protected static java.lang.String	AT_CHARSET_HEX
protected static java.lang.String	AT_CHECK_LOGIN
protected static java.lang.String	AT_CMD_MODE
protected static java.lang.String	AT_DELETE_MESSAGE
protected static java.lang.String	AT_DISABLE_INDICATIONS
protected static java.lang.String	AT_ECHO_OFF
protected static java.lang.String	AT_ERICSSON_DISABLE_INDICATIONS
protected static java.lang.String	AT_ERICSSON_SMS_STORAGE
protected static java.lang.String	AT_IMSI
protected static java.lang.String	AT_KEEP_LINK_OPEN
protected static java.lang.String	AT_LIST
protected static java.lang.String	AT_LOGIN
protected static java.lang.String	AT_MANUFACTURER
protected static java.lang.String	AT_MODEL
protected static java.lang.String	AT_OK
protected static java.lang.String	AT_PDU_MODE
protected static java.lang.String	AT_READY
protected static java.lang.String	AT_SEND_MESSAGE

protected static java.lang.String	AT_SERIALNO
protected static java.lang.String	AT_SIEMENS_SMS_STORAGE
protected static java.lang.String	AT_SIGNAL
protected static java.lang.String	AT_SOFTWARE

Constructor Summary

CATCommands()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

AT_OK
protected static final java.lang.String AT_OK
See Also:
Constant Field Values

AT_AT

protected static final java.lang.String AT_AT
See Also:
Constant Field Values

AT_ECHO_OFF

protected static final java.lang.String AT_ECHO_OFF
See Also:
Constant Field Values

AT_CMD_MODE

protected static final java.lang.String AT_CMD_MODE
See Also:
Constant Field Values

AT_DISABLE_INDICATIONS

protected static final java.lang.String AT_DISABLE_INDICATIONS
See Also:
Constant Field Values

AT_SIEMENS_SMS_STORAGE

protected static final java.lang.String AT_SIEMENS_SMS_STORAGE
See Also:
Constant Field Values

AT_ERICSSON_SMS_STORAGE

protected static final java.lang.String AT_ERICSSON_SMS_STORAGE
See Also:
Constant Field Values

AT_ERICSSON_DISABLE_INDICATIONS

protected static final java.lang.String AT_ERICSSON_DISABLE_INDICATIONS
See Also:
Constant Field Values

AT_MANUFACTURER

protected static final java.lang.String AT_MANUFACTURER
See Also:
Constant Field Values

AT_MODEL

protected static final java.lang.String AT_MODEL
See Also:
Constant Field Values

AT_SERIALNO

protected static final java.lang.String AT_SERIALNO
See Also:
Constant Field Values

AT_IMSI

protected static final java.lang.String AT_IMSI
See Also:
Constant Field Values

AT_BATTERY

protected static final java.lang.String AT_BATTERY
See Also:
Constant Field Values

AT_SIGNAL

protected static final java.lang.String AT_SIGNAL
See Also:
Constant Field Values

AT_SOFTWARE

protected static final java.lang.String AT_SOFTWARE
See Also:
Constant Field Values

AT_LIST

protected static final java.lang.String AT_LIST
See Also:
Constant Field Values

AT_SEND_MESSAGE

protected static final java.lang.String AT_SEND_MESSAGE
See Also:
Constant Field Values

AT_KEEP_LINK_OPEN

protected static final java.lang.String AT_KEEP_LINK_OPEN
See Also:
Constant Field Values

AT_DELETE_MESSAGE

protected static final java.lang.String AT_DELETE_MESSAGE
See Also:
Constant Field Values

AT_ASCII_MODE

protected static final java.lang.String AT_ASCII_MODE
See Also:
Constant Field Values

AT_PDU_MODE

protected static final java.lang.String AT_PDU_MODE
See Also:
Constant Field Values

AT_CHARSET_HEX

protected static final java.lang.String AT_CHARSET_HEX
See Also:
Constant Field Values

AT_CHECK_LOGIN

protected static final java.lang.String AT_CHECK_LOGIN
See Also:
Constant Field Values

AT_LOGIN

protected static final java.lang.String AT_LOGIN
See Also:
Constant Field Values

AT_READY

protected static final java.lang.String AT_READY
See Also:
Constant Field Values

Constructor Detail

CATCommands

public CATCommands()

Class CDeviceInfo.CStatistics

java.lang.Object
└ org.jsmsengine.CDeviceInfo.CStatistics
Enclosing class:
CDeviceInfo

public class CDeviceInfo.CStatistics
extends java.lang.Object
This subclass keeps counters for incoming / outgoing messages.
See Also:
CService.refreshDeviceInfo(), CService.getDeviceInfo()

Field Summary

protected int	totalIn
protected int	totalOut

Constructor Summary

CDeviceInfo.CStatistics()
Default constructor of the class.

Method Summary

int	getTotalIn() Returns the total number of incoming messages processed by jSMSEngine.
int	getTotalOut() Returns the total number of outgoing messages dispatched by jSMSEngine.
protected void	incTotalIn()
protected void	incTotalOut()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

totalIn
protected int totalIn

totalOut
protected int totalOut

Constructor Detail

CDeviceInfo.CStatistics
public CDeviceInfo.CStatistics()
Default constructor of the class.

Method Detail

getTotalIn
public int getTotalIn()
Returns the total number of incoming messages processed by jSMSEngine.
Returns:

the number of incoming messages.

---

getTotalOut
public int getTotalOut()
Returns the total number of outgoing messages dispatched by jSMSEngine.
Returns:
the number of outgoing messages.

---

incTotalIn
protected void incTotalIn()

---

incTotalOut
protected void incTotalOut()

Class CGSMAlphabets

java.lang.Object
└ org.jsmsengine.CGSMAlphabets

---

class CGSMAlphabets
extends java.lang.Object
This class contains the conversion routines to and from the standard 7bit GSM alphabet.

Every normal ASCII character must be converted according to the GSM 7bit default alphabet before dispatching through the GSM device. The opposite conversion is made when a message is received.

Since some characters in 7bit alphabet are in the position where control characters exist in the ASCII alphabet, each message is represented in HEX format as well (field hexText in CMessage class and descendants). When talking to the GSM device, either for reading messages, or for sending messages, a special mode is used where each character of the actual message is represented by two hexadecimal digits. So there is another conversion step here, in order to get the ASCII character from each pair of hex digits, and vice verca.

Note: currently, only GSM default 7Bit character set is supported. In all routines, you may assume the "charSet" parameter as constant.

Field Summary	
private static java.lang.String	alphabet
protected static int	GSM7BITDEFAULT

Constructor Summary	
(package private)	CGSMAlphabets()

Method Summary	
protected static java.lang.String	char2Hex(char c, int charSet) Converts an ASCII character to its hexadecimal pair.
protected static char	hex2Char(int index, int charSet) Converts a hexadecimal value to the ASCII character it represents.
protected static char	hex2ExtChar(int ch, int charSet) Converts a int value to the extended ASCII character it represents.
protected static java.lang.String	hex2Text(java.lang.String text, int charSet) Converts the given string of hexadecimal pairs to its ASCII equivalent string.
protected static java.lang.String	text2Hex(java.lang.String text, int charSet) Converts the given ASCII string to a string of hexadecimal pairs.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

GSM7BITDEFAULT
protected static final int GSM7BITDEFAULT
See Also:
Constant Field Values

---

alphabet
private static final java.lang.String alphabet
See Also:
Constant Field Values

Constructor Detail

CGSMAlphabets
CGSMAlphabets()

Method Detail

char2Hex
protected static java.lang.String char2Hex(char c,
int charSet)
Converts an ASCII character to its hexadecimal pair.
Parameters:
c - the ASCII character.
charSet - the target character set for the conversion.
Returns:
the two hex digits which represent the character in the specific character set.

---

hex2Char
protected static char hex2Char(int index,
int charSet)
Converts a hexadecimal value to the ASCII character it represents.
Parameters:
index - the hexadecimal value.
charSet - the character set in which "index" is represented.
Returns:
the ASCII character which is represented by the hexadecimal value.

---

hex2ExtChar
protected static char hex2ExtChar(int ch,
int charSet)
Converts a int value to the extended ASCII character it represents.
Parameters:
ch - the int value.
charSet - the character set in which "ch" is represented.
Returns:
the extended ASCII character which is represented by the int value.

---

text2Hex
protected static java.lang.String text2Hex(java.lang.String text,
int charSet)
Converts the given ASCII string to a string of hexadecimal pairs.
Parameters:
text - the ASCII string.
charSet - the target character set for the conversion.
Returns:
the string of hexadecimals pairs which represent the "text" parameter in the specified "charSet".

---

hex2Text
protected static java.lang.String hex2Text(java.lang.String text,
int charSet)
Converts the given string of hexadecimal pairs to its ASCII equivalent string.
Parameters:
text - the hexadecimal pair string.
charSet - the target character set for the conversion.
Returns:
the ASCII string.

Class CIncomingMessage

java.lang.Object
└ org.jsmsengine.CMessage
└ org.jsmsengine.CIncomingMessage

---

public class CIncomingMessage
extends CMessage
This class represents an incoming SMS message, i.e. message read from the GSM device.
See Also:
CMessage, COutgoingMessage, CService.readMessages(LinkedList, int)

Field Summary

static int CLASS_ALL

static int CLASS_REC_READ

static int	CLASS_REC_UNREAD
static int	CLASS_STO_SENT
static int	CLASS_STO_UNSENT

Fields inherited from class org.jsmsengine.CMessage

date, id, memIndex, MESSAGE_ENCODING_7BIT, MESSAGE_ENCODING_8BIT, MESSAGE_ENCODING_UNICODE, messageEncoding, originator, recipient, text, TYPE_INCOMING, TYPE_OUTGOING

Constructor Summary

	CIncomingMessage(java.util.Date date, java.lang.String originator, java.lang.String text, int memIndex) Default constructor of the class.
protected	CIncomingMessage(java.lang.String pdu, int memIndex)

Method Summary

java.lang.String	getOriginator() Returns the originator's phone number (international format).
private java.lang.String	pduToText(java.lang.String pdu)
void	setOriginator(java.lang.String originator) Set the phone number of the originator.

Methods inherited from class org.jsmsengine.CMessage

getDate, getHexText, getId, getMemIndex, getMessageEncoding, getText, getType, setDate, setId, setMessageEncoding, setText, toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

CLASS_ALL
public static final int CLASS_ALL
See Also:
Constant Field Values

CLASS_REC_UNREAD
public static final int CLASS_REC_UNREAD
See Also:
Constant Field Values

CLASS_REC_READ
public static final int CLASS_REC_READ
See Also:
Constant Field Values

CLASS_STO_UNSENT
public static final int CLASS_STO_UNSENT
See Also:
Constant Field Values

CLASS_STO_SENT
public static final int CLASS_STO_SENT
See Also:
Constant Field Values

Constructor Detail

CIncomingMessage
public CIncomingMessage(java.util.Date date,
java.lang.String originator,
java.lang.String text,
int memIndex)
Default constructor of the class.
Parameters:
date - the creation date of the message.
originator - the originator's number.

text - the actual text of the message.
memIndex - the index of the memory location in the GSM device where this message is stored.

Notes:
Phone numbers are represented in their international format (e.g. +306974... for Greece).

CIncomingMessage
protected CIncomingMessage(java.lang.String pdu,
int memIndex)

Method Detail

setOriginator
public void setOriginator(java.lang.String originator)
Set the phone number of the originator. Applicable to incoming messages.
Parameters:
originator - the originator's phone number (international format).

---

getOriginator
public java.lang.String getOriginator()
Returns the originator's phone number (international format). Applicable only for incoming messages.
Returns:
the originator's phone number.

---

pduToText
private java.lang.String pduToText(java.lang.String pdu)

Class CMessage
java.lang.Object
└ org.jsmsengine.CMessage
Direct Known Subclasses:
CIncomingMessage, COutgoingMessage

---

public class CMessage
extends java.lang.Object
This class encapsulates the basic characteristics of an SMS message. A message is further subclassed to an "Incoming" message and an "Outgoing" message.

This class is never used directly. Please use one of its descendants.
See Also:
CIncomingMessage, COutgoingMessage, CPhoneBook

Field Summary

protected java.util.Date	date
protected java.lang.String	id
protected int	memIndex
static int	MESSAGE_ENCODING_7BIT
static int	MESSAGE_ENCODING_8BIT
static int	MESSAGE_ENCODING_UNICODE
protected int	messageEncoding
protected java.lang.String	originator
protected java.lang.String	recipient
protected java.lang.String	text
private int	type
static int	TYPE_INCOMING

static int	TYPE_OUTGOING

Constructor Summary

CMessage(int type, java.util.Date date, java.lang.String originator, java.lang.String recipient, java.lang.String text, int memIndex)
Default constructor of the class.

Method Summary

java.util.Date	getDate() Returns the date of the message.
java.lang.String	getHexText() Returns the text of the message, in hexadecimal format.
java.lang.String	getId() Returns the id of the message.
int	getMemIndex() Returns the memory index of the GSM device, where the message is stored.
int	getMessageEncoding() Returns the encoding method of the message.
java.lang.String	getText() Returns the actual text of the message (ASCII).
int	getType() Returns the type of the message.
void	setDate(java.util.Date date) Set the date of the message.
void	setId(java.lang.String id) Set the id of the message.
void	setMessageEncoding(int messageEncoding) Set the message encoding.
void	setText(java.lang.String text) Set the text of the message.
java.lang.String	toString()

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

MESSAGE_ENCODING_7BIT
public static final int MESSAGE_ENCODING_7BIT
See Also:
Constant Field Values

MESSAGE_ENCODING_8BIT
public static final int MESSAGE_ENCODING_8BIT
See Also:
Constant Field Values

MESSAGE_ENCODING_UNICODE
public static final int MESSAGE_ENCODING_UNICODE
See Also:
Constant Field Values

TYPE_INCOMING
public static final int TYPE_INCOMING
See Also:
Constant Field Values

TYPE_OUTGOING
public static final int TYPE_OUTGOING
See Also:
Constant Field Values

---

type
private int type

---

id

protected java.lang.String id

---

memIndex
protected int memIndex

---

date
protected java.util.Date date

---

originator
protected java.lang.String originator

---

recipient
protected java.lang.String recipient

---

text
protected java.lang.String text

---

messageEncoding
protected int messageEncoding

Constructor Detail

CMessage
public CMessage(int type,
java.util.Date date,
java.lang.String originator,
java.lang.String recipient,
java.lang.String text,
int memIndex)
Default constructor of the class.
Parameters:
type - the type (incoming/outgoing) of the message.
date - the creation date of the message.
originator - the originator's number. Applicable only for incoming messages.
recipient - the recipient's number. Applicable only for outgoing messages.
text - the actual text of the message.
memIndex - the index of the memory location in the GSM device where this message is stored. Applicable only for incoming messages.

Notes:
Phone numbers are represented in their international format (e.g. +306974... for Greece).
"Recipient" may be an entry from the phonebook.

Method Detail

getType
public int getType()
Returns the type of the message. Type is either incoming or outgoing, as denoted by the class' static values INCOMING and OUTGOING.
Returns:
the type of the message.

---

getId
public java.lang.String getId()
Returns the id of the message.
Returns:
the id of the message.

---

getMemIndex
public int getMemIndex()
Returns the memory index of the GSM device, where the message is stored. Applicable only for incoming messages.
Returns:
the memory index of the message.

---

getDate
public java.util.Date getDate()
Returns the date of the message. For incoming messages, this is the sent date. For outgoing messages, this is the creation date.
Returns:
the date of the message.

---

getText
public java.lang.String getText()
Returns the actual text of the message (ASCII).
Returns:
the text of the message.

getHexText

public java.lang.String getHexText()
Returns the text of the message, in hexadecimal format.
Returns:
the text of the message (HEX format).

---

getMessageEncoding
public int getMessageEncoding()
Returns the encoding method of the message. Returns of the constants MESSAGE_ENCODING_7BIT, MESSAGE_ENCODING_8BIT, MESSAGE_ENCODING_UNICODE. This is meaningful only when working in PDU mode.
Returns:
the message encoding.

---

setId
public void setId(java.lang.String id)
Set the id of the message.
Parameters:
id - the id of the message.

---

setText
public void setText(java.lang.String text)
Set the text of the message.
Parameters:
text - the text of the message.

---

setDate
public void setDate(java.util.Date date)
Set the date of the message.
Parameters:
date - the date of the message.

---

setMessageEncoding
public void setMessageEncoding(int messageEncoding)
Set the message encoding. Should be one of the constants MESSAGE_ENCODING_7BIT, MESSAGE_ENCODING_8BIT, MESSAGE_ENCODING_UNICODE. This is meaningful only when working in PDU mode - default is 7bit.
Parameters:
messageEncoding - one of the message encoding contants.

---

toString
public java.lang.String toString()

Class COutgoingMessage

java.lang.Object
  └ org.jsmsengine.CMessage
    └ org.jsmsengine.COutgoingMessage

---

public class COutgoingMessage
extends CMessage
This class represents an outgoing SMS message, i.e. message created for dispatch from the GSM device.
See Also:
CMessage, CIncomingMessage, CPhoneBook, CService.sendMessage(COutgoingMessage), CService.sendMessage(LinkedList)

Field Summary

private java.util.Date dispatchDate

Fields inherited from class org.jsmsengine.CMessage

date, id, memIndex, MESSAGE_ENCODING_7BIT, MESSAGE_ENCODING_8BIT, MESSAGE_ENCODING_UNICODE, messageEncoding, originator, recipient, text, TYPE_INCOMING, TYPE_OUTGOING

Constructor Summary

COutgoingMessage()
Default constructor of the class.

COutgoingMessage(java.lang.String recipient, java.lang.String text)
Constructor of the class.

Method Summary

java.util.Date	getDispatchDate() Returns the dispatch date of the message.
protected java.lang.String	getPDU(java.lang.String smscNumber)
java.lang.String	getRecipient() Returns the recipient's phone number (international format).
protected void	setDispatchDate(java.util.Date date) Sets the dispatch date of the message.
void	setRecipient(java.lang.String recipient) Set the phone number of the recipient.
private java.lang.String	textToPDU(java.lang.String text)
private java.lang.String	toBCDFormat(java.lang.String s)

Methods inherited from class org.jsmsengine.CMessage

getDate, getHexText, getId, getMemIndex, getMessageEncoding, getText, getType, setDate, setId, setMessageEncoding, setText, toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

dispatchDate
private java.util.Date dispatchDate

Constructor Detail

COutgoingMessage
public COutgoingMessage()
Default constructor of the class.

COutgoingMessage
public COutgoingMessage(java.lang.String recipient,
java.lang.String text)
Constructor of the class.
Parameters:
recipient - the recipients's number.
text - the actual text of the message.

Notes:
Phone numbers are represented in their international or national format.
If you use a phonebook, the phone number may be a string starting with the '~' character, representing an entry in the phonebook.
By default, a created message is set to be encoded in 7bit. If you want to change that, be sure to operate in PDU mode, and change the encoding with setMessageEncoding() method.

Method Detail

setRecipient
public void setRecipient(java.lang.String recipient)
Set the phone number of the recipient. Applicable to outgoing messages.
Parameters:
recipient - the recipient's phone number (international format).

---

getRecipient
public java.lang.String getRecipient()
Returns the recipient's phone number (international format). Applicable only for outgoing messages.
This may be an entry from the phonebook.
Returns:
the type of the message.

---

setDispatchDate
protected void setDispatchDate(java.util.Date date)
Sets the dispatch date of the message.
Parameters:
date - the dispatch date of the message.

getDispatchDate
public java.util.Date getDispatchDate()
Returns the dispatch date of the message.
Returns:
the dispatch date of the message.

getPDU
protected java.lang.String getPDU(java.lang.String smscNumber)

textToPDU
private java.lang.String textToPDU(java.lang.String text)

toBCDFormat
private java.lang.String toBCDFormat(java.lang.String s)

Class CPhoneBook
java.lang.Object
└ org.jsmsengine.CPhoneBook

class CPhoneBook
extends java.lang.Object
This class handles the operation of the phonebook.

The phone book is an XML file, which holds information about destinations. The phone book is created and maintained by you. When you use a phone book, it is possible to send messages to "nicknames" define in the book, instead of real phone numbers. Apart from nicknames, you can also create groups of nicknames, in order to send an SMS message to more than one destinations, with only one API call.

Note: the phone book is optional.

In the "misc" directory of the distribution tree, you will find a sample phone book file. A phone book contains:
<phonebookentry> entries, which define the association of a person with a mobile number. For each entry, you must define the code (i.e. nickname), the name (description) and the actual phone.
<group> entries. These entries group together one or more phone book entries. This way, you can define a group as the recipient of your SMS message, and your message will be send to all individual members of the group.

When you create a message and you want to use a phonebook nickname (for example "thanasis"), use it with a "~" symbol in front. This means, set the recipient to value "~thanasis". When jSMSEngine sees a recipient value starting with "~", it will know that you mean a nickname, and not the actual phone. However, please keep in mind that the "~" character does not appear in the phonebook XML definition file.

This class contains all the relevant function for loading the XML phonebook file in memory (linked lists), and for resolving the names to the respected numbers.

All functions of the class are used internally by jSMSEngine API and are not accecible to the user.

Comments left to be added in next release.
See Also:
CService.setPhoneBook(java.lang.String), CService.sendMessage(org.jsmsengine.COutgoingMessage), COutgoingMessage

Nested Class Summary	
(package private) class	CPhoneBook.CParser
(package private) class	CPhoneBook.CPhoneBookEntry
(package private) class	CPhoneBook.CPhoneBookGroupEntry

Field Summary	
private java.util.LinkedList	entries
private static int	ENTRY_TYPE_ENTRY
private static int	ENTRY_TYPE_GROUP
private static int	ENTRY_TYPE_NOTFOUND
private java.util.LinkedList	groups
private static char	PHONE_BOOK_INDICATOR

Constructor Summary

CPhoneBook()	
Method Summary	
protected java.util.LinkedList	expandPhoneBookEntries(COutgoingMessage message)
protected java.util.LinkedList	expandPhoneBookEntries(java.util.LinkedList inList)
private CPhoneBook.CPhoneBookEntry	getEntry(java.lang.String code)
private java.lang.String	getEntryName(java.lang.String code)
private java.lang.String	getEntryPhone(java.lang.String code)
private int	getEntryType(java.lang.String code)
private CPhoneBook.CPhoneBookGroupEntry	getGroupEntry(java.lang.String code)
private java.util.LinkedList	getGroupMembers(java.lang.String code)
private java.lang.String	getGroupName(java.lang.String code)
protected boolean	isLoaded()
protected boolean	load(java.lang.String file)
Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

Field Detail

PHONE_BOOK_INDICATOR
private static final char PHONE_BOOK_INDICATOR
See Also:
Constant Field Values

ENTRY_TYPE_NOTFOUND
private static final int ENTRY_TYPE_NOTFOUND
See Also:
Constant Field Values

ENTRY_TYPE_ENTRY
private static final int ENTRY_TYPE_ENTRY
See Also:
Constant Field Values

ENTRY_TYPE_GROUP
private static final int ENTRY_TYPE_GROUP
See Also:
Constant Field Values

entries
private java.util.LinkedList entries

groups
private java.util.LinkedList groups

Constructor Detail

CPhoneBook
public CPhoneBook()

Method Detail

load
protected boolean load(java.lang.String file)

isLoaded
protected boolean isLoaded()

expandPhoneBookEntries
protected java.util.LinkedList expandPhoneBookEntries(COutgoingMessage message)

---

expandPhoneBookEntries
protected java.util.LinkedList expandPhoneBookEntries(java.util.LinkedList inList)

---

getEntryType
private int getEntryType(java.lang.String code)

---

getEntryName
private java.lang.String getEntryName(java.lang.String code)

---

getEntryPhone
private java.lang.String getEntryPhone(java.lang.String code)

---

getGroupName
private java.lang.String getGroupName(java.lang.String code)

---

getGroupMembers
private java.util.LinkedList getGroupMembers(java.lang.String code)

---

getEntry
private CPhoneBook.CPhoneBookEntry getEntry(java.lang.String code)

---

getGroupEntry
private CPhoneBook.CPhoneBookGroupEntry getGroupEntry(java.lang.String code)

Class CSerialDriver
java.lang.Object
└ org.jsmsengine.CSerialDriver
All Implemented Interfaces:
java.util.EventListener, javax.comm.SerialPortEventListener

---

class CSerialDriver
extends java.lang.Object
implements javax.comm.SerialPortEventListener
This class handles the operation the serial port.

This class contains all the necessary (low-level) functions that handle COMM API and are responsible for the serial communication with the GSM device.

Comments left to be added in next release.

Field Summary	
private int	baud
private static int	BUFFER_SIZE Input/Output buffer size for serial communication.
private int	dataBits
private static int	DELAY_BETWEEN_CHARS Delay (20ms) after each character sent.
private java.io.InputStream	inStream
private java.util.logging.Logger	log
private java.io.OutputStream	outStream
private int	parity
private java.lang.String	port
private javax.comm.CommPortIdentifier	portId
private static int	RECV_TIMEOUT Timeout period for the phone to respond to jSMSEngine.

private javax.comm.SerialPort	serialPort
private int	stopBits

Constructor Summary

CSerialDriver(java.lang.String port, int baud, java.util.logging.Logger log)

Method Summary

void	clearBuffer()
void	close()
boolean	dataAvailable()
int	getBaud()
int	getDataBits()
int	getParity()
java.lang.String	getPort()
java.lang.String	getResponse()
int	getStopBits()
boolean	open()
void	send(char c)
void	send(java.lang.String s)
void	serialEvent(javax.comm.SerialPortEvent event)
void	setPort(java.lang.String port)
void	skipBytes(int numOfBytes)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

RECV_TIMEOUT
private static final int RECV_TIMEOUT
Timeout period for the phone to respond to jSMSEngine.
See Also:
Constant Field Values

BUFFER_SIZE
private static final int BUFFER_SIZE
Input/Output buffer size for serial communication.
See Also:
Constant Field Values

DELAY_BETWEEN_CHARS
private static final int DELAY_BETWEEN_CHARS
Delay (20ms) after each character sent. Seems that some mobile phones get confused if you send them the commands without any delay, even in slow baud rate.
See Also:
Constant Field Values

port
private java.lang.String port

baud
private int baud

dataBits
private int dataBits

stopBits
private int stopBits

parity
private int parity

portId
private javax.comm.CommPortIdentifier portId

serialPort
private javax.comm.SerialPort serialPort

inStream
private java.io.InputStream inStream

outStream
private java.io.OutputStream outStream

log
private java.util.logging.Logger log

Constructor Detail

CSerialDriver
public CSerialDriver(java.lang.String port,
int baud,
java.util.logging.Logger log)

Method Detail

setPort
public void setPort(java.lang.String port)

getPort
public java.lang.String getPort()

getBaud
public int getBaud()

getDataBits
public int getDataBits()

getStopBits
public int getStopBits()

getParity
public int getParity()

open
public boolean open()
throws java.lang.Exception
Throws:
java.lang.Exception

close
public void close()

serialEvent
public void serialEvent(javax.comm.SerialPortEvent event)
Specified by:
serialEvent in interface javax.comm.SerialPortEventListener

clearBuffer
public void clearBuffer()
throws java.lang.Exception
Throws:
java.lang.Exception

send
public void send(java.lang.String s)
throws java.lang.Exception
Throws:
java.lang.Exception

---

send
public void send(char c)
throws java.lang.Exception
Throws:
java.lang.Exception

---

skipBytes
public void skipBytes(int numOfBytes)
throws java.lang.Exception
Throws:
java.lang.Exception

---

dataAvailable
public boolean dataAvailable()
throws java.lang.Exception
Throws:
java.lang.Exception

---

getResponse
public java.lang.String getResponse()
throws java.lang.Exception
Throws:
java.lang.Exception

---

class CService
java.lang.Object
└ org.jsmsengine.CService

---

public class CService
extends java.lang.Object
This class provides all the functionality of jSMSEngine API to the developer.

The class CService provides all the interface routines to jSMSEngine. It is responsible for initialization of the communication with the GSM device, reading and sending messages, setting the phonebook.

The sequence of actions that need to be done are:
Call initialize() to setup the service.
Call connect() to connect with the GSM device.
Call sendMessage(), or readMessages() to send or receive messages from the device. Call deleteMessage() to delete a message from the device's memory.
Call refreshDeviceInfo() to get updated GSM device specific information.
Call disconnect() to disconnect from the GSM device.

Nested Class Summary

private class CService.CReceiveThread

Field Summary

static java.lang.String	_name Internal Software Name.
static java.lang.String	_reldate Release Date.
private java.lang.Object	_SYNC_ Synchronization object for critical sections of the API.
static java.lang.String	_version Version.
private java.lang.String	cacheDir
private boolean	connected
static java.lang.String	DEFAULT_VALUE_NOT_REPORTED Default value for information that is not reported by the GSM device.

private CDeviceInfo	deviceInfo
static int	ERR_CANNOT_DISABLE_INDICATIONS This error value is returned when the GSM device does not support the AT+CNMI command for disabling indications to TE.
static int	ERR_CHARSET_HEX_NOT_SUPPORTED This error value is returned when the GSM device does not support HEX mode.
static int	ERR_COMM_NOT_SUPPORTED This error value is returned when the GSM device does not support ASCII or PDU mode.
static int	ERR_GENERIC_ERROR This is a generic error, which is not classified yet.
static int	ERR_INVALID_DIR This error value is returned when the given directory is invalid.
static int	ERR_MESSAGE_NOT_FOUND This error value is returned when the specific message was not found.
static int	ERR_NO_CACHE This error value on attempting to connect to the GSM device without first having defined the cache directories.
static int	ERR_NOT_CONNECTED This error value is returned when the service is not connected to the GSM device.
static int	ERR_NOT_INITIALIZED This error value is returned when the service is not initialized yet.
static int	ERR_NOT_SUPPORTED This error value is returned when the specified operation is not supported by jSMSEngine API.
static int	ERR_OK This error value is returned when the operation was succesfull.
static int	ERR_PHONEBOOK_NOT_LOADED This error value is returned when the specified phonebook file did not load.
static int	ERR_SEND_FAILED This error value is returned when a send-message operation failed.
static int	ERR_SIM_PIN_ERROR This error value is returned when the GSM device asks for a PIN number, however the PIN given is invalid.
private boolean	initialized
private static java.util.logging.Logger	log Logging facilities.
static int	MAX_SMS_LEN_7BIT
static int	MAX_SMS_LEN_8BIT
static int	MAX_SMS_LEN_UNICODE
static int	MODE_ASCII Constant value for ASCII operation mode.
static int	MODE_PDU Constant value for PDU operation mode.
private int	operationMode
private CPhoneBook	phoneBook
static int	RECEIVE_MODE_ASYNC

static int	RECEIVE_MODE_SYNC Receive modes: Synchronous and Ascynchronous.
private int	receiveMode
private CService.CReceiveThread	receiveThread
private CSerialDriver	serialDriver
private java.lang.String	simPin
private static int	SMS_PARTS
private static java.lang.String	SMS_SPLIT_SIGNATURE
private java.lang.String	smscNumber
private static int	smsSplitId
private int	supportedModes

Constructor Summary

CService(java.lang.String port, int baud) Default constructor of the class.

Method Summary

int	connect() Connects to the GSM device.
int	deleteMessage(CIncomingMessage message) Deletes an SMS message from the GSM device memory.
int	deleteMessage(int memIndex) Deletes an SMS message from the GSM device memory.
int	disconnect() Disconnects to the GSM device.
private int	getBatteryLevel()
java.lang.String	getCacheDir() Returns the cache directory for messages.
boolean	getConnected() Returns TRUE if the service is connected with the GSM device.
CDeviceInfo	getDeviceInfo() Returns a CDeviceInfo object that holds information about the GSM device in use.
private java.lang.String	getImsi()
boolean	getInitialized() Returns TRUE if the service has already been initialized.
private java.lang.String	getManufacturer()
private java.lang.String	getModel()
int	getOperationMode() Returns the operation mode of the GSM device, i.e. one of the values MODE_ASCII, MODE_PDU.
int	getReceiveMode() Returns the reception mode.
private java.lang.String	getSerialNo()

private int	getSignalLevel()
java.lang.String	getSimPin() Returns the SIM pin number.
java.lang.String	getSmscNumber() Returns the Short Message Service Center (SMSC) number you have previously defined with setSmscNumber().
private int	getSmsSplitId()
private java.lang.String	getSwVersion()
int	initialize() Initializes the service.
private boolean	isIncomingMessage(java.lang.String pdu) Checks if the message is SMS-DELIVER (incoming) or SMS-SUBMIT
static void	main(java.lang.String[] args)
int	readMessages(java.util.LinkedList messageList, int messageClass) Reads SMS from the GSM device's memory.
boolean	received(CIncomingMessage message) Virtual method, called upon receipt of a message (Asynchronous mode only!)
int	refreshDeviceInfo() Refreshes the GSM device specific information.
int	sendMessage(COutgoingMessage message) Send an SMS message from the GSM device.
int	sendMessage(java.util.LinkedList messageList) Send an series of SMS messages from the GSM device.
int	setCacheDir(java.lang.String dir) Sets the cache directory for messages.
private void	setConnected(boolean connected)
private void	setInitialized(boolean initialized)
boolean	setOperationMode(int mode) Sets the operation mode of the GSM device
int	setPhoneBook(java.lang.String phoneBookFile) Loads the phonebook.
void	setReceiveMode(int receiveMode) Sets the reception mode.
void	setSimPin(java.lang.String simPin) Sets the SIM pin number.
void	setSmscNumber(java.lang.String smscNumber) Sets the Short Message Service Center (SMSC) number.
private java.util.LinkedList	splitLargeMessages(java.util.LinkedList messageList)

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

_name
public static final java.lang.String _name
Internal Software Name.
See Also:
Constant Field Values

_version
public static final java.lang.String _version
Version.
See Also:
Constant Field Values

_reldate
public static final java.lang.String _reldate
Release Date.
See Also:
Constant Field Values

---

log
private static java.util.logging.Logger log
Logging facilities.

ERR_OK
public static final int ERR_OK
This error value is returned when the operation was succesfull.
See Also:
Constant Field Values

ERR_GENERIC_ERROR
public static final int ERR_GENERIC_ERROR
This is a generic error, which is not classified yet. More error classifications may be introduced at a later stage.
See Also:
Constant Field Values

ERR_NOT_INITIALIZED
public static final int ERR_NOT_INITIALIZED
This error value is returned when the service is not initialized yet. You should call method initialize().
See Also:
Constant Field Values

ERR_NOT_CONNECTED
public static final int ERR_NOT_CONNECTED
This error value is returned when the service is not connected to the GSM device. You should call method connect().
See Also:
Constant Field Values

ERR_COMM_NOT_SUPPORTED
public static final int ERR_COMM_NOT_SUPPORTED
This error value is returned when the GSM device does not support ASCII or PDU mode. This is a fatal error, in the sense that jSMSEngine can work only with GSM devices supporting ASCII or PDU Mode.
See Also:
Constant Field Values

ERR_CHARSET_HEX_NOT_SUPPORTED
public static final int ERR_CHARSET_HEX_NOT_SUPPORTED
This error value is returned when the GSM device does not support HEX mode. This is a fatal error, in the sense that jSMSEngine can work only with GSM devices supporting HEX Mode when in ASCII mode. In order to get around this error, switch to PDU mode.
See Also:
Constant Field Values

ERR_CANNOT_DISABLE_INDICATIONS
public static final int ERR_CANNOT_DISABLE_INDICATIONS
This error value is returned when the GSM device does not support the AT+CNMI command for disabling indications to TE.
See Also:
Constant Field Values

ERR_MESSAGE_NOT_FOUND
public static final int ERR_MESSAGE_NOT_FOUND
This error value is returned when the specific message was not found. Double-check your message and/or memory index used.
See Also:
Constant Field Values

ERR_SEND_FAILED
public static final int ERR_SEND_FAILED
This error value is returned when a send-message operation failed. This could be attributed to a number of reasons: Coverage problems, invalid recipient phone number, GSM device malfunction.
See Also:
Constant Field Values

ERR_PHONEBOOK_NOT_LOADED
public static final int ERR_PHONEBOOK_NOT_LOADED
This error value is returned when the specified phonebook file did not load. Recheck your XML file for errors in its structure.

See Also:
Constant Field Values

ERR_INVALID_DIR
public static final int ERR_INVALID_DIR
This error value is returned when the given directory is invalid.
See Also:
Constant Field Values

ERR_NO_CACHE
public static final int ERR_NO_CACHE
This error value on attempting to connect to the GSM device without first having defined the cache directories.
See Also:
Constant Field Values

ERR_SIM_PIN_ERROR
public static final int ERR_SIM_PIN_ERROR
This error value is returned when the GSM device asks for a PIN number, however the PIN given is invalid. Please check your PIN.
See Also:
Constant Field Values

ERR_NOT_SUPPORTED
public static final int ERR_NOT_SUPPORTED
This error value is returned when the specified operation is not supported by jSMSEngine API.
See Also:
Constant Field Values

MODE_ASCII
public static final int MODE_ASCII
Constant value for ASCII operation mode.
See Also:
Constant Field Values

MODE_PDU
public static final int MODE_PDU
Constant value for PDU operation mode.
See Also:
Constant Field Values

RECEIVE_MODE_SYNC
public static final int RECEIVE_MODE_SYNC
Receive modes: Synchronous and Ascynchronous.
See Also:
Constant Field Values

RECEIVE_MODE_ASYNC
public static final int RECEIVE_MODE_ASYNC
See Also:
Constant Field Values

DEFAULT_VALUE_NOT_REPORTED
public static final java.lang.String DEFAULT_VALUE_NOT_REPORTED
Default value for information that is not reported by the GSM device.
See Also:
Constant Field Values

MAX_SMS_LEN_7BIT
public static final int MAX_SMS_LEN_7BIT
See Also:
Constant Field Values

MAX_SMS_LEN_8BIT
public static final int MAX_SMS_LEN_8BIT
See Also:
Constant Field Values

MAX_SMS_LEN_UNICODE
public static final int MAX_SMS_LEN_UNICODE
See Also:
Constant Field Values

SMS_SPLIT_SIGNATURE
private static final java.lang.String SMS_SPLIT_SIGNATURE
See Also:

Constant Field Values

SMS_PARTS
private static final int SMS_PARTS
See Also:
Constant Field Values

smsSplitId
private static int smsSplitId

cacheDir
private java.lang.String cacheDir

smscNumber
private java.lang.String smscNumber

simPin
private java.lang.String simPin

operationMode
private int operationMode

supportedModes
private int supportedModes

receiveMode
private int receiveMode

serialDriver
private CSerialDriver serialDriver

initialized
private boolean initialized

connected
private boolean connected

phoneBook
private CPhoneBook phoneBook

deviceInfo
private CDeviceInfo deviceInfo

receiveThread
private CService.CReceiveThread receiveThread

SYNC
private java.lang.Object _SYNC_
Synchronization object for critical sections of the API.

Constructor Detail

CService
public CService(java.lang.String port,
int baud)
Default constructor of the class.
Parameters:
port - the serial port where the GSM device is connected (e.g. "com1").
baud - the connection speed (i.e. 9600, 19200 etc).

Notes:
Use one of the standard values for baud. Most GSM devices work well at 9600 or 19200. Some may handle speeds up to 115200 (like Nokia mobile phone model 6210 does). The connection speed is not that important to the speed at which jSMSEngine processes messages. Personally, I work at 9200 to avoid pushing the mobile. Dedicated GSM modems may handle higher speeds better than mobile phones do.

Method Detail

getInitialized
public boolean getInitialized()
Returns TRUE if the service has already been initialized.
Returns:
TRUE if the service has already been initialized.

getConnected
public boolean getConnected()
Returns TRUE if the service is connected with the GSM device.
Returns:

TRUE if the service is connected with the GSM device.

---

getDeviceInfo
public CDeviceInfo getDeviceInfo()
Returns a CDeviceInfo object that holds information about the GSM device in use.
Returns:
a CDeviceInfo object.
See Also:
CDeviceInfo

---

setCacheDir
public int setCacheDir(java.lang.String dir)
Sets the cache directory for messages.
Parameters:
dir - The directory which will act like a cache.
Returns:
One of ERR_* values.

---

setSmscNumber
public void setSmscNumber(java.lang.String smscNumber)
Sets the Short Message Service Center (SMSC) number. Please use international format. If you don't want to set the SMSC and use the one defined in your GSM device, use an empty string parameter. Another way to do the same, is to pass a null parameter. Some phones may prefer one way or the other - please test your phone.
Parameters:
smscNumber - the SMSC number.

---

getSmscNumber
public java.lang.String getSmscNumber()
Returns the Short Message Service Center (SMSC) number you have previously defined with setSmscNumber().
Returns:
the SMSC number.

---

setSimPin
public void setSimPin(java.lang.String simPin)
Sets the SIM pin number. This is used if and when the GSM device asks for it. If you set it to null, then the API does not give any PIN to the device (in order to avoid locking it up), and returns ERR_SIM_PIN_ERROR.
Parameters:
simPin - the SIM pin number.

---

getSimPin
public java.lang.String getSimPin()
Returns the SIM pin number.
Returns:
the SIM pin number.

---

setOperationMode
public boolean setOperationMode(int mode)
Sets the operation mode of the GSM device
Parameters:
mode - the mode of operation (one of values MODE_ASCII, MODE_PDU).
Returns:
TRUE if the change of mode succeded.
See Also:
getOperationMode()

---

getOperationMode
public int getOperationMode()
Returns the operation mode of the GSM device, i.e. one of the values MODE_ASCII, MODE_PDU.
Returns:
the operation mode.
See Also:
setOperationMode(int)

---

setReceiveMode
public void setReceiveMode(int receiveMode)
Sets the reception mode. There are two reception modes, the synchronous and the asynchronous. In synchronous mode, you should call readMessages() function on demand, where you want to check for new messages. In asynchronous mode, the engine automatically calls the received() method (which you should override) for every received message.
By default, the reception mode is the synchronous one.
Parameters:
receiveMode - the reception mode (one of values RECEIVE_MODE_ASYNC, RECEIVE_MODE_SYNC).
See Also:
getReceiveMode()

getReceiveMode
public int getReceiveMode()
Returns the reception mode.
Returns:
the reception mode (one of values RECEIVE_MODE_ASYNC, RECEIVE_MODE_SYNC).
See Also:
setReceiveMode(int)

---

getCacheDir
public java.lang.String getCacheDir()
Returns the cache directory for messages.
Returns:
the caching directory.
See Also:
setCacheDir(String)

---

initialize
public int initialize()
Initializes the service. This should be the first method call.
Returns:
ERR_OK (for this version).
See Also:
connect()

---

connect
public int connect()
Connects to the GSM device. Opens the serial port, and sends the appropriate AT commands to initialize the operation mode of the GSM device. Retrieves information about the GSM device. This method should be called after initialize() has been called.
By default, jSMSEngine API sets your GSM device to PDU mode. If you want to switch to ASCII mode (I don't see any reason why, but anyway...), use the setOperationMode() method.

Notes:
The GSM device specific information (read by the call to refreshDeviceInfo() function is called once from this method. Since some information changes with time (such as battery or signal level), its your responsibility to call refreshDeviceInfo() periodically in order to have the latest information. Otherwise, you will get the information snapshot taken at the time of the initial connection.
Returns:
One of ERR_* values.
See Also:
CDeviceInfo, refreshDeviceInfo(), disconnect(), initialize(), setOperationMode(int)

---

disconnect
public int disconnect()
Disconnects to the GSM device. Closes the serial port.
Returns:
ERR_OK value.
See Also:
connect()

---

setPhoneBook
public int setPhoneBook(java.lang.String phoneBookFile)
Loads the phonebook. The phonebook is an XML file containing associations of name and phone numbers.

The phonebook is optional.
Parameters:
phoneBookFile - The XML full-path name which keeps the phonebook.
Returns:
One of ERR_* values.
See Also:
CPhoneBook, sendMessage(COutgoingMessage), sendMessage(LinkedList)

---

refreshDeviceInfo
public int refreshDeviceInfo()
Refreshes the GSM device specific information. This method is called once during connection. Its up to the developer to call it periodically in order to get the latest information.
Returns:
One of ERR_* values.
See Also:
CDeviceInfo, connect(), getDeviceInfo()

---

readMessages
public int readMessages(java.util.LinkedList messageList,
int messageClass)

Reads SMS from the GSM device's memory. You should call this method when you want to read messages from the device. In the MessageList object you pass, the method will add objects of type CIncomingMessage, as many of them as the messages pending to be read. The class defines which types of messages should be read.

Notes:
The method does not delete the messages it reads from the GSM device. It's your responsibility to delete them, if you don't want them. Otherwise, on the next call of this function you will read the same messages.
IMPORTANT NOTE: This version of jSMSEngine will read and process only received messages, and not stored messages - regardless of the class you requested.
Parameters:
messageList - a LinkedList object which will be loaded with the messages.
messageClass - one of the CLASS_* values defined in CIncomingMessage class which define what type of messages are to be read.
Returns:
One of ERR_* values.
See Also:
CIncomingMessage, deleteMessage(CIncomingMessage), deleteMessage(int)

---

sendMessage
public int sendMessage(COutgoingMessage message)
Send an SMS message from the GSM device. Once connected, you can create a COutgoingMessage object with the message you want to send, and pass it to this function.

Notes:
If you have set a phonebook, you can create the COutgoingMessage object with a nickname, instead of the actual phone number.
Parameters:
message - a COutgoingMessage containing the message you wish to send.
Returns:
One of ERR_* values.
See Also:
COutgoingMessage, CPhoneBook, sendMessage(LinkedList), setPhoneBook(String)

---

sendMessage
public int sendMessage(java.util.LinkedList messageList)
Send an series of SMS messages from the GSM device. This method is used when you want to send more than one message as a batch. If your GSM device support the feature of keeping the GSM link open during message dispatch, this method should work faster than calling the sendMessage(COutgoingMessage) method many times.
Just create a LinkedList object, add as many COutgoingMessage objects you wish and call the method.

Notes:
If you have set a phonebook, you can create the COutgoingMessage object with a nickname, instead of the actual phone number.
Parameters:
messageList - a LinkedList filled with COutgoingMessage objects.
Returns:
One of ERR_* values.
See Also:
COutgoingMessage, CPhoneBook, sendMessage(COutgoingMessage), setPhoneBook(String)

---

deleteMessage
public int deleteMessage(CIncomingMessage message)
Deletes an SMS message from the GSM device memory.

Notes:
A deleted message cannot be recovered.
Parameters:
message - a valid CIncomingMessage object, i.e. an object which is previously read with readMessages() from the GSM device.
Returns:
One of ERR_* values.
See Also:
CIncomingMessage, deleteMessage(int)

---

deleteMessage
public int deleteMessage(int memIndex)
Deletes an SMS message from the GSM device memory.

Notes:
A deleted message cannot be recovered.
It is highly recommended to use the other form of the deleteMessage() method.
Parameters:
memIndex - the memory index of the GSM device's memory from where the message (if there is any message there) should be deleted.
Returns:

One of ERR_* values.
See Also:
deleteMessage(CIncomingMessage)

---

received
public boolean received(CIncomingMessage message)
Virtual method, called upon receipt of a message (Asynchronous mode only!)

Notes:
If you plan to use jSMSEngine API in asynchronous mode, you should override this method, making it do your job upon message receipt.
Parameters:
message - the received message.
Returns:
return true if you wish the message to be deleted from the GSM device's memory. Otherwise false.
See Also:
setReceiveMode(int)

---

isIncomingMessage
private boolean isIncomingMessage(java.lang.String pdu)
Checks if the message is SMS-DELIVER (incoming) or SMS-SUBMIT
Parameters:
pdu - the message pdu
Returns:
true if the message is SMS-DELIVER

---

setConnected
private void setConnected(boolean connected)

---

setInitialized
private void setInitialized(boolean initialized)

---

getManufacturer
private java.lang.String getManufacturer()
throws java.lang.Exception
Throws:
java.lang.Exception

---

getModel
private java.lang.String getModel()
throws java.lang.Exception
Throws:
java.lang.Exception

---

getSerialNo
private java.lang.String getSerialNo()
throws java.lang.Exception
Throws:
java.lang.Exception

---

getImsi
private java.lang.String getImsi()
throws java.lang.Exception
Throws:
java.lang.Exception

---

getSwVersion
private java.lang.String getSwVersion()
throws java.lang.Exception
Throws:
java.lang.Exception

---

getBatteryLevel
private int getBatteryLevel()
throws java.lang.Exception
Throws:
java.lang.Exception

---

getSignalLevel
private int getSignalLevel()
throws java.lang.Exception
Throws:
java.lang.Exception

splitLargeMessages

private java.util.LinkedList splitLargeMessages(java.util.LinkedList messageList)

---

getSmsSplitId
private int getSmsSplitId()

---

main
public static void main(java.lang.String[] args)

Class CUtils
java.lang.Object
└ org.jsmsengine.CUtils

---

public class CUtils
extends java.lang.Object
This class has some general purpose functions.

Constructor Summary

CUtils()

Method Summary

static java.lang.String	substituteSymbol(java.lang.String text, java.lang.String symbol, java.lang.String value) String substitution routine.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

CUtils
public CUtils()

Method Detail

substituteSymbol

public static java.lang.String substituteSymbol(java.lang.String text,
java.lang.String symbol,
java.lang.String value)
String substitution routine.
Parameters:
text - the initial text.
symbol - the string to be substituted.
value - the string that the "symbol" will be substituted with, in the "text" (all occurences).
Returns:
the changed text.

โปรแกรมติดตามรถพยาบาล

All Classes

AbstractMap
AbstractMapEvent
AbstractMapEventListener
AbstractMapFactory
AgeContext
AgeContextFactory
AICommand
ALabel
AlgoAssociation
AlgoParams
AlgoProperties
AlgoStatDialog
AlgoStatDisplayer
AlgoStatDisplayerFactory
AlgoStatPanel
AmbId
Ambulance
AmbulanceCreationTest
AmbulanceStation
AProgressDialog
AProgressMonitor
APSS
APSSFactory
APSSI
APSSPathFinder
AStarAlgo
AStarBasic
AStarBasicHeur
AStarForProduction
AStarFrame
AStarFudge
AStarInterface
AStarSequential
AStarSpeedOptimized
AStarSpeedOptimizedHeur
AuthentificationException
BlankEnv
BreachedCellGroup
CACControler
CAClientException
CACMaintenanceService
CacMediatorI
CACMissionRecordingException
CACNetworkException
CacParams
CaCWorld
AMBAmbulanceChooser

AMBConnection
AMBConnectionFactory
AMBMissionManager
AMBMissionTracker
CAGUIframe
CAGUIFrame2
CAGUIFrameAWT
CAGUIPanel
Car
CAUserInterface
Cell
CellGroup
CellMap
CellRectangle
CellSquare
ChangeDebuLevelDialog
ClassicCellMap
ClassicCellMapNoDiag
ClassicStatusBar
CoherenceChecker
Command
Commandable
ComputerCommand
ConnectDialog
ConnectionData
ConnectionDataException
ConnectionDataFactory
ConnectionReport
ConnectionReportFactory
ConnectionReportI
ConnectionToObjectException
ControlPanel
CoreTimeMaster
CustomMapsFileFilter
CustomMapsPanelChooser
DarkForestEnv
DeapSeaEnv
DefaultGenome
DefaultOwner
DesignMapFeaturesPanel
DirtRoadEnv
Drawable2DObject
DumForm
DumGPS
DumMDT
DumMissionTracker
DummyIncident
DumPersistentDataLinkI
DumSimViewer
EarthEnv

EmergencyVehicule
EmergencyVehiculeFactory
EmergencyVehiculeI
EnhancedDialog
EnvType
ErrorMessages
FactoryException
FieldSet
FieldSetFactory
FieldSetI
Filler
FillerCatalog
FillerServer
FillerServerFactory
Form
FormException
FormFactory
FormI
Gaia
GaiaGenome
GenericMapsPanelChooser
Genomable
GPS
gps_serverI
gps_serverI_Stub
GPSCentralizer
GPSCentralizerFactory
GPSConnector
GPSDeviceI
GPSPanel
GPSPosition
GPSPositionFactory
GPSSim
GPSTask
GrassEnv
Gravity
GravityFactory
GravityI
GridCell
HorCircularCellMap
Hospital
Id
IdFactory
IdI
IDPathUser
ImageLibrary
ImagesFileNameFilter
Incident
IncidentFactory
IncidentI

IntegerSemaphor
LargeStoneRoadEnv
LocalTestSimNoDb
LondonGenome
ROIETService
MainFrame
MapCatalog
MapDisplayer
MapDisplayerFactory
MapFieldPanel
MapFileFilter
MapPanelChooser
MapToolBar
MapUIEvent
MapUIEventListener
MapUnit
MDT
mdt_serverI
mdt_serverI_Stub
MDTCentralizer
MDTCentralizerFactory
MDTConnector
MDTConnectorFactory
MDTDeviceI
MDTFrame
MDTPanel
MDTSim
MDTStatus
MDTTask
MersenneTwister
MersenneTwisterFast
MiniMap
MiniMapContainer
MiniMapDisplayer
MissionManager
MissionManagerFactory
MissionRecorder
MissionRecorderFactory
MissionRecorderI
MissionTracker
MissionTrackerFactory
ModernAge
MoveModStats
MTAdapter
MTFAdapter
MyUnit
MyWorld
NetworkAcceptor
NetworkAcceptorFactory
NetworkConnector

NetworkConnectorFactory
NonImplPanel
ObjectDisplayer
ObjectMediator
ObjectProducer
OpenSavePanel
Owner
OwnerFactory
ParamControler
PassiveStructure
Path
PathException
PathFindable
PathFinder
PathFinderFactory
PathPlace
PathPlaceEvent
PathPlaceEventListener
PathPlaceSet
PathUser
PersistentDataException
PersistentDataLink
PersistentDataLinkFactory
PersistentDataLinkI
Place
PlaceFactory
PlaceI
PlayPanel
RandomGenerator
RandomGeneratorFactory
Region
RMIConnectionData
RMIConnectionReport
RMIConnector
RMILink
RMINetworkAcceptor
RMINetworkAcceptor_Skel
RMINetworkAcceptor_Stub
Run
RunMainServer
Selection
ServerFrame
ServerUIFactory
ShowDialog
SimAccident
SimAmbulance
SimEnvironment
SimEnvironmentFactory
SimGPSPanel
SimMDTPanel

SimMediator
SimObject
SimObjectDataPanel
SimObjectImagePanel
SimObjectServer
SimObjectServerFactory
SimObjectTask
SimPanel
SimParams
SimPartsFactory
SimToolBar
SimulatedObject
SimulationFactory
Simulator
SimUnitFeaturePanel
SimViewer
SimWorldCAC
SoftSandEnv
SphericalCellMap
SplashWindow
StatComputer
StatException
StatServer
StatServerFactory
Status
StatusFactory
StatusTextDisplayer
StringsParam
Structure
StructureCatalog
StructureChooserPanel
StructureFactory
StructureMediator
TabbedFeaturesChooser
Team
TerrainPanelChooser
TerrainType
TerrainTypeControlerException
TerrainTypeProducer
test
test
test
test
TestAccidentStats
TestAlgoPackage
TestAstarSequential
testGPSSim
TestMap
testMDTSim
TestRunning

TestSimMainServerConnection
TestTracking
TestUnitStats
TextPanePanel
TopControl
ToroidalCellMap
ToroidalCellMapNoDiag
TrackerReport
Tui
Type
TypeFactory
TypeI
UIMediator
UIParams
UnitChooserPanel
UnitControler
UnitException
UnitMediator
UnitStats
UserCommand
UserInterface
VacuumEnv
VehiculeAssociationException
VehiculeChooser
VehiculeChooserFactory
VehiculeException
VerCircularCellMap
Villager
WhitePaperMapPanel
WorldFileFilter