ระบบสารสนเทศภูมิศาสตร์ผ่านเครือข่ายอินเทอร์เน็ต เพื่อเลือกเส้นทางที่ใช้เวลาน้อยที่สุด โดยระบบขนส่งมวลชนหลายรูปแบบ

นายมนต์ศักดิ์ โซ่เจริญธรรม

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมสำรวจ ภาควิชาวิศวกรรมสำรวจ
คณะวิศวกรรมศาตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2549
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

INTERNET GIS FOR SELECTING SHORTEST TRAVELING TIME ROUTE IN MULTIMODE MASS TRANSPORTATION SYSTEM

Mr. Monsak Socharoentum

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Survey Engineering
Department of Survey Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2006

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	ระบบสารสนเทศภูมิศาสตร์ผ่านเครือข่ายอินเทอร์เน็ต เพื่อเลือกเส้นทางที่ใช้เวลาน้อยที่สุดโดยระบบขนส่งมวลชนหลายรูปแบบ
โดย	นายมนต์ศักดิ์ โซ่เจริญธรรม
สาขาวิชา	วิศวกรรมสำรวจ
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์สุทธิพงศ์ วิญญูประดิษฐ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ดร.ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์สวัสดิ์ชัย เกรียงไกรเพชร)

..... อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์สุทธิพงศ์ วิญญูประดิษฐ์)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ดร.ชนินทร์ ทินนโชติ)

..... กรรมการ
(รองศาสตราจารย์วิชัย เยี่ยงวีรชน)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

มนต์ศักดิ์ โชเจริญธรรม: ระบบสารสนเทศภูมิศาสตร์ผ่านเครือข่ายอินเทอร์เน็ต เพื่อเลือกเส้นทางที่ใช้เวลาน้อยที่สุดโดยระบบขนส่งมวลชนหลายรูปแบบ. (INTERNET GIS FOR SELECTING SHORTEST TRAVELING TIME ROUTE IN MULTIMODE MASS TRANSPORTATION SYSTEM) อ. ที่ปรึกษา : ผศ.สุทธิพงศ์ วิญญูประดิษฐ์, 134 หน้า.

งานวิจัยนี้เป็นการศึกษาพฤติกรรมของระบบขนส่งมวลชน 5 ประเภท คือ รถโดยสารประจำทาง เรือโดยสาร เรือข้ามฟาก รถไฟฟ้าบีทีเอส และรถไฟฟ้ามหานคร (รถไฟฟ้าใต้ดิน) ของกรุงเทพมหานคร เพื่อจะวิเคราะห์ และออกแบบแบบจำลองข้อมูลสำหรับรองรับและจำลองพฤติกรรมของระบบขนส่งมวลชนดังกล่าว โดยเน้นเพื่อให้สามารถใช้คำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดบนระบบขนส่งมวลชนในลักษณะผสมผสานกันได้

ในกระบวนการวิจัยได้มีวิเคราะห์ ออกแบบ และนำเสนอแบบจำลองข้อมูลที่สามารถจัดเก็บและจำลองพฤติกรรมของระบบขนส่งมวลชนดังกล่าว ตลอดจนข้อมูลประกอบอื่นๆ ที่จำเป็นต่อการคำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดบนระบบขนส่งมวลชนหลายรูปแบบในลักษณะผสมผสานกัน เมื่อได้โครงสร้างข้อมูลที่เหมาะสมแล้ว ได้ทดลองพัฒนาโปรแกรมซึ่งทำงานบนเว็บเบราว์เซอร์และเครือข่ายอินเทอร์เน็ต สำหรับทำการสืบค้น คำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด เพื่อทดสอบแบบจำลองข้อมูลที่ออกแบบ รวมทั้งทดลองประยุกต์ใช้ทฤษฎีทางด้านการวิเคราะห์โครงข่าย

แบบจำลองข้อมูลและโปรแกรมที่พัฒนาขึ้น สามารถคำนวณเปรียบเทียบและคัดเลือกเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด โดยเป็นลักษณะผสมผสานกันของระบบขนส่งมวลชนหลายประเภทต่อการเดินทางแต่ละเส้นทาง สามารถพิจารณาและรองรับการเปลี่ยนแปลงค่าตัวแปรระยะเวลารอคอยยานพาหนะ ระยะเวลาจอดรับส่งผู้โดยสาร ตารางเวลาการให้บริการของระบบขนส่งมวลชนแต่ละรูปแบบ อัตราเร็วยานพาหนะแยกอิสระกันเป็นเส้นทางย่อยๆ ตลอดจนอัตราเร็วในการเดินเท้าเพื่อเปลี่ยนรูปแบบการเดินทาง ได้

ภาควิชา วิศวกรรมสำรวจ ลายมือชื่อนิสิต มนต์ศักดิ์ โชเจริญธรรม

สาขาวิชา วิศวกรรมสำรวจ ลายมือชื่ออาจารย์ที่ปรึกษา

ปีการศึกษา 2549 ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

4670438421 : MAJOR SURVEY ENGINEERING

KEY WORD: MULTIMODE PUBLIC TRANSPORT / SHORTEST TRAVELING TIME / INTELLIGENT TRANSPORT SYSTEMS / DIJKSTRA

MONSAK SOCHAROENTUM : INTERNET GIS FOR SELECTING SHORTEST TRAVELING TIME ROUTE IN MULTIMODE MASS TRANSPORTATION SYSTEM. THESIS ADVISOR : ASSISTANT PROFESSOR SOOTTIPONG WINYOOPRADIST, 134 pp.

This research is to study the behaviour of five public transports in Bangkok (Bus, Sky Train, Subway, Ferry, Chaophraya Express Boat) and to analyse and design a suitable data model for their behaviours. This research intend to develop a localized data model that support computing and finding shortest travelling time route on multimodal public transport in Bangkok.

After the study and analysis, a data model was created. The purpose of this data model is to contain mass transport and traveling behaviors as well as other related attributes such as speed, waiting time, available time, etc. In the latter part of the research, the development of computer program for testing the proposed data model and conducting an experiment on applying network analysis theories to the data model.

The synergy of the proposed data structure and the developed computer program is the capability to compare and find the shortest traveling time route by one or more public transport modes. They are also able to compute traveling time base on dynamic value of vehicle speed in each section of route and waiting time for each type of vehicle. The starting date and time are also inputted and used to consult the service period recorded in database about the available of the public transports. A further research may be the development of processes that cab consider every traveling time factors based on time and date constraint.

Department .....Survey Engineering..... Student's signature ..Monsak Socharoentum

Field of study.....Survey Engineering..... Advisor's signature.....

Academic year...2006..... Co-advisor's signature.....

กิตติกรรมประกาศ

ผู้เขียนขอขอบพระคุณอาจารย์ที่ปรึกษาวิทยานิพนธ์ ผศ. สุทธิพงศ์ วิญญูประดิษฐ์ ที่ช่วยให้คำปรึกษาแนะนำพร้อมทั้งแนวคิดต่างๆ เป็นอย่างดี

ขอขอบพระคุณบิดามารดา ผู้ซึ่งให้การสนับสนุน ดูแลข้าพเจ้า ตลอดระยะเวลาการศึกษา อีกทั้งอบรมสั่งสอนและสนับสนุนบูรพวิทยาแก่ข้าพเจ้า จนสามารถเติบใหญ่และสำเร็จการศึกษาขั้นสูงในที่สุด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ง

บทคัดย่อภาษาอังกฤษ.....จ

กิตติกรรมประกาศ.....ฉ

สารบัญ.....ช

บทที่ 1 บทนำ.....1

ความเป็นมาและความสำคัญของปัญหา.....1

วัตถุประสงค์ของการวิจัย.....2

ขอบเขตของการวิจัย.....2

ประโยชน์จากการวิจัย.....3

ขั้นตอนดำเนินการวิจัย.....3

ข้อจำกัดของการวิจัย.....4

บทที่ 2 แนวคิด เอกสารและงานวิจัยที่เกี่ยวข้อง.....9

แนวคิดและทฤษฎีที่เกี่ยวข้อง.....9

เอกสารและงานวิจัยที่เกี่ยวข้อง.....16

บทที่ 3 วิธีดำเนินการวิจัย.....34

ขั้นตอนต่างๆ ในการวิจัย.....34

การจำลองสภาพความเป็นจริงให้มาอยู่ในแบบจำลองและโครงสร้างข้อมูลซึ่งเป็นนามธรรม.....36

ผลการศึกษาพฤติกรรมของระบบขนส่งมวลชนและการเดินทาง.....38

ความแตกต่างระหว่างพฤติกรรมจริงและพฤติกรรมที่พยายามจำลอง.....38

การออกแบบแบบจำลองและโครงสร้างข้อมูล.....46

เครื่องมือที่ใช้ในการวิจัย.....59

ข้อมูลที่เกี่ยวข้อง.....62

การจัดเตรียมและปรับแก้ข้อมูล.....63

การพัฒนาโปรแกรมเพื่อคำนวณเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด.....73

หน้า

บทที่ 4 ผลการวิเคราะห์ข้อมูล……78
ผลการวิเคราะห์ข้อมูลพฤติกรรมของระบบขนส่งมวลชน
(การออกแบบโครงสร้างข้อมูล)……78
ผลการออกแบบและพัฒนาโปรแกรมฯ…… 80
ตัวอย่างผลลัพธ์จากการทดลองสืบค้นเส้นทาง…….... 81

บทที่ 5 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ……...91
สรุปผลการวิจัย…….. 91
อภิปรายผลการวิจัย…… 92
ข้อเสนอแนะ…….. 95

รายการอ้างอิง……...98

ภาคผนวก
ภาคผนวก ก รหัสโปรแกรมสำหรับตรวจสอบข้อมูลโครงข่ายในตารางความสัมพันธ์……102
ภาคผนวก ข รหัสโปรแกรมสำหรับคำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด.....108
ภาคผนวก ค ตารางการเดินรถและให้บริการรถไฟฟ้าบีทีเอส……117
ภาคผนวก ง ตารางเวลาให้บริการเรือด่วนเจ้าพระยา…….. 119
ภาคผนวก จ ผังแสดงเส้นทางและท่าเรือสำหรับบริการเรือด่วนเจ้าพระยา……120
ภาคผนวก ฉ รหัสโปรแกรมสำหรับสร้างตารางและนำเข้าข้อมูลสู่ตาราง
transport_node และ transport_station…… 121
ภาคผนวก ช รหัสโปรแกรมสำหรับรวมข้อมูล arc จากแต่ละชั้นข้อมูลลงในตาราง
transport_geom……... 126
ภาคผนวก ซ การปรับปรุงแก้ไขข้อมูลเชิงเรขาคณิต ตำแหน่งที่ตั้งสถานี (vertex) หรือลักษณะ
รูปร่างของเส้น (arc)…….. 129
ภาคผนวก ฌ การปรับปรุงแก้ไขข้อมูลความสัมพันธ์เชิงโครงข่าย…… 132

ประวัติผู้เขียนวิทยานิพนธ์…….. 134

บทที่ 1
บทนำ

ความเป็นมาและความสำคัญของปัญหา

การเดินทางไปยังสถานที่ต่างๆ ถือเป็นกิจวัตรที่สำคัญอย่างหนึ่งในชีวิตประจำวันของมนุษย์ ซึ่งในกรุงเทพมหานครมีระบบขนส่งมวลชนที่ให้บริการประชาชนอยู่หลายประเภทเช่น รถประจำทาง รถไฟฟ้า เรือโดยสาร เรือข้ามฟาก รถไฟ รถตู้มวลชน โดยทั่วไปมนุษย์ต้องการเดินทางไปยังจุดหมายปลายทางภายใต้ข้อกำหนดบางอย่าง เช่น ต้องการระยะเวลาการเดินทางที่น้อยที่สุด ต้องการเดินทางด้วยระยะทางที่สั้นที่สุด ต้องการเส้นทางหรือวิธีที่ใช้ค่าใช้จ่ายต่ำสุด เป็นต้น

เนื่องจากปัจจัยด้านเวลาเป็นปัจจัยที่มีความสำคัญมากสำหรับชีวิตที่เร่งรีบอย่างในเมืองใหญ่แบบกรุงเทพมหานคร และในบางกรณีถึงแม้ผู้เดินทางจะเลือกเส้นทางที่สั้นที่สุดในการเดินทางก็ตาม แต่อาจจะไม่ใช่เส้นทางที่ถึงจุดหมายเร็วที่สุดก็เป็นได้ ทั้งนี้ขึ้นกับปัจจัยประกอบอีกหลายปัจจัยเช่น ปริมาณการจราจร จำนวนสัญญาณไฟจราจร จำนวนทางแยกทางเลี้ยว เป็นต้น

การเดินทางโดยอาศัยระบบขนส่งมวลชนนั้นแตกต่างจากการเดินทางโดยรถยนต์ส่วนบุคคลหลายประการ เช่น ผู้เดินทางไม่สามารถเลือกเส้นทางโดยอิสระ จะต้องเลือกเส้นทางและเดินทางไปตามสายทางของระบบขนส่งที่มีอยู่เท่านั้น การเดินทางหนึ่งครั้งอาจต้องเกิดจากการเดินทางย่อยๆ ผสมรวมรูปแบบการเดินทางหลายรูปแบบ หรือเส้นทางที่สั้นที่สุดอาจมิได้ทำให้ผู้เดินทางไปถึงจุดหมายปลายทางเร็วที่สุด เนื่องจากประเภทของระบบขนส่งมวลชนที่เลือกมีผลต่อระยะเวลาที่ใช้เดินทางด้วย นอกเหนือจากปัจจัยด้านปริมาณการจราจรบนท้องถนนแล้ว

ในต่างประเทศ ได้มีการพยายามพัฒนาแบบจำลองและโครงสร้างข้อมูล พัฒนามาตรฐานตลอดจนระบบแนะนำเส้นทางทั้งสำหรับรถยนต์ส่วนตัวและรถโดยสารสาธารณะอยู่พอสมควร อย่างไรก็ตามแนวคิดและผลงานต่างๆ ได้พัฒนาขึ้นภายใต้บริบทของต่างประเทศ ในขณะที่ประเทศไทยยังมีงานวิจัยจำนวนน้อยที่มุ่งศึกษาและพัฒนาแบบจำลองข้อมูลเพื่อการแนะนำเส้นทางการเดินทางสำหรับระบบขนส่งมวลชนหลายรูปแบบที่ผสมผสานกัน (Multimodal)

จึงเป็นที่มาของแนวความคิดที่จะศึกษาพฤติกรรมของระบบขนส่งมวลชนในประเทศไทย เน้นไปที่พื้นที่กรุงเทพมหานคร ทำการออกแบบแบบจำลองและโครงสร้างข้อมูลสำหรับการจำลองพฤตติกรรม พร้อมทั้งทดลองพัฒนาโปรแกรมคอมพิวเตอร์ที่ช่วยการตัดสินใจเลือกเส้นทางที่ใช้ระยะเวลาเดินทางน้อยที่สุดโดยอาศัยระบบขนส่งมวลชนหลายรูปแบบผสมผสานกัน ทั้งนี้ได้กำหนดขอบเขตการศึกษาไปที่การเดินทางด้วยระบบขนส่งมวลชนเพียง 5 ประเภท คือ รถโดยสารประจำทาง เรือโดยสาร เรือข้ามฟาก รถไฟฟ้าบีทีเอส และรถไฟฟ้ามหานคร (รถไฟฟ้าใต้ดิน) ในพื้นที่กรุงเทพมหานคร

วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาพฤติกรรมของระบบขนส่งมวลชน 5 ประเภท คือ รถโดยสารประจำทาง เรือโดยสาร เรือข้ามฟาก รถไฟฟ้าบีทีเอส และรถไฟฟ้ามหานคร (รถไฟฟ้าใต้ดิน)
2. เพื่อศึกษา วิเคราะห์ และออกแบบแบบจำลองและโครงสร้างข้อมูลที่เหมาะสมกับพฤติกรรมของระบบขนส่งมวลชน 5 ประเภท ดังกล่าว โดยเน้นจัดเก็บข้อมูลที่จำเป็นต่อการคำนวณหรือสืบค้นในเชิงโครงข่ายการเดินทาง ในที่นี้คือเพื่อการค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดบนระบบขนส่งมวลชนผสมผสานกันหลายประเภท
3. เพื่อวิเคราะห์และประยุกต์ใช้ทฤษฎีทางด้านการวิเคราะห์โครงข่าย (Network Analysis) เข้ากับปัญหาซึ่งมีกรอบเงื่อนไขบางอย่างจากสภาพความเป็นจริง
4. ทดลองพัฒนาโปรแกรมสำหรับคำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดบนระบบขนส่งมวลชนหลายประเภท เพื่อทดสอบโครงสร้างข้อมูลที่ได้ออกแบบขึ้น และทดสอบการประยุกต์ใช้ทฤษฎีทางด้านการวิเคราะห์โครงข่าย

ขอบเขตของการวิจัย

ศึกษาและวิเคราะห์พฤติกรรมของระบบขนส่งมวลชน 5 ประเภท คือ รถโดยสารประจำทาง เรือโดยสาร เรือข้ามฟาก รถไฟฟ้าบีทีเอส และรถไฟฟ้ามหานคร (รถไฟฟ้าใต้ดิน) ในพื้นที่กรุงเทพมหานคร และอยู่ภายใต้สมมติฐานว่าระบบขนส่งมวลชนแต่ละประเภทเชื่อมโยงเข้าด้วยกันด้วยรูปแบบการเดินทางที่พื้นฐานที่สุด คือการเดิน

ออกแบบและนำเสนอแบบจำลองและโครงสร้างข้อมูลที่สามารถจัดเก็บและจำลองพฤติกรรมของระบบขนส่งมวลชนดังกล่าว ตลอดจนข้อมูลประกอบอื่นๆ ที่จำเป็นต่อการคำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดบนระบบขนส่งมวลชนหลายประเภทในลักษณะผสมผสานกัน เมื่อได้โครงสร้างข้อมูลที่เหมาะสมแล้วได้ทดลองพัฒนาโปรแกรมซึ่งทำงานบนเว็บเบราว์เซอร์และผ่านเครือข่ายอินเทอร์เน็ต สำหรับทำการสืบค้น คำนวณระยะเวลาและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด เพื่อทดสอบโครงสร้างข้อมูลที่ออกแบบ รวมทั้งทดลองประยุกต์ใช้ทฤษฎีทางด้านการวิเคราะห์โครงข่ายในการปฏิบัติจริง

ประโยชน์จากการวิจัย

องค์ความรู้และต้นแบบแบบจำลองและโครงสร้างข้อมูลที่สามารถจัดเก็บและจำลองพฤติกรรมของระบบขนส่งมวลชน 5 ประเภท รวมทั้งแนวทางและข้อเสนอแนะสำหรับการพัฒนาโปรแกรมคอมพิวเตอร์เพื่อการค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดบนระบบขนส่งมวลชนหลายประเภทในลักษณะผสมผสานกัน

ขั้นตอนดำเนินการวิจัย

1. ศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องดังนี้
    1.1 Graph network theory
    1.2 Dijkstra's algorithm
    1.3 Database design and implementation
    1.4 Internet programming and server side script application (PHP)
    1.5 Client side script application (JavaScript)
    1.6 Traffic flow theory
    1.7 งานวิจัยที่เกี่ยวข้องประเภทระบบสารเทศภูมิศาสตร์ที่นำมาประยุกต์ใช้ในด้านการแนะนำเส้นทางการเดินทาง
2. กำหนดขอบเขตการวิจัย และจุดประสงค์
3. รวบรวมข้อมูลที่จำเป็นจากหน่วยงานเจ้าของข้อมูลและออกเก็บข้อมูลภาคสนามเพิ่มเติม หน่วยงานที่เกี่ยวข้องได้แก่
    3.1 องค์การขนส่งมวลชนกรุงเทพ
    3.2 สำนักการจราจรและขนส่ง กรุงเทพมหานคร
    3.3 กรมการขนส่งทางน้ำและพาณิชยนาวี
    3.4 บริษัทรถไฟฟ้ากรุงเทพ จำกัด
    3.5 ศูนย์ข้อมูลกรุงเทพมหานคร
4. ศึกษาและวิเคราะห์พฤติกรรมของระบบขนส่งมวลชนทั้ง 5 ประเภท
5. ออกแบบแบบจำองข้อมูลเพื่อจัดเก็บและจำลองพฤติกรรมของระบบขนส่งมวลชนทั้ง 5 ประเภท ตลอดจนกิจกรรมที่เกิดขึ้นในการเดินทาง
6. ออกแบบขั้นตอนวิธีที่ใช้ประกอบการคำนวณหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด

6.1 ประมวลรายการตัวแปรต่างๆ ที่มีผลต่อการคำนวณหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด แล้วทำการคัดเลือกตัวแปรที่ต้องการและไม่ต้องการ

6.2 ออกแบบวิธีการคำนวณระยะเวลาในการเดินทางจากตัวแปรต่างๆ ที่มีอยู่

6.3 ออกแบบฐานข้อมูลสำหรับจัดเก็บข้อมูลโครงข่ายเส้นทางและข้อมูลอรรถาธิบายที่เกี่ยวข้องเพื่อใช้สนับสนุนการประมวลผล

7. นำข้อมูลเชิงพื้นที่ (เส้นทางให้บริการระบบขนส่งมวลชน) มาทำการปรับแก้และสร้างเป็นข้อมูลชุดใหม่ซึ่งมีนัยเชิงโครงข่าย จัดเก็บความสัมพันธ์ในแง่ของโครงข่าย (Network topology) โดยมิได้มองในเชิงรูปร่างทางเรขาคณิตเท่านั้น
8. จัดทำข้อมูลอรรถาธิบายที่จำเป็น เช่น ระยะเวลารอคอยพาหนะ อัตราเร็วพาหนะแต่ละประเภท อัตราเร็วในการเดินเพื่อเปลี่ยนรูปแบบการเดินทาง เป็นต้น
9. พัฒนาโปรแกรมประยุกต์สำหรับคำนวณหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด
   9.1 พัฒนาโปรแกรมประยุกต์ส่วนที่ทำงานบนเครื่องคอมพิวเตอร์แม่ข่าย
   9.2 พัฒนาส่วนการติดต่อกับฐานข้อมูล
   9.3 พัฒนาส่วนการคำนวณและประมวลผล
   9.4 พัฒนาส่วนการส่งผลลัพธ์ไปยังเครื่องคอมพิวเตอร์ลูกข่าย
   9.5 พัฒนาโปรแกรมประยุกต์ส่วนที่ทำงานบนเครื่องคอมพิวเตอร์ลูกข่าย
      9.5.1 พัฒนาส่วนการรับคำสั่งและโต้ตอบกับผู้ใช้
      9.5.2 พัฒนาส่วนการส่งข้อมูลจากผู้ใช้ไปยังเครื่องคอมพิวเตอร์แม่ข่าย
      9.5.3 พัฒนาส่วนการแสดงผลลัพธ์
10. ทดสอบและปรับปรุงโปรแกรมประยุกต์
11. ปรับปรุงการทำงานของโปรแกรมประยุกต์ให้ถูกต้อง
12. สรุปผลการวิจัยและเรียบเรียงวิทยานิพนธ์

ข้อจำกัดของการวิจัย

1. การจำลองกิจกรรมระหว่างเดินทางของผู้เดินทางให้ง่ายลง

เนื่องจากงานวิจัยเป็นการวิเคราะห์และออกแบบโครงสร้างข้อมูลเพื่อการคำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด ดังนั้นปัจจัยหรือตัวแปรต่างๆ ที่มีผลต่อระยะเวลาการเดินทางจึงต้องนำมาพิจารณา แต่ในการเดินทางจริงนั้นมีกิจกรรมและรายละเอียดมากมายที่ไม่อาจนำมาใช้หรือบรรจุลงในโครงสร้างข้อมูลทั้งหมดได้ เช่น ระยะเวลาที่ใช้ต่อแถวซื้อตั๋วโดยสาร ระยะเวลาจอดรอให้ผู้โดยสารขึ้นลงรถประจำทาง อัตราเร็วจริงของรถไฟฟ้าซึ่งอาจไม่คงที่ เป็นต้น

ดังนั้นตัวแปรบางอย่างจึงมิได้นำมาใส่ลงไปในโครงสร้างข้อมูล โดยพยายามเลือกเฉพาะตัวแปรที่มีขนาดของระยะเวลามาก หรือเป็นตัวชี้เฉพาะของการเดินทางในระบบขนส่งมวลชนนั้นๆ หรือมีการบันทึกไว้ชัดเจน หรือสามารถกะประมาณคำนวณได้ เป็นต้น

2. การจำลองพฤติกรรมของระบบขนส่งมวลชนให้ง่ายลง

ระบบขนส่งมวลชนมีรายละเอียดมาก อีกทั้งมีหลายๆ ตัวแปรที่ในความเป็นจริงนั้นสามารถแปรเปลี่ยนได้ไม่คงที่ นอกจากนี้ลักษณะการสอบถามเส้นทางในการเดินทางนั้น จะเป็นการถามถึงเหตุการณ์ที่เป็นปัจจุบันหรืออนาคตเท่านั้น เช่น ต้องการสอบถามเส้นทางสำหรับการเดินทางในขณะนั้น ต้องการทราบเส้นทางการเดินทางสำหรับในอีก 2 ชั่วโมงข้างหน้า หรือในวันรุ่งขึ้นเป็นต้น ไม่มีโอกาสที่จะมีลักษณะคำถามที่เป็นอดีต

เนื่องจากปริมาณข้อมูลมีมาก อีกทั้งความหลากหลายของข้อมูล ผู้วิจัยจึงไม่สามารถรวบรวมและนำเข้าข้อมูลที่เกิดขึ้นตามเวลาจริงทั้งหมด และกรอบของงานวิจัยนี้ไม่สามารถครอบคลุมไปถึงการสร้างกระบวนการเพื่อใช้สำหรับทำนายค่าในอนาคตได้ อีกทั้งกระบวนการในเก็บบันทึกและวิเคราะห์เพื่อให้ได้มาซึ่งข้อมูลที่ถูกต้องตลอดจนกระบวนการทำนายค่าตัวแปรในอนาคตนั้นเป็นงานที่อยู่นอกเหนือจากขอบเขตของงานวิจัยนี้

ดังนั้นสำหรับค่าตัวแปรต่างๆ ที่มีลักษณะพฤติกรรมเชิงพลวัต เช่น อัตราเร็วของรถโดยสารประจำทางซึ่งขึ้นกับปริมาณการจราจรบนท้องถนน ตารางเวลาให้บริการของสายรถประจำทางแต่ละสายที่แตกต่างกัน กรณีรถโดยสารประจำทางบางสายอาจไม่รับส่งผู้โดยสารทุกป้าย อัตราการปล่อยรถโดยสารหรือเรือที่ไม่แน่นอน เป็นต้น ในงานวิจัยนี้จึงได้กำหนดให้ค่าตัวแปรบางตัวมีค่าคงที่ (ค่าโดยปริยายเบื้องต้น) หรือตัดทอนพฤติกรรมบางอย่างออกไป เช่น กำหนดสมมติฐานว่ารถโดยสารประจำทางทุกสายจะเข้าจอดรับส่งผู้โดยสารทุกป้าย หรือกำหนดสมมติฐานว่าตารางเวลาการให้บริการของรถโดยสารประจำทางทุกสายเหมือนกัน ซึ่งจะได้กล่าวรายละเอียดในบทที่ 3 ของวิทยานิพนธ์

3. ความจำเป็นของข้อมูลเส้นทางระบบขนส่งมวลชนซึ่งมีนัยเชิงโครงข่าย จัดเก็บความสัมพันธ์ในแง่ของโครงข่าย โดยมิได้มองในเชิงรูปร่างทางเรขาคณิตเท่านั้น

การจัดทำข้อมูลแผนที่เส้นทางคมนาคมและระบบขนส่งมวลชนนั้น เท่าที่หาได้และมีการเผยแพร่สู่สาธารณะ ยังไม่มีการเผยแพร่ข้อมูลที่มีโครงสร้างในลักษณะเพื่อการวิเคราะห์เชิงโครงข่าย หรือเพื่อการสืบค้นเส้นทางการเดินทางที่ใช้เวลาน้อยที่สุดโดยเฉพาะ

ดังนั้นในงานวิจัยนี้ ผู้วิจัยจึงต้องจัดทำข้อมูลดังกล่าวขึ้นมาเอง โดยอาศัยข้อมูลเส้นทางคมนาคมและระบบขนส่งมวลชนเดิมที่มีเผยแพร่อยู่เป็นฐาน แล้วจึงทำการปรับแก้ หรือจัดทำเพิ่มในบางส่วน เพื่อให้สอดคล้องกับความต้องการของงาน

อย่างไรก็ตามข้อมูลที่ปรับแก้และจัดทำเพิ่มนั้นเป็นเพียงส่วนหนึ่งของข้อมูลทั้งหมดสำหรับระบบเต็มรูปแบบเท่านั้น ทั้งนี้โดยมีจุดประสงค์เพื่อใช้ประกอบการวิจัย และไม่ควรนำไปใช้อ้างอิงใดๆ เนื่องจากอาจมีความคลาดเคลื่อนทั้งในเชิงคุณภาพและปริมาณ เช่น หมายเลขสายรถประจำทางที่ผ่านแต่ละป้าย ระยะทางระหว่างแต่ละสถานีหรือป้าย จำนวนป้ายหรือท่าเรือ หรือความความถูกต้องเชิงตำแหน่ง เป็นต้น

<u>คำจำกัดความสำหรับศัพท์เฉพาะที่ใช้ในวิทยานิพนธ์ (อภิธานศัพท์)</u>

Arc

ดู Link

Cost (ISO 19133, 2005)

ค่าใช้จ่ายในการเดินทาง หน่วยของ cost อาจจะเป็นในรูปของเงิน เวลา ระยะทาง หรือในแง่อื่นๆ

Cost function (ISO 19133, 2005)

ฟังก์ชั่นทางคณิตศาสตร์ซึ่งทำหน้าที่ประเมินหรือคำนวณค่าใช้จ่ายในการเดินทางที่เกิดขึ้นในเส้นทางการเดินทาง

Data Model หรือแบบจำลองข้อมูล (Illingworth และ Pyle, 1996)

เป็นการพยายามสร้างตัวแทนเชิงนามธรรมจากสถานการณ์บางอย่างที่อยู่ในโลกของความเป็นจริง หรือจากขอบเขตความสนใจเกี่ยวกับข้อสนเทศที่ต้องการนำไปจัดเก็บในฐานข้อมูล

Headway

ระยะทางระหว่างยานพาหนะสองคัน ที่ถูกปล่อยออกจากท่ารถ หรือเรือ ซึ่งมักใช้กับกรณีรถขนส่งสาธารณะ

Junction (ISO 19133, 2005)

Node โดดเดี่ยวเชิง topology ใน network โดยจะมีคุณสมบัติที่มาพร้อมกันด้วยคือ turns link ที่มุ่งเข้าสู่ และมุ่งออกจาก junction

Link (ISO 19133, 2005)
การเชื่อมโยงเชิง topology แบบมีทิศทางเฉพาะระหว่างสอง nodes (junctions)

Dwell time (ผู้วิจัย)
ระยะเวลาที่ยานพาหนะใช้ในการจอดรอให้ผู้โดยสารเคลื่อนย้ายเข้าสู่หรือออกจากยานพาหนะที่ป้าย สถานี หรือท่าเรือ

Network (ISO 19133, 2005)
โครงสร้างเชิงนามธรรมซึ่งประกอบขึ้นจากเซ็ตของวัตถุที่ไม่มีมิติ (0-dimensional) ที่เรียกว่าทางแยก (junction) และจากเซ็ตของวัถตุหนึ่งมิติ (1-dimensional) ที่เรียกว่า links ซึ่งทำหน้าที่เชื่อมทางแยกเข้าด้วยกัน แต่ละ link จะระบุถึงทางแยกที่เป็นจุดเริ่มและทางแยกที่เป็นจุดสิ้นสุดของ link

Node
ดู Junction

Private transportation (ระบบขนส่งส่วนตัว) (ผู้วิจัย)
เป็นระบบที่ผู้เดินทาง (ผู้ขับขี่) ทางสามารถเลือกได้โดยอิสระว่าต้องการเลือกใช้เส้นทางใด หรือเดินทาง ณ เวลาใด และมักจะเดินทางโดยยานพาหนะซึ่งสงวนสิทธิเฉพาะบุคคล (ในช่วงเวลาที่เดินทาง)

Public transportation (ระบบขนส่งมวลชน) (ผู้วิจัย)
เป็นระบบซึ่งมีการกำหนดเส้นทางการเดินทางหรือเส้นทางให้บริการไว้แน่นอน และใช้ยานพาหนะที่สามารถให้บริการต่อคนจำนวนมากได้ในคราวเดียว ผู้เดินทางหรือผู้ขับขี่ยานพาหนะจะต้องเดินทางไปตามเส้นทางที่กำหนดและประกาศไว้ต่อสาธารณะเท่านั้น

Routing (ISO 19133, 2005)
การสืบค้นเส้นทางซึ่งเหมาะสมที่สุด (optimal) โดยเส้นทางนั้นจะต้องมีค่า cost ต่ำสุด

Turn (ISO 19133, 2005)
ส่วนหนึ่งของเส้นทาง (route) หรือ network ซึ่งประกอบด้วย junction และ link ที่พุ่งเข้าหรือพุ่งออกจาก junction นั้น

Waiting time (ISO 19133, 2005)
ระยะเวลาการเดินทางที่เพิ่มขึ้นเนื่องจากการรอคอยหรือหยุดรอ ณ จุดต่างๆ ใน Netowrk
หมายเหตุ สำหรับในวิทยานิพนธ์นี้ Waiting time เกิดจากผลรวมของระยะเวลาที่ยานพาหนะใช้เคลื่อนที่ผ่าน headway distance รวมกับระยะเวลา dwell time ที่เกิดขึ้นตามจุดสถานีจอดยานพาหนะในช่วง headway นั้น (ผู้วิจัย)

จุดเริ่มต้นการเดินทาง Origin (Miller และ Shaw, 2001: 57)
ตำแหน่งที่เป็นจุดเริ่มเกิดการเดินทาง

จุดสิ้นสุดการเดินทาง Destination (Miller และ Shaw, 2001: 57)
ตำแหน่งที่เป็นจุดสิ้นสุดการเดินทาง

ข้อมูลความสัมพันธ์เชิงโครงข่าย (ผู้วิจัย)
ข้อมูลซึ่งเกิดจากการแปลงข้อมูลเชิงเรขาคณิตของเครือข่ายเส้นทางคมนาคม เช่น ตำแหน่งป้ายหยุดรถโดยสารประจำทาง สถานีรถไฟฟ้า เส้นทางรถไฟฟ้า ถนน ให้อยู่ที่รูปแบบของ node และ arc ทั้งนี้เพื่อบ่งบอกความเชื่อมโยงกันของ node แต่ละ node ในโครงข่าย (Network)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
แนวคิด เอกสารและงานวิจัยที่เกี่ยวข้อง

แนวคิดและทฤษฎี

ทฤษฎี Graph (Miller และ Shaw, 2001: 54)

เป็นทฤษฎีที่ว่าด้วยเรื่องของความสัมพันธ์หรือความเชื่อมต่อกันทางกายภาพของสมาชิกต่างๆ ภายในเซ็ตๆ หนึ่ง คุณสมบัติของ Graph มีดังนี้

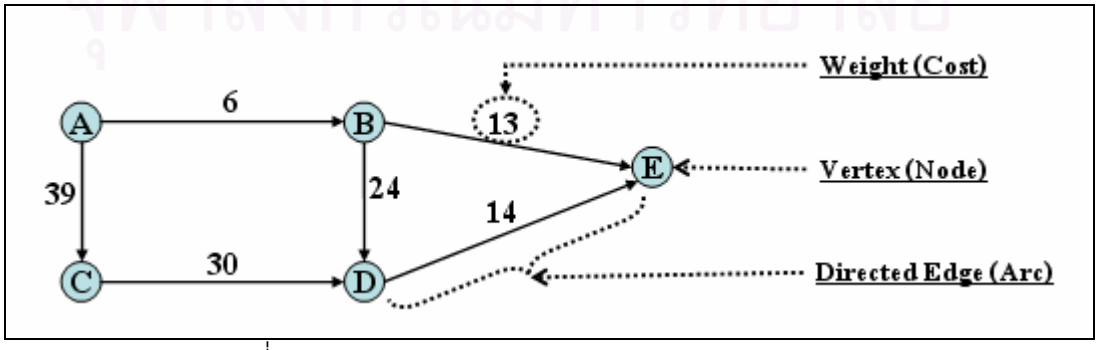
Graph ประกอบด้วยจุดที่เรียกว่า Vertices ซึ่งจะแทนสมาชิกใดๆ ในเซ็ตและจะมีชื่อสำหรับเรียกอ้างอิงสมาชิกแต่ละตัวได้โดยชื่อเหล่านี้จะประจำกับแต่ละ Vertex และมีความเป็นเอกลักษณ์

Edges (Miller และ Shaw, 2001: 54) จะใช้แสดงความสัมพันธ์เชิงตรรก (Logical relation) ของความเชื่อมต่อกันทางกายภาพระหว่าง Vertices เราสามารถเขียนสัญลักษณ์แทน Edges ได้โดยการระบุถึง Vertices 2 ตัว ที่ Edge นั้นๆ ทำหน้าที่เชื่อมโยงอยู่ เช่น {a,b} ทั้งนี้ a และ b เป็นชื่อเรียกของ vertices ทั้งสอง

Edges สามารถเป็นลักษณะการเชื่อมต่อแบบทางเดียวก็ได้ (one way or directed) ซึ่งหมายความว่าความสัมพันธ์ระหว่างสอง Vertices นั้น เป็นไปในทิศทางตามที่ Edges ระบุ หากไม่ได้ระบุจะถือว่าไม่มีการบังคับทิศทาง

Planar Graph (Miller และ Shaw, 2001: 54) เป็น Graph ที่มีคุณสมบัติเฉพาะคือถือว่าวางตัวอยู่บน Euclidian Space กล่าวคือถ้ามีสอง edges ใดตัดกันจะต้องมี vertex เกิดขึ้นที่จุดนั้นเสมอ หรืออีกนัยหนึ่งคือ edges สอง edges จะลอดใต้หรือข้ามกันไม่ได้

Graph จะพิจารณาถึงความสัมพันธ์ว่าสมาชิกแต่ละตัวใน Graph นั้นเชื่อมต่อกันกับสมาชิกอื่นใดบ้างหรือไม่เท่านั้น มิได้สนใจถึงรูปร่างทางเรขาคณิตของ Graph หรือตำแหน่งของสมาชิกแต่ละตัว (Worboys, 1997:135) ตัวอย่างพร้อมส่วนประกอบของกราฟแสดงอยู่รูปที่ 2-1



รูปที่ 2-1 ตัวอย่างของกราฟและส่วนประกอบต่างๆ ในกราฟ

Network (Miller และ Shaw, 2001: 54) คือ Graph ที่ใช้แสดงถึงพฤติกรรมเกี่ยวกับการเคลื่อนที่ต่างๆ Network นั้นจะประกอบด้วย node กับ arc โดย node จะใช้แทนจุดต่างๆ ที่เป็นจุดเริ่มต้น สิ้นสุด หรือหน่วงของการเคลื่อนตัวที่เกิดใน Network ในขณะที่ arc จะเปรียบเสมือนท่อหรือเส้นทางการไหลหรือเคลื่อนที่ที่เกิดขึ้นระหว่างสอง node ตัวอย่างของ node และ arc เช่น การเดินทางจากสถานีรถไฟหนึ่ง (node ที่หนึ่ง) ผ่านรางรถไฟ (arc ที่เชื่อมต่อระหว่าง node) ไปยังอีกสถานีหนึ่ง (node ที่สอง) หากมีการกำหนดทิศทางการเคลื่อนที่ใน arc จะระบุทิศทางดังกล่าวโดยใช้ลำดับของ node เป็นตัวกำหนดทิศทาง

ความแตกต่างสำคัญอีกประการระหว่าง Network และ Graph คือ Network สามารถกำหนดน้ำหนักให้กับแต่ละ arc ได้ ซึ่งโดยทั่วไปน้ำหนักดังกล่าวมักจะพิจารณาแปรผันกับค่าใช้จ่ายหรือความสามารถในการเคลื่อนที่ผ่าน arc นั้นๆ

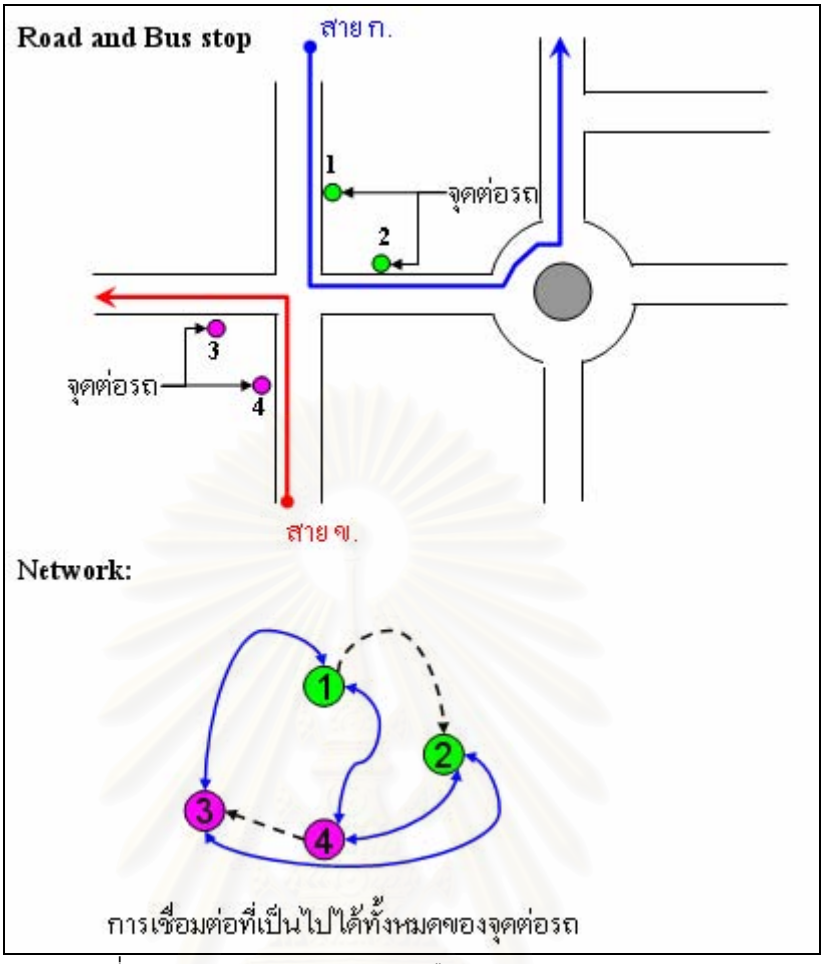
<u>แนวคิดการแทนระบบการขนส่ง (Transportation system) ด้วย Network</u> (Sheffi, 1985: 10-18)

การแทนโครงข่ายการขนส่งด้วยแบบจำลอง Planar network หรือบางครั้งเรียกว่า Node-arc representation เป็นการพยายามใช้ directed networks (ซึ่งประกอบไปด้วย arc ที่กำหนดทิศทางการเคลื่อนที่) แทนหรืออธิบายทิศทางการคลื่อนที่ไปบนโครงข่ายคมนาคมขนส่ง เช่น ถนนเดินรถทางเดียว หรือกำหนดคุณสมบัติระยะเวลาเดินทางที่แตกต่างกันแปรเปลี่ยนไปตามช่วงเวลาในแต่ละวัน

Node-arc representation เป็นการสร้าง Network ขึ้นเพื่อเป็นตัวแทนระบบโครงข่ายการขนส่งโดยใช้พื้นฐานทางคณิตศาตร์ โดย Network อาจประกอบด้วย โครงข่ายการขนส่งย่อยหลายโครงข่ายรวมกันอยู่ เช่น private transportation (เช่น โครงข่ายถนน) และ public transportation (เช่น โครงข่ายรถโดยสารประจำทาง รถไฟฟ้า) และมีTransfer arc จะทำหน้าที่เชื่อมโยงโครงข่ายย่อยเข้าด้วยกัน

ในกรณี private transportation นั้น node ใน network มักจะเป็นตัวแทนของทางแยก และ arc เป็นตัวแทนของช่วงถนนที่เชื่อมระหว่างแยก หากมีการเดินรถแบบ two-way ก็จะสร้าง directed arc สอง arc แทนการเดินทางทั้งสองทิศทาง มีการกำหนดฟังก์ชั่นได้หลายๆ รูปแบบเพื่อใช้คำนวณระยะเวลาหรือค่าใช้จ่ายในการเดินทาง

ในกรณี public transportation นั้น node ใน network มักจะเป็นตัวแทนของป้ายหรือสถานีของระบบขนส่งต่างๆ ในขณะที่ arc ใช้แทนเส้นทางซึ่งเชื่อมระหว่างป้ายหรือสถานีของระบบขนส่ง นอกจากนี้ยังอาจจะมีการสร้าง arc พิเศษ เพื่อเชื่อมการเดินทางระหว่างจุดเริ่มต้นเดินทางเข้าสู่ระบบขนส่งมวลชน และออกจากระบบขนส่งมวลชน



รูปที่ 2-2 การแปลงจากระบบเครือข่ายถนนมาเป็น Network

ตัวอย่างการแปลงจากเครือข่ายถนนมาเป็น Network แสดงอยู่ในรูปที่ 2-2 จุดต่อรถถูกแปลงเป็น node และเส้นทางเชื่อมโยงระหว่างจุดต่อรถถูกแปลงเป็น arc

<u>แนวคิดที่ใช้ในการบันทึกจัดเก็บข้อมูลสำหรับการประมวลผล</u> ซึ่งใช้ในงานวิจัยนี้ประกอบด้วย 3 ส่วนคือ

1. ส่วนของข้อมูลเชิงพื้นที่ที่เป็นเส้นทางคมนาคมและตำแหน่งสำคัญในโครงข่าย เช่น ถนน ทางรถไฟฟ้า เส้นทางจราจรทางน้ำ สถานีรถไฟฟ้า ท่าเรือ ตำแหน่งสถานที่สำคัญ เป็นต้น ซึ่งจะสนใจคุณสมบัติทางด้าน Topology เป็นหลัก ข้อมูลดังกล่าวจะถูกจำลองและจัดเก็บในรูปแบบของ Network ซึ่งมีรายละเอียดของทฤษฎีดังกล่าวในหัวข้อทฤษฎีกราฟ (Graph) เพื่อใช้สำหรับการคำนวณและวิเคราะห์เชิงโครงข่าย
2. ส่วนข้อมูลพิกัดเส้นทางเพื่อแสดงรูปร่างในเชิงเรขาคณิต เช่น ค่าพิกัดของเส้นทางถนน จุดเริ่มต้นและสิ้นสุดของเส้นทาง หรือตำแหน่งสถานีขนส่งมวลชน จะจัดเก็บแยกต่างหาก และใช้เพื่อการแสดงผลในเชิงแผนที่ หรือเพื่อการแนะนำเส้นทาง

3. ส่วนของข้อมูลอรรถาธิบาย (คุณสมบัติประกอบอื่นๆ ที่เกี่ยวเนื่อง) ซึ่งบ่งบอกข้อมูลที่จำเป็นในการคำนวณรวมถึงกฎเกณฑ์ต่างๆ เช่น อัตราเร็วเฉลี่ยของพาหนะในระบบขนส่งมวลชนแต่ละประเภท เวลาที่ใช้ในการเปลี่ยนรูปแบบการเดินทาง อัตราการมาถึงของยานพาหนะในระบบส่งมวลชนแต่ละประเภท หมายเลขรถโดยสารประจำทางที่ผ่านแต่ละเส้นทาง เป็นต้น ข้อมูลเหล่านี้จะถูกเก็บในลักษณะแบบจำลองข้อมูลเชิงสัมพันธ์ (Relational data model) โดยจัดเก็บเป็นสดมภ์ในลักษณะตารางข้อมูลโครงข่าย (Miller, 2001: 59-60)

ตัวอย่างของแบบจำลองข้อมูลเชิงสัมพันธ์ (Relational data model) ของทั้งสามส่วน แสดงใน รูปที่ 2-3 โดยตารางด้านบนซ้าย เป็นตารางซึ่งใช้ระบุความสัมพันธ์เชิงโครงข่ายหรือความเชื่อมต่อกันระหว่างโหนดต่างๆ ในโครงข่าย กำหนดทิศทางโดยการระบุ from node และ to node หมายถึงสามารถเดินทางผ่าน arc ได้จาก “from node” ไปยัง “to node” นอกจากนี้ยังจัดเก็บค่าคุณสมบัติต่างๆ ของแต่ละ arc ไว้ที่สดมภ์ถัดๆ ไป คุณสมบัติเหล่านี้สามารถกำหนดได้ว่าจะจัดเก็บคุณสมบัติใดบ้าง โดยจะแตกต่างกันไป ขึ้นกับความต้องการของผู้ออกแบบ (Miller, 2001: 59-60)

Arc id	From node	To node	No. of Lane	Type	Speed limit
55	100	102	3	highway	60
67	108	112	4	highway	60
78	120	104	4	expressway	90
80	132	124	6	expressway	90

Node id	Coordinate
100	x y
102	…
104	…
108	…
112	…
120	…
124	…
132	…

Arc id	Coordinate
55	x1 y1, x2 y2, x3 y3, x4 y4, …
67	…
78	…
80	…

รูปที่ 2-3 แบบจำลองข้อมูลเชิงสัมพันธ์ซึ่งใช้แทนข้อมูลโครงข่าย

ตารางซ้ายมือด้านล่างจัดเก็บข้อมูลเชิงเรขาคณิตของเส้น arc โดยจัดเก็บเป็นพิกัดจากจุดเริ่มต้นเรียงไปตามลำดับจนถึงจุดสุดท้าย และมี arc id เป็น primary key และยังเป็น key อ้างอิงเชื่อมโยงไปยังตารางอื่นๆ ด้วย ส่วนตารางด้านขวามือ เป็นค่าพิกัดของ node ในโครงข่าย มี Node id เป็น primary key

แนวคิดการทำงานของคอมพิวเตอร์ Client -Server Architectures

Client-Server Architectures (Dennis A., 2002: 36) เป็นการทำงานรูปแบบหลักอย่างหนึ่งของเครือข่ายอินเทอร์เน็ต โดยฝั่ง เครื่องลูกข่าย จะทำหน้าที่จัดการด้านการแสดงผล สื่อสารกับผู้ใช้และ/หรือประมวลผลข้อมูลบางส่วนหรือทั้งหมด ส่วนฝั่งเครื่องแม่ข่าย จะทำหน้าที่จัดเก็บข้อมูลและให้บริการการเข้าถึงข้อมูลแก่ เครื่องลูกข่าย และ/หรือประมวลผลข้อมูลเหล่านั้นแล้วส่งผลลัพธ์ให้แก่ เครื่องลูกข่าย

ระบบคอมพิวเตอร์ที่พัฒนาขึ้นภายใต้แนวคิดแบบนี้มีข้อดีคือระบบสามารถให้บริการแก่ผู้ใช้หลายคนพร้อมๆ กันได้ การบำรุงรักษาและปรับปรุงระบบทำได้ง่าย นอกจากนี้ยังมีความสอดคล้องตรงกันของข้อมูลกลางที่ใช้ร่วมกัน เนื่องจากข้อมูลกลางถูกจัดเก็บที่เครื่องแม่ข่ายเพียงแห่งเดียว

การพัฒนาโปรแกรมประยุกต์เพื่อใช้งานใน Client-Server Architectures นั้น ผู้พัฒนาจะต้องพัฒนาโปรแกรมประยุกต์ขึ้น 2 ชุด คือชุดที่ติดตั้งและทำงานที่เครื่องคอมพิวเตอร์ฝั่งเครื่องแม่ข่าย กับชุดที่ติดตั้งและทำงานที่เครื่องคอมพิวเตอร์ฝั่งเครื่องลูกข่ายโปรแกรมประยุกต์ทั้ง 2 ชุด จะทำหน้าที่แตกต่างกันดังกล่าวแล้ว

ตัวแปลภาษาที่ทำหน้าที่แปลภาษาให้สามารถทำงานบนเครื่องคอมพิวเตอร์ภายใต้ Client-Server Architectures นั้นมีอยู่มากมาย ทั้งนี้สามารถแบ่งออกเป็น 2 กลุ่มใหญ่ๆ คือ โปรแกรมเชิงพาณิชย์และโปรแกรมแบบรหัสเปิด ซึ่งในงานวิจัยนี้จะเลือกใช้โปรแกรมชนิดรหัสเปิด เนื่องจากมีผู้วิจัยมีความคุ้นเคยมาก่อนและมีความสามารถต่างๆ ครบถ้วนตามที่ผู้วิจัยต้องการ

ในกรณีของฝั่งเครื่องแม่ข่าย นั้นผู้วิจัยได้เลือกใช้ตัวแปลภาษาและไวยากรณ์ของภาษาชื่อ PHP ส่วนในกรณีฝั่งเครื่องลูกข่ายนั้น ผู้วิจัยเลือกใช้ JavaScript ซึ่งเป็นไวยากรณ์ของภาษาหนึ่งที่ทำงานภายใต้สภาวะแวดล้อมของโปรแกรม web browser

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

<u>ทฤษฎีเกี่ยวกับการคำนวณหาเส้นทางการเดินทางที่สั้นที่สุด (Shortest path)</u>

ขั้นตอนวิธีส่วนที่ใช้คำนวณหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด จะใช้ขั้นตอนวิธีที่เรียกว่า Dijkstra's Algorithm (Weiss, 1994) ซึ่งเป็นขั้นตอนวิธีที่เป็นที่รู้จักดีสำหรับการคำนวณหาเส้นทางที่มีค่าใช้จ่ายน้อยที่สุด ทั้งนี้จะต้องมีการกำหนดค่าใช้จ่ายสำหรับ arc ต่างๆ ในโครงข่ายเสียก่อน โดยค่าใช้จ่ายดังกล่าวมิได้หมายความว่าจะต้องอยู่ในรูปตัวเงินเสมอไปอาจเป็นในรูปของระยะทาง หรือระยะเวลาที่ใช้ในการเดินทางก็ได้แล้วแต่การกำหนดความหมาย ในงานวิจัยนี้ค่าใช้จ่ายดังกล่าว (ระยะเวลาที่ใช้ในการเดินทาง) ได้พิจารณาจากผลของตัวแปรหลายสามตัวร่วมกันคือ อัตราเร็วของยานพาหนะ ระยะทาง และระยะเวลาที่ใช้ในการเปลี่ยนรูปแบบการเดินทาง

Dijkstra's algorithm เป็นขั้นตอนวิธีที่ใช้สำหรับคำนวณหาเส้นทางที่สั้นที่สุด (โดยพิจารณาจากน้ำหนักที่กำหนดให้กับ arc) ระหว่างจุดสองจุดใน Network (Worboys, 1997: 235) โดยสามารถอธิบายได้ดังนี้

Dijkstra's algorithm (Cormen, Leiserson และ Rivest, 1990) จะใช้ได้กับ Planar Graph ที่มีการกำหนดทิศทางของ arc และน้ำหนักของเส้นทุกเส้นต้องไม่น้อยกว่าศูนย์ นั่นคือ $w(u,v) > 0$ ทุกๆ (u,v) เป็นสมาชิกของ Graph E(G) หลักการคิดวิเคราะห์ มีดังนี้

ให้ S เป็นเซ็ตของจุด vertex โดยมีค่าเริ่มต้นเป็นเซ็ตว่าง

ให้ d (v) เป็นค่าของระยะทางจากจุดเริ่มต้นถึงจุดปลายทาง v

ให้ Q เป็นเซ็ตของจุด vertex ที่ยังไม่เข้าวนซ้ำ (loop) โดยมีค่าเริ่มต้นเป็น V(G)

การเลือกจุด u ที่จะเข้าวนซ้ำ จะเลือกจากจุดที่อยู่ใน Q ซึ่งมีค่า d(u) ต่ำที่สุด และเมื่อเลือกแล้วจะลบจุดนี้ออกจาก Q นำไปใส่ใน S แทน

ตรวจสอบทุกจุด v ที่มีเส้นลากจาก u ไปถึงจุด v ว่าถ้า $d(v) > d(u) + w(u,v)$ แล้วจะต้องเปลี่ยน $d(v) = d(u) + w(u,v)$ และแก้ตัวชี้ว่าจุด vertex v ต้องมาจาก u

วนซ้ำจนกระทั่ง Q เป็นเซ็ตว่าง จะได้เส้นทางที่สั้นที่สุดตามที่ต้องการ

ขั้นตอนวิธีของ Dijkstra สามารถเขียนในรูปแบบรหัสเทียม (pseudo code) ได้ดังนี้

1. Initialize-Graph (G, S)

```
S←ϕ
Q←V(G)
While Q≠ϕ
        Do u←Extract-Min(Q)
        S←S U u
        For each vertex v ∈ Adj(u)
```

Do Relax (u, v, w)

2. Initialize-Single-Source (G, S)

For each vertex v∈ V(G)

Do d(v)←∞

π(v) ←Null

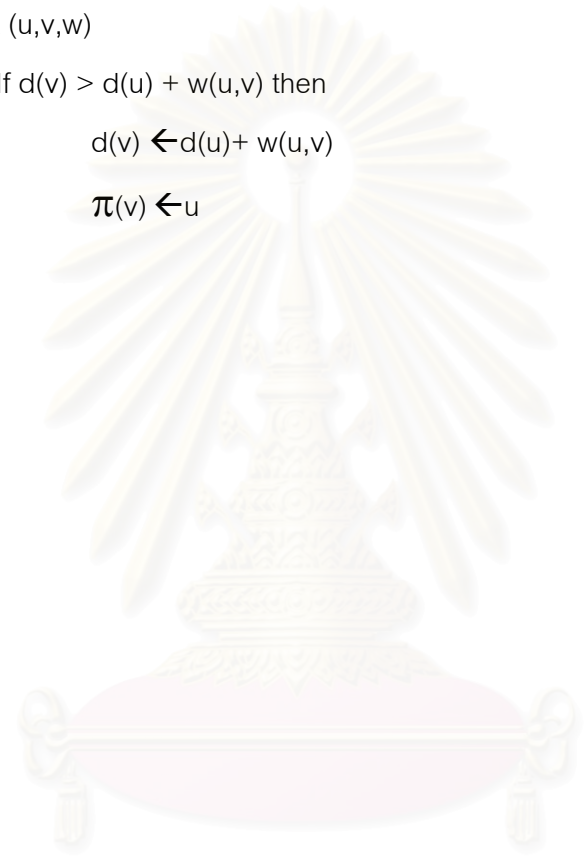
d(e) ←0

3. Relax (u,v,w)

If d(v) > d(u) + w(u,v) then

d(v) ←d(u)+ w(u,v)

π(v) ←u



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เอกสารและงานวิจัยที่เกี่ยวข้อง

1. โปรแกรมสำเร็จรูปสำหรับแนะนำเส้นทางเดินรถโดยสารประจำทาง (วศิน สินธุภิญโญ, 2538)

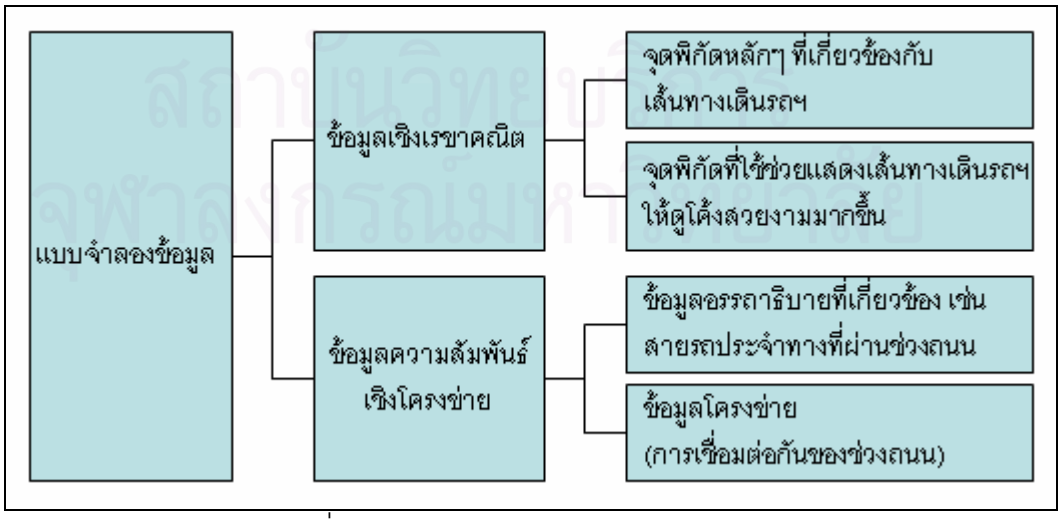
งานวิจัยนี้ กล่าวถึงการพัฒนาโปรแกรมสำเร็จรูปสำหรับแนะนำเส้นทางเดินรถโดยสารประจำทางให้กับผู้ใช้รถโดยสารประจำทาง เงื่อนไขที่ผู้ใช้สามารถเลือกได้คือ เส้นทางที่สั้นที่สุด หรือเส้นทางที่ต่อรถน้อยครั้งที่สุด และต้องการเน้นรถโดยสารประจำทางปรับอากาศหรือรถธรรมดา อีกทั้งสามารถระบุตำแหน่งที่ต้องการผ่านและไม่ต้องการผ่านได้ด้วย

ในการเลือกตำแหน่งต้นทาง ตำแหน่งปลายทาง ตำแหน่งที่ต้องการผ่านและตำแหน่งที่ไม่ต้องการผ่านสามารถเลือกได้โดยตรงจากแผนที่หรือเลือกโดยชื่อสถานที่ ซึ่งสถานที่แบ่งเป็น 5 กลุ่มคือ สถานที่สำคัญ โรงแรม โรงพยาบาล สถานีตำรวจและห้างสรรพสินค้า

นอกจากนี้ในโปรแกรมสำเร็จรูปยังเอื้ออำนวยให้ผู้มีอำนาจ สามารถนำเข้าข้อมูลใหม่ตลอดจนบำรุงรักษาข้อมูลที่เก็บอยู่ได้ ทั้งโดยการเพิ่มเติม ลบ หรือการเคลื่อนย้ายจุดหรือเส้นต่างๆ ที่ประกอบกันเป็นแผนที่ การเพิ่มเติมหรือการเคลื่อนย้ายตำแหน่งสถานที่ต่างๆ เปลี่ยนภาพสัญลักษณ์สถานที่สำคัญและในการทำงานยังสามารถเลือกเฉพาะส่วนของแผนที่ที่อยู่ในความสนใจมาขยาย เพื่อดูแผนที่ที่มีความละเอียดยิ่งขึ้น ทำให้ทำงานได้ถูกต้องแม่นยำยิ่งขึ้น

<u>แนวคิดเกี่ยวกับแบบจำลองข้อมูลที่ใช้</u>

วศิน สินธุภิญโญ (2538) ได้ทำการสร้างแบบจำลองข้อมูลเพื่อจัดเก็บข้อมูลรถโดยสารประจำทาง และความสัมพันธ์เชิงโครงข่ายของเส้นทางเดินรถไว้โดย แบบจำลองข้อมูลแยกออกเป็นสองกลุ่มใหญ่ๆ คือ ส่วนที่จัดเก็บข้อมูลเชิงเราขาคณิต (พิกัดของเส้นทางเดินรถ) และส่วนที่จัดเก็บความสัมพันธ์เชิงโครงข่าย (node และ arc) ดังแสดงในรูปที่ 2-4



รูปที่ 2-4 แบบจำลองข้อมูลแยกเป็นประเภท

ส่วนของข้อมูลเชิงเรขาคณิตนั้นแยกย่อยเป็นเป็นสองกลุ่ม คือกลุ่มที่เป็นจุดพิกัดหลักๆ ที่เกี่ยวข้องกับเส้นทางเดินรถฯ และกลุ่มจุดพิกัดที่ใช้ช่วยแสดงเส้นทางเดินรถฯ ให้ดูโค้งสวยงามมากขึ้น จุดพิกัดที่เกี่ยวข้องกับเส้นทางเดินรถประกอบด้วยสองประเภทคือ จุดที่เป็นตำแหน่งเชื่อมช่วงถนน ได้แก่ทางแยกต่างๆ และจุดที่เป็นตำแหน่งของสถานีต้นทางและปลายทาง

ส่วนของความสัมพันธ์เชิงโครงข่ายนั้นดังแสดงในรูปที่ 2-5 แยกเป็นส่วนข้อมูลของโครงข่ายและข้อมูลอรรถาธิบายที่เกี่ยวข้อง ข้อมูลของโครงข่ายประกอบด้วย จุดต้นช่วงถนน จุดปลายช่วงถนน ประเภทถนน ส่วนข้อมูลระยะทางระหว่างช่วงถนน จะใช้วิธีการคำนวณจากจุดพิกัดที่บันทึกไว้ และจะกระทำทุกครั้งเมื่อมีการรันโปรแกรม และสายรถโดยสารประจำทางที่ผ่านช่วงถนน ข้อมูลอรรถาธิบายประกอบด้วย รหัสรถโดยสารประจำทาง จุดสถานีต้นทาง จุดสถานีปลายทาง รายละเอียดของรถโดยสารประจำทาง

ข้อมูลความสัมพันธ์เชิงโครงข่าย	
รายการ	คำอธิบาย
จุดต้นช่วงถนน	รหัสอ้างอิงเพื่อเชื่อมโยงไปยังข้อมูลพิกัดตำแหน่ง
จุดปลายช่วงถนน	รหัสอ้างอิงเพื่อเชื่อมโยงไปยังข้อมูลพิกัดตำแหน่ง
ประเภทถนน	แบ่งเป็นสองประเภท คือ r: ถนนปกติ, h: ทางด่วน
สายรถโดยสารประจำทางที่ผ่านช่วงถนน	สายรถเขียนเรียงกันไป เช่น 54, 67, ...

ข้อมูลอรรถาธิบายประกอบ	
รายการ	คำอธิบาย
รายละเอียดของรถโดยสารประจำทาง	รายละเอียด เช่น ประเภทรถปรับอากาศ รถธรรมดา
จุดสถานีต้นทาง	รหัสสถานีต้นทางของสายรถนั้นๆ
จุดสถานีปลายทาง	รหัสสถานีปลายทางของสายรถนั้นๆ
รหัสรถโดยสารประจำทาง	หมายเลขหรือรหัสสายรถ

รูปที่ 2-5 ข้อมูลความสัมพันธ์เชิงโครงข่ายและข้อมูลอรรถาธิบาย

2. การประยุกต์ใช้ระบบสารสนเทศภูมิศาสตร์ในการแนะนำเส้นทางรถประจำทางที่สะดวก (ศิริพร ชวนิช, 2545)

งานวิจัยนี้เป็นการพัฒนาระบบสารสนเทศภูมิศาสตร์สำหรับสนับสนุนศูนย์บริการสอบถามเส้นทางทางโทรศัพท์ 184 ขององค์การขนส่งมวลชนกรุงเทพมหานคร (ขสมก.) ในการแนะนำเส้นทางรถประจำทางที่สะดวก โดยทำการศึกษาความต้องการด้วยวิธีสัมภาษณ์และสังเกตจากพนักงานรับโทรศัพท์ และจากบุคคลทั่วไปที่ใช้รถประจำทาง พบว่ามีความต้องแบ่งออกเป็น 4 ส่วน คือ ความต้องการทราบสายรถประจำทางจากจุดต้นทางถึงจุดปลายทาง ความต้องการตรวจสอบสายรถประจำทาง ความต้องการทราบสายรถประจำทางจากจุดที่กำหนด และความต้องการทราบป้ายรถประทางที่อยู่ใกล้ที่สุด

จากผลการศึกษาดังกล่าว ได้นำมาออกแบบและจัดสร้างฐานข้อมูล การวิเคราะห์กระบวนการ รวมถึงการจัดทำโปรแกรมประยุกต์เฉพาะทางขึ้นใหม่โดยใช้โปรแกรมประยุกต์ ArcInfo 8.0 เป็นฐาน โดยโปรแกรมประยุกต์เฉพาะทางที่จัดทำขึ้นมีความสามารถดังนี้

- ทำการค้นหาสายรถประจำทางจากสถานที่ต้นทางไปยังสถานที่ปลายทาง โดยสามารถระบุจำนวนครั้งการต่อรถที่ผู้ใช้ต้องการได้ตั้งแต่ 1 ถึง 3 ต่อ
- ระบุป้ายรถประจำทางที่ขึ้นลง ทิศทางและระยะทางที่จะต้องเดินเท้าไปขึ้นรถยังป้ายรถประจำทาง
- ตรวจสอบสายรถและค้นหาสายรถประจำทางจากจุดต่างๆ ที่กำหนด รวมทั้งสามารถค้นหาป้ายรถประจำทางที่ใกล้ที่สุด
- สามารถระบุระยะทางที่จะต้องเดินเท้าไปป้ายหยุดรถประจำทางหรือเพื่อเปลี่ยนสายรถ รวมถึงสามารถกำหนดระยะเดินเท้าที่ยอมให้เกิดขึ้นได้อีกด้วย

<u>แนวคิดเกี่ยวกับแบบจำลองข้อมูลที่ใช้</u>

ศิริพร ชวนิช (2545) ได้ทำการสร้างแบบจำลองข้อมูลเพื่อจัดเก็บข้อมูลระบบขนส่งมวลชนเพียงระบบเดียวคือ รถโดยสารประจำทาง โดยมีคุณสมบัติของแบบจำลองดังนี้

- จัดเก็บชั้นข้อมูลป้ายรถโดยสารประจำทางเป็นข้อมูลชนิด Point และมีข้อมูลอรรถาธิบายต่างๆ ที่เกี่ยวข้อง ได้แก่ รหัสป้าย ชื่อป้าย จุดสังเกต ถนน แขวง เขต โดยจุดสังเกตหมายถึงจุดเด่นที่สามารถสังเกตได้ในพื้นที่เพื่อใช้ช่วยอ้างอิงตำแหน่งป้ายและช่วยในการสื่อสารระหว่างเจ้าหน้าที่และผู้ที่โทรเข้ามาสอบถามเส้นทาง

- เส้นทางรถประจำทางจัดเป็นเป็นข้อมูลชนิด Line โดยแยกตามสายรถประจำทาง (1 Line ต่อ 1 สาย) และข้อมูลอรรถาธิบายคือ ชื่อสายรถประจำทาง ชื่อป้ายเริ่มต้น ชื่อป้ายสิ้นสุด เลขลำดับป้าย ชนิดเที่ยวการเดินรถ ชนิดรถ
- ผู้วิจัยได้กำหนดให้ป้ายรถฯ ทุกป้ายมีชื่อป้ายไม่ซ้ำกันโดยชื่อป้ายเกิดจากชื่อถนนรวมตัวอักษร “P” กับหมายเลขป้าย ตัวอย่างเช่น “สุขุมวิท P-02”
- หมายเลขป้ายจะเป็นเลขคู่ (02, 04, 06, …) สำหรับเส้นทางรถเข้าตัวเมืองและเป็นเลขคี่ (01, 03, 05, …) สำหรับเส้นทางรถวิ่งออกจากตัวเมือง
- หมายเลขป้ายจะเป็นตัวเลขเรียงลำดับไปเรื่อยๆ โดยป้ายที่กำหนดให้เป็นป้ายแรกสำหรับแต่เส้นทางนั้นจะพิจารณาจากการที่เมื่อรถเคลื่อนที่เข้าสู่ถนนสายนั้นแล้วพบกับป้ายใดเป็นป้ายแรก ไล่ลำดับไปเรื่อยๆ จนกระทั่งพบป้ายสุดท้ายของถนนสายนั้น
- หากมีการเพิ่มป้ายขึ้นมาภายหลังจะใช้เครื่องหมาย “/” แล้วตามด้วยตัวเลข เช่น “สุขุมวิท P-02/1” แต่ในกรณีที่มีการลบป้ายออก ก็จะลบรายการนั้นออกจากฐานข้อมูลไปเลย
- ยังมิได้คำนึงถึง Network topology ระหว่างป้ายแต่ละป้ายเนื่องจากมิได้ออกแบบเพื่อการวิเคราะห์เชิงโครงข่าย แต่ออกแบบเพื่อให้รองรับการสืบค้นป้ายหรือเส้นทางรถโดยสารประจำทางที่มีเงื่อนไขตรงตามความต้องการจากฐานข้อมูล เช่น เมื่อผู้ใช้สอบถามเส้นทางรถโดยสารประจำทางที่ผ่านจุดสองจุดเข้ามา โปรแกรมก็จะทำการสืบค้นหาสายรถที่ผ่านหรือใกล้กับจุดทั้งสองดังกล่าว เพื่อตอบเป็นสายรถที่สามารถใช้เดินทางจากจุดต้นทางไปยังจุดปลายทาง
- ยังมิได้มีการออกแบบให้รองรับการจัดเก็บค่าตัวแปรที่มีผลต่อระยะเวลาในการเดินทาง เช่น อัตราเร็ว ระยะเวลารอคอย หรือระยะทาง และหากต้องการเปรียบเทียบระยะทางของเส้นทางที่เป็นตัวเลือกนั้นใช้การคำนวณโดยอ้อม โดยอาศัยการนับจำนวนป้ายตลอดเส้นทางและอาศัยสมมติฐานว่าเส้นทางที่มีจำนวนป้ายมากกว่าย่อมไกลกว่า

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

3. Intelligent transport system using GIS (Kumar, Reddy และ Singh, 2003)

งานวิจัยนี้กล่าวถึงการพัฒนาระบบสารสนเทศแนะนำด้านการเดินทางสำหรับพื้นที่เมือง Hyderabad ประเทศอินเดีย ภายใต้เทคโนโลยีด้านระบบสารสนเทศภูมิศาสตร์ และจากการศึกษางานวิจัยดังกล่าว ผู้วิจัยสามารถสรุปประเด็นสำคัญๆ ออกมาได้ดังนี้

- ระบบสารสนเทศดังกล่าวพัฒนาขึ้นโดยใช้เครื่องมือดังนี้ ArcView GIS version 3.1, ArcView Network Analyst version 1.1b, Avenue programming language
- การนำเข้าข้อมูลเชิงพื้นที่ ใช้ข้อมูลแผนที่ภูมิประเทศ, ภาพถ่ายทางอากาศ และภาพถ่ายดาวเทียมที่ผ่านการลงทำเบียนจุดภาพแล้วเป็นฐานในการ digitize ข้อมูลเชิงพื้นที่ต่างๆ ที่สนใจ
- ระบบสารสนเทศดังกล่าวสามารถให้บริการ แนะนำเส้นทางการเดินทางที่สั้นที่สุดโดยพิจารณาจากระยะทาง หรือระยะเวลาการเดินทางระหว่างจุดสองจุด, แนะนำเส้นทางการเดินทางที่สั้นที่สุดโดยพิจารณาจากระยะทาง หรือระยะเวลาการเดินทางไปยังสิ่งอำนวยความสะดวกต่างๆ เช่น โรงพยาบาล สนามกีฬา เป็นต้น, แสดงเส้นทางเดินรถโดยสารประจำทาง, การสืบค้นตำแหน่งที่ตั้งสิ่งอำนวยความสะดวกต่างๆ, แสดงตารางเวลาการให้บริการของรถโดยสารระหว่างเมือง รถไฟ สายการบิน และช่วยการวางแผนเส้นทางเดินทางท่องเที่ยวโดยเสนอเส้นทางที่สั้นที่สุดและผ่านจุดต่างๆ ที่นักท่องเที่ยวต้องการ
- ระบบสารสนเทศดังกล่าวสามารถนำไปติดตั้งยังสถานที่ต่างๆ เช่น ที่จอดรถโดยสารประจำทาง, สถานีรถไฟ, สนามบิน, ศูนย์ข้อมูลนักท่องเที่ยว โดยจะต้องทำงานบนเครื่องคอมพิวเตอร์ส่วนบุคคลแบบตั้งโต๊ะ

<u>แนวคิดเกี่ยวกับแบบจำลองข้อมูลที่ใช้</u>

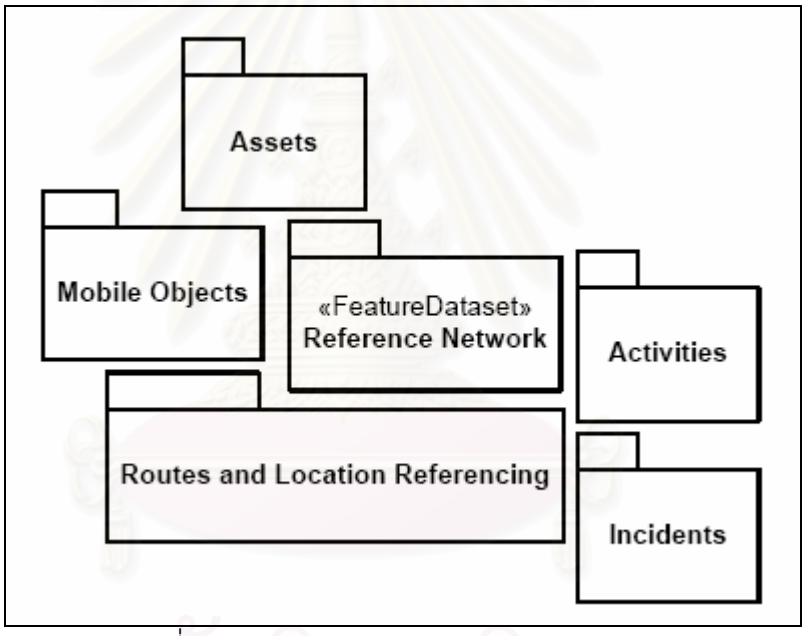
งานวิจัยนี้เป็นการใช้ แบบจำลองข้อมูลอ้างอิงจาก ArcView Network Analyst version 1.1b ของ ESRI ซึ่งเป็นแบบจำลองข้อมูลสำเร็จรูปพร้อมซอฟต์แวร์เสริมที่สามารถสร้างข้อมูลโครงข่ายโดยอัตโนมัติจากข้อมูลเส้นทางถนนเดิมที่อยู่ในรูป shape file จุดที่เส้นตัดกันจะถูกกำหนดให้เป็น node ส่วนเส้นถนนที่เชื่อมระหว่างจุดตัดกันสองจุดจะกำหนดให้เป็น arc

ต่อมาในปี คศ. 2003 บริษัท ESRI ได้พัฒนาแบบจำลองข้อมูลรุ่นใหม่ออกมาชื่อ UNETRANS ซึ่งมีการเผยแพร่รายละเอียดและมีความก้าวหน้ากว่ามาก ผู้วิจัยจึงขอยกยอดไปกล่าวรายละเอียดในหัวข้อถัดไปเรื่อง “ArcGIS Transportation Data Model”

4. ArcGIS Transportation Data Model (UNETRANS)

เมื่อสิงหาคม ปี 2003 คณะทำงานกลุ่มหนึ่ง (Environmental Systems Research Institute, 2003) ซึ่งประกอบไปด้วยกลุ่มผู้ใช้ และผู้มีส่วนได้ส่วนเสีย นักวิจัยจาก NCGIA (National Center for Geographic Information Analysis) นักพัฒนาจากบริษัท ESRI ได้ร่วมกันพัฒนาและเผยแพร่แบบจำลองข้อมูลที่ชื่อว่า UNETRANS ออกมา ซึ่งเป็นแบบจำลองข้อมูลระบบสารสนเทศภูมิศาสตร์สำหรับการขนส่งและจราจร (GIS-T, Geographic Information System for Transportation)

UNETRANS มิได้มุ่งจะสร้างมาตรฐานใหม่หากแต่เป็นการพยายามรองรับมาตรฐานที่มีอยู่เดิมด้วยการออกแบบฐานข้อมูลเพื่อการใช้งานเชิงปฏิบัติซึ่งมุ่งให้ทำงานสอดคล้องได้ดีกับซอฟต์แวร์ตระกูล ArcGIS ของ บริษัท ESRI (ESRI, 2003: 25)



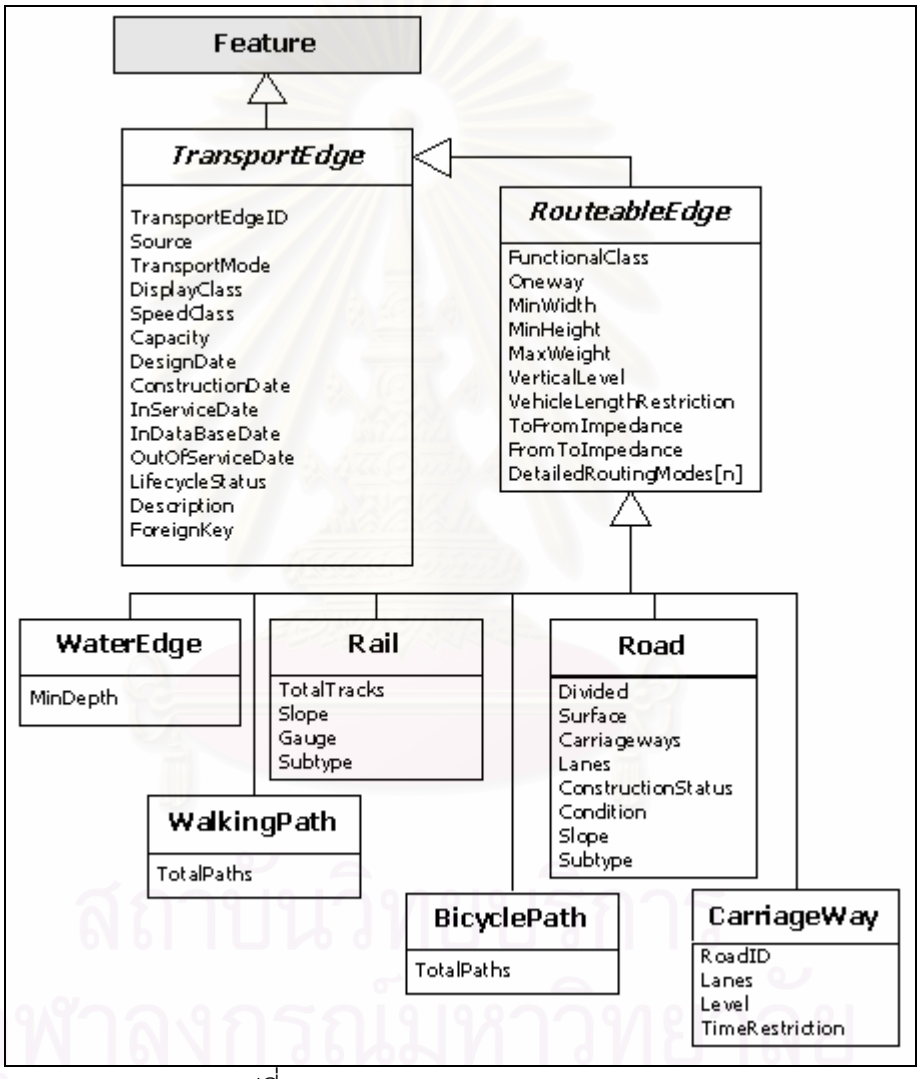
รูปที่ 2-6 ภาพรวมของ UNETRANS packages

UNETRANS ประกอบด้วย packages หลักทั้งหมด 6 packages ได้แก่ Reference Network, Routes and Location Referencing, Assets, Activities, Incidents และ Mobile Objects ดังแสดงในรูปที่ 2-6 แต่ละ package ประกอบด้วยเซ็ตของ object กับ feature class และความสัมพันธ์ระหว่างแต่ละ class

จากรูปที่ 2-7 Reference Network package เป็น package ที่กำหนดชนิดและคุณสมบัติของ Edge ซึ่งทำหน้าที่เป็นเส้นทางขนส่งหรือทางเดินในโครงข่าย โดยเริ่มจาก Object class ชื่อ TransportEdge ซึ่งประกอบด้วยค่าอรรถาธิบายต่างๆ เบื้องต้น และมี class ชื่อ RoutableEdge

เป็น class ลูกซึ่ง inherit คุณสมบัติต่างๆ มาจาก TransportEdge นอกจากนี้ RoutableEdge ยังมี class ลูก อีกได้แก่ WaterEdge, Rail, Road, WalkingPath, BicyclePath, และ CarriageWay

แม้จะได้มีการกำหนด class ต่างๆ พร้อมรายการอรรถาธิบายไว้ล่วงหน้าแล้ว แต่ผู้ใช้ก็สามารถนำมาปรับเปลี่ยนเพิ่ม/ลดให้ตรงตามลักษณะงานหรือความต้องการของตนได้ เช่น การแก้อรรถาธิบายชื่อ “MinDepth” ใน WaterEdge เป็นอย่างอื่น หรือแม้กระทั่งเพิ่ม subclass ใหม่เข้ามา

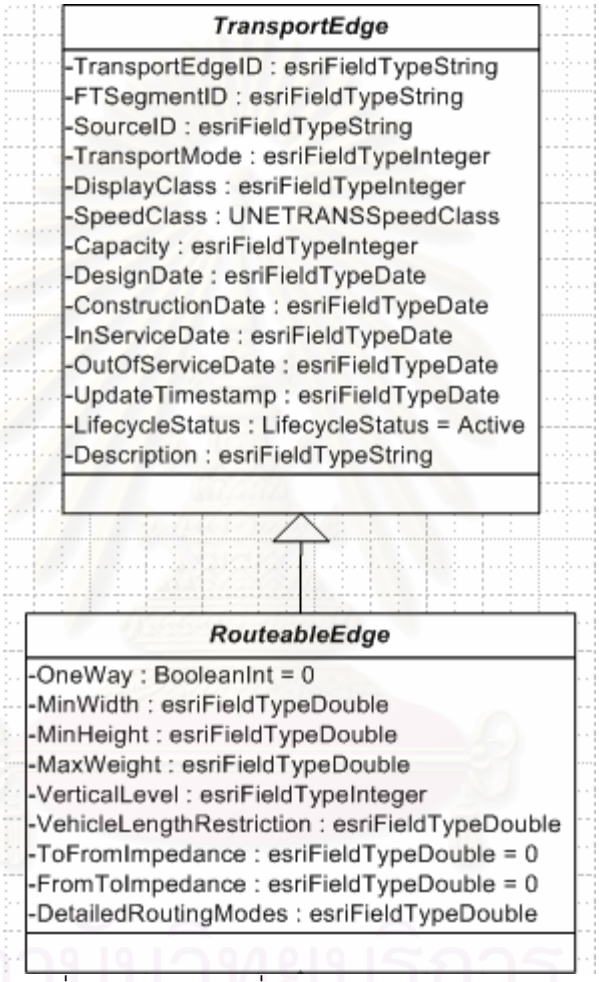


รูปที่ 2-7 Reference Network package

อย่างไรก็ตามในเอกสารที่เผยแพร่มานั้นเป็นเพียงการอธิบายโดยคร่าวๆ เท่านั้น ยังมิได้อธิบายลงลึกในรายละเอียด เช่น ความหมายของอรรถาธิบายแต่ละรายการในแต่ละ subclass จะมีก็เพียงชนิดข้อมูลที่ใช้จัดเก็บอรรถาธิบายนั้นๆ ดังตัวอย่างแสดงในรูปที่ 2-8 เช่น

TransportEdgeID ก็เพียงบอกว่าเป็นข้อมูลชนิด esriFieldTypeString ซึ่งเป็นชนิดข้อมูลที่นิยามขึ้นใช้ในชุดซอฟต์แวร์ของบริษัท ESRI

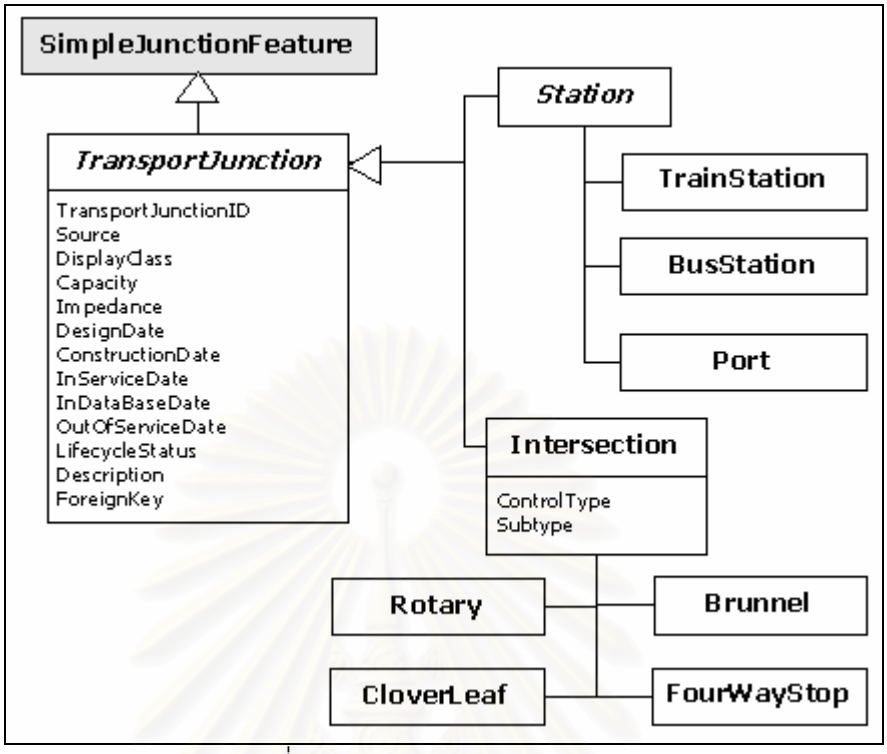
จากกรณีดังกล่าวอาจสันนิษฐานได้ว่าตัวมาตรฐานนั้นต้องการให้ผู้ใช้คิดวิธีการกำหนดค่าหรืออธิบายความหมายในรายละเอียดด้วยตนเอง ทั้งนี้เพื่อให้สอดคล้องกับความต้องการใช้งานและสภาวะเงื่อนไขที่แตกต่างกันไป



รูปที่ 2-8 ชนิดข้อมูลที่ใช้จัดเก็บข้อมูลอรรถาธิบาย

นอกจาก TransportEdge แล้ว ใน Reference Network package ยังมี class ชื่อ TransportJunction ด้วย ซึ่งใช้อธิบาย junction ที่เชื่อมระหว่าง edge ในโครงข่าย แบบจำลองข้อมูลของ TransportJunction แสดงโครงสร้างในรูปที่ 2-9

TransportJunction แบ่งออกเป็นสองชนิดคือ station และ intersection โดย station คือสถานีหรือป้ายสำหรับเข้าหรือออกจากระบบขนส่งมวลชน junction ชนิดนี้จะวางตัวไปตามแนวยาวของ edge ต่างกับกรณี intersection ซึ่งเป็นส่วนหนึ่งของ edge เลย โดย intersection เกิดจากการตัดกันของ edge



รูปที่ 2-9 TransportJunction class

สำหรับ package อื่นๆ นั้นก็จะเป็นในลักษณะเดียวกันกับ TransportEdge โดย Routes and Location Referencing เป็น class ที่ใช้จัดเก็บข้อมูลเพิ่มเติมสำหรับเงื่อนไขต่างๆ เพื่อการสืบค้นเส้นทางที่เหมาะสมในโครงข่ายตามเงื่อนไขที่กำหนด (routing) เช่น ช่วงเวลาที่จำกัดการเข้าสู่ edge กฎจราจร หรือข้อจำกัดการเลี้ยวที่ทางแยกต่างๆ การจำกัดยานพาหนะบางประเภทสำหรับบางเส้นทาง ดังแสดงโครงสร้างในรูปที่ 2-10

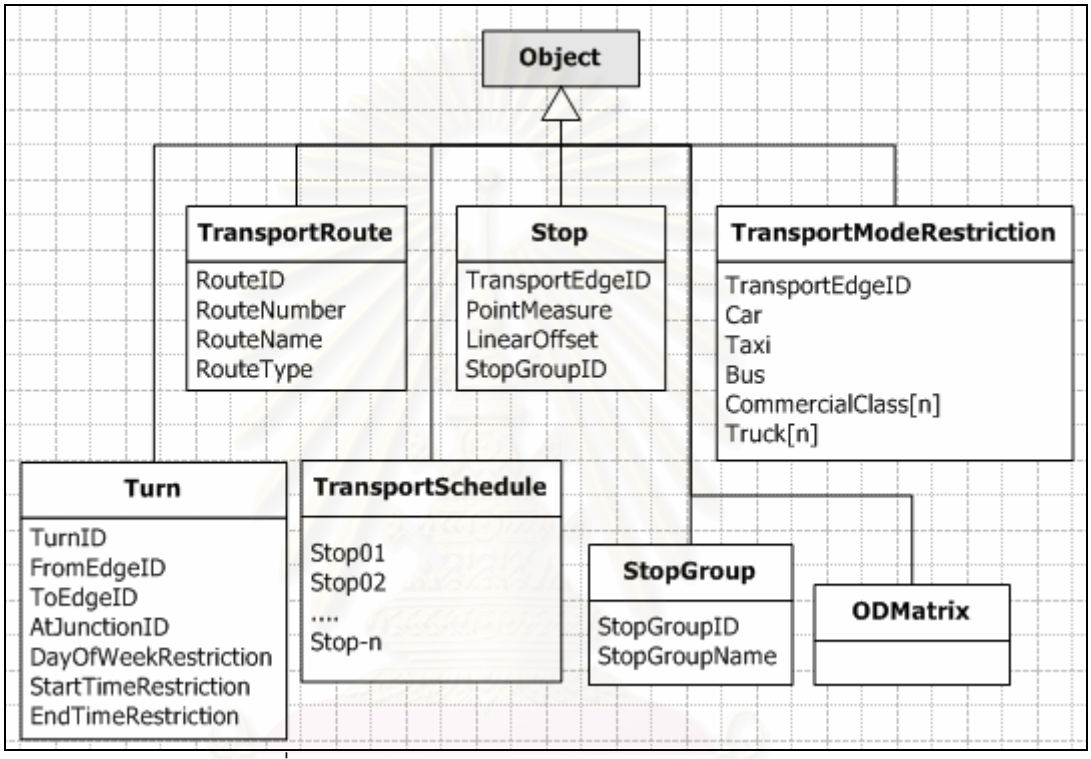
สำหรับกรณี package อื่นๆ ที่แสดงอยู่ในรูปที่ 2-4 สามารถอธิบายโดยสรุปได้ดังนี้

กรณี Assets ใช้แสดงสิ่งที่มีอยู่จริงเชิงกายภาพอาจเป็นในรูป จุด เส้น หรือพื้นที่ก็ได้ โดย Assets มิได้ถือเป็นส่วนหนึ่งของ Network แต่มีความสัมพันธ์และมีผลกระทบต่อการเดินทางใน Network ตัวอย่างของ Asset เช่น สัญญาณไฟจราจร ป้ายสัญลักษณ์จราจร ขอบทางเท้า แผงกั้นทางรถไฟ พื้นที่จอดรถ เป็นต้น

กรณี Activities จะค่อนข้างคล้ายคลึงกับ Asset เพียงแต่ Activities จะใช้ในกรณีที่ต้องการอธิบายถึง feature ที่มีอิทธิพลต่อ Network ในช่วงเวลาหนึ่งเท่านั้นมิได้อยู่ถาวรเหมือนกรณี Asset ตัวอย่างของ Activities เช่น การก่อสร้างซ่อมแซมทาง การปิดถนน หรืองานพิธีต่างๆ ที่มีผลต่อการจราจร เป็นต้น

กรณี Incidents จะคล้ายคลึงกับ Activities ต่างกันตรงที่ incident จะเป็นเหตุการณ์ในลักษณะที่ไม่อาจคาดการณ์หรือบอกช่วงเวลาการดำรงอยู่ได้ล่วงหน้า เช่น อุบัติเหตุรถยนต์สารเคมี หรือการก่อเหตุระเบิด เป็นต้น

กรณี Mobile เป็นการมุ่งจำแนกและอธิบายวัตถุที่เคลื่อนที่ไปใน Network เช่น รถยนต์ส่วนบุคคล ขบวนรถขนสินค้า รถจักรยาน หรือ คนเดินเท้า เป็นต้น



รูปที่ 2-10 Routes and Location Referencing class

5. Intelligent transport systems – Geographic Data Files- Overall data specification (GDF 4.0, ISO/FDIS 14825)

ISO/FDIS 14825 (2003) เป็นมาตรฐานสากลที่ออกโดย International Organization for Standardization ว่าด้วยเรื่องของแบบจำลองข้อมูลสำหรับข้อมูลระบบสารสนเทศภูมิศาสตร์ในงานด้านขนส่งและจราจร โดยถือเป็นแบบจำลองข้อมูลในระดับ conceptual และ logical และรวมถึงรูปแบบข้อมูลกลางเพื่อการแลกเปลี่ยนและสื่อสารกัน

ในมาตรฐานดังกล่าว มีรายละเอียดมากมายและครอบคลุมสิ่งต่างๆ ที่อยู่ในโลกความจริงกว้างมาก โดยจะเน้นอธิบายความหมายของคำจำกัดความต่างๆ การจัดกลุ่ม การแยกประเภทของสิ่งต่างๆ ทั้งหมดที่เกี่ยวข้องทั้งโดยตรงและโดยอ้อมกับระบบขนส่งและจราจร เช่น ถนน เส้นทางน้ำ ทางรถไฟ สถานีขนส่งประเภทต่างๆ สถานที่สำคัญประเภทต่างๆ ระบบอ้างอิงที่อยู่ทาง

ไปรษณีย์ สิ่งปลูกสร้าง ยานพาหนะชนิดต่างๆ ขอบเขตการปกครองประเภทต่างๆ พื้นที่แหล่งน้ำ พื้นที่การเกษตร ระบบการออกเสียงอ่านหรือเรียกชื่อสิ่งต่างๆ ฯลฯ

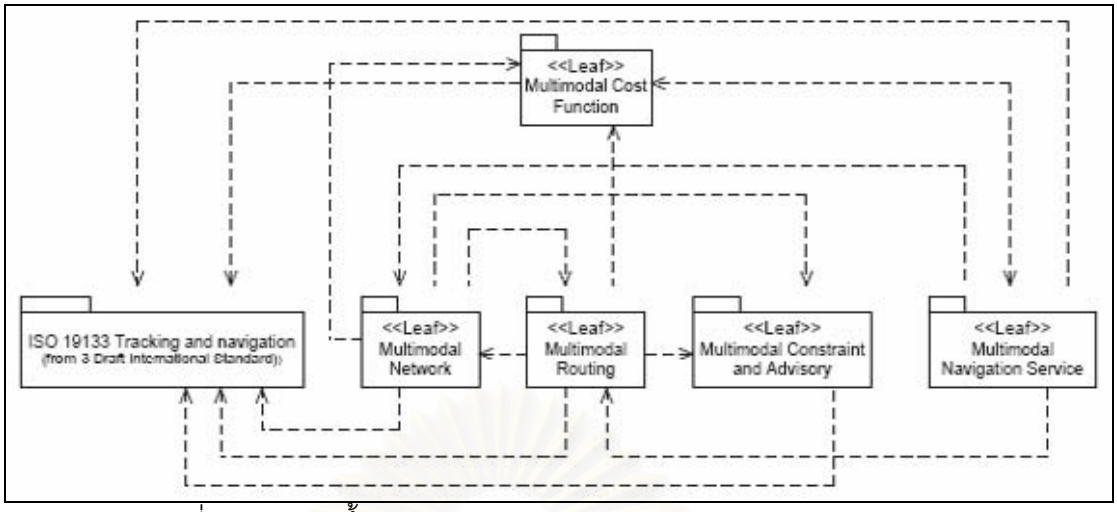
แบบจำลองข้อมูลในมาตรฐาน GDF จัดอยู่ในระดับเชิงกรอบแนวคิด (conceptual) และเชิงตรรกะ (logical) เท่านั้น ยังมิได้ลงลึกถึงระดับการออกแบบแบบจำลองข้อมูลเพื่อพัฒนาระบบหรือการบริการเฉพาะด้าน เช่น การพัฒนาแบบจำลองข้อมูลเพื่อการนำทาง หรือ การค้นหาเส้นทางที่เหมาะสมสำหรับกรณีต่างๆ อนึ่งแบบจำลองข้อมูลที่สร้างขึ้นในวิทยานิพนธ์นี้เน้นไปที่การออกแบบเพื่อใช้สืบค้นและคำนวณเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดผ่านระบบขนส่งมวลชนหลายรูปแบบ

6. Geographic Information - Location based services- Multimodal routing and navigation (ISO/DIS 19134)

ISO/DIS 19134 อ้างอิงแบบจำลองข้อมูลหลายส่วนมาจาก ISO 19133 ซึ่งเป็นมาตรฐานสากลว่าด้วยเรื่องการตรวจจับและรายงานตำแหน่งยานพาหนะ (Tracking) และระบบนำทาง (Navigation) โดยใน ISO/DIS 19134 เป็นเอกสารซึ่งอธิบายถึงมาตรฐานสากลสำหรับแบบจำลองข้อมูลเพื่อการพัฒนาบริการ Location based services สำหรับ Multimodal routing and navigation ดังนั้นจึงประกอบด้วยสองส่วนคือ การค้นหาเส้นทาง (Routing) และ การนำทาง (Navigation) ทั้งนี้ผู้วิจัยจะเน้นอธิบายเฉพาะส่วนของการค้นหาเส้นทางซึ่งเป็นส่วนที่เกี่ยวเนื่องกับงานวิจัยโดยตรง

ISO/DIS 19134 แบ่งกลุ่มแบบจำลองข้อมูลออกเป็นกลุ่มย่อยห้ากลุ่มคือ Multimodal Network, Multimodal Routing, Multimodal Constraint and Advisory, Multimodal Cost Function, และ Multimodal Navigation Service ดังแสดงความสัมพันธ์ด้วยแผนผัง UML ในรูปที่ 2-11

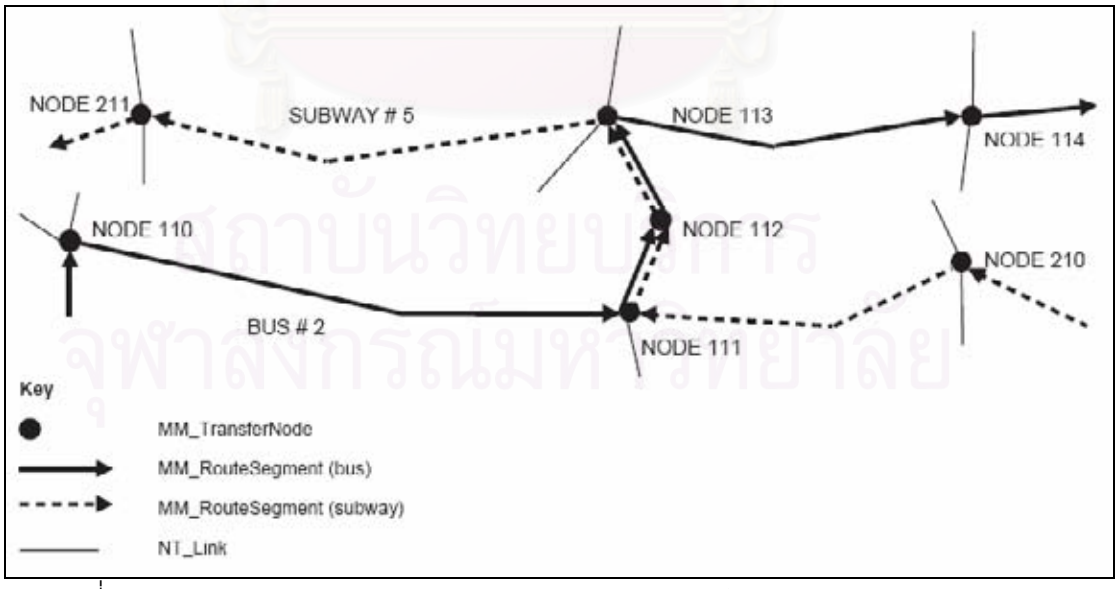
เป็นการอธิบายแบบจำลองข้อมูลสำหรับระบบขนส่งหลายรูปแบบในแง่ต่างๆ คือ ข้อมูลโครงข่าย (Network) การค้นหาเส้นทาง(Routing) ข้อบังคับและคำแนะนำสำหรับการเดินทาง (Constraint and Advisory) ฟังก์ชั่นสำหรับการคำนวณค่าใช้จ่าย (Cost Function) และ การบริการนำทาง (Navigation Service)



รูปที่ 2-11 ความขึ้นต่อกัน (dependencies) ของกลุ่มแบบจำลองข้อมูล

โครงข่ายระบบขนส่งหลายรูปแบบ (Multimodal Network)

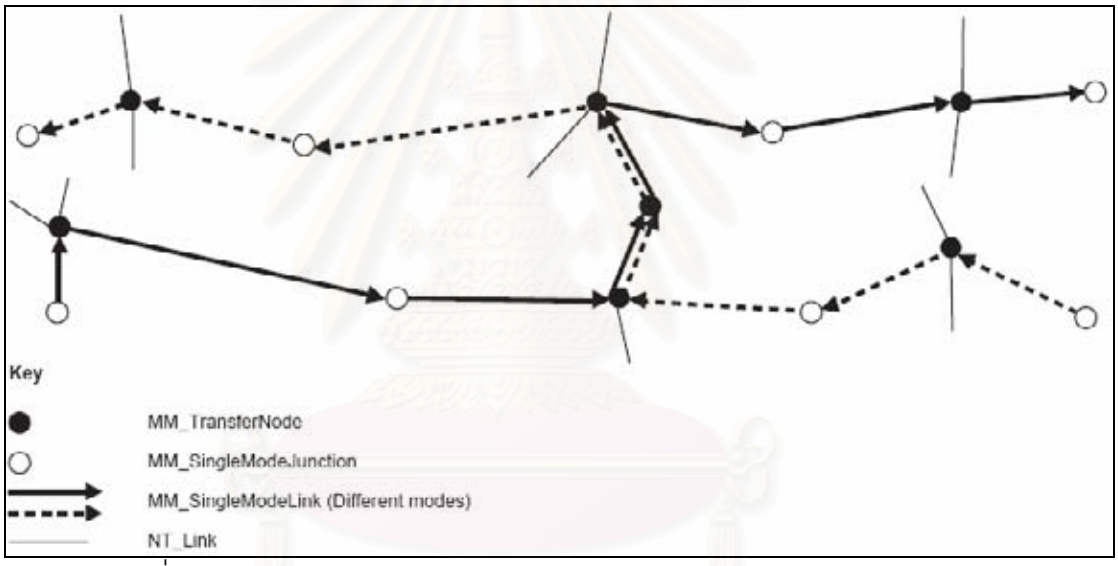
โครงข่ายระบบขนส่งหลายรูปแบบดังแสดงใน รูปที่ 2-12 ซึ่งประกอบด้วยช่วงของเส้นทาง (route segment, MM_RouteSegment) และ node สำหรับเปลี่ยนหรือเข้าสู่ระบบขนส่งหลายรูปแบบ (transfer nodes, MM_TransferNode) เส้นทึบในรูปเป็นเส้นทางของสายรถโดยสารประจำทาง และเส้นประในรูปเป็นเส้นทางของสายรถไฟฟ้าใต้ดิน ตัวอย่างการเปลี่ยนรูปแบบการเดินทาง เช่นต้องการเปลี่ยนจากรถโดยสารประจำทางหมายเลข 2 (Bus #2) ไปยังรถไฟใต้ดินหมายเลข 5 (Subway #5) ผู้เดินทางจะต้องมาเปลี่ยนที่ node 112



รูปที่ 2-12 MM_TransferNode และ MM_RouteSegment ใน MM_MultimodalNetwork

โครงข่ายระบบขนส่งหลายรูปแบบสามารถอธิบายให้ละเอียดมากขึ้นด้วยรูปที่ 2-13 ซึ่งแสดง MM_SingleModeLinks (อ้างอิงคุณสมบัติมาจาก NT_Link ซึ่งระบุอยู่ใน ISO 19133) และ MM_SingleModeJunctions ซึ่งเป็น junction ที่เชื่อมระหว่างระบบขนส่งระบบเดียวกัน ส่วน MM_transferNode เป็น junction ที่เชื่อมระหว่างระบบขนส่งคนละรูปแบบในที่นี้คือเส้นทึบและเส้นประหนาหมายถึงระบบขนส่งมวลชนสองระบบ

เส้น NT_Link แสดงเป็นเส้นทึบบางๆ เชื่อมออกไปจาก MM_TransferNode ซึ่ง NT_Link ในที่นี้หมายถึงโครงข่ายเส้นทางทั่วไป เช่น ถนนหลวง ในกรณีที่ผู้เดินทางต้องการเปลี่ยนจากระบบขนส่งสาธารณะเป็นระบบส่วนตัวก็จะต้องใช้ MM_TransferNode เป็นจุดเปลี่ยน ตัวอย่างในเชิงกายภาพของ MM_TransferNode อาจหมายถึงอาคารจอดรถเพื่อเปลี่ยนจากรถยนต์ส่วนตัวเข้าสู่ระบบขนส่งสาธารณะหรือกลับกัน

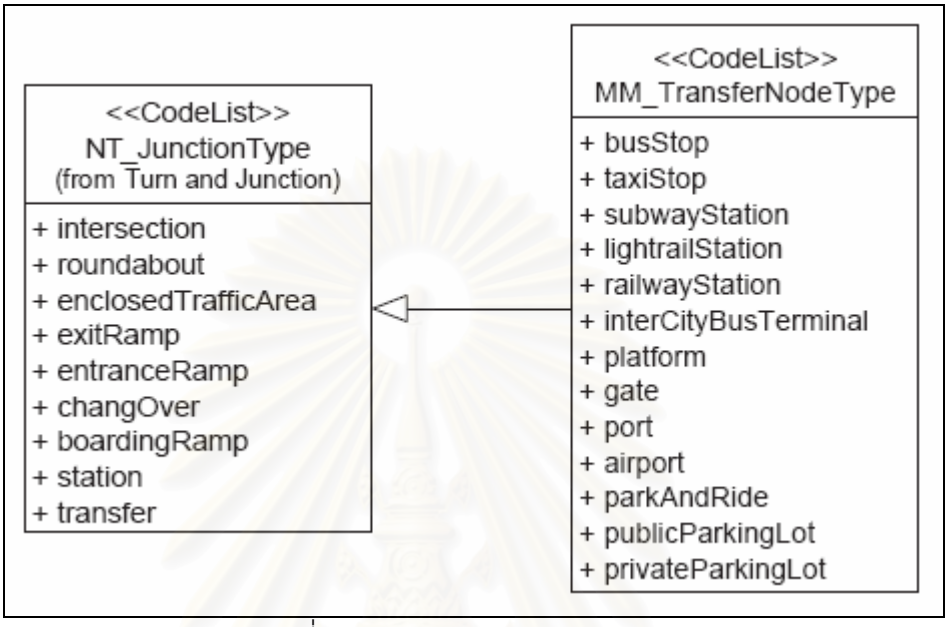


รูปที่ 2-13 MM_SingleModeLinks และ MM_SingleModeJunctions ภายใน MM_MultimodalNetwork

ในแง่ของโครงข่ายถนนปกติหรือโครงข่ายขนส่งส่วนตัว (Private transportation network) เมื่อกล่าวถึง junction (node) โดยทั่วไปจะหมายความถึงจุดที่ถนนตัดกันเป็นทางแยก (Miller, H. J. and Shaw, S., 2001: 55)

แต่ในกรณีระบบขนส่งสาธารณะ (Public transportation network) เมื่อกล่าวถึง junction (node) มักจะใช้แทนจุด สถานี หรือสถานที่สำหรับเข้าสู่หรือออกจากระบบขนส่งมวลชน เพื่อเปลี่ยนระหว่างระบบขนส่งมวลชนด้วยกัน (Miller, H. J. and Shaw, S., 2001: 56) หรือเปลี่ยนระหว่างระบบขนส่งสาธารณะและส่วนตัวดังแสดงในรูปที่ 2-14 ซึ่งนำมาจากเอกสาร ISO 19133 ที่ได้ออกแบบแบบจำลองข้อมูลสำหรับรองรับ junction ของทั้งสองระบบไว้ โดย

MM_TransferNodeType เป็นการแตกออกมาจาก NT_JunctionType ใน ISO 19133 สังเกตว่า MM_TransferNodeType จะครอบคลุมรวมทั้งสถานีของระบบขนส่งสาธารณะ เช่น subwayStation และ จุดจอดแล้วจร (park-and-ride) เช่น publicParkingLot



รูปที่ 2-14 MM_TransferNodeType

นอกจากการอธิบายสิ่งต่างๆ (Node และ Link) ที่ประกอบกันเป็น network แล้ว ISO/DIS 19134 ยังได้อธิบายและออกแบบแบบจำลองข้อมูลเพื่อรองรับการเลือก Link ที่เกิดขึ้นที่ node ไว้อีกหรือเรียกตามศัพท์เฉพาะว่า Turn

Turn เกิดจากการรวมกันของ junction และ link ที่วิ่งเข้าหรือวิ่งออกจาก junction นั้น (ISO 19133, 2005) หรืออาจกล่าวอีกนัยหนึ่งว่า Turn ใช้อธิบายถึงจำนวนความเป็นไปได้ที่จะเดินทางผ่าน junction (node) ใน network

ทั้งนี้ในกรณีที่เป็น Private transportation network จำนวนความเป็นไปได้ดังกล่าวจะหมายถึงจำนวนความเป็นไปได้ของการเคลื่อนตัวของยานพาหนะผ่าน node ใดๆ ใน network (เลี้ยวซ้าย เลี้ยวขวา ตรงไป หรือกลับรถ เป็นต้น) แต่หากเป็นกรณี Public transportation network จำนวนความเป็นไปได้ดังกล่าวจะหมายถึงตัวเลือกหรือทางเลือกของระบบขนส่งมวลชนที่มีให้ผู้เดินทาง (มนุษย์) เลือกใช้ ณ node นั้น ซึ่งความเป็นไปได้ดังกล่าวมีตั้งแต่ศูนย์ (ไม่มีระบบขนส่งมวลชนให้เลือกเลย ณ สภาวะขณะนั้น) เป็นต้นไป

ค่าใช้จ่ายในการเดินทาง (Multimodal Cost Function)

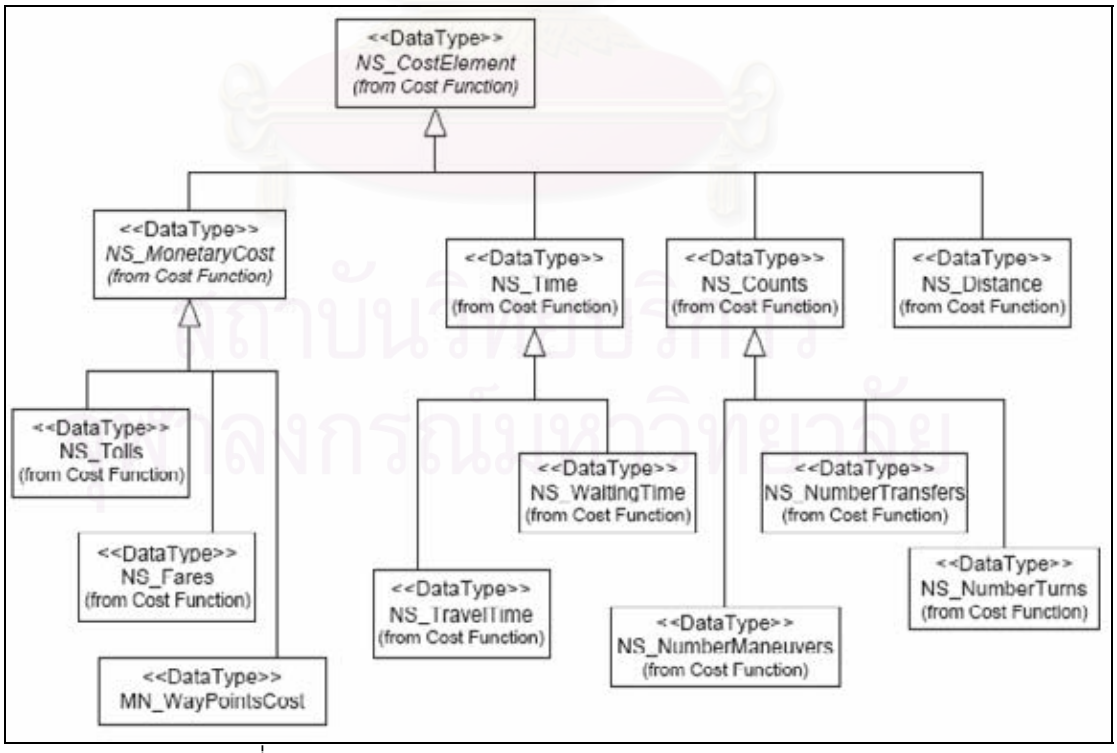
ค่าใช้จ่ายในการเดินทาง เป็นอีกประเด็นที่น่าสนใจ ในเอกสาร ISO 19133 ได้จำแนกค่าใช้จ่ายในการเดินทางออกเป็นประเภทดังแสดงในรูปที่ 2-15 โดยแบ่งออกเป็นค่าใช้จ่ายที่เป็นตัวเงิน (NS_Monetary) ค่าใช้จ่ายด้านระยะเวลา (NS_Time) ค่าใช้จ่ายเรื่องจำนวนครั้ง (NS_Count) และค่าใช้จ่ายด้านระยะทาง (NS_Distance)

ค่าใช้จ่ายที่เป็นตัวเงิน ประกอบด้วย ค่าผ่านทาง (NS_Tolls) ค่าธรรมเนียม (NS_Fares) และ ค่าเดินทาง (NMN_WayPointsCost) เช่น ค่าที่จอดรถ ค่าที่พัก ค่าอาหาร เป็นต้น

ค่าใช่จ่ายด้านระยะเวลา ประกอบด้วย ระยะเวลารอคอยหรือหยุดตามจุดต่างๆ ระหว่างทาง (NS_WaitingTime) และระยะเวลาเดินทาง (NS_TravelTime)

ค่าใช้จ่ายเรื่องจำนวนครั้ง ประกอบด้วย จำนวนครั้งในการเปลี่ยนรูปแบบการเดินทาง (NS_NumberTransfers) จำนวนครั้ง (ความซับซ้อน) ในการเดินทาง (NS_NumberManeuvers) เช่น จำนวนการเลี้ยว จำนวนการขึ้นหรือลงทางลาด หรือจำนวนครั้งการกลับรถ เป็นต้น ซึ่งโดยปกติ การเปลี่ยนแปลงหนึ่งครั้งจะนับเป็นหนึ่ง และสุดท้ายคือจำนวนทางเลือก turn (NS_NumberTurns) ซึ่งโดยปกติหนึ่งทางเลือกจะนับเป็นหนึ่ง หากเป็นเส้นทางตรงตลอดจะไม่มีการนับ

ค่าใช้จ่ายด้านระยะทาง เป็นการคำนวณค่าใช้จ่ายจากระยะทางของเส้นทางที่พิจารณา



รูปที่ 2-15 ค่าใช้จ่าย (cost) สำหรับระบบขนส่งหลายรูปแบบ

สำหรับค่าใช้จ่ายที่อยู่ในขอบเขตของวิทยานิพนธ์นี้จะคำนึงถึงเฉพาะ ค่าใช่จ่ายด้านระยะเวลา ได้แก่ ระยะเวลารอคอยหรือหยุดตามจุดต่างๆ ระหว่างทาง (เฉพาะระยะเวลาที่ใช้รอคอยยานพาหนะ) และระยะเวลาเดินทาง ซึ่งคำนวณจากอัตราเร็วเฉลี่ยและระยะทาง

การระบุข้อจำกัดและการให้คำแนะนำเส้นทาง (Multimodal Constraint and Advisory)

กรณี Multimodal Constraint เป็นการกำหนดคุณสมบัติของ Link หรือ Turn ต่างๆ โดยอิงจาก NT_Constraint ใน ISO 19133 แล้วกำหนดคุณลักษณะอื่นสำหรับ Multimodal transportation เพิ่มเติมเข้ามาใน ISO 19134 คุณลักษณะพื้นฐาน เช่น ทางห้ามผ่านในบางช่วงเวลา ห้ามเลี้ยวขวา ยานพาหนะที่อนุญาตหรือไม่อนุญาตให้ผ่าน เช่น ห้ามรถที่มีขนาดหรือน้ำหนักเกินพิกัด ห้ามรถจักรยานยนต์ เป็นต้น หรือการระบุลักษณะของทางชันหรือทางโค้ง ส่วนคุณลักษณะที่กำหนดเพิ่มคือ รูปแบบการเดินทางที่ผู้เดินทางต้องการใช้หรือเลือกเป็นการเฉพาะ เช่น ต้องการเดินทางโดยรถโดยสารประจำทางเท่านั้น

ส่วนกรณี Multimodal Advisory เป็นการออกแบบแบบจำลองข้อมูลให้สามารถจัดเก็บคำอธิบายต่างๆ สำหรับผู้ขับขี่ยานพาหนะ เพื่อช่วยในการเดินทาง หรือช่วยนำทาง เช่น จุดสังเกต ทางเชื่อม ทางแยก ทางลัด เป็นต้น และใน ISO 19134 ก็ได้มีการเพิ่มคำอธิบายสำหรับจุดที่เข้าสู่และออกจากระบบขนส่ง สำหรับกรณีผู้เดินทางผ่าน Multimodal transportation

การให้บริการนำทาง (Multimodal Navigation Service)

เป็นการออกแบบให้แบบจำลองข้อมูลมีส่วนที่รองรับการร้องขอและโต้ตอบเพื่อแนะนำเส้นทางซึ่งในส่วนนี้ผู้วิจัย ไม่ขอกล่าวในรายละเอียด เนื่องจาก อยู่นอกขอบเขตและมีรายละเอียดมาก ผู้สนใจสามารถค้นคว้าเพิ่มเติมได้จาก เอกสาร ISO 19133 และ ISO 19134 ในหัวข้อ Navigation Service และ Multimodal Navigation Service ตามลำดับ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ข้อคิดเห็นของผู้วิจัยต่องานวิจัยที่มีมาก่อน

- งานวิจัยสองงานแรกให้ข้อมูลเฉพาะรถโดยสารประจำทางเท่านั้น ยังไม่รวมระบบขนส่งมวลชนประเภทอื่น นอกจากนี้ยังไม่สามารถกำหนดให้โปรแกรมประยุกต์เลือกเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดได้
- งานวิจัยที่สามไม่สามารถระบุให้แนะนำเส้นทางที่เหมาะสมในการใช้บริการระบบขนส่งมวลชนได้
- โปรแกรมประยุกต์ในงานวิจัยแรกและงานวิจัยที่สามยังทำงานโดยใช้แฟ้มข้อมูลเป็นฐาน จึงทำให้ไม่สามารถปรับปรุงค่าตัวแปรและข้อมูลอรรถาธิบายต่างๆ ที่เกี่ยวข้องในการประมวลผลให้เป็นแบบพลวัตได้ เนื่องจากการทำงานกับแฟ้มข้อมูลนั้นจะถูกจำกัดสิทธิโดยระบบปฏิบัติการ ให้มีโปรแกรมประยุกต์เข้าถึงแฟ้มข้อมูลได้ครั้งละหนึ่งโปรแกรมเท่านั้น ทำให้ไม่สามารถปรับปรุงค่าตัวแปรและให้บริการสอบถามเส้นทางพร้อมกันได้
- งานวิจัยที่หนึ่งและสาม ไม่ได้คำนึงถึงการเดินเท้าซึ่งเป็นรูปแบบการเดินทางหลักที่ใช้เชื่อมต่อระหว่างการเดินทางแต่ละช่วง (การต่อรถ หรือเปลี่ยนรูปแบบยานพาหนะ) มีเพียงงานวิจัยที่สองที่คำนึงถึงการเดินเท้าด้วย

ส่วนในงานวิจัยนี้จะมุ่งเน้นภายใต้กรอบการเลือกรูปแบบการเดินทางแบบผสมผสานเพื่อให้ได้เส้นทางที่มีระยะเวลาการเดินทางน้อยที่สุดเป็นหลัก และใช้สมมติฐานว่าการเดินเท้าเป็นรูปแบบการเดินทางที่ใช้เชื่อมต่อระหว่างรูปแบบการเดินทางแต่ละรูปแบบ และระหว่างสายทางภายในรูปแบบการเดินทางเดียวกัน

ความสอดคล้องและความแตกต่างระหว่างสมมติที่ใช้ในวิทยานิพนธ์กับมาตรฐานสากล ISO 19133 และ ISO 19134

แนวคิดที่ได้จากมาตรฐาน ISO 19133 และ ISO 19134 ที่ได้นำมาใช้ในงานวิจัยนี้มีความสามารถสรุปเป็นประเด็นได้ดังนี้

1. Junction (node) ในกรณีของ ISO19134 สำหรับกรณี Multimodal จะกำหนดให้ node หมายถึงจุดสถานีรับส่งผู้โดยสาร ป้ายหยุดรถโดยสารประจำทาง หรือท่าเรือ ไม่ใช่ทางแยกที่เกิดจากการตัดกันของถนน
2. เมื่อจะมีเปลี่ยนรูปแบบการเดินทาง เช่น จากรถโดยสารประจำทางไปเป็นรถไฟฟ้า ผู้เดินทางจะต้องเดินทางมายัง node ที่เรียกว่า TransferNode ซึ่งเป็น node ที่สังกัดอยู่ในระบบขนส่งมวลชนทั้งสองประเภท และจะต้องมีการ arc เชื่อมจาก node ของระบบขนส่งแต่ละรูปแบบเข้าสู่ TransferNode นี้ เพียงแต่ในกรณีของมาตรฐานสากลนั้นจะถือรวม

เอา node ที่เกิดการเปลี่ยนรูปแบบการเดินทางจาก Private transportation มาเป็น Public transportation รวมอยู่ใน TransferNode ด้วย

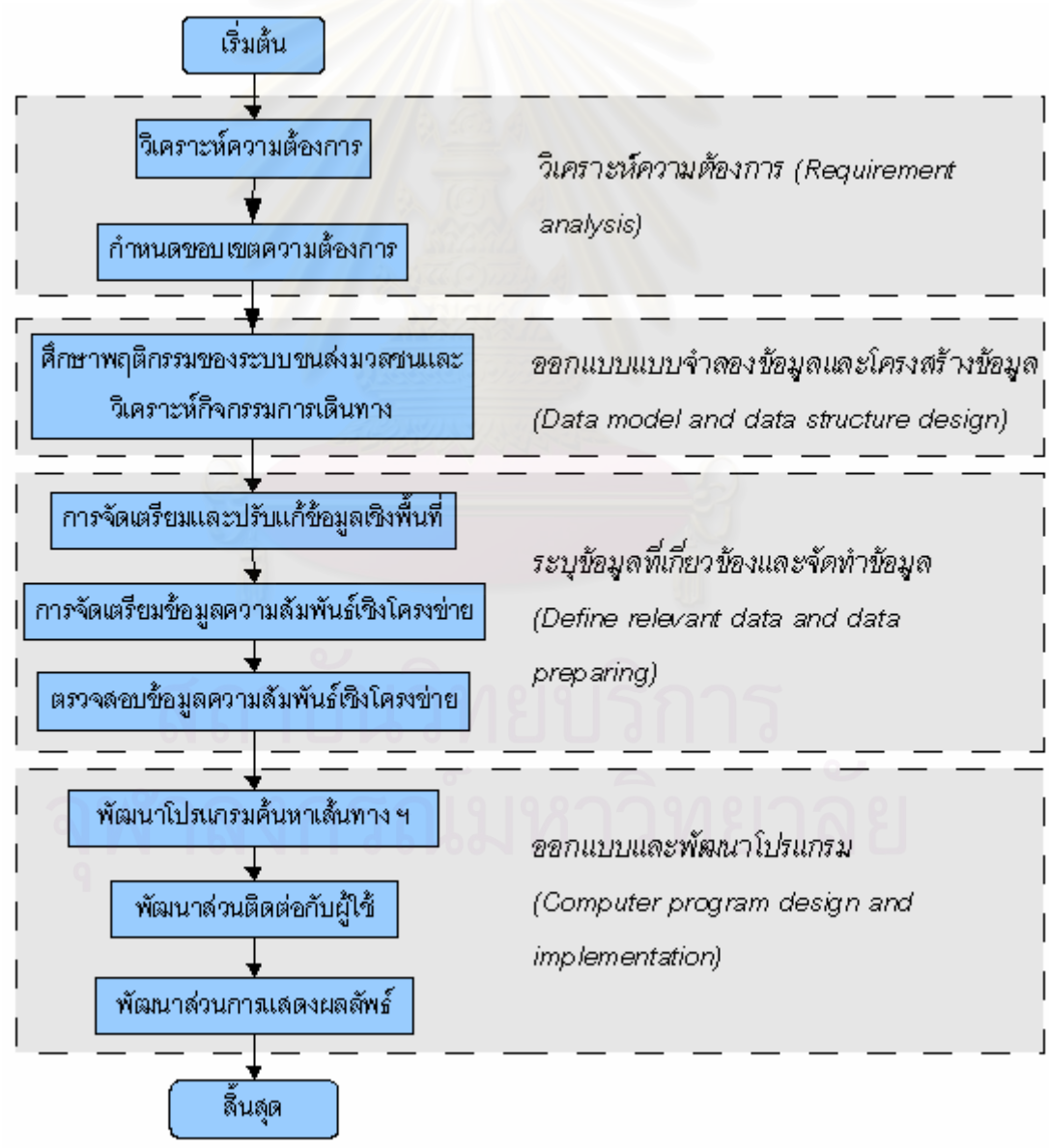
3. กรณีของ Turn ในกรณีของโครงข่ายถนนทั่วๆ ไป จะหมายถึงทางแยกที่ทางเลือกให้ผู้ขับขี่เลือกตัดสินใจว่าจะเลี้ยวซ้าย ขวา ตรงไป ขึ้นหรือลงทางลาด เป็นต้น แต่ในกรณีระบบขนส่งสาธารณะจะหมายถึงความเป็นไปได้ที่ผู้เดินทางจะตัดสินใจเลือกรูปแบบการเดินทาง ณ node ใดๆ ในโครงข่าย
4. Multimodal Network ถือเป็นส่วนหนึ่งของ link ปกติในโครงข่าย (NT_Link) ดังนั้นโครงข่ายทั้งสองจึงมีความสัมพันธ์ต่อกันโดยตรง เช่น เมื่อมีการเปลี่ยนแปลงใดๆ เกิดขึ้นกับโครงข่ายถนนปกติ ความเปลี่ยนแปลงนั้นก็จะส่งผลต่อโครงข่ายระบบขนส่งสาธารณะด้วย เช่นรถโดยสารประจำทาง ซึ่งในวิทยานิพนธ์นี้ยังมิได้ได้จัดทำโปรแกรมหรือกระบวนการตรวจจับความเปลี่ยนแปลงดังกล่าว และยังไม่สามารถปรับแก้โครงข่ายระบบขนส่งสาธารณะแบบอัตโนมัติได้
5. ค่าใช้จ่าย (cost) ที่พิจารณาในวิทยานิพนธ์นี้คือค่าใช้จ่ายในส่วนของเวลา ระยะเวลารอคอย (NT_WaitingTime) และ ระยะเวลาเดินทาง (NS_TravelTime)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
วิธีดำเนินการวิจัย

ขั้นตอนต่างๆ ในการวิจัย

ขั้นตอนต่างๆ ในงานวิจัยแสดงในรูปที่ 3-1 โดยประกอบด้วยการศึกษาพฤติกรรมของระบบขนส่งมวลชน ศึกษาพฤติกรรมการเดินทาง การจัดเตรียมและปรับแก้ข้อมูลเชิงพื้นที่ การจัดเตรียมข้อมูลความสัมพันธ์เชิงโครงข่าย ตรวจสอบข้อมูลความสัมพันธ์เชิงโครงข่าย พัฒนาโปรแกรมค้นหาเส้นทางฯ พัฒนาส่วนติดต่อกับผู้ใช้ และพัฒนาส่วนการแสดงผลลัพธ์ โดยรายละเอียดจะได้กล่าวในส่วนต่อไป



รูปที่ 3-1 ผังขั้นตอนการดำเนินการวิจัย

ขั้นตอนการดำเนินการวิจัยแบ่งเป็นสี่ขั้นตอนย่อยๆ ดังนี้

1. วิเคราะห์ความต้องการ (Requirement analysis)

เป็นการวิเคราะห์ความต้องการของการใช้งานของผู้ใช้ปลายทาง (ผู้สอบถามเส้นทางเดินทาง) ทั้งนี้โดยอาศัยการวิเคราะห์จากวัตถุประสงค์และขอบเขตการวิจัย สามารถสรุปเป็นความต้องการได้ดังนี้

1.1 การรับค่าจากผู้สอบถามเส้นทาง จะเป็นรับผ่านทางแผนที่บนหน้าจอคอมพิวเตอร์

1.2 เส้นทางที่แนะนำแก่ผู้สอบถาม จะเป็นเส้นทางที่ประกอบไปด้วยรูปแบบการเดินทางหลายรูปแบบหรือหลายสายรถประจำทางทั้งนี้ เป็นการผสมผสานกันเพื่อให้ได้เส้นทางที่ใช้เวลาเดินทางน้อยที่สุด

1.3 การเดินจะถูกใช้เป็นรูปแบบการเดินทางที่เชื่อมต่อระหว่างรูปแบบการเดินทางที่แตกต่างกัน

1.4 การสืบค้นเส้นทางจะต้องรองรับการตรวจสอบช่วงที่ให้บริการของระบบขนส่งมวลชนแต่ละประเภทด้วย

1.5 โปรแกรมและโครงสร้างข้อมูลจะต้องเอื้ออำนวยให้ผู้ดูแลระบบสามารถเปลี่ยนแปลงค่าระยะเวลารอคอยและอัตราเร็วเฉลี่ยของยานพาหนะแต่ละรูปแบบในแต่ละเส้นทางย่อย (arc) ได้

1.6 การแสดงผลลัพธ์เส้นทางที่แนะนำจะแสดงทั้งในรูปแผนที่เส้นทางและตาราง

2. ออกแบบแบบจำลองข้อมูลและโครงสร้างข้อมูล (Data model design)

เป็นขั้นตอนการศึกษาและวิเคราะห์พฤติกรรมของระบบขนส่งมวลชนที่สนใจ และวิเคราะห์กิจกรรมต่างๆ ที่เกิดขึ้นระหว่างเดินทาง เพื่อใช้เป็นปัจจัยพื้นฐานสำหรับออกแบบแบบจำลองข้อมูล พร้อมทั้งทำการออกแบบโครงสร้างข้อมูลสำหรับใช้งานร่วมกับโปรแกรมที่พัฒนาขึ้น รายละเอียดในส่วนนี้จะกล่าวถึงในหัวข้อ “*การจำลองสภาพความเป็นจริงให้มาอยู่ในแบบจำลองและโครงสร้างข้อมูลซึ่งเป็นนามธรรม*” “*ผลการศึกษาพฤติกรรมของระบบขนส่งมวลชนและการเดินทาง*” และ “*การออกแบบแบบจำลองและโครงสร้างข้อมูล*”

3. ระบุข้อมูลที่เกี่ยวข้องและจัดทำข้อมูล (Define relevant data and data preparing)

หลังจากได้โครงสร้างข้อมูลแล้ว เป็นขั้นตอนของการจัดทำตัวเนื้อข้อมูลตามโครงสร้างข้อมูลที่ได้ออกแบบไว้ กำหนดค่าตัวแปรต่างๆ ที่จำเป็น การตรวจสอบความ

ถูกต้องของข้อมูลที่จัดทำขึ้น รายละเอียดส่วนนี้กล่าวถึงในหัวข้อ “*ข้อมูลที่เกี่ยวข้อง*” และ “*การจัดเตรียมและปรับแก้ข้อมูล*”

4. ออกแบบและพัฒนาโปรแกรม (Computer program design and implementation)

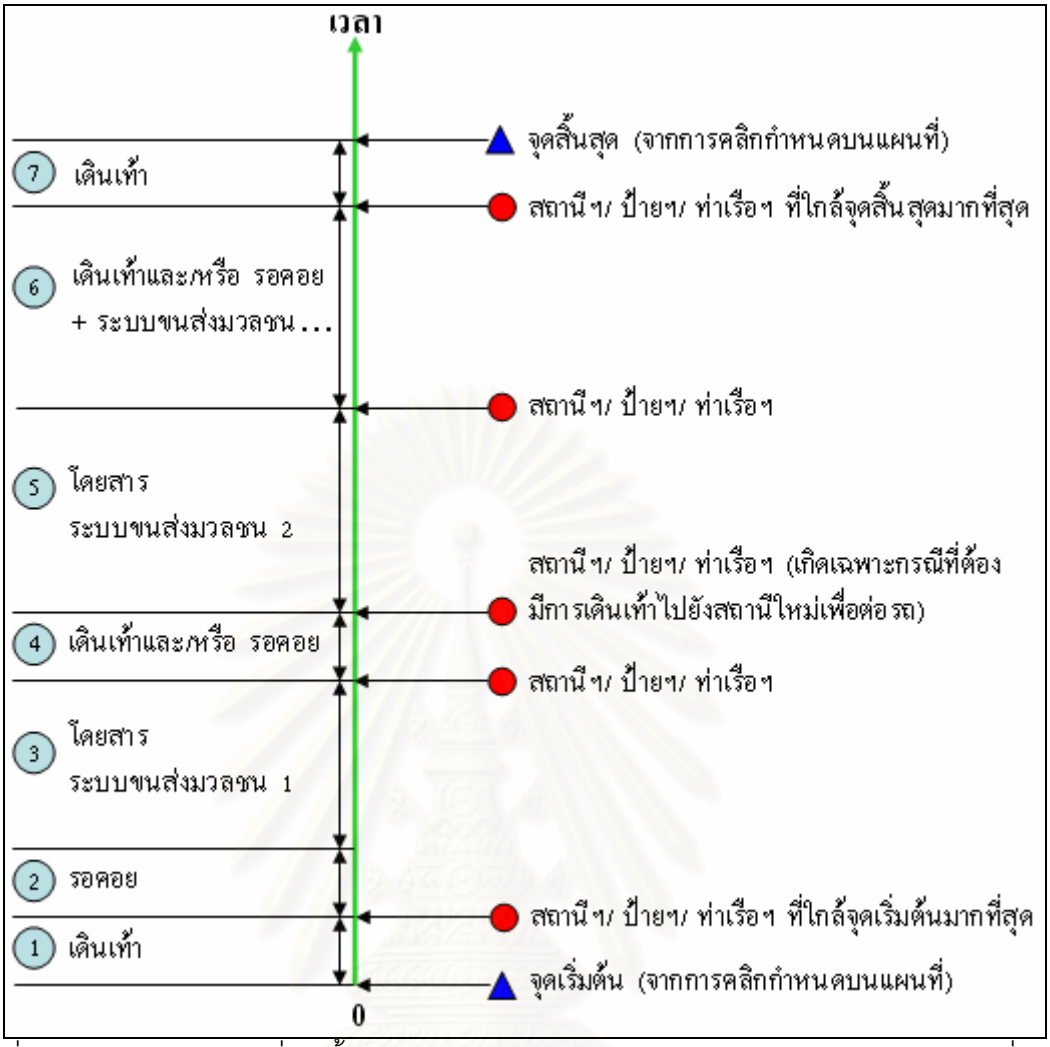
เป็นขั้นตอนการออกแบบอัลกอริทึม และพัฒนาโปรแกรมคอมพิวเตอร์เพื่อสืบค้นเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดบนระบบขนส่งมวลชนหลายรูปแบบ รายละเอียดในส่วนนี้ กล่าวถึงในหัวข้อ “*การพัฒนาโปรแกรมเพื่อคำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด*”

การจำลองสภาพความเป็นจริงให้มาอยู่ในแบบจำลองและโครงสร้างข้อมูลซึ่งเป็นนามธรรม

ในการเดินทางครั้งหนึ่งๆ โดยระบบขนส่งมวลชนนั้นประกอบด้วยกิจกรรมต่างๆ มากมายจนมิอาจจะนำมาบรรยายในที่นี้ได้ทั้งหมด ดังนั้นเพื่อให้สามารถเข้าใจภาพรวมได้โดยง่ายจึงได้เขียนแผนผังแสดงกิจกรรมต่างๆ ที่เกิดขึ้นตั้งแต่เริ่มต้นการเดินทางจนกระทั่งสิ้นสุดการเดินทาง ดังแสดงในรูปที่ 3-2 โดยแสดงกิจกรรมที่เกิดขึ้นในมิติของเวลาและเลือกแสดงเฉพาะกิจกรรมที่อยู่ในอยู่ในแบบจำลองข้อมูลเท่านั้น

แกนตั้งที่เป็นเส้นหนาตรงกลางเป็นแกนของเวลาชี้ขึ้นด้านบน ด้านซ้ายมือของแกน เป็นชื่อของกิจกรรมที่เกิดขึ้นในแต่ละช่วงการเดินทาง พร้อมหมายเลขประจำช่วง แสดงเรียงตามลำดับเหตุการณ์ตามแกนเวลา ส่วนด้านขวามือเป็นตำแหน่งต่างๆ ที่ปรากฏหรือผ่านตามเส้นทางการเดินทาง

จุดวงกลมใช้แทนสถานีรถไฟฟ้า ป้ายหยุดรถโดยสารประจำทาง หรือ ท่าเรือ ซึ่งบางจุดอาจจะไม่ปรากฏ หากช่วงการเดินทางนั้นไม่ต้องมีการเดินเท้าจากสถานีของระบบขนส่งมวลชนหนึ่งไปสู่อีกระบบหนึ่ง กล่าวอีกนัยหนึ่งคือเมื่อลงจากยานพาหนะออกมาที่สถานีฯ ป้ายฯ หรือท่าเรือฯ แล้วสามารถเข้าใช้ระบบขนส่งมวลชนสายต่อไปได้เลย (อาจจะเป็นระบบเดิมหรือเปลี่ยนระบบก็ได้) โดยไม่ต้องมีการเดินเท้าต่อไปยังสถานีฯ ป้ายฯ หรือท่าเรือฯ อื่น เช่น กรณีที่มีการเปลี่ยนสายรถโดยสารประจำทาง โดยใช้ป้ายสุดท้าย ณ ตำแหน่งที่ลงจากรถ เป็นป้ายเริ่มต้นสำหรับการขึ้นรถโดยสารประจำทางอีกสายหนึ่ง ตัวอย่างที่แสดงในรูปที่ 3-2 คือ ช่วงการเดินทางที่ 4 ซึ่งระบุว่าเป็นช่วงที่มีการเดินทางเท้าและ/หรือ รอคอย คือหมายถึงอาจเป็นการเดินเท้าและการรอคอยทั้งสองอย่าง หรืออาจจะเป็นการรอคอยเพียงอย่างเดียวก็ได้



รูปที่ 3-2 ผังแสดงกิจกรรมที่เกิดขึ้นในช่วงการเดินทาง ในมิติของระยะเวลา (เฉพาะกิจกรรมที่อยู่ในกรอบแบบจำลองข้อมูล)

รูปสามเหลี่ยมใช้แทนจุดเริ่มต้นและจุดสิ้นสุดการเดินทาง ในกรณีของงานวิจัยนี้ จุดทั้งสองเกิดจากการคลิกระบุตำแหน่งบนแผนที่โดยอิสระ ไม่จำเป็นต้องเป็นจุดที่มีการลงทะเบียนไว้ล่วงหน้า และโปรแกรมที่พัฒนาขึ้นในงานวิจัยจะทำหน้าที่ค้นหาสถานีฯ ป้ายฯ หรือท่าเรือฯ ที่อยู่ใกล้ที่สุดให้โดยอัตโนมัติเพื่อใช้เป็นจุดเริ่มต้นและจุดสิ้นสุดที่จะเข้าและออกจากโครงข่ายระบบขนส่งมวลชน

ผลการศึกษาพฤติกรรมของระบบขนส่งมวลชนและการเดินทาง

ความแตกต่างระหว่างพฤติกรรมจริงและพฤติกรรมที่พยายามจำลอง

เนื่องจากงานวิจัยนี้เป็นการพยายามจำลองพฤติกรรมการเดินทางและพฤติกรรมของระบบขนส่งมวลชนลงในโครงสร้างข้อมูลในระบบคอมพิวเตอร์ ซึ่งการจำลองนั้นไม่สามารถนำรายละเอียดหรือความเป็นจริงทั้งหมดมาได้ จึงมีความแตกต่างกันระหว่างพฤติกรรมจริงและพฤติกรรมที่จำลองลงในโครงสร้างข้อมูลอยู่ ดังที่จะกล่าวต่อไปนี้

<u>พฤติกรรมจริง</u>

การเดินเท้า

การเดินในที่นี้หมายถึงการเดินเพื่อจะเปลี่ยนรูปแบบจากระบบขนส่งมวลชนระบบหนึ่งไปสู่อีกระบบหนึ่ง การเดินจากจุดใดๆ เพื่อไปยังจุดที่จะเริ่มต้นเข้าสู่ระบบขนส่งมวลชน และรวมถึงการเดินจากสถานีของระบบขนส่งมวลชน ไปยังจุดหมายปลายทางสุดท้าย ซึ่งประกอบด้วย

1. เดินลัดเลาะไปตามทางเท้า
2. เดินขึ้นสะพานลอย ขึ้นสะพานข้ามแม่น้ำหรือคลอง หรือลอดอุโมงค์
3. เดินข้ามถนน
4. เดินภายในพื้นที่สถานีขนส่งมวลชน เช่น สถานีรถไฟฟ้าใต้ดิน หรือลอยฟ้า เพื่อจะไปขึ้นพาหนะหรือออกสู่ถนน
5. เดินลัดเลาะตามตรอกซอกซอย
6. เดินผ่านพื้นที่โล่งกว้างที่มิได้มีการสร้างเส้นทางเดิน หรือสร้างทางเท้าไว้
7. อัตราเร็วในการเดินขึ้นกับแต่ละบุคคล (เพศ วัย บุคคลิก) สภาพอากาศ และสภาพเส้นทาง

รถไฟฟ้าบีทีเอส

รถไฟฟ้าบีทีเอสดำเนินการโดยบริษัท ระบบขนส่งมวลชนกรุงเทพ จำกัด (มหาชน) เป็นรถไฟไฟฟ้า ที่วิ่งบนรางเฉพาะซึ่งถูกสร้างยกระดับสูงจากพื้นถนนตลอดเส้นทาง ในเส้นทางส่วนใหญ่จะเป็นระบบรางคู่ แต่มีเส้นทางบางส่วนเป็นระบบรางเดี่ยว เช่น บริเวณสถานีตากสิน หรือสถานีหมอชิต ซึ่งเป็นสถานีปลายทาง ในกรณีเป็นรางเดี่ยวหมายความว่ารางบริเวณนั้น เป็นการเดินรถไฟสองทางไปและกลับบนรางเดียวกัน กรณีบริเวณที่เป็นรางคู่ ก็จะมีรถไฟสองขบวนวิ่งในทิศทางเดียวในแต่ละรางและเป็นอิสระจากกัน และที่สถานีใดๆ จะมีรถไฟวิ่งในทั้งสองทิศทางเสมอ คือไปและกลับ ยกเว้นสถานีปลายทาง

อัตราเร็วของยานพาหนะจะถูกควบคุมให้เหมาะสมโดยระบบคอมพิวเตอร์ และค่อนข้างใกล้เคียงกันตลอดเส้นทาง ประมาณ 35 กิโลเมตรต่อชั่วโมง ทำให้สามารถคำนวณระยะเวลาเดินทางได้ค่อนข้างแน่นอน

ความถี่ของการปล่อยรถจะแปรผันตามช่วงเวลา โดยในชั่วโมงเร่งด่วนจะมีการปล่อยรถถี่กว่าช่วงเวลาปกติ และยังแปรผันไปตามแต่สายรถไฟฟ้าและแต่ละวันของสัปดาห์ ทำให้ระยะเวลารอคอยแปรผันตามในอัตราเดียวกัน ช่วงเวลาให้บริการคือ 06:00 น. - 24:00 น. (บริษัทระบบขนส่งมวลชนกรุงเทพ จำกัด (มหาชน), 2550) รายละเอียดการให้บริการรถไฟฟ้าบีทีเอสแสดงอยู่ในภาคผนวก ค

รถไฟฟ้ามหานคร (รถไฟใต้ดิน)

รถไฟฟ้ามหานคร ดำเนินการโดยการรถไฟฟ้าขนส่งมวลชนแห่งประเทศไทย หรือ รฟม. รถไฟฟ้ามหานคร เป็นรถไฟไฟฟ้า ที่วิ่งไปบนรางที่วางตัวอยู่ใต้ดิน โดยจะเป็นสองรางแยกกันเสมอ ทำให้ที่สถานีใดๆ จะมีรถวิ่งในทั้งสองทิศทางเป็นอิสระจากกัน ระยะการเดินระหว่างสถานีของระบบขนส่งมวลชนอื่นกับสถานีของระบบขนส่งมวลชนชนิดนี้ค่อนข้างยาวกว่าคู่การเปลี่ยนรูปแบบการเดินทางอื่นๆ ทั้งนี้เนื่องจากตัวสถานีจะอยู่ลึกลงไปใต้ดินมาก

อัตราเร็วของยานพาหนะจะถูกควบคุมให้เหมาะสมโดยระบบคอมพิวเตอร์ จะค่อนข้างใกล้เคียงกันตลอดเส้นทาง โดยเฉลี่ยประมาณ 35 กิโลเมตรต่อชั่วโมง (การรถไฟฟ้าขนส่งมวลชนแห่งประเทศไทย, -) ทำให้สามารถคำนวณระยะเวลาเดินทางได้ค่อนข้างแน่นอน

ความถี่ของการปล่อยรถจะแปรผันตามช่วงเวลา โดยในชั่วโมงเร่งด่วนจะมีปล่อยรถถี่ (2 - 4 นาที/ขบวน) กว่าช่วงเวลาปกติ (4 - 6 นาที/ขบวน) ทำให้ระยะเวลารอคอยแปรผันตามในอัตราเดียวกัน ช่วงเวลาให้บริการคือ 05:00 น. - 24:00 น.

รถโดยสารประจำทาง (รถเมล์)

รถโดยสารประจำทางในที่นี้หมายถึงรถโดยสารประจำทางขนาดใหญ่ที่ให้บริการโดยองค์การขนส่งมวลชนกรุงเทพ ไม่รวมรถโดยสารขนาดเล็กและรถตู้

รถโดยสารประจำทางจะมีการเส้นทางเดินรถที่แน่นอน แยกออกเป็นสายและมีหมายเลขประจำสาย พฤติกรรมหรือคุณสมบัติของรถโดยสารประจำทางตามข้อเท็จจริงที่ได้ศึกษาและวิเคราะห์ มีดังนี้

1. รถโดยสารประจำทางจะวิ่งเป็นทิศทางเดียวเสมอ โดยแยกฝั่งกันสำหรับถนนที่มีการเดินรถสองทิศทาง หากเป็นถนนที่เดินรถทางเดียวก็จะวิ่งเป็นทิศทางเดียว ตำแหน่งป้าย

หยุดรถประจำทางมักจะติดตั้งอยู่บริเวณริมทางเท้า อาจมีศาลาหรือที่พักผู้โดยสารบ้างในบางป้าย

2. รถโดยสารประจำทางบางประเภทอาจไม่เข้าจอดรับส่งผู้โดยสารทุกป้าย เช่นรถโดยสารประจำทางด่วนพิเศษจะเข้าจอดเฉพาะป้ายที่มีการกำหนดไว้เท่านั้น (จากการสุ่มสำรวจโดยผู้วิจัย)
3. ทุกครั้งที่เกิดการตัดกันหรือซ้อนทับกันของสายรถโดยสารประจำทางมากกว่าหนึ่งสาย ก็จะเกิดโอกาสที่ผู้เดินทางสามารถลงจากรถสายเดิมและเดินไปยังป้ายใหม่เพื่อเปลี่ยนสายได้ หรืออาจไม่ต้องเดิน หากป้ายที่ลงนั้นเป็นป้ายที่เป็นจุดเริ่มการแยกกันของสายการเดินรถโดยสารประจำทาง
4. การเปลี่ยนสายรถประจำทางสามารถทำที่ป้ายใดก็ได้ ไม่จำเป็นต้องเป็นป้ายแรกที่เป็นจุดตัดหรือจุดเริ่มต้นซ้อนทับกันของเส้นทางเดินรถสองสาย
5. ช่วงเวลาการให้บริการแต่ละสายจะไม่เหมือนกัน กล่าวคือ บางสายให้บริการตลอด 24 ชั่วโมง บางสายให้บริการเป็นช่วงเวลา สายที่ให้บริการตลอด 24 ชั่วโมง เช่น สาย 4 สาย 9 (องค์การขนส่งมวลชนกรุงเทพ, 2549)
6. อัตราเร็วของรถโดยสารประจำทางอยู่ที่ 8-17 กิโลเมตรต่อชั่วโมงในช่วงวันทำงาน และ 10-20 กิโลเมตรต่อชั่วโมงสำหรับช่วงวันหยุด (ชรัด พิริยะวัฒน์, 2543)
7. ความถี่ของการปล่อยรถหรือระยะเวลารอคอยจะไม่มีความแน่นอน โดยในชั่วโมงเร่งด่วนแม้จะมีการปล่อยรถมาก แต่เนื่องจากการจราจรติดขัด ทำให้ระยะเวลารอคอยไม่ได้สั้นลง หรืออาจยาวนานขึ้นด้วยซ้ำ เนื่องจากอัตราเร็วของรถต่ำมาก ในขณะที่นอกช่วงเวลาเร่งด่วนแม้จะมีปริมาณรถวิ่งในระบบน้อยแต่อัตราเร็วของรถก็สูง ทำให้ช่วยชดเชยกับปริมาณรถที่น้อยลง นอกจากนี้หากการจราจรติดขัดมากทำให้ไม่มีรถวิ่งกลับเข้ามาที่จุดปล่อย ก็จะยังไม่สามารถปล่อยรถได้ แม้จะครบกำหนดเวลาปล่อยรถก็ตาม

เรือด่วนตามแม่น้ำเจ้าพระยา

เรือด่วนตามแม่น้ำเจ้าพระยาให้บริการโดยบริษัทเรือด่วนเจ้าพระยา จำกัด เป็นการเดินเรือเลียบไปตามสองฝั่งแม่น้ำเจ้าพระยา มีมีเรือหลายประเภท ได้แก่ เรือด่วนพิเศษ (ธงเหลือง) สาย นนทบุรี-สาทร นนทบุรี-ราษฎร์บูรณะ เรือด่วนพิเศษ (ธงฟ้า) สาย นนทบุรี-สาทร เรือด่วนพิเศษ (ธงส้ม) สายนนทบุรี-วัดราชสิงขร และเรือด่วนประจำทางสายนนทบุรี-วัดราชสิงขร (บริษัท เรือด่วนเจ้าพระยา จำกัด, 2549)

เส้นทางเรือที่นำมาใช้ในงานวิจัยเป็นบางส่วนของเส้นทางเรือด่วนประจำทางสายนนทบุรี-วัดราชสิงขร โดยเริ่มต้นตั้งแต่ท่าโอเรียนเต็ลจนถึงท่าสะพานกรุงธน

พฤติกรรมหรือคุณสมบัติของเรือด่วนตามแม่น้ำเจ้าพระยาตามข้อเท็จจริงที่ได้ศึกษาและวิเคราะห์ มีดังนี้

1. ระยะเวลารอคอยมีความผันแปรน้อยกว่ารถโดยสารประจำทาง และมีการประกาศตารางหรือเกณฑ์การปล่อยเรือไว้ในเว็บไซต์ของบริษัท รายละเอียดแสดงอยู่ในภาคผนวก ง
2. อัตราเร็วแม้ไม่ไม่มีการควบคุมโดยระบบคอมพิวเตอร์เหมือนรถไฟฟ้า แต่ก็ถือว่าค่อนข้างคงที่ เนื่องจาก การจราจรทางน้ำเบาบางมาก เกือบจะถือได้ว่าการเดินเรือมีทางแล่นเฉพาะของตนเองแยกต่างหากเหมือนระบบราง อีกทั้งผู้ควบคุมเรือมีความชำนาญควบคุมให้อัตราเร็วค่อนข้างสม่ำเสมอ
3. การขึ้นเรือจะต้องขึ้นท่าที่เหมาะสมด้วย คือเป็นท่าที่เรือชนิดนั้นๆ เข้าจอดรับส่งผู้โดยสาร จึงต้องมีการศึกษาตำแหน่งท่าเรือและระบบการเดินเรือให้ดีก่อนเข้าใช้บริการ

เรือข้ามฟาก

เรือข้ามฟากหมายถึงเรือที่ใช้ข้ามฟากระหว่างสองฝั่งของแม่น้ำเจ้าพระยา ผู้ดูแลบริหารและให้บริการจะเป็นบริษัทเอกชน โดยกรมเจ้าท่าอาจจัดเจ้าหน้าที่เข้ามาช่วยกำกับดูแลเรื่องความปลอดภัยบ้างในท่าที่มีการใช้บริการหนาแน่น ทั้งนี้อาจมีทุ่นหรือเรือข้ามฟากอื่นอีกซึ่งใช้ข้ามฝั่งคลอง เช่น คลองแสนแสบจะไม่ขอกล่าวถึงในงานวิจัยนี้

พฤติกรรมหรือคุณสมบัติของเรือข้ามฟากตามข้อเท็จจริงที่ได้ศึกษาและวิเคราะห์ มีดังนี้

1. จำนวนเรือข้ามฟากที่ให้บริการต่อหนึ่งคู่ของฟากนั้นจะแปรผันตามเวลา คือ ในชั่วโมงเร่งด่วนอาจมีเรือให้บริการถึง 3 ลำ ส่วนในชั่วโมงปกติจะมีให้บริการ 1-2 ลำ (ข้อมูลจากการสำรวจในพื้นที่ ณ ท่าเรือสี่พระยา-คลองสาน)
2. อัตราเร็วของเรือข้ามฟากค่อนข้างคงที่ เนื่องจากมีพื้นที่ของช่องจราจรขนาดใหญ่และยืดหยุ่นมาก อีกทั้งปริมาณการจราจรทางน้ำถือว่าน้อยมากเมื่อเทียบกับทางบก
3. อัตราเร็วของเรืออาจลดลงโดยความตั้งใจของผู้ควบคุมเรือ เนื่องจากต้องรอให้เรืออีกลำที่ประจำอยู่ที่ท่าเป้าหมายออกไปก่อน บางครั้งจึงมีการชะลอเรืออยู่กลางแม่น้ำ และบางครั้งยังอาจต้องชะลอเรืออยู่กลางแม่น้ำเพื่อให้เรือขนส่งสินค้าซึ่งแล่นตามแนวลำน้ำแล่นผ่านไปก่อน
4. ระยะเวลารอคอยจะสั้นในชั่วโมงเร่งด่วน และจะยาวนานขึ้นในชั่วโมงปกติ เนื่องจากในชั่วโมงเร่งด่วนแม้จะมีผู้ใช้บริการมากแต่ก็มีจำนวนเรือมากแล่นสับเปลี่ยนกันเข้า

มารับผู้โดยสารถี่กว่าปกติ อีกทั้งระยะเวลาจอดรอให้ผู้โดยสารขึ้นเต็มลำเรือก็จะสั้นกว่าปกติด้วย

<u>พฤติกรรมที่จำลอง หรือกรอบที่ทำให้ง่ายลง</u>

<u>รายละเอียดพฤติกรรมการเดินทางที่จำลองหรือทำให้ง่ายลง</u>

การเดินเท้า

1. การสร้างเส้นทางเดินเชื่อมระหว่างระบบขนส่งมวลชนสองระบบใดๆ จะใช้เส้นตรงต่อเชื่อมระหว่างสองสถานีหรือสองจุดใดๆ หรืออาจจะวางตัวใกล้เคียงกับแนวถนน และตั้งสมมติฐานว่าเป็นการเดินในแนวราบเท่านั้น
2. ไม่มีการจัดทำแผนที่หรืออธิบายเส้นทางเดินในรายละเอียดให้กับผู้ใช้
3. อัตราเร็วเฉลี่ยของการเดินใช้อยู่ที่ 5 กิโลเมตรต่อชั่วโมง (ค่าโดยปริยายเริ่มต้น)
4. ทุกเส้นทางเดินสามารถเดินผ่านได้ตลอด 24 ชั่วโมง
5. ทุกครั้งที่มีการสอบถามเส้นทางจะมีการสร้างเส้นทางเดินเสมือนขึ้นมาชั่วคราวสองเส้นทางเสมอ คือเส้นทางที่จะเดินจากจุดเริ่มต้นใดๆ เพื่อเริ่มต้นเข้าสู่ระบบขนส่งมวลชน และเส้นทางเดินออกจากระบบขนส่งมวลชน เพื่อไปยังจุดหมายปลายทางสุดท้าย ทั้งนี้ได้กำหนดระยะการเดินเท้ายาวที่สุดที่ยอมรับได้คือ 300 เมตร เป็นค่าตัวแปรอยู่ในโปรแกรมฯ

รถโดยสารประจำทาง

1. ถือว่าสายรถทุกสายจอดทุกป้ายที่รถฯ วิ่งผ่านไม่ขึ้นกับประเภทหรือสายรถโดยสารประจำทาง
2. ยังมิได้คำนึงถึงช่วงเวลาการให้บริการ แยกเป็นสาย โดยมีสมมติว่าทุกสายบริการในช่วงเวลาเดียวกัน
3. กรณีการเปลี่ยนสายรถ การเปลี่ยนสายรถจะถูกนำไปพิจารณาโดยโปรแกรมฯ ก็ต่อเมื่อมีการสร้างเส้นทางเดินเชื่อมต่อระหว่างสาย (ป้าย) เอาไว้ล่วงหน้า ซึ่งอาจมีความเป็นไปได้ของคู่ป้ายมากกว่าหนึ่งคู่ในบริเวณเดียวกัน หรือมีเส้นทางเดินที่เป็นไปได้มากกว่าหนึ่งเส้นทาง ในงานวิจัยนี้อาจไม่ได้สร้างเส้นทางดังกล่าวเตรียมไว้ครบทั้งหมด โดยผู้จัดทำข้อมูลได้พยายามเลือกคู่ของป้ายหรือเส้นทางเดินเท้าที่สั้นที่สุด (ในส่วนนี้มีประเด็นการอภิปรายอยู่ในบทที่ 5 หัวข้ออภิปรายผลการวิจัย)

4. ป้ายหยุดรถโดยสารประจำทาง ทำหน้าที่เป็น vertex ส่วน arc คือเส้นที่เชื่อมระหว่างป้าย ดังนั้น vertex และ arc ของรถที่วิ่งในทิศทางตรงข้ามกัน (คนละฝั่งถนน) จะถูกบริหารจัดการเป็นอิสระจากกัน ทำให้การเปลี่ยนแปลง เลื่อน เพิ่ม ลด ป้าย สามารถทำได้อย่างอิสระ หมายความว่าที่หนึ่งคู่ของ vertex ใดๆ จะมีเพียงหนึ่ง arc เชื่อมอยู่และสามารถเดินทางผ่านได้เพียงทิศทางเดียว
5. อัตราเร็วเฉลี่ยของแต่ละ arc ถูกจัดเก็บแยกต่างหากจากกัน เพื่อให้สามารถรองรับการกำหนดอัตราเร็วเฉลี่ยของแต่ละ arc ให้เหมาะสมกับสภาพการจราจรแต่ละพื้นที่ได้ อัตราเร็วเฉลี่ยโดยปริยายที่ใช้คือ 20 กิโลเมตรต่อชั่วโมง (ค่าโดยปริยายเริ่มต้น)
6. ช่วงเวลาที่สามารถเดินทางผ่านแต่ละ arc จะถูกจัดเก็บแยกต่างหากจากกัน เพื่อให้สามารถกำหนดช่วงเวลาให้บริการของแต่ละ arc ให้เหมาะสมกับสภาพการจราจรแต่ละพื้นที่ได้ เช่นในกรณีที่มีการปิดช่องทางจราจร การก่อสร้าง หรือ อุบัติเหตุ (ช่วงเวลาให้บริการนี้สามารถใช้กำหนดเป็นช่วงเวลาให้บริการของรถโดยสารประจำทางได้ด้วย)
7. ระยะเวลารอคอยใช้ระยะเวลารอคอยที่ครึ่งหนึ่งของระยะเวลารอคอยที่นานที่สุด คือ 9 นาที (จากการทดลองสุ่มสำรวจระยะเวลารอคอยนานสุดคือ 18 นาที) ใช้เป็นค่าโดยปริยายเริ่มต้นสำหรับทุก arc และสายรถทุกสายใช้ค่าเดียวกัน
8. ช่วงเวลาให้บริการคือ 05:00 น. – 23:00 น.

รถไฟฟ้าบีทีเอส

1. ทุกๆ arc ที่เชื่อมระหว่างสถานี (vertex) จัดทำเป็นสองทิศทางทั้งหมดเนื่องจากจุดขึ้นรถไฟฟ้าทั้งสองฝั่งถูกสร้างและบริหารจัดการภายในสถานีเดียวกันอยู่แล้ว หมายความว่าที่หนึ่งคู่ของ vertex ใดๆ จะมีสอง arc เชื่อมอยู่ โดยมีทิศการเดินทางตรงข้ามกัน
2. อัตราเร็วเฉลี่ยของรถไฟฟ้าบีทีเอส จะมิได้ขึ้นกับสภาพการจราจร ดังนั้นอัตราเร็วเฉลี่ยในแต่ละ arc จึงถือได้ว่าคงที่ แต่ในทางปฏิบัติก็ยังได้จัดเก็บแยกต่างหากจากกันอยู่ (เป็นผลพลอยได้จากการใช้โครงสร้างเดียวกับกรณีรถโดยสารประจำทาง)
3. ช่วงเวลาที่สามารถเดินทางผ่านแต่ละ arc จะถูกจัดเก็บแยกต่างหากจากกัน เพื่อให้สามารถกำหนดช่วงเวลาให้บริการของแต่ละ arc ให้เป็นอิสระจากกันได้ (เป็นผลพลอยได้จากการใช้โครงสร้างเดียวกับกรณีรถโดยสารประจำทาง)
4. อัตราเร็วที่ใช้เป็นอัตราเร็วเฉลี่ยคือที่ 35 กิโลเมตรต่อชั่วโมงเป็นค่าโดยปริยายเริ่มต้นสำหรับทุก arc (ภาคผนวก ค)

5. ระยะเวลารอคอยใช้ระยะเวลารอคอยที่ครึ่งหนึ่งของระยะเวลารอคอยที่นานที่สุด คือ 4 นาที (คำนวณจากอัตราการปล่อยรถที่ 8 นาทีต่อขบวน, ภาคผนวก ค) ใช้เป็นค่าโดยปริยายเริ่มต้นสำหรับทุก arc
6. ช่วงเวลาให้บริการคือ 05:00 น. - 24:00 น.

รถไฟฟ้ามหานคร (รถไฟใต้ดิน)

1. ทุกๆ arc ที่เชื่อมระหว่างสถานี (vertex) จัดทำเป็นสองทิศทางทั้งหมดเนื่องจากจุดขึ้นรถไฟฟ้าทั้งสองฝั่งถูกสร้างและบริหารจัดการภายในสถานีเดียวกันอยู่แล้ว หมายความว่าที่ 1 คู่ของ vertex ใดๆ จะมี 2 arc เชื่อมอยู่ โดยมีทิศการเดินทางตรงข้ามกัน
2. อัตราเร็วเฉลี่ยของรถไฟฟ้ามหานคร มิได้ขึ้นกับสภาพการจราจร ดังนั้นอัตราเร็วเฉลี่ยในแต่ละ arc จึงถือได้ว่าคงที่ แต่ในทางปฏิบัติก็ยังได้จัดเก็บแยกต่างหากจากกันอยู่ (เป็นผลพลอยได้จากการใช้โครงสร้างเดียวกับกรณีรถโดยสารประจำทาง)
3. ช่วงเวลาที่สามารถเดินทางผ่านแต่ละ arc จะถูกจัดเก็บแยกต่างหากจากกัน เพื่อให้สามารถกำหนดช่วงเวลาให้บริการของแต่ละ arc ให้เป็นอิสระจากกันได้ (เป็นผลพลอยได้จากการใช้โครงสร้างเดียวกับกรณีรถโดยสารประจำทาง)
4. อัตราเร็วที่ใช้เป็นอัตราเร็วเฉลี่ยคือที่ 35 กิโลเมตรต่อชั่วโมงเป็นค่าโดยปริยายเริ่มต้นสำหรับทุก arc
5. ระยะเวลารอคอยใช้ระยะเวลารอคอยที่ครึ่งหนึ่งของระยะเวลารอคอยที่นานที่สุด คือ 4 นาที (คำนวณจากอัตราการปล่อยรถที่ 8 นาทีต่อขบวน) ใช้เป็นค่าโดยปริยายเริ่มต้นเดียวกันสำหรับทุก arc
6. ช่วงเวลาให้บริการคือ 05:00 น. - 24:00 น.

เรือข้ามฟาก

1. ทุกๆ arc ที่เชื่อมระหว่างท่าเรือ (vertex) จัดทำเป็นสองทิศทางทั้งหมด หมายความว่าที่ 1 คู่ของ vertex ใดๆ จะมี 2 arc เชื่อมอยู่ โดยมีทิศการเดินทางตรงข้ามกัน
2. อัตราเร็วเฉลี่ยของเรือข้ามฟาก แม้มิได้ผูกติดกับปริมาณการจราจรมากนักเนื่องจากปริมาณการจราจรทางน้ำมีน้อยและช่องจราจรกว้าง เรือสามารถแล่นลัดเลี้ยวได้อิสระ แต่ผู้ให้บริการก็มิได้กำหนดอัตราเร็วไว้ชัดเจน อย่างไรก็ตามอัตราเร็วเฉลี่ยของแต่ละ arc ยังคงถูกจัดเก็บแยกต่างหากจากกัน (เป็นผลพลอยได้จากการใช้โครงสร้างเดียวกับกรณีรถโดยสารประจำทาง)

3. อัตราเร็วที่ใช้เป็นอัตราเร็วเฉลี่ยคือที่ 15 กิโลเมตรต่อชั่วโมง เป็นค่าปริยายเริ่มต้นสำหรับทุก arc (โดยการกะประมาณและทดลองเดินทางโดยผู้วิจัย)
4. ช่วงเวลาที่สามารถเดินทางผ่านแต่ละ arc ได้ ถูกจัดเก็บแยกต่างหากจากกัน (เป็นผลพลอยได้จากการใช้โครงสร้างเดียวกับกรณีรถโดยสารประจำทาง)
5. ระยะเวลารอคอยใช้ระยะเวลารอคอยที่ครึ่งหนึ่งของระยะเวลารอคอยที่นานที่สุด คือ 6 นาที (คำนวณจากอัตราการปล่อยเรือที่ 12 นาทีต่อลำ) เป็นค่าปริยายเริ่มต้นสำหรับทุก arc
6. ช่วงเวลาให้บริการคือ 05:00 น. – 23:00 น.

เรือด่วนตามแม่น้ำเจ้าพระยา

1. ทุกๆ arc ที่เชื่อมระหว่างท่าเรือ (vertex) จัดทำเป็นสองทิศทางทั้งหมด หมายความว่าที่ 1 คู่ของ vertex ใดๆ จะมี 2 arc เชื่อมอยู่ โดยมีทิศการเดินทางตรงข้ามกัน
2. อัตราเร็วเฉลี่ยของเรือข้ามฟาก แม้จะมิได้ผูกติดกับปริมาณการจราจรมากนัก แต่ผู้ให้บริการก็มิได้กำหนดไว้ชัดเจน แต่อย่างไรก็ตามอัตราเร็วเฉลี่ยของแต่ละ arc ถูกจัดเก็บแยกต่างหากจากกัน (เป็นผลพลอยได้จากการใช้โครงสร้างเดียวกับกรณีรถโดยสารประจำทาง)
3. อัตราเร็วที่ใช้เป็นอัตราเร็วเฉลี่ยคือที่ 30 กิโลเมตรต่อชั่วโมง เป็นค่าปริยายเริ่มต้นสำหรับทุก arc (โดยการกะประมาณและทดลองเดินทางโดยผู้วิจัย)
4. ช่วงเวลาที่สามารถเดินทางผ่านแต่ละ arc ถูกจัดเก็บแยกต่างหากจากกัน (เป็นผลพลอยได้จากการใช้โครงสร้างเดียวกับกรณีรถโดยสารประจำทาง)
5. ระยะเวลารอคอยใช้ระยะเวลารอคอยที่ครึ่งหนึ่งของระยะเวลารอคอยที่นานที่สุด คือ10 นาที (ภาคผนวก ง) ใช้ค่าเดียวกันทั้งหมดสำหรับทุก arc
6. ช่วงเวลาให้บริการคือ 06:00 น. – 18:00 น.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

การออกแบบแบบจำลองและโครงสร้างข้อมูล

คำศัพท์เฉพาะที่ใช้ในงานวิจัย เพื่อใช้อธิบายแบบจำลอง โครงสร้างข้อมูลและพฤติกรรมของระบบขนส่งมวลชน ในรูปแบบความสัมพันธ์เชิงโครงข่าย แสดงสรุปคำจำกัดความของศัพท์ในตารางที่ 3-1 ซึ่งต่อไปนี้จะใช้คำเหล่านี้ในการอธิบายโครงสร้างข้อมูลและพฤติกรรมของระบบขนส่งมวลชน

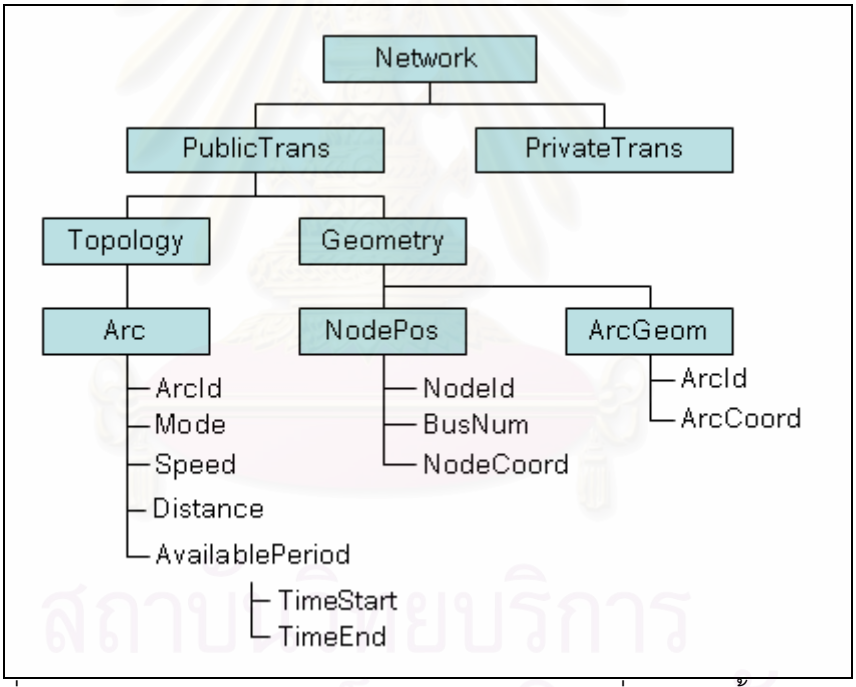
ตาราง 3-1 ความหมายศัพท์เฉพาะที่ใช้ในโครงสร้างข้อมูลและแทนพฤติกรรมของระบบขนส่งมวลชน

คำศัพท์	ความหมาย
vertex	จุด ซึ่งมองในแง่ความสัมพันธ์เชิงโครงข่ายทำหน้าที่เป็นจุดเชื่อมต่อของเส้นทางในโครงข่าย ในที่นี้หมายถึง ป้ายหยุดรถโดยสารประจำทาง สถานีรถไฟฟ้า สถานีรถไฟฟ้าใต้ดิน และท่าเรือ
arc	เส้นทาง (เชิงตรรกะ) ที่เชื่อมระหว่าง vertex หาก vertex คู่ใดมี arc เชื่อมระหว่างกัน เท่ากับเป็นการระบุว่าสอง vertex นั้น สามารถเดินทางถึงกันได้
source	Vertex ที่ทำหน้าที่เป็น จุดต้นทางของ arc
target	Vertex ที่ทำหน้าที่เป็น จุดปลายทางของ arc
distance	ระยะทางของเส้นทางทางกายภาพ ที่เชื่อมระหว่าง source กับ target และระยะทางของเส้นทางทางกายภาพนี้จะใช้เป็นค่าสำหรับกำหนดคุณสมบัติให้กับระยะทางของ arc ด้วย
speed	อัตราเร็วของยานพาหนะที่สามารถเคลื่อนผ่านเส้นทาง (ทางกายภาพ) ที่เชื่อมระหว่าง source กับ target
cost	ค่าใช้จ่ายในการเดินทางผ่าน arc ในที่นี้คือระยะเวลาที่ใช้ โดยคำนวณจากระยะทาง (distance) หารด้วยอัตราเร็ว (speed)
transport_mode	รูปแบบระบบขนส่งมวลชนที่ให้บริการสำหรับ arc ใดๆ ได้แก่ รถโดยสารประจำทาง (bus), รถไฟฟ้าบีทีเอส (bts), รถไฟฟ้ามหานคร (mrt), เรือข้ามฟาก (ferry), เรือด่วนตามแม่น้ำเจ้าพระยา (boat) และการเดินเท้า (walk)
time_start	ระยะเวลาเริ่มต้นการให้บริการของระบบขนส่งมวลชนที่ให้บริการใน arc ใดๆ
time_end	ระยะเวลาสิ้นสุดการให้บริการของระบบขนส่งมวลชนที่ให้บริการใน arc ใดๆ
bus_num_start	หมายเลขสายรถประจำทางที่ผ่าน source vertex เฉพาะกรณี vertex ที่แทนตำแหน่งป้ายหยุดรถโดยสารประจำทาง
bus_num_end	หมายเลขสายรถประจำทางที่ผ่าน target vertex เฉพาะกรณี vertex ที่แทนตำแหน่งป้ายหยุดรถโดยสารประจำทาง
bus_num	หมายเลขสายรถประจำทางที่ผ่าน arc ซึ่งเชื่อมระหว่าง source เฉพาะกรณีเป็น arc ที่เชื่อมระหว่าง vertex ซึ่งแทนตำแหน่งป้ายหยุดรถโดยสารประจำทาง

แบบจำลองข้อมูลที่ได้ออกแบบขึ้นในงานวิจัยนี้แสดงในรูปที่ 3-3 เริ่มต้นจาก Network ซึ่งใช้แทนโครงข่ายการคมนาคมขนส่งทั้งหมด แล้วจึงมองแยกออกเป็นโครงข่ายระบบขนส่งมวลชน (PublicTrans) และ ระบบขนส่งส่วนตัว (PrivateTrans)

ระบบขนส่งมวลชน แยกจำลองเป็นข้อมูลความสัมพันธ์เชิงโครงข่าย (Topology) และข้อมูลเชิงเรขาคณิต (Geometry)

ข้อมูลความสัมพันธ์เชิงโครงข่ายเป็นการระบุถึงความสัมพันธ์ของ arc ต่างๆ ในโครงข่าย โดยแต่ละ arc จะมีอรรถาธิบายคือ รหัสประจำ arc (ArcId) ซึ่งเป็นเอกลักษณ์ รูปแบบการเดินทาง (Mode) ที่มีอยู่บน arc นั้น (1 arc จะมีได้เพียงหนึ่งรูปแบบ) อัตราเร็วเฉลี่ยของยานพาหนะที่วิ่งบน arc นั้น (Speed) ระยะทางของเส้น arc (Distance) และช่วงเวลาที่รูปแบบการขนส่งใน arc นั้นเปิดให้บริการ (AvailablePeriod) โดยแยกออกเป็น เวลาเริ่มต้นให้บริการ (TimeStart) และเวลาสิ้นสุดให้บริการ (TimeEnd)



รูปที่ 3-3 แบบจำลองข้อมูลสำหรับระบบขนส่งสาธารณะที่ออกแบบขึ้นในงานวิจัย

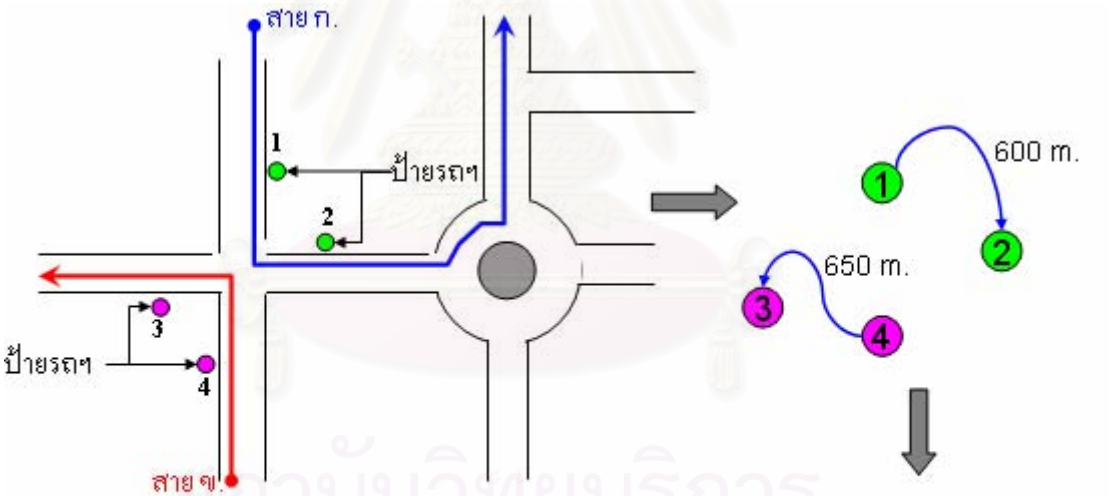
ข้อมูลเรขาคณิตแบ่งออกเป็นข้อมูลเรขาคณิตของ node (NodePos) และข้อมูลเรขาคณิตของ arc (ArcGeom)

ข้อมูลเรขาคณิตของ node ประกอบด้วยอรรถาธิบาย รหัสประจำ node (NodeId) ซึ่งต้องเป็นเอกลักษณ์ หมายเลขสายรถประจำทางที่ผ่าน node (BusNum) ซึ่งหากเป็น node ของระบบขนส่งมวลชนประเภทอื่น อรรถาธิบายตัวนี้จะเว้นว่างไว้ และค่าพิกัดตำแหน่งของ node บนแผนที่ (NodeCoord) ซึ่งจัดเก็บในรูปแบบพิกัดจุดค่า x,y

ข้อมูลเรขาคณิตของ arc ประกอบด้วยอรรถาธิบาย รหัสประจำ arc (ArcId) และคำพิกัดเส้น arc (ArcCoord) ซึ่งจัดเก็บเป็นข้อมูลประเภท Line String

การออกแบบแบบจำลองข้อมูลเริ่มต้นจากการจำลองสภาพความเป็นจริงให้ง่ายลงแล้วแทนคุณสมบัติและพฤติกรรมต่างๆ ด้วยโครงสร้างของ Network จากนั้นจึงทำการออกแบบแบบจำลองสำหรับการจัดเก็บข้อมูลต่างของ Network ขั้นตอนดังกล่าวสามารถแสดงให้เห็นความสัมพันธ์โดยสังเขปได้ในรูปที่ 3-4 โดยรูปบนซ้ายเป็นภาพแสดงเส้นทางเดินรถโดยสารประจำทางบนถนนที่ทางแยก มีตำแหน่งป้ายหยุดรถโดยสารประจำทางสี่ป้าย และสายรถสองสายวิ่งในสองทิศทาง

รูปบนขวาแสดงการจำลองความสัมพันธ์ของป้ายหยุดรถโดยสารประจำทาง (vertex) และเส้นทางเดินรถเชื่อมระหว่างป้าย (arc) รูปล่างขวาแสดงตารางซึ่งทำหน้าเก็บบันทึกความสัมพันธ์ดังกล่าว ประกอบด้วยสดมภ์ source, target และ distance โดย source หมายถึงป้ายหยุดรถโดยสารประจำทางที่ต้นทางและ target หมายถึงป้ายหยุดรถโดยสารประจำทางที่ปลายทาง สัมพันธ์กับทิศทางการเดินรถฯ



source	target	distance	...
1	2	600	...
4	3	650	...
...	...	...	...

รูปที่ 3-4 การแปลงจากสภาพพฤติกรรมจริงมาสู่โครงสร้างข้อมูลความสัมพันธ์เชิงโครงข่าย

เพื่อให้การอ้างอิง vertex สามารถทำได้รวดเร็วและสื่อถึงคุณสมบัติหรือชนิดของ vertex นั้นจึงได้มีการกำหนดเลขรหัสของ vertex ที่ใช้เป็นตัวแทนป้าย สถานี หรือ ท่าเรือในระบบขนส่งมวลชนโดยประกอบด้วยตัวเลขหกตัว สามตัวแรกเป็นการระบุถึงประเภทของระบบขนส่งมวลชน

และสามตัวหลังเป็นตัวเลขใช้นับจำนวน vertex ที่ใช้แทนป้าย สถานี หรือท่าเรือ ของระบบขนส่งมวลชนแต่ละประเภท สรุปและแสดงอยู่ในตาราง 3-2 สำหรับกรณีการเดินเท้านั้น เนื่องจากใช้เป็นรูปแบบการเดินทางที่เชื่อมระบบขนส่งมวลชนต่างระบบเข้าด้วยกัน ดังนั้นเลขสามหลักแรกของค่ารหัสจึงเป็นได้ ตั้งแต่ 001-999 จึง

ตาราง 3-2 รหัสของ vertex ที่ใช้แทนป้าย สถานี หรือท่าเรือ สำหรับระบบขนส่งมวลชนแต่ละประเภท

ประเภทระบบขนส่งมวลชน	ช่วงค่าของรหัสที่เป็นไปได้ของ node หัวและท้าย arc	
	สามตัวแรก	สามตัวหลัง
การเดินเท้า	001 ถึง 999	001 ถึง 999
รถโดยสารประจำทาง	050 ถึง 777	001 ถึง 999
รถไฟฟ้าบีทีเอส	030 ถึง 040	001 ถึง 999
รถไฟฟ้ามหานคร	001 ถึง 020	001 ถึง 999
เรือข้ามฟาก	888	001 ถึง 999
เรือด่วนตามแม่น้ำเจ้าพระยา	999	001 ถึง 999

สำหรับช่วงค่าของตัวเลขสามหลักแรกนั้นนอกจากจะใช้ระบุถึงประเภทของระบบขนส่งมวลชนแล้วยังใช้ระบุถึงเส้นทางด้วย โดยในกรณีของรถโดยสารประจำทางตัวเลขที่แตกต่างกันจะหมายถึง vertex นั้นๆ อยู่บนถนนคนละสายกันด้วย ส่วนกรณีของรถไฟฟ้าตัวเลขที่แตกต่างกันหมายถึง vertex นั้น อยู่บนสายรถไฟฟ้าคนละสายกัน

การที่ออกแบบเช่นนี้ก็เพื่อให้สามารถตรวจสอบได้ว่า arc ที่เกิดจากการจับคู่กันของ vertex (node) ประจำอยู่บนถนนหรือเส้นทางใดเพื่อจะได้สามารถอ้างอิงค่าต่างๆ จากระบบอื่นมาใช้ได้ เช่น หากทราบว่า arc นั้นๆ อยู่บนถนนเส้นใดก็จะสามารถอ้างอิงปริมาณจราจร อุบัติเหตุ หรือการปิดถนน มาใช้เพื่อปรับแก้ค่าอัตราเร็วยานพาหนะและช่วงเวลาให้บริการของระบบขนส่งมวลชนที่ผ่านถนนเส้นนั้นได้ และนอกจากนี้ยังเพื่อให้สามารถตรวจสอบความถูกต้องของข้อมูลความสัมพันธ์เชิงโครงข่ายด้วยโปรแกรมคอมพิวเตอร์สะดวกขึ้น

อย่างไรก็ตามการอ้างอิงในระดับของถนนหรือเส้นทางนั้นอาจยังไม่ละเอียดเพียงพอ เนื่องจากปริมาณการจราจรอาจเปลี่ยนแปลงและแตกต่างกันมากแม้จะอยู่บนถนนเส้นเดียวกัน จึงอาจต้องมีการกำหนดรหัสที่สามารถอ้างอิงได้ละเอียดกว่านี้ เช่น ระดับส่วนย่อย (segment) ของถนน เป็นต้น

เท่าที่ผู้วิจัยได้ศึกษาและตรวจสอบพบว่า ในปัจจุบัน (พ.ศ. 2550) ยังไม่มีมาตรฐานกลางของค่ารหัสสำหรับการอ้างอิงช่วงย่อยของถนนระหว่างระบบหรือระหว่างฐานข้อมูลที่แตกต่างกัน

ประกาศใช้อย่างเป็นทางการ การออกแบบและวิธีอ้างอิงที่ใช้ในงานวิจัยนี้จึงเป็นเพียงการริเริ่มแนวความคิดดังกล่าวเท่านั้น

นอกจากนี้ช่วงค่ารหัสที่กำหนดขึ้นใช้ยังเป็นค่าที่กำหนดขึ้นภายใต้ปริมาณข้อมูลที่ใช้ในการทำวิจัยและอยู่ภายใต้กรอบความสนใจของงานวิจัยเป็นหลัก ดังนั้นจึงอาจยังไม่ครอบคลุมความต้องการในด้านอื่นๆ และอาจจะต้องมีการปรับปรุงวิธีการกำหนดค่ารหัสใหม่ก็เป็นได้ ทั้งนี้สำหรับในเบื้องต้นเมื่อปริมาณข้อมูลมีมากขึ้น อาจใช้วิธีการเพิ่มจำนวนหลักของตัวเลขขึ้นเพื่อให้สามารถรองรับทั้งประเภทของระบบขนส่งมวลชนและจำนวน vertex ที่มากขึ้นได้

จากข้อกำหนดเบื้องต้นของแบบจำลองข้อมูล จึงนำมาจัดทำเป็นโครงสร้างข้อมูลโดยแยกแสดงสำหรับระบบขนส่งมวลชนประเภทต่างๆ ดังนี้

1. โครงสร้างข้อมูลที่จัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องสำหรับการเดินเท้า แสดงอยู่ในตาราง 3-3

ตาราง 3-3 โครงสร้างข้อมูลสำหรับจัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องของการเดินเท้า

ชื่อรายการ	ชนิดข้อมูล	คุณลักษณะ	ตัวอย่าง	ค่าโดยปริยาย
source	character	เลขรหัสหกหลักของ vertex ต้นทาง xxxyyy (001<=xxx<=999, 001<=yyy<=999)	053002	-
target	character	เลขรหัสหกหลักของ vertex ปลายทาง xxxyyy (001<=xxx<=999, 001<=yyy<=999)	036003	-
distance	real	เลขทศนิยม ระยะทางของ arc ที่เชื่อมระหว่าง vertex ต้นทางและปลายทาง (>=0, หน่วยเป็นเมตร)	222.45	-
cost	real	เลขทศนิยม ระยะเวลาที่ใช้เดินทางผ่าน arc นั้นๆ (>=0, หน่วยเป็นวินาที) = 3600* distance/(speed *1000)	32	-
transport_mode	character	ตัวอักษร ระบุชนิดของระบบขนส่งมวลชน ความยาวไม่เกิน 5 ตัวอักษร	walk	walk
speed	real	เลขทศนิยม อัตราเร็วในการเดินผ่าน arc นั้นๆ (หน่วยเป็นกิโลเมตรต่อชั่วโมง)	5	5
time_start	character	เลขสี่หลักบอกเวลาเริ่มต้นที่สามารถเดินเท้าผ่าน arc นั้นๆ	00:00	00:00
time_end	character	เลขสี่หลักบอกเวลาสิ้นสุดที่สามารถเดินเท้าผ่าน arc นั้นๆ	24:00	24:00

2. โครงสร้างข้อมูลที่จัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องสำหรับรถโดยสารประจำทาง แสดงอยู่ในตาราง 3-4

ตาราง 3-4 โครงสร้างข้อมูลสำหรับจัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องของรถโดยสารประจำทาง

ชื่อรายการ	ชนิดข้อมูล	คุณลักษณะ	ตัวอย่าง	ค่าโดยปริยาย
source	character	เลขรหัสหกหลักของ vertex ต้นทาง xxxyyy (050<=xxx<=777, 001<=yyy<=999)	053002	-
target	character	เลขรหัสหกหลักของ vertex ปลายทาง xxxyyy (050<=xxx<=777, 001<=yyy<=999)	053003	-
distance	real	เลขทศนิยม ระยะทางของ arc ที่เชื่อมระหว่าง vertex ต้นทางและปลายทาง (>=0, หน่วยเป็นเมตร)	222.45	-
cost	real	เลขทศนิยม ระยะเวลาที่ใช้เดินทางผ่าน arc นั้นๆ (>=0, หน่วยเป็นวินาที) = 3600* distance/(speed *1000)	32	-
transport_mode	character	ตัวอักษร ระบุชนิดของระบบขนส่งมวลชน ความยาวไม่เกิน 5 ตัวอักษร	bus	bus
speed	real	เลขทศนิยม อัตราเร็วของยานพาหนะที่วิ่งผ่าน arc นั้นๆ (หน่วยเป็นกิโลเมตรต่อชั่วโมง)	17	20
time_start	character	เลขสี่หลักบอกเวลาเริ่มต้นให้บริการ สำหรับรถโดยสารประจำทาง และสำหรับ arc นั้นๆ	05:00	05:00
time_end	character	เลขสี่หลักบอกเวลาสิ้นสุดให้บริการ สำหรับรถโดยสารประจำทาง และสำหรับ arc นั้นๆ	23:00	23:00
bus_num_start	text	ตัวเลขสายรถประจำทางที่วิ่งผ่าน vertex ต้นทาง หากมีมากกว่าหนึ่งสายให้คั่นด้วยเครื่องหมาย “,”	23,34, 18	-
bus_num_end	text	ตัวเลขสายรถประจำทางที่วิ่งผ่าน vertex ปลายทาง หากมีมากกว่าหนึ่งสายให้คั่นด้วยเครื่องหมาย “,”	23,34,5	-
bus_num	text	ตัวเลขสายรถประจำทางที่วิ่งผ่าน arc นั้นๆ หากมีมากกว่าหนึ่งสายให้คั่นด้วยเครื่องหมาย “,”	23,34	-

3. โครงสร้างข้อมูลที่จัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องสำหรับรถไฟฟ้าบีทีเอส แสดงอยู่ในตาราง 3-5

ตาราง 3-5 โครงสร้างข้อมูลสำหรับจัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องของรถไฟฟ้าบีทีเอส

ชื่อรายการ	ชนิดข้อมูล	คุณลักษณะ	ตัวอย่าง	ค่าโดยปริยาย
source	character	เลขรหัสหกหลักของ vertex ต้นทาง xxxyyy (030<=xxx<=040, 001<=yyy<=999)	036002	-
target	character	เลขรหัสหกหลักของ vertex ปลายทาง xxxyyy (030<=xxx<=040, 001<=yyy<=999)	036003	-
distance	real	เลขทศนิยม ระยะทางของ arc ที่เชื่อมระหว่าง vertex ต้นทางและปลายทาง (>=0, หน่วยเป็นเมตร)	422.45	-
cost	real	เลขทศนิยม ระยะเวลาที่ใช้เดินทางผ่าน arc นั้นๆ (>=0, หน่วยเป็นวินาที) = 3600* distance/(speed *1000)	82	-
transport_mode	character	ตัวอักษร ระบุชนิดของระบบขนส่งมวลชน ความยาวไม่เกิน 5 ตัวอักษร	bts	bts
speed	real	เลขทศนิยม อัตราเร็วของยานพาหนะที่วิ่งผ่าน arc นั้นๆ (หน่วยเป็นกิโลเมตรต่อชั่วโมง)	35	35
time_start	character	เลขสี่หลักบอกเวลาเริ่มต้นให้บริการ สำหรับรถไฟฟ้าบีทีเอส และสำหรับ arc นั้นๆ	06:00	06:00
time_end	character	เลขสี่หลักบอกเวลาสิ้นสุดให้บริการ สำหรับรถไฟฟ้าบีทีเอส และสำหรับ arc นั้นๆ	24:00	24:00

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

4. โครงสร้างข้อมูลที่จัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องสำหรับรถไฟฟ้ามหานคร แสดงอยู่ในตาราง 3-6

ตาราง 3-6 โครงสร้างข้อมูลสำหรับจัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องของรถไฟฟ้ามหานคร

ชื่อรายการ	ชนิดข้อมูล	คุณลักษณะ	ตัวอย่าง	ค่าโดยปริยาย
source	character	เลขรหัสหกหลักของ vertex ต้นทาง xxxyyy (001<=xxx<=020, 001<=yyy<=999)	001002	-
target	character	เลขรหัสหกหลักของ vertex ปลายทาง xxxyyy (001<=xxx<=020, 001<=yyy<=999)	001003	-
distance	real	เลขทศนิยม ระยะทางของ arc ที่เชื่อมระหว่าง vertex ต้นทางและปลายทาง (>=0, หน่วยเป็นเมตร)	422.45	-
cost	real	เลขทศนิยม ระยะเวลาที่ใช้เดินทางผ่าน arc นั้นๆ (>=0, หน่วยเป็นวินาที) = 3600* distance/(speed *1000)	82	-
transport_mode	character	ตัวอักษร ระบุชนิดของระบบขนส่งมวลชน ความยาวไม่เกิน 5 ตัวอักษร	mrt	mrt
speed	real	เลขทศนิยม อัตราเร็วของยานพาหนะที่วิ่งผ่าน arc นั้นๆ (หน่วยเป็นกิโลเมตรต่อชั่วโมง)	35	35
time_start	character	เลขสี่หลักบอกเวลาเริ่มต้นให้บริการ สำหรับรถไฟฟ้าใต้ดิน และสำหรับ arc นั้นๆ	06:00	06:00
time_end	character	เลขสี่หลักบอกเวลาสิ้นสุดให้บริการ สำหรับรถไฟฟ้าใต้ดิน และสำหรับ arc นั้นๆ	24:00	24:00

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

5. โครงสร้างข้อมูลที่จัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องสำหรับเรือข้ามฟาก แสดงอยู่ในตาราง 3-7

ตาราง 3-7 โครงสร้างข้อมูลสำหรับจัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องของเรือข้ามฟาก

ชื่อรายการ	ชนิดข้อมูล	คุณลักษณะ	ตัวอย่าง	ค่าโดยปริยาย
source	character	เลขรหัสหกหลักของ vertex ต้นทาง xxxyyy (xxx=888, 001<=yyy<=999)	888002	-
target	character	เลขรหัสหกหลักของ vertex ปลายทาง xxxyyy (xxx=888, 001<=yyy<=999)	888003	-
distance	real	เลขทศนิยม ระยะทางของ arc ที่เชื่อมระหว่าง vertex ต้นทางและปลายทาง (>=0, หน่วยเป็นเมตร)	422.45	-
cost	real	เลขทศนิยม ระยะเวลาที่ใช้เดินทางผ่าน arc นั้นๆ (>=0, หน่วยเป็นวินาที) = 3600* distance/(speed *1000)	82	-
transport_mode	character	ตัวอักษร ระบุชนิดของระบบขนส่งมวลชนความยาวไม่เกิน 5 ตัวอักษร	ferry	ferry
speed	real	เลขทศนิยม อัตราเร็วของยานพาหนะที่วิ่งผ่าน arc นั้นๆ (หน่วยเป็นกิโลเมตรต่อชั่วโมง)	15	15
time_start	character	เลขสี่หลักบอกเวลาเริ่มต้นให้บริการ สำหรับเรือข้ามฟาก และสำหรับ arc นั้นๆ	06:00	06:00
time_end	character	เลขสี่หลักบอกเวลาสิ้นสุดให้บริการ สำหรับเรือข้ามฟาก และสำหรับ arc นั้นๆ	24:00	24:00

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

6. โครงสร้างข้อมูลที่จัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องสำหรับเรือด่วนตามแม่น้ำเจ้าพระยา แสดงอยู่ในตาราง 3-8

ตาราง 3-8 โครงสร้างข้อมูลสำหรับจัดเก็บพฤติกรรมและค่าต่างๆ ที่เกี่ยวข้องของเรือด่วนตามแม่น้ำเจ้าพระยา

ชื่อรายการ	ชนิดข้อมูล	คุณลักษณะ	ตัวอย่าง	ค่าโดยปริยาย
source	character	เลขรหัสหกหลักของ vertex ต้นทาง xxxyyy (xxx=999, 001<=yyy<=999)	999002	-
target	character	เลขรหัสหกหลักของ vertex ปลายทาง xxxyyy (xxx=999, 001<=yyy<=999)	999003	-
distance	real	เลขทศนิยม ระยะทางของ arc ที่เชื่อมระหว่าง vertex ต้นทางและปลายทาง (>=0, หน่วยเป็นเมตร)	422.45	-
cost	real	เลขทศนิยม ระยะเวลาที่ใช้เดินทางผ่าน arc นั้นๆ (>=0, หน่วยเป็นวินาที) = 3600* distance/(speed *1000)	82	-
transport_mode	character	ตัวอักษร ระบุชนิดของระบบขนส่งมวลชน ความยาวไม่เกิน 5 ตัวอักษร	boat	boat
speed	real	เลขทศนิยม อัตราเร็วของยานพาหนะที่วิ่งผ่าน arc นั้นๆ (หน่วยเป็นกิโลเมตรต่อชั่วโมง)	30	30
time_start	character	เลขสี่หลักบอกเวลาเริ่มต้นให้บริการ สำหรับเรือด่วนตามแม่น้ำเจ้าพระยา และสำหรับ arc นั้นๆ	06:00	06:00
time_end	character	เลขสี่หลักบอกเวลาสิ้นสุดให้บริการ สำหรับเรือด่วนตามแม่น้ำเจ้าพระยา และสำหรับ arc นั้นๆ	24:00	24:00

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ระยะเวลารอคอย (Waiting Time)

ระยะเวลาที่ใช้ในการเข้าสู่สถานะที่พร้อมจะเปลี่ยนจากระบบขนส่งมวลชนระบบหนึ่งไปยังอีกระบบหนึ่ง อาจจะประกอบไปด้วยหลายส่วน เช่น การซื้อตั๋วโดยสาร การเดินเท้าเพื่อไปยังสถานีรถไฟฟ้า หรือท่าเรือ เป็นต้น แต่ในที่นี้จะพิจารณาเฉพาะระยะเวลาที่ใช้รอรถประจำทาง เรือโดยสาร หรือรถไฟฟ้า และระยะเวลาที่ใช้เพื่อเดินไปยังจุดสถานีหรือป้ายหยุดรถโดยสารประจำทางเท่านั้น

ระยะเวลารอคอยยานพาหนะ (Waiting time) ในที่นี้หมายถึงระยะเวลาของสองกิจกรรมรวมกัน คือระยะเวลาที่พาหนะใช้ในการเดินทางบนระยะทางที่เท่ากับระยะห่างการปล่อยยานพาหนะแต่ละคัน รวมกับระยะเวลาจอดรับส่งผู้โดยสารระหว่างระยะทางช่วงดังกล่าว

ระยะเวลารอคอยยานพาหนะดังกล่าวจะเกิดขึ้นก็ต่อเมื่อมีการเปลี่ยนรูปแบบการเดินทางจากระบบขนส่งมวลชนประเภทหนึ่งไปยังระบบขนส่งมวลชนอีกประเภทหนึ่งหรือการเปลี่ยนสายรถประจำทาง (ยกเว้นการเปลี่ยนจากระบบขนส่งมวลชนอื่นๆ ไปสู่การเดิน) สำหรับกรณีที่เป็นรูปแบบการเดินทางเดียวกันตลอดจะมีเฉพาะระยะเวลาจอดรอรับส่งผู้โดยสาร (Dwell time) เท่านั้น คำอธิบายส่วนนี้แสดงสรุปอยู่ในตารางที่ 3-9

ตาราง 3-9 สรุปความเป็นไปได้ของรูปแบบการเดินทางที่มีผลต่อระยะเวลารอคอยและระยะเวลาจอดรอรับส่งผู้โดยสาร

ความเป็นไปได้ของรูปแบบการเดินทางจากช่วงที่หนึ่ง (A) ไปสู่ ช่วงที่สอง (B)	ระยะเวลารอคอย (Waiting time)	ระยะเวลาจอดรอรับส่งผู้โดยสาร (Dwell time)
[A=B] และ [A =“การเดิน” และ B=”การเดิน”]	0	0
[A=B] และ [A ≠ “การเดิน” และ B≠”การเดิน”]	0	✓
[A≠B] และ [A =“การเดิน” และ B≠”การเดิน”]	✓	✓
[A≠B] และ [A ≠ “การเดิน” และ B=”การเดิน”]	0	0
[A≠B] และ [A ≠ “การเดิน” และ B≠”การเดิน”]	✓	0

ระยะเวลารอคอยยานพาหนะและระยะเวลาจอดรอรับส่งผู้โดยสารของยานพาหนะแต่ละประเภทถูกจัดเก็บแยกเป็นตารางต่างหากในฐานข้อมูล และเป็นค่าที่สามารถเปลี่ยนแปลงปรับแก้ได้แบบพลวัต ตาราง 3-10 สรุปค่าโดยปริยายของระยะเวลารอคอยยานพาหนะและระยะเวลาจอดรอรับส่งผู้โดยสารของระบบขนส่งมวลชนแต่ละประเภทที่ใช้ในงานวิจัย

ตาราง 3-10 สรุปค่าโดยปริยายของระยะเวลารอคอยยานพาหนะและระยะเวลาจอดรอรับส่งผู้โดยสารของยานพาหนะแต่ละประเภท

ประเภทระบบขนส่งมวลชน	ระยะเวลารอคอย (วินาที)	ระยะเวลาจอดรอ (วินาที)
การเดินเท้า	ไม่มี	ไม่มี
รถโดยสารประจำทาง	540	10
รถไฟฟ้าบีทีเอส	240	10
รถไฟฟ้ามหานคร	240	10
เรือข้ามฟาก	360	300
เรือด่วนตามแม่น้ำเจ้าพระยา	600	15

ข้อมูลต่างๆ จะถูกจัดเก็บลงในตารางตามโครงสร้างข้อมูลที่แสดงในตาราง 3-3 ถึง 3-8 ลักษณะการปรับแก้ที่อนุญาตให้กระทำได้สำหรับแต่ละรายการได้แสดงสรุปไว้ในตาราง 3-11 โดยที่ค่าต่างๆ ที่จัดเก็บสามารถเปลี่ยนแปลงได้ในสองลักษณะ คือ การเปลี่ยนแปลงที่ต้องจัดเตรียมไว้ล่วงหน้า และการเปลี่ยนแปลงแบบพลวัตที่สามารถทำได้ระหว่างคำนวณและค้นหาเส้นทาง

ตาราง 3-11 ลักษณะการปรับแก้ค่าคุณสมบัติหรือพฤติกรรมของระบบขนส่งมวลชน

ชื่อรายการ	การปรับแก้ค่าในตารางข้อมูล		
	ผู้ดูแลต้องจัดเตรียมล่วงหน้า		ปรับแก้ได้แบบพลวัตตลอดเวลา และไม่ต้องตรวจสอบความสัมพันธ์เชิงโครงข่าย
	ต้องตรวจสอบความสัมพันธ์เชิงโครงข่าย	ไม่ต้องตรวจสอบความสัมพันธ์เชิงโครงข่าย	
source	✓		
target	✓		
distance		✓	
cost			✓
transport_mode	✓		
speed			✓
time_start			✓
time_end			✓
bus_num_start	✓		
bus_num_end	✓		
bus_num	✓		

การเปลี่ยนแปลงที่ต้องจัดเตรียมไว้ล่วงหน้า แบ่งเป็นการเปลี่ยนแปลงที่ต้องมีการตรวจสอบความสัมพันธ์เชิงโครงข่าย ก่อนการเริ่มต้นคำนวณและค้นหาเส้นทางในแต่ละครั้ง เนื่องจากจะมีผลกระทบต่อความสัมพันธ์และเส้นทางที่ได้บันทึกไว้ก่อนหน้า และการเปลี่ยนแปลงที่ไม่ต้องตรวจสอบความสัมพันธ์เชิงโครงข่าย เนื่องจากเป็นค่าตัวแปรซึ่งไม่มีผลกระทบ สำหรับรายละเอียดในการตรวจสอบนั้นได้กล่าวไว้ใน บทที่ 3 ในหัวข้อ “การจัดเตรียมและปรับแก้ข้อมูล” หัวข้อย่อยที่ 2 และ 3 เรื่อง “การจัดเตรียมข้อมูลความสัมพันธ์เชิงโครงข่าย” และ “การตรวจสอบข้อมูลความสัมพันธ์เชิงโครงข่าย”

การเปลี่ยนแปลงแบบพลวัตที่สามารถทำได้ระหว่างคำนวณและค้นหาเส้นทาง โดยเป็นการเปลี่ยนแปลงเพื่อให้มีความเหมาะสมหรือใกล้เคียงสภาพความเป็นจริง ณ ขณะนั้น เช่น อัตราเร็วรถโดยสารประจำทางอาจช้าลงหากการจราจรใน arc นั้นติดขัดมาก หรือ ช่วงเวลาที่ให้บริการเปลี่ยนแปลงเนื่องจากเหตุขัดข้องหรือมีการปิดการจราจร เป็นต้น

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

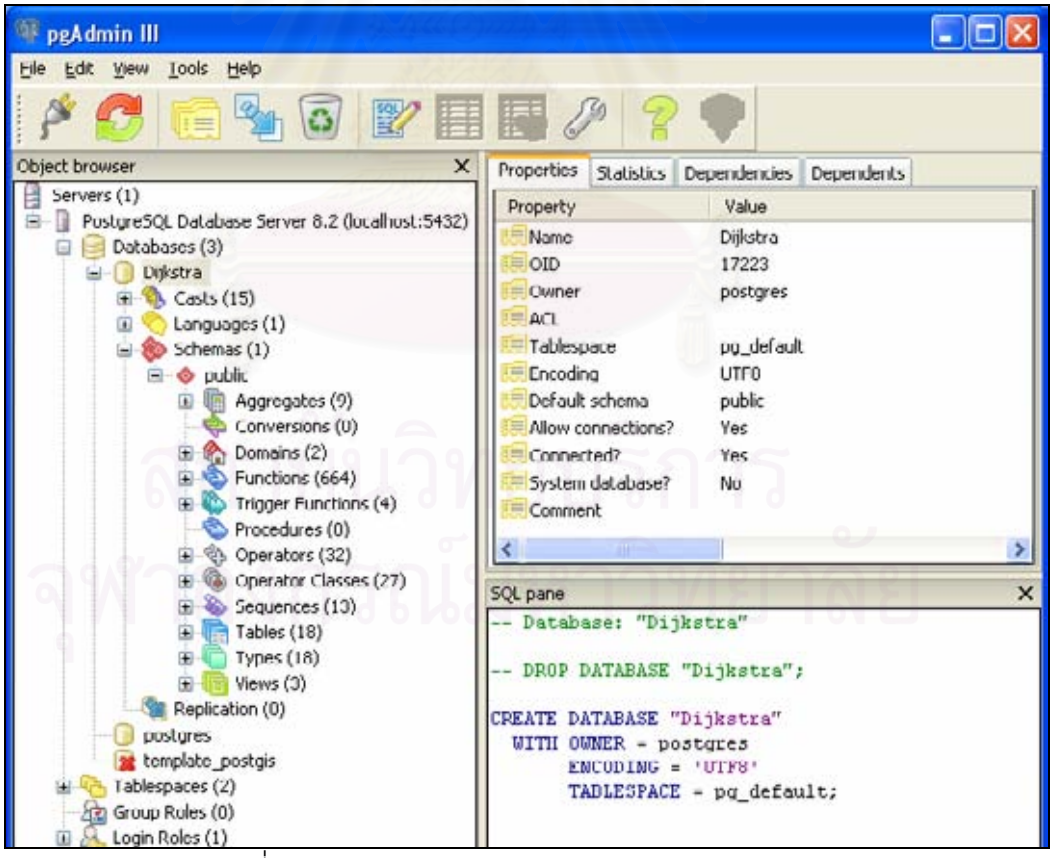
เครื่องมือที่ใช้ในการวิจัย

เครื่องมือด้านซอฟต์แวร์ที่ใช้ในงานวิจัยนี้

1. ซอฟต์แวร์สำหรับบริหารจัดการฐานข้อมูล

ซอฟต์แวร์สำหรับบริหารจัดการฐานข้อมูลที่ใช้ คือ PostgreSQL ซึ่งเป็นซอฟต์แวร์แบบเปิดเผยรหัส (Open Source) พัฒนาและเผยแพร่โดยภาควิชาวิทยาการคอมพิวเตอร์ University of California at Berkeley ปัจจุบันมีผู้นิยมใช้ทั่วโลก เนื่องจากสามารถใช้ได้ฟรีโดยไม่มีค่าลิขสิทธิ์ และยังเป็นซอฟต์แวร์ที่ถูกออกแบบและพัฒนามาเพื่อใช้ทดสอบทดลองแนวความคิดหรือเทคนิคใหม่ๆ ด้านฐานข้อมูล สามารถดาวน์โหลดได้ฟรีที่ http://www.postgresql.org

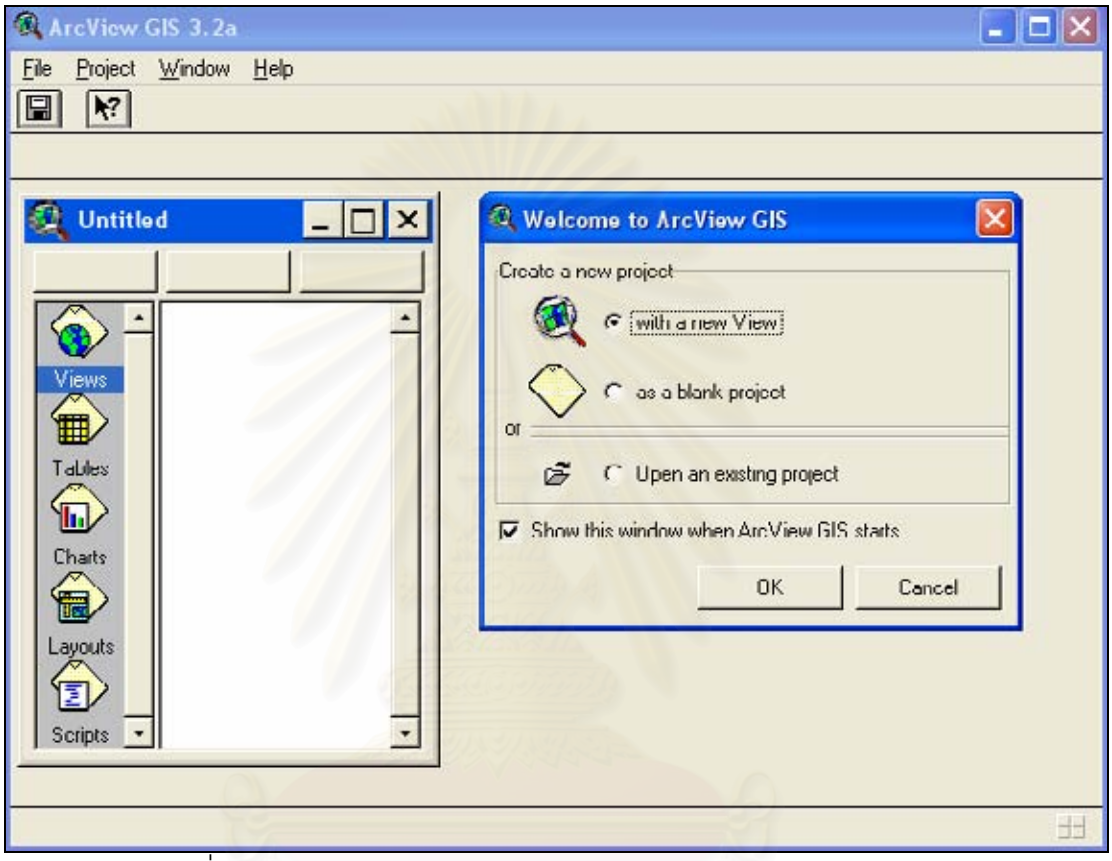
รุ่นที่ใช้ในงานวิจัยคือรุ่นที่ 8.2 และทำงานบนระบบปฏิบัติการ Microsoft Windows XP เมื่อดาวน์โหลดและติดตั้งลงในเครื่องคอมพิวเตอร์แล้วจะได้ตัวซอฟต์แวร์บริหารจัดการฐานข้อมูล PostgreSQL และซอฟต์แวร์ที่ทำหน้าเป็นส่วนติดต่อผู้ใช้แบบกราฟิกเรียกว่า pgAdmin ซึ่งจะช่วยให้ผู้ดูแลฐานข้อมูลมีความสะดวกสะบายในการใช้งานยิ่งขึ้นดังแสดงในรูปที่ 3-5



รูปที่ 3-5 หน้าต่างสำหรับโต้ตอบกับผู้ใช้ของ pgAdmin

2. ซอฟต์แวร์สำหรับจัดการข้อมูลเชิงพื้นที่

ซอฟต์แวร์สำหรับจัดการข้อมูลเชิงพื้นที่ที่ใช้ในงานวิจัยนี้เป็นซอฟต์แวร์เชิงพาณิชย์ชื่อ ArcView โดยเป็นรุ่นที่ 3.2a พัฒนาโยบริษัท อีเอสอาร์ไอ มีความสามารถทั่วๆ ไปด้านการจัดการข้อมูลเชิงพื้นที่ เช่น เปิดดูข้อมูล สืบค้นข้อมูล และปรับแก้ไขข้อมูล เป็นต้น แสดงในรูปที่ 3-6



รูปที่ 3-6 ส่วนโต้ตอบกับผู้ใช้แบบกราฟิกของซอฟต์แวร์ ArcView 3.2a

3. ภาษาสำหรับการพัฒนาโปรแกรม

ภาษาสำหรับการพัฒนาโปรแกรมที่ใช้ในงานวิจัยนี้เป็นภาษาสคริปต์ที่ชื่อว่า PHP ("PHP: Hypertext Preprocessor") ใช้สำหรับการพัฒนาโปรแกรมประยุกต์ที่ทำงานบนเครื่องแม่ข่ายและสื่อสารหรือแสดงผลลัพธ์ผ่านเว็บ การส่งและรับคำสั่งจะกระทำผ่านเครือข่ายอินเทอร์เน็ต สามารถดาวน์โหลดมาใช้ได้ฟรีจาก http://www.php.net รุ่นที่ใช้ในงานวิจัยคือ 5.2.0

4. ซอฟต์แวร์สำหรับควบคุมดูแลการทำงานของเว็บเซอร์ฟเวอร์

ซอฟต์แวร์สำหรับควบคุมดูแลการทำงานของเว็บเซอร์ฟเวอร์ที่ใช้ในงานวิจัยนี้ เป็นซอฟต์แวร์แบบเปิดเผยรหัส ชื่อว่า Apache Web Server สามารถดาวน์โหลดมาใช้ได้ฟรีจาก http://www.apache.org รุ่นที่ใช้ในงานวิจัยคือ 2.2.3

5. ซอฟต์แวร์สำหรับให้บริการแผนที่ผ่านเครือข่ายอินเทอร์เน็ต

ซอฟต์แวร์สำหรับให้บริการแผนที่ผ่านเครือข่ายอินเทอร์เน็ตที่ใช้ในงานวิจัยนี้เป็นซอฟต์แวร์แบบเปิดเผยรหัส ชื่อว่า Map Server พัฒนาโดยมหาวิทยาลัย Minnesota สามารถดาวน์โหลดมาใช้ได้ฟรีจาก http://www.maptools.org/ms4w/index.phtml?page=downloads.html รุ่นที่ใช้ในงานวิจัยคือ 4.10.0

6. ซอฟต์แวร์สำหรับทำงานตอบโต้กับผู้ใช้บนหน้าจอ และแสดงผลลัพธ์

ซอฟต์แวร์สำหรับทำงานตอบโต้กับผู้ใช้บนหน้าจอ และแสดงผลลัพธ์ที่ใช้ในงานวิจัยนี้เป็นซอฟต์แวร์แบบเปิดเผยรหัส ชื่อว่า Openlayers เป็นชุดของกลุ่มโปรแกรมย่อยที่เขียนด้วยภาษา Java Script ซึ่งจะทำหน้าที่ในการตอบโต้กับผู้ใช้พร้อมทั้งติดต่อกับเครื่องแม่ข่ายเพื่อร้องขอภาพแผนที่ในบริเวณที่สอดคล้องกับกรอบการแสดงผลมาแสดงบนเว็บเบราว์เซอร์ สามารถดาวน์โหลดมาใช้ได้ฟรีจาก http://www.openlayers.org/ รุ่นที่ใช้ในงานวิจัยนี้คือ 2.3

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ข้อมูลที่เกี่ยวข้อง

ในงานวิจัยนี้จะต้องมีการศึกษาและวิเคราะห์ข้อมูลต่างๆ ที่เกี่ยวข้อง โดยอาศัยการตีความจากทฤษฎี Graph และวิเคราะห์หาข้อมูลอรรถาธิบายที่จะมีอิทธิพลต่อระยะเวลาการเดินทาง ซึ่งจากการศึกษาและวิเคราะห์สรุปได้เป็นสองกลุ่มใหญ่ๆ คือ ระยะเวลาที่ใช้เดินทางบนเส้นทาง และระยะเวลาที่ใช้ไปกับการรอคอยต่างๆ เช่น รอคอยยานพาหนะ ระยะเวลาจอดรอรับผู้โดยสาร

ในงานวิจัยนี้จำเป็นต้องอาศัยข้อมูลพื้นฐานทางด้านเส้นทางการคมนาคมขนส่งที่ได้มีการจัดทำไว้แล้วโดยหน่วยงานที่รับผิดชอบ มาพัฒนาและปรับแก้ให้เหมาะสมกับจุดประสงค์เฉพาะในงานวิจัย ข้อมูลสืบเนื่องอื่นที่ยังมิได้มีการจัดทำขึ้นเป็นกิจลักษณะก็ได้จัดทำเพิ่มเติมขึ้นมา โดยกลุ่มของข้อมูลที่จำเป็นมีดังนี้

ข้อมูลที่สนใจใช้งานนั้นจะเป็นเส้นทางบางเส้นทางในพื้นที่กรุงเทพมหานคร โดยมีรายการดังนี้

1. ข้อมูลเชิงตำแหน่ง
    1.1 ข้อมูลเส้นทางเดินรถโดยสารประจำทางของ ขสมก.พร้อมตำแหน่งป้ายหยุดรถโดยสาร
    1.2 ข้อมูลเส้นทางเดินเรือโดยสาร และเรือข้ามฟาก ของกรมการขนส่งทางน้ำและพาณิชยนาวีพร้อมตำแหน่งท่าเรือ
    1.3 ข้อมูลเส้นทางเดินรถไฟฟ้าบีทีเอส พร้อมตำแหน่งที่ตั้งสถานี
    1.4 ข้อมูลเส้นทางเดินรถไฟฟ้ามหานคร พร้อมตำแหน่งที่ตั้งสถานี
    1.5 ข้อมูลแนวถนน รางรถไฟฟ้า และเส้นทางการจราจรทางน้ำ
    1.6 เส้นทางเดินเพื่อเปลี่ยนจากรูปแบบการเดินทางหนึ่งไปสู่อีกรูปแบบหนึ่ง
2. ข้อมูลอรรถาธิบาย
    2.1 อัตราเร็วเฉลี่ยของรถโดยสารประจำทาง
    2.2 อัตราเร็วเฉลี่ยของรถไฟฟ้าบีทีเอส
    2.3 อัตราเร็วเฉลี่ยของรถไฟฟ้ามหานคร
    2.4 อัตราเร็วเฉลี่ยของเรือโดยสาร และเรือข้ามฟาก
    2.5 อัตราเร็วเฉลี่ยของการเดินเท้า
    2.6 ความถี่ของการมาถึงของยวดยานในระบบขนส่งมวลชนแต่ละแบบ
    2.7 ระยะทางที่ผู้เดินทางใช้ในการเดินเพื่อเปลี่ยนจากรูปแบบการเดินทางหนึ่งไปสู่อีกรูปแบบหนึ่ง
    2.8 ทิศทางการจราจรในถนน ทางน้ำและรางรถไฟฟ้า

การจัดเตรียมและปรับแก้ข้อมูล

1. การจัดเตรียมและปรับแก้ข้อมูลเชิงพื้นที่

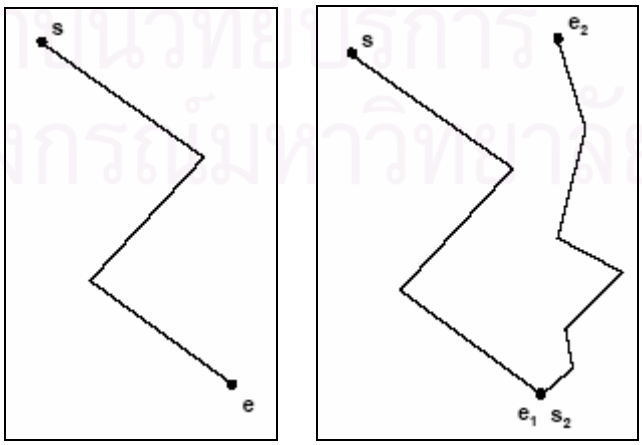
ข้อมูลเชิงตำแหน่งและข้อมูลอรรถาธิบายที่จำเป็นจะถูกปรับแก้และจัดเตรียมโดยซอฟต์แวร์ทางด้านระบบสารสนเทศภูมิศาสตร์ ArcView 3.2a และซอฟต์แวร์ทางด้าน Spreadsheet ซึ่งมีแพร่หลายทั่วไปและเป็นที่รู้จักกันดี

เนื่องจากข้อมูลด้านโครงข่ายคมนาคมและระบบขนส่งมวลชนที่มีจัดทำแพร่หลายอยู่แล้วนั้น มิได้จัดทำขึ้นภายใต้วัตถุประสงค์หรือแนวคิดเพื่อการคำนวณและสืบค้นหาเส้นทางการเดินทางที่ใช้เวลาน้อยที่สุดบนระบบขนส่งมวลชนตั้งแต่แรก เป็นเพียงการจัดทำเพื่อจุดประสงค์ด้านสารสนเทศภูมิศาสตร์ทั่วๆ ไป ดังนั้นในงานวิจัยนี้จึงมีขั้นตอนสำคัญขั้นตอนหนึ่งคือการปรับแก้ และเตรียมข้อมูลฐานดังกล่าวให้มีโครงสร้างและรูปแบบที่เหมาะสมก่อนจะนำไปใช้ประมวลผลในขั้นตอนถัดไป

การปรับแก้ดังกล่าวเช่น การระบุถึงสิ่งที่เป็น vertex และ arc ตามนิยามของทฤษฎีทางการวิเคราะห์โครงข่าย รวมถึงข้อมูลปลีกย่อยอื่นๆ ที่เกี่ยวข้อง ซึ่งจะต้องมีการวิเคราะห์และตีความข้อมูลเชิงตำแหน่งและข้อมูลอรรถธิบายที่มีอยู่แล้วแปลงเป็นรูปแบบหรือมุมมองตามทฤษฎีดังกล่าวเพื่อจัดเก็บลงในซอฟต์แวร์ Spreadsheet

นอกจากนี้ข้อมูลเชิงพื้นที่หรือแผนที่เวกเตอร์ซึ่งได้จากหน่วยงานต่างๆ ยังมีข้อจำกัดทำให้ต้องทำการแก้ไขดังนี้

- มีความปะปนกันระหว่างเส้นที่มีลักษณะเป็น Multilinestring ซึ่งเป็นเซ็ตของเส้น Linestring หลายเส้น ซึ่งมีจุดหัวหรือท้ายมากกว่าหนึ่งจุด และกรณี Linestring ซึ่งเป็นเส้นที่มีจุดหัวและท้ายเพียงอย่างละหนึ่งจุด (OGC,1999: 2-5 ถึง 2-7) แสดงตัวอย่างเปรียบเทียบในรูปที่ 3-7



รูปที่ 3-7 เส้นชนิด Linestring รูปด้านซ้ายมือ (se) และเส้นชนิด Multilinestring รูปด้านขวามือ (se_1, e_2s_2)

- การจัดเก็บข้อมูลเป็นการจัดเก็บเส้นในเชิงรูปร่างของเส้นทางคมนาคมพร้อมข้อมูลอรรถาธิบายบางอย่าง เช่น ชื่อถนน ชื่อสถานีรถไฟฟ้า สายรถประจำทางที่ผ่าน ยังมิได้จัดเก็บในเชิงหรือมุมมองด้านโครงข่าย

รายการปรับแก้และจัดเตรียมข้อมูลมีดังนี้

- กำหนดรหัสประจำป้ายหยุดรถโดยสารประจำทาง สถานีรถไฟฟ้า สถานีรถไฟฟ้าใต้ดิน และท่าเรือ โดยรหัสนี้ต้องมีความเป็นเอกลักษณ์โดยตนเองและต้องสามารถแยกแยะจากรหัสได้ว่าเป็นสถานีของระบบขนส่งมวลชนประเภทใด เพื่อให้ง่ายต่อการตรวจสอบในภายหลัง ป้ายหรือสถานีเหล่านี้ถือเป็นตัวแทนของ vertex ในมุมมองของการวิเคราะห์เชิงโครงข่าย
- กำหนด arc โดยการตัดแยกเส้นทางคมนาคมเดิมออกเป็นส่วนย่อยๆ ส่วนย่อยๆ เหล่านี้จะทำหน้าที่เชื่อมระหว่างป้ายหรือสถานีหรือท่าเรือเข้าด้วยกัน จุดประสงค์ของขั้นตอนนี้คือเพื่อให้ได้ระยะทางของเส้นทางที่เชื่อมระหว่างป้าย สถานี หรือท่าเรือ และให้ได้รูปร่างเส้นทางของเส้นเชื่อมนี้ เพื่อใช้สำหรับแสดงผลลัพธ์เส้นทางต่อไป ตลอดจนเป็นการระบุว่าระหว่างสองป้ายหรือสองสถานีใดๆ มีคู่ใดบ้างที่มีระบบขนส่งมวลชนเชื่อมถึงกัน
- ทำการสร้าง arc เพิ่มในกรณีที่เป็นการเดินทางแบบสองทิศทางซ้อนทับกัน เช่น รถไฟฟ้าหรือเรือข้ามฟาก ที่เป็น arc ซึ่งสามารถเดินทางไปกลับได้ทั้งสองทิศทาง
- กำหนดรหัสประจำ arc โดยใช้รหัสประจำ vertex สอง vertex หัวท้ายของ arc เป็นตัวบอกทิศทางของระบบขนส่งมวลชน (vertex ตัวหนึ่งทำหน้าที่เป็น source และอีก vertex หนึ่งทำหน้าที่เป็น target) เช่น หากต้องการระบุว่า arc นี้ใช้สำหรับบ่งบอกการเดินทางจาก vertex 1 ไปยัง vertex 2 ก็ให้กำหนดรหัสสำหรับ arc นี้เป็น “12” หากเป็นการเดินทางกลับทิศก็ให้สลับตำแหน่งตัวเลขเป็น “21” รหัสประจำ arc นี้จะต้องเป็นค่าที่เป็นเอกลักษณ์เช่นเดียวกับกรณีรหัสของ vertex
- สร้าง arc เพิ่มเติมสำหรับเส้นทางเดินที่ใช้เชื่อมต่อระหว่างระบบขนส่งมวลชนแต่ละประเภท (โดยการ digitize เส้นเวกเตอร์เพิ่ม) arc ของการเดินนี้จะสามารถเดินได้ทั้งสองทิศทางไปและกลับเสมอ เส้น arc นี้จะเชื่อมระหว่าง vertex สอง vertex ที่เป็นของระบบขนส่งมวลชนคนละประเภทกัน
- เส้น arc ทุกเส้นจะต้องเป็น Linestring (single part) ทั้งหมด เนื่องจากจะต้องเป็นเส้นที่มีจุดหัวและจุดท้ายอย่างละ 1 จุดเท่านั้น จึงจะสามารถกำหนดทิศทางได้
- Vertex หนึ่งคู่ใดๆ จะมี arc เชื่อมได้มากที่สุดสอง arc เท่านั้น คือในทิศทางไปและกลับ
- รหัสต่างๆ เหล่านี้จะจัดเก็บเป็นข้อมูลอรรถาธิบายในแฟ้มข้อมูล dbf

2. การจัดเตรียมข้อมูลความสัมพันธ์เชิงโครงข่าย

ข้อมูลความสัมพันธ์เชิงโครงข่ายในที่นี้หมายถึง ข้อมูลที่ระบุว่า vertex แต่ละ vertex มีความสัมพันธ์ (เชื่อมต่อ) กับ vertex อื่นๆ อย่างไร vertex ตัวใดควรจะทำหน้าที่เป็น source และตัวใดควรจะทำหน้าที่เป็น target โดยเป็นการคัดแยกและแปลจากข้อมูลแผนที่ที่ปรับแก้ให้มีนัยเชิงโครงข่ายแล้ว

ข้อมูลรายละเอียดของความสัมพันธ์ ได้แก่

2.1 ระยะทางของ arc ที่เชื่อมระหว่าง vertex คู่ใดๆ โดย vertex ตัวหนึ่งจะทำหน้าที่เป็น source อีกตัวจะทำหน้าที่เป็น target และมีเส้น arc เชื่อมระหว่าง source และ target

2.2 อัตราเร็วของยานพาหนะแต่ละชนิดที่วิ่งไปบน arc จัดเก็บแยกเป็นอิสระในแต่ละ arc (speed)

2.3 ช่วงเวลาที่มีระบบขนส่งหรือยานพาหนะวิ่งให้บริการอยู่ในแต่ละ arc (time_start, time_end)

2.4 หมายเลขสายรถโดยสารประจำทางที่วิ่งผ่านแต่ละ vertex เฉพาะกรณีที่เป็น vertex ของป้ายหยุดรถโดยสารประจำทาง (bus_num_start, bus_num_end) โดยจัดเก็บแยกกันสำหรับ vertex ต้นและปลาย

2.5 หมายเลขสายรถโดยสารประจำทางที่วิ่งผ่านแต่ละ arc เฉพาะกรณีที่เป็น arc ซึ่งเชื่อมระหว่าง vertex ของป้ายหยุดรถโดยสารประจำทาง (bus_num) กล่าวอีกนัยหนึ่งคือเป็นหมายเลขสายรถประจำทางที่วิ่งผ่านทั้ง vertex ต้นทางและปลายทาง

ตัวอย่างข้อมูลความสัมพันธ์เชิงโครงข่ายได้จัดทำในซอฟต์แวร์ประเภท Spreadsheet โดยตัวอย่างข้อมูลที่จัดทำแสดงอยู่ในรูปที่ 3-8 ซึ่งเป็นกรณีสำหรับการเดินเท้า

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

source	target	distance	cost	transport_mode	speed	time_start	time_end	bus_num_start	bus
001003	037002	112.59	81	walk	5	00:00	24:00	-	
001016	036022	320.06	230	walk	5	00:00	24:00	-	
001012	037014	293.92	212	walk	5	00:00	24:00	-	
344100	001003	559.24	403	walk	5	00:00	24:00	-	
344041	001004	196.22	141	walk	5	00:00	24:00	-	
459001	001004	175.16	126	walk	5	00:00	24:00	-	
459036	001005	148.43	107	walk	5	00:00	24:00	-	
459037	001005	106.95	77	walk	5	00:00	24:00	-	
459006	001005	145.98	105	walk	5	00:00	24:00	-	
459007	001005	177.96	128	walk	5	00:00	24:00	-	
420040	001006	138.26	100	walk	5	00:00	24:00	-	
420015	001007	88.39	64	walk	5	00:00	24:00	-	
420043	001008	215.76	155	walk	5	00:00	24:00	-	
420048	001009	200.79	145	walk	5	00:00	24:00	-	
420047	001009	100.14	72	walk	5	00:00	24:00	-	
420050	001010	20.62	15	walk	5	00:00	24:00	-	
364017	001011	454.81	327	walk	5	00:00	24:00	-	
364018	001011	310.24	223	walk	5	00:00	24:00	-	
364003	001011	456.63	329	walk	5	00:00	24:00	-	
364004	001011	340.44	245	walk	5	00:00	24:00	-	
559004	001012	343.39	247	walk	5	00:00	24:00	-	
559005	001012	337.05	243	walk	5	00:00	24:00	-	
420018	001013	131.10	94	walk	5	00:00	24:00	-	

รูปที่ 3-8 ข้อมูลความสัมพันธ์เชิงโครงข่ายที่จัดเก็บในรูปแบบตารางในซอฟต์แวร์ Spreadsheet

3. การตรวจสอบข้อมูลความสัมพันธ์เชิงโครงข่าย

เนื่องจากข้อมูลต่างๆ ที่ใช้ในงานวิจัยนี้ทำการสร้างด้วยมือ มิได้สร้างโดยอัตโนมัติด้วยโปรแกรมเฉพาะ ดังนั้นเมื่อได้ข้อมูลความสัมพันธ์เชิงโครงข่ายแล้ว จะต้องมีการตรวจสอบอีกขั้นหนึ่งก่อน เพื่อตรวจสอบความผิดพลาดที่อาจเกิดขึ้นได้จากความผิดพลาดของมนุษย์ โดยกฎที่ใช้ในการตรวจสอบมีดังนี้

- จะต้องไม่มีรหัสของ arc ที่ซ้ำซ้อนกัน
- ทิศทางของ arc สำหรับรถไฟฟ้า เรือ และการเดินเท้าเพื่อเปลี่ยนประเภทของระบบขนส่งมวลชนจะต้องเป็นสองทิศทางเสมอ หมายความว่ามี arc สองเส้นเชื่อมระหว่าง vertex หนึ่งคู่ และค่ารหัสของ arc จะทำหน้าที่บ่งบอกทิศทาง
- ทิศทางของ arc สำหรับรถโดยสารประจำทางจะต้องเป็นทิศทางเดียวเสมอ หมายความว่ามี arc ได้เพียงเส้นเดียวที่เชื่อมระหว่าง vertex หนึ่งคู่ใดๆ
- ระยะทางของ arc ในสองทิศทางสำหรับ vertex หนึ่งคู่ใดๆ จะต้องเท่ากัน (ถ้ามี)

รหัสโปรแกรมที่ใช้สำหรับตรวจสอบข้อมูลโครงข่ายในตารางความสัมพันธ์ แสดงอยู่ในภาคผนวก ก

4. การนำข้อมูลที่จัดเตรียมไว้เข้าจัดเก็บใน PostGIS (ลักษณะรูปร่างเส้น และความสัมพันธ์)

เมื่อการปรับแก้และจัดเตรียมข้อมูลเสร็จเรียบร้อยแล้ว ขั้นตอนต่อไปเป็นการนำข้อมูลที่จัดเตรียมไว้เข้าสู่ฐานข้อมูลสำหรับทำงานร่วมกับโปรแกรมคำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดที่ได้พัฒนาขึ้น งานวิจัยนี้ใช้ซอฟต์แวร์บริหารจัดการฐานข้อมูลที่ชื่อว่า PostgreSQL พร้อมกับส่วนต่อขยายที่ชื่อว่า PostGIS เพื่อทำหน้าที่จัดเก็บข้อมูลพิกัดของเส้น arc และความสัมพันธ์เชิงโครงข่าย

โปรแกรมคำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดจะใช้ข้อมูลความสัมพันธ์เชิงโครงข่าย เพื่อการคำนวณและสืบค้น เมื่อได้ลำดับของ vertex ในการเดินทางแล้ว ก็จะนำลำดับนั้น ไปสืบค้นต่อในตารางที่จัดเก็บข้อมูลพิกัดของเส้น arc เพื่อค้นคืน รูปร่างเส้นทางที่เชื่อมระหว่าง vertex แต่ละคู่แล้วใช้แสดงผลเป็นแผนที่เส้นทางผลลัพธ์ออกบนจอภาพต่อไป ดังนั้นการนำเข้าข้อมูลจึงประกอบด้วยสองตารางดังกล่าว ในส่วนเทคนิควิธีการ เป็นดังนี้

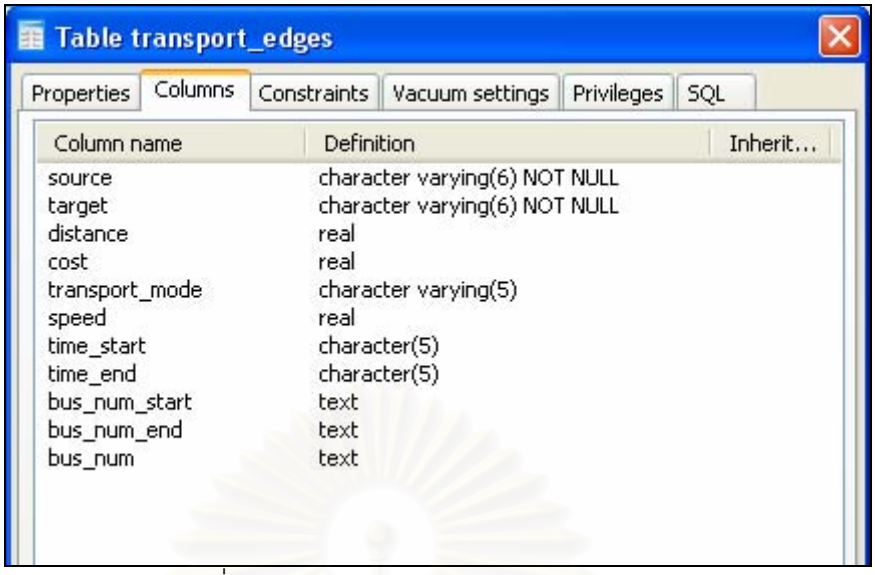
5. การนำเข้าข้อมูลความสัมพันธ์เชิงโครงข่าย

ทำการแปลงข้อมูลความสัมพันธ์เชิงโครงข่ายที่ได้จัดทำเตรียมไว้ในซอฟต์แวร์ Spreadsheet ออกเป็นคำสั่ง SQL ดังตัวอย่างแสดงในรูปที่ 3-9

```
1  INSERT INTO transport_edges VALUES('001003','037002','112.593','81.07','walk','5','1','00:00','24:00','');
2  INSERT INTO transport_edges VALUES('001016','036022','320.059','230.44','walk','5','1','00:00','24:00','');
3  INSERT INTO transport_edges VALUES('001012','037014','293.919','211.62','walk','5','1','00:00','24:00','');
4  INSERT INTO transport_edges VALUES('344100','001003','559.237','402.65','walk','5','1','00:00','24:00','');
5  INSERT INTO transport_edges VALUES('344041','001004','196.215','141.27','walk','5','1','00:00','24:00','');
6  INSERT INTO transport_edges VALUES('459001','001004','175.156','126.11','walk','5','1','00:00','24:00','');
7  INSERT INTO transport_edges VALUES('459036','001005','148.426','106.87','walk','5','1','00:00','24:00','');
8  INSERT INTO transport_edges VALUES('459037','001005','106.948','77.00','walk','5','1','00:00','24:00','');
9  INSERT INTO transport_edges VALUES('459006','001005','145.982','105.11','walk','5','1','00:00','24:00','');
10 INSERT INTO transport_edges VALUES('459007','001005','177.957','128.13','walk','5','1','00:00','24:00','');
11 INSERT INTO transport_edges VALUES('420040','001006','138.261','99.55','walk','5','1','00:00','24:00','');
12 INSERT INTO transport_edges VALUES('420015','001007','88.391','63.64','walk','5','1','00:00','24:00','');
13 INSERT INTO transport_edges VALUES('420043','001008','215.762','155.35','walk','5','1','00:00','24:00','');
14 INSERT INTO transport_edges VALUES('420048','001009','200.788','144.57','walk','5','1','00:00','24:00','');
15 INSERT INTO transport_edges VALUES('420047','001009','100.144','72.10','walk','5','1','00:00','24:00','');
16 INSERT INTO transport_edges VALUES('420050','001010','20.618','14.84','walk','5','1','00:00','24:00');
17 INSERT INTO transport_edges VALUES('364017','001011','454.809','327.46','walk','5','1','00:00','24:00');
18 INSERT INTO transport_edges VALUES('364018','001011','310.235','223.37','walk','5','1','00:00','24:00');
19 INSERT INTO transport_edges VALUES('364003','001011','456.628','328.77','walk','5','1','00:00','24:00');
20 INSERT INTO transport_edges VALUES('364004','001011','340.444','245.12','walk','5','1','00:00','24:00');
```

รูปที่ 3-9 ข้อมูลความสัมพันธ์เชิงโครงข่ายซึ่งถูกแปลงเป็นชุดคำสั่ง SQL แล้ว

สร้างตาราง transport_edges สำหรับเก็บข้อมูลความสัมพันธ์เชิงโครงข่าย โดยมีโครงสร้างแสดงในรูปที่ 3-10



รูปที่ 3-10 โครงสร้างตาราง transport_edges

ผลจากการประมวลผลคำสั่ง SQL จะได้ข้อมูลในตาราง transport_edges ดังแสดงในรูปที่ 3-11

	source [PK] ch.	target [PK] cha	distance real	cost real	transport_mode character varying	speed real	time_start character(5)	time_end character(5)
227	148003	084004	853.51	614.527	walk	5	00:00	24:00
228	148003	148001	686.05	123.489	bus	20	05:00	23:00
229	148003	888005	494.29	355.889	walk	5	00:00	24:00
230	148004	148003	296.04	53.2872	bus	20	05:00	23:00
231	148005	148004	232.75	41.895	bus	20	05:00	23:00
232	148006	148005	389.15	70.047	bus	20	05:00	23:00
233	148007	148006	158.36	28.5048	bus	20	05:00	23:00
234	148008	084004	1046.57	753.53	walk	5	00:00	24:00
235	148008	148010	679.59	122.326	bus	20	05:00	23:00
236	148010	084008	856.22	616.478	walk	5	00:00	24:00
237	148010	148011	530.23	95.4414	bus	20	05:00	23:00
238	148010	888005	494.89	356.321	walk	5	00:00	24:00
239	148011	148012	474.31	85.3758	bus	20	05:00	23:00
240	148012	148013	264.24	47.5632	bus	20	05:00	23:00
241	186002	186001	468.76	84.3768	bus	20	05:00	23:00
242	186003	186004	462.47	83.2446	bus	20	05:00	23:00
243	250001	036023	276.47	199.058	walk	5	00:00	24:00
244	250001	548005	185.49	133.553	walk	5	00:00	24:00

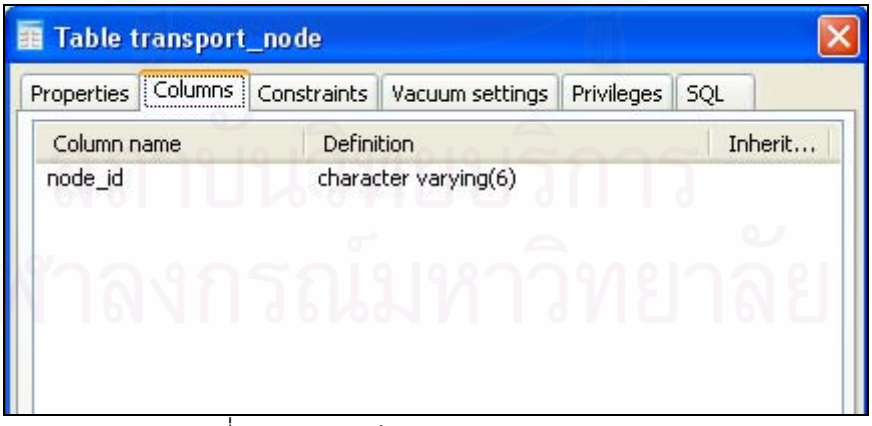
รูปที่ 3-11 ตัวอย่างข้อมูลในตาราง transport_edges หลังจากประมวลผลชุดคำสั่ง SQL

bus_num_start text	bus_num_end text	bus_num text
-	-	-
89,111,84,88,6	530,149,89,ปอ.พ.9,177,164,ปอ.พ.12,3	89, 111, 84, 88, 6
-	-	-
89,111,84,88,6	89,111,84,88,6	89, 111, 84, 88, 6
89,111,84,88,6	89,111,84,88,6	89, 111, 84, 88, 6
89,111,84,88,6	89,111,84,88,6	89, 111, 84, 88, 6
89,111,84,88,6	89,111,84,88,6	89, 111, 84, 88, 6
-	-	-
530,149,89,ปอ.พ.9,177,164,ปอ.พ.12,3,	89,111,84,88,6	89, 111, 84, 88, 6
-	-	-
89,111,84,88,6	89,111,84,88,6	89, 111, 84, 88, 6
-	-	-
89,111,84,88,6	89,111,84,88,6	89, 111, 84, 88, 6
89,111,84,88,6	89,111,84,88,6	89, 111, 84, 88, 6
541,529,172,514,73,163,ปอ.พ.9,157,54	541,529,172,514,73,163,ปอ.พ.9,157,5	541, 529, 172, 514, 73, 163,
541,529,172,514,73,163,ปอ.พ.9,157,54	541,529,172,514,73,163,ปอ.พ.9,157,5	541, 529, 172, 514, 73, 163,
-	-	-
-	-	-

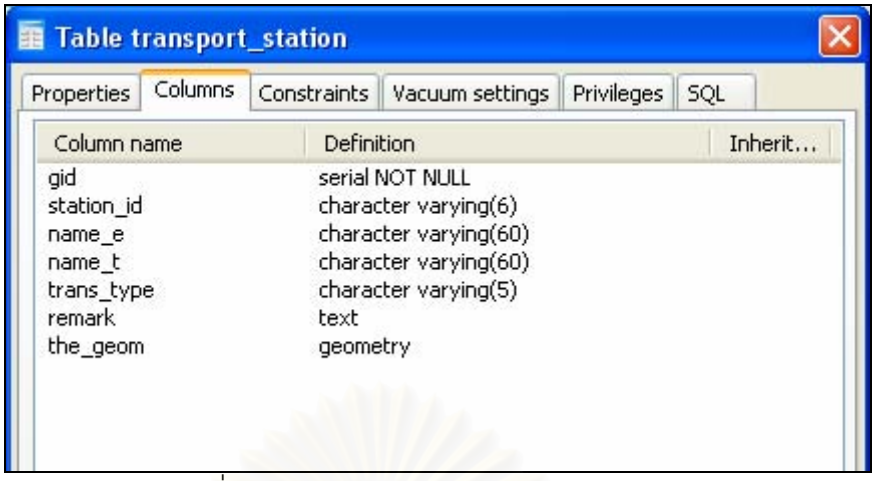
รูปที่ 3-11 (ต่อทางด้านข้าง)

สร้างตารางย่อยอีกสองตารางเพื่อใช้ช่วยประกอบการคำนวณ ตารางย่อยเหล่านี้สร้างขึ้นจากข้อมูลบางส่วนใน transport_edges รหัสโปรแกรมสำหรับสร้างตารางและนำเข้าข้อมูลสู่ตาราง transport_node และ transport_station แสดงอยู่ในภาคผนวก ฉ

ตารางย่อยอีกสองตารางคือ transport_node และ transport_station แสดงโครงสร้างตารางอยู่ในรูปที่ 3-12 และ 3-13



รูปที่ 3-12 โครงสร้างตาราง transport_node

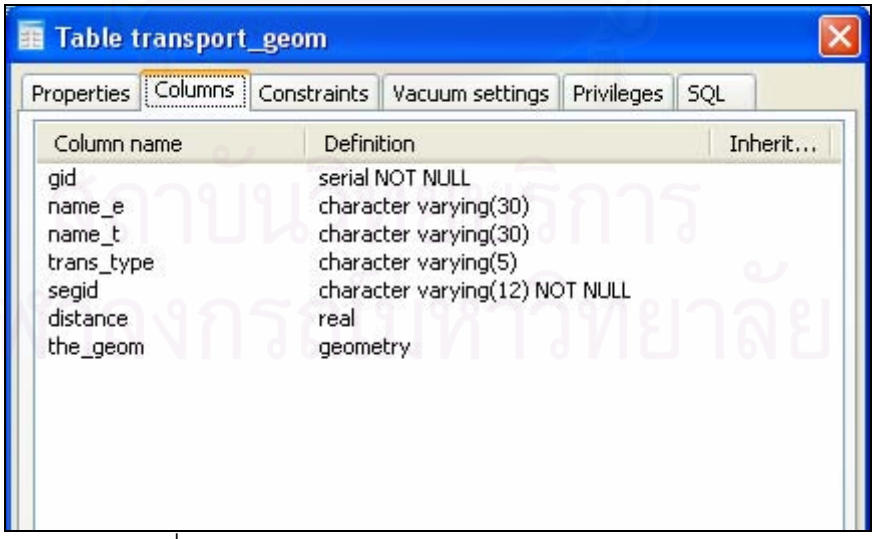


รูปที่ 3-13 โครงสร้างตาราง transport_station

6. การนำเข้าข้อมูลพิกัดของเส้น arc เข้าสู่ตาราง transport_geom

เป็นขั้นตอนเตรียมข้อมูลเวกเตอร์ โดยจัดเก็บแบ่งแยกเป็นส่วนย่อยๆ เรียกว่า arc มีค่ารหัสประจำตัวของตนเองเป็นเอกลักษณ์ และเชื่อมโยงกับตาราง transport_edges โดยรหัสประจำตัวของ arc ใน transport_geom จะเกิดจากสดมภ์ชื่อ source และ target ในตาราง transport_edges

ข้อมูลเวกเตอร์ที่จัดเตรียมไว้ใน transport_geom นี้จะใช้สำหรับสืบค้นรูปร่างของเส้นทางเพื่อใช้แสดงต่อผู้ใช้ หลังจากที่ได้ลำดับของ arc แล้ว โครงสร้างตาราง transport_geom แสดงในรูปที่ 3-14



รูปที่ 3-14 โครงสร้างตาราง transport_geom

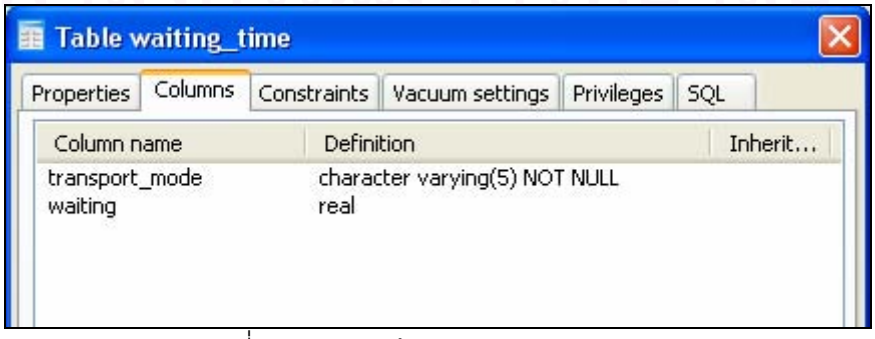
การนำเข้าข้อมูลนั้นใช้โปรแกรมย่อยที่มาพร้อมกับ PostGIS ที่ชื่อ shp2pgsql เพื่อแปลงเป็นชุดคำสั่ง SQL แล้วจึงชุดคำสั่ง SQL นี้ ไปประมวลผลโดย pgAdmin ต่อไป

และเนื่องจากการจัดทำข้อมูลเวกเตอร์ย่อยๆ ของ arc นั้นเป็นการจัดทำแยกกันเป็นชั้นข้อมูล เช่น arc ของเส้นทางรถโดยสารประจำทาง arc ของเส้นทางรถไฟฟ้า arc ของเส้นทางเดินเรือ arc ของเส้นทางเดินเท้า เป็นต้น ดังนั้นจึงต้องมีโปรแกรมย่อยอีกหนึ่งโปรแกรมทำหน้าที่รวมตารางย่อยเหล่านี้เข้าใน transport_geom อีกต่อหนึ่งตัวอย่างข้อมูลหลังจากนำเข้าแล้วแสดงในรูปที่ 3-15 รหัสโปรแกรมสำหรับรวมข้อมูล arc จากแต่ละชั้นข้อมูล แสดงอยู่ในภาคผนวก ช

	gid serial	name_e character	name_t character va	trans_type character va	segid [PK] characte	distance real	the_geom geometry
1	356	Subway	รถไฟฟ้าใต้ดิน	MRT	001001001002	992.617	0102000020877
2	373	Subway	รถไฟฟ้าใต้ดิน	MRT	001002001001	992.617	0102000020877
3	357	Subway	รถไฟฟ้าใต้ดิน	MRT	001002001003	859.317	0102000020877
4	374	Subway	รถไฟฟ้าใต้ดิน	MRT	001003001002	859.317	0102000020877
5	358	Subway	รถไฟฟ้าใต้ดิน	MRT	001003001004	1601.47	0102000020877
6	74	-	walk	Walk	001003037002	112.593	0102000020877
7	167	-	walk	Walk	001003344100	559.237	0102000020877
8	375	Subway	รถไฟฟ้าใต้ดิน	MRT	001004001003	1601.47	0102000020877
9	359	Subway	รถไฟฟ้าใต้ดิน	MRT	001004001005	1290.1	0102000020877
10	168	-	walk	Walk	001004344041	195.149	0102000020877
11	169	-	walk	Walk	001004459001	175.157	0102000020877
12	376	Subway	รถไฟฟ้าใต้ดิน	MRT	001005001004	1290.1	0102000020877
13	360	Subway	รถไฟฟ้าใต้ดิน	MRT	001005001006	1079.1	0102000020877
14	172	-	walk	Walk	001005459006	145.982	0102000020877
15	173	-	walk	Walk	001005459007	177.957	0102000020877
16	170	-	walk	Walk	001005459036	148.426	0102000020877
17	171	-	walk	Walk	001005459037	106.948	0102000020877

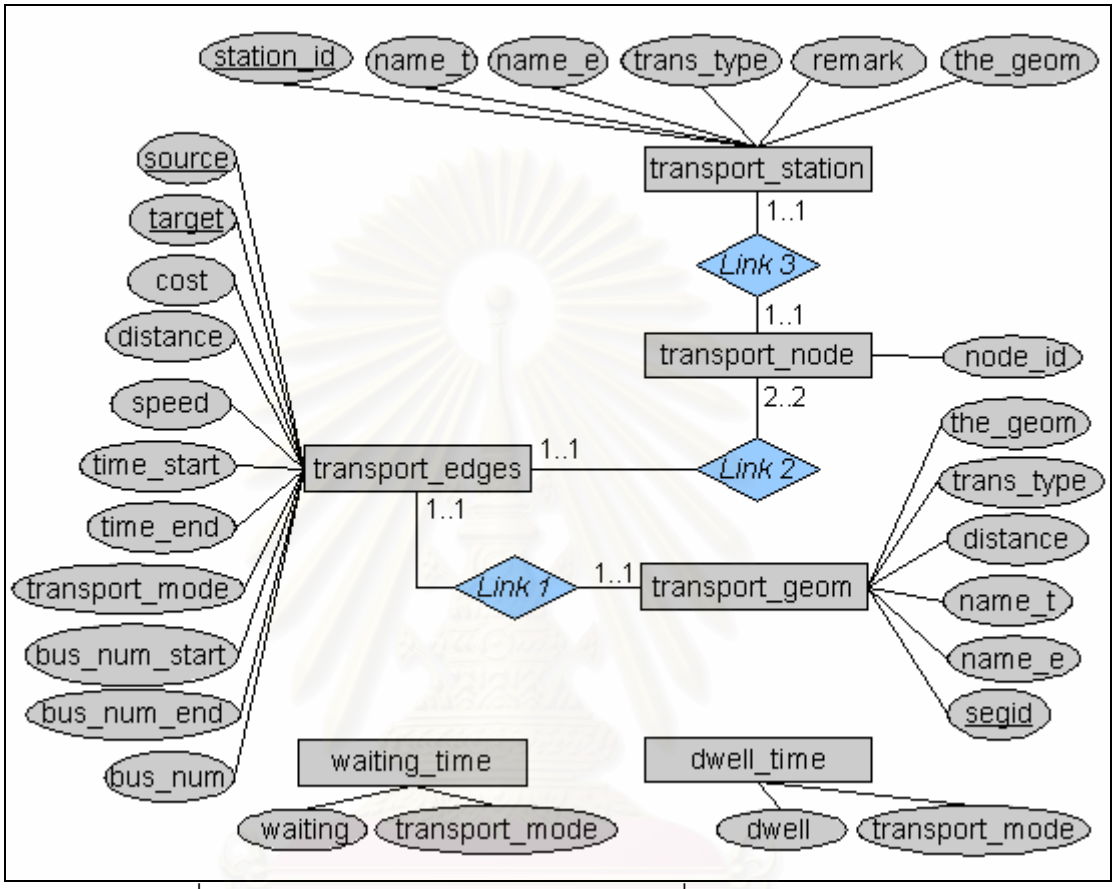
รูปที่ 3-15 ตัวอย่างข้อมูลในตาราง transport_geom

สร้างตาราง waiting_time เพื่อจัดเก็บข้อมูลระยะเวลารอคอยสำหรับระบบขนส่งมวลชนแต่ละประเภท โครงสร้างตาราง waiting_time แสดงอยู่ในรูปที่ 3-16 ระยะเวลารอคอยที่เป็นค่าโดยปริยายสำหรับระบบขนส่งมวลชนแต่ละประเภท ได้แสดงไว้ก่อนหน้านี้แล้วในตาราง 3-10



รูปที่ 3-16 โครงสร้างตาราง waiting_time

โดยสรุปแล้วมีตารางทั้งหมดห้าตาราง คือ transport_edges, transport_station, transport_geom, transport_node และ waiting_time และมีความสัมพันธ์กันดังแสดงสรุปในแผนผัง E-R Diagram ในรูปที่ 3-17



รูปที่ 3-17 E-R Diagram ของโครงสร้างข้อมูลที่จัดเก็บลงฐานข้อมูลแล้ว

จากรูปที่ 3-17 ตาราง transport_edges และ transport_geom มีความสัมพันธ์กันแบบ 1 ต่อ 1 กล่าวคือทุกๆ หนึ่ง arc ในตาราง transport_edges ซึ่งบ่งบอกความสัมพันธ์เชิงโครงข่ายจะเชื่อมโยงไปยัง arc เดียวกันที่จัดเก็บพิกัดเวกเตอร์เส้นทางของ arc นั้นๆ ในตาราง transport_geom โดยผ่านความสัมพันธ์ที่ชื่อว่า “Link 1”

ตาราง transport_edges และ transport_node มีความสัมพันธ์กันแบบ 1 ต่อ 2 กล่าวคือทุกๆ หนึ่ง arc ในตาราง transport_edges ซึ่งบ่งบอกความสัมพันธ์เชิงโครงข่าย จะเชื่อมโยงไปยัง node สอง node ในตาราง transport_node (arc ใดๆ ประกอบด้วยสอง node เชื่อมต่อกันเสมอ) โดยผ่านความสัมพันธ์ที่ชื่อว่า “Link 2”

ในขณะเดียวกัน ตาราง transport_station ยังมีความสัมพันธ์กันแบบ 1 ต่อ 1 กับตาราง transport_node กล่าวคือทุกๆ node ในตาราง transport_station ซึ่งจัดเก็บข้อมูลอรรถาธิบาย

ของสถานีในระบบขนส่งมวลชน จะเชื่อมโยงไปยัง node เดียวกันที่จัดเก็บพิกัดตำแหน่งของสถานีนั้นๆ ในตาราง transport_node โดยผ่านความสัมพันธ์ที่ชื่อว่า “Link 3”

การพัฒนาโปรแกรมเพื่อคำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด

<u>เครื่องมือที่ใช้ในการประมวลผลเส้นทางการเดินทาง</u>

โปรแกรมประมวลผลข้อมูลเพื่อคำนวณและค้นหาเส้นทางการเดินทางที่ใช้เวลาน้อยที่สุดจะทำงานบนเครื่องแม่ข่าย รับคำสั่งและส่งผลลัพธ์ผ่านเครือข่ายอินเทอร์เน็ต/อินทราเน็ต เพื่อให้สามารถให้บริการกับผู้ใช้หลายคนพร้อมกันได้ โดยการเป็นออกแบบและพัฒนาโดยผู้วิจัยเอง

ขั้นตอนวิธี (Algorithm) ที่ใช้ประกอบการคำนวณและค้นหาเส้นทางการเดินทางที่ใช้เวลาน้อยที่สุดนั้น สร้างขึ้นโดยพยายามอิงจากกิจกรรมต่างๆ ที่เกิดขึ้นจริงในการเดินทางเพื่อให้ผลลัพธ์จากการคำนวณมีความใกล้เคียงกับการเดินทางจริง นอกจากนี้ยังได้ออกแบบโครงสร้างข้อมูลเกี่ยวกับโครงข่ายระบบขนส่งมวลชนให้มีการจัดเก็บค่าของตัวแปรต่างๆ ไว้ในฐานข้อมูลด้วย เพื่อให้ไม่จำเป็นต้องเป็นค่าตายตัว สามารถปรับเปลี่ยนได้โดยแก้ไขค่าที่บันทึกอยู่ในฐานข้อมูล (หากมีข้อมูลที่แปรผันตาม วัน เวลาและสถานที่ป้อนให้กับระบบ หรือมีกฎเกณฑ์ในการทำนายค่าดังกล่าวมาให้)

สำหรับขั้นตอนการทำงานของโปรแกรมฯ ประกอบด้วยสามส่วนหลักๆ คือ

1. ส่วนการรับค่าจากผู้ใช้และเตรียมข้อมูลเบื้องต้น
2. ส่วนการค้นหาเส้นทาง
3. ส่วนการแสดงผลลัพธ์

1. ส่วนการรับค่าจากผู้ใช้และเตรียมข้อมูลเบื้องต้น

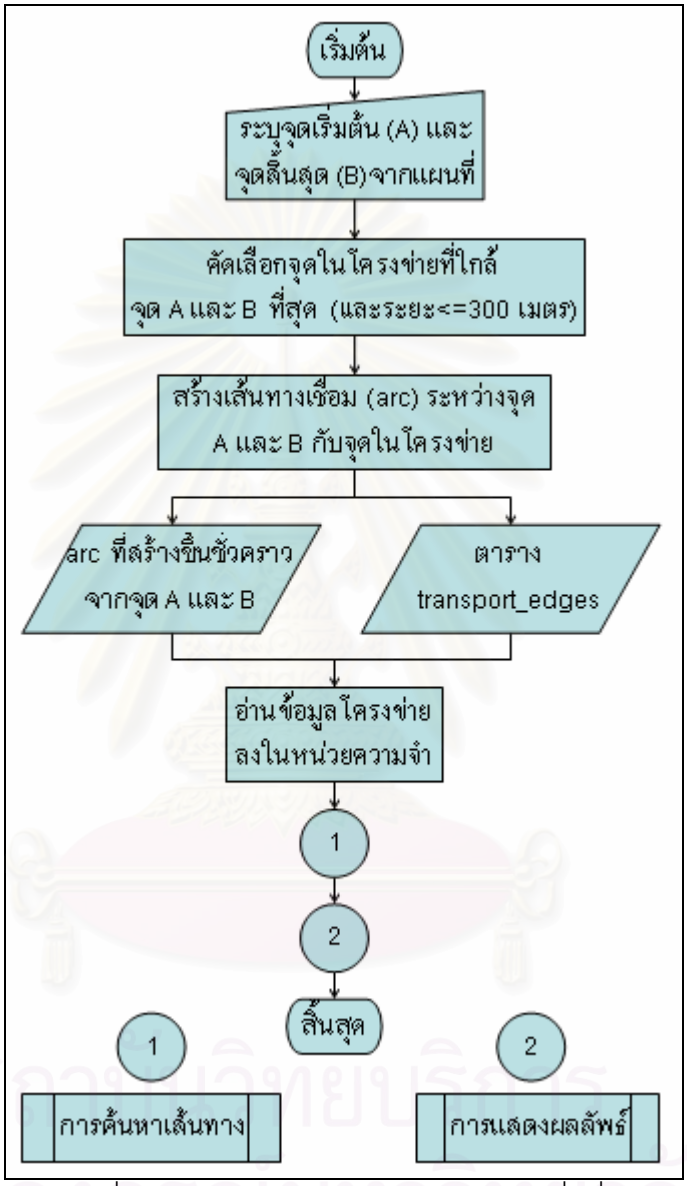
ส่วนการรับค่าจากผู้ใช้และเตรียมข้อมูลเบื้องต้น แสดงอยู่ใน Flow chart ในรูปที่ 3-18 โดยมีขั้นตอนดังนี้

1.1 รับค่าพิกัดตำแหน่งจุดเริ่มต้นและจุดสิ้นสุดการเดินทางจากผู้ใช้ ผ่านการคลิกกำหนดตำแหน่งบนแผนที่

1.2 คัดเลือก node จากในโครงข่าย (ตำแหน่งป้ายหรือสถานีของระบบขนส่งมวลชน) ที่อยู่ใกล้กับจุดเริ่มต้น (A) และจุดสิ้นสุด (B) มากที่สุด ทั้งนี้เนื่องจากตั้งสมมติฐานว่าผู้เดินทางใช้การเดินเท้าเพื่อเข้าสู่ระบบขนส่งมวลชนที่สถานีแรก และเดินเท้าจากสถานีสุดท้ายไปยังจุดสิ้นสุดการเดินทาง ต้องการระยะการเดินเท้าที่สั้นที่สุด

1.3 สร้างเส้นทางเชื่อมขึ้นชั่วคราวระหว่าง node ในโครงข่าย กับ จุด A และ จุด B ทั้งนี้เพื่อไว้ใช้แสดงเส้นทางการเดินสำหรับผลลัพธ์

1.4 เพิ่มเส้นทางเชื่อมสองเส้นทางที่ได้จากข้อ 1.3 เข้าเป็นส่วนหนึ่งในโครงข่าย แล้วอ่านข้อมูลความสัมพันธ์เชิงโครงข่ายทั้งหมดเข้าสู่หน่วยความจำ



รูปที่ 3-18 Flow chart ของการทำงานส่วนที่หนึ่ง

2. ส่วนการค้นหาเส้นทาง

ส่วนการค้นหาเส้นทาง แสดงอยู่ใน Flow chart ในรูปที่ 3-19 โดยมีขั้นตอนดังนี้

2.1 การจัดเตรียมค่าตัวแปรต่างๆ ที่จำเป็นในการคำนวณ เป็นการอ่านข้อมูลความสัมพันธ์เชิงโครงข่ายบรรจุลงในตัวแปร และกำหนดค่าเริ่มต้นให้กับตัวแปรต่างๆ ที่ใช้ประกอบการคำนวณ

2.2 อ่านค่าความสัมพันธ์ของ node และ arc และข้อมูลอรรถาธิบายอื่นที่เกี่ยวข้องเพื่อนำเข้ามาพิจารณา

2.3 ตรวจสอบช่วงเวลาให้บริการของระบบขนส่งมวลชนบน arc นั้น โดยหากเวลาเมื่อเดินทางไปถึง node ต้นทางของ arc นั้นๆ แล้วพบว่าอยู่นอกช่วงเวลาให้บริการ ก็จะไม่นำ arc นั้นมาพิจารณาในกระบวนการของ Dijkstra's algorithm

2.4 หากอยู่ในช่วงเวลาให้บริการ ให้ตรวจสอบชนิดของระบบขนส่งมวลชนที่กำลังพิจารณาหากเป็นรถโดยสารประจำทางให้เข้าสู่กระบวนการพิจารณาหมายเลขสายรถโดยสารประจำทางที่เหมาะสมก่อน เพื่อคัดเลือกสายรถที่ทำให้เกิดจำนวนครั้งในการต่อรถน้อยที่สุด พร้อมทั้งเตรียมพิจารณาเรื่องระยะเวลารอคอยหากตรวจสอบพบว่าจำเป็นต้องเปลี่ยนสายรถ

2.5 อ่านค่าระยะเวลารอคอยและระยะเวลาจอดรอรับส่งผู้โดยสาร อัตราเร็วและระยะทาง ที่สอดคล้องกับระบบขนส่งมวลชนที่กำลังพิจารณา มาร่วมคำนวณระยะเวลาเดินทางผ่าน arc นั้นๆ แล้วจึงเข้าสู่กระบวนวิธีการคัดเลือก node ตาม Dijkstra's Algorithm

2.6 เมื่อพิจารณา arc จนหมดทั้งโครงข่ายแล้วจะได้ค่าใหม่ของชุดตัวแปรในข้อ 2.1 โดยเป็นผลจากการประมวลผลด้วย Dijkstra's algorithm

2.7 อ่านค่าลำดับของ node ที่ประกอบกันเป็นเส้นทางผลลัพธ์ออกจากชุดตัวแปรในข้อ 2.4

3. ส่วนการแสดงผลลัพธ์

ส่วนการแสดงผลลัพธ์แสดงอยู่ใน Flow chart ในรูปที่ 3-20 โดยมีขั้นตอนดังนี้

3.1 เริ่มต้นจากลำดับของ node และ arc ที่ประกอบกันเป็นเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด (ผลจากส่วนการทำงานที่สองของโปรแกรมฯ)

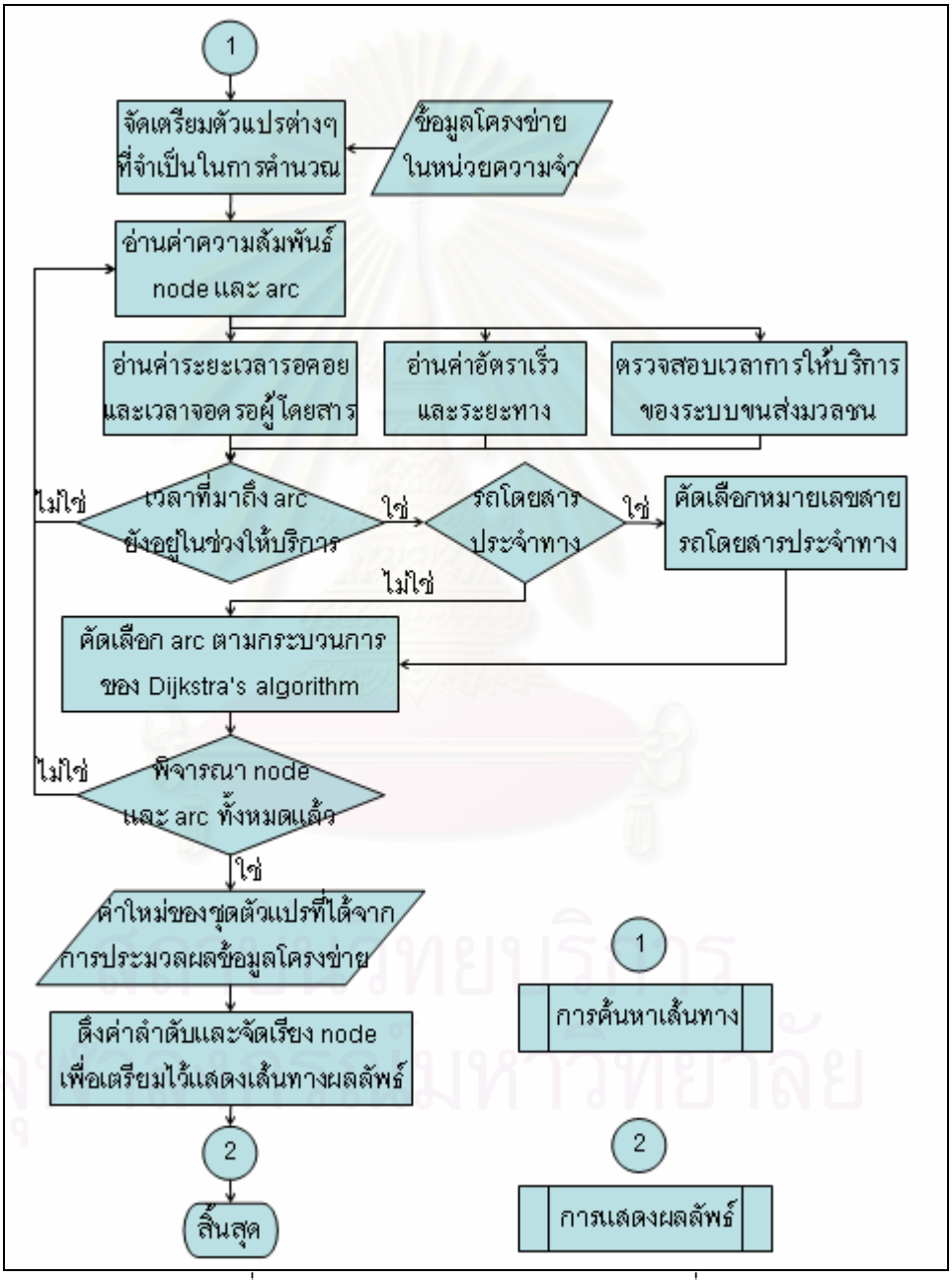
3.2 ดึงค่าพิกัดของเส้นทาง arc ออกมาโดยอาศัยค่ารหัสประจำ arc เป็นค่าสำหรับอ้างอิงไปยังตาราง transport_geom ซึ่งใช้จัดเก็บเวกเตอร์ของเส้นทาง arc ทั้งหมดในโครงข่าย

3.3 สั่งให้ Minesota Map Server สร้างเส้นแผนที่เส้นทางจากชุดพิกัดเวกเตอร์ที่ได้จากข้อ 3.2

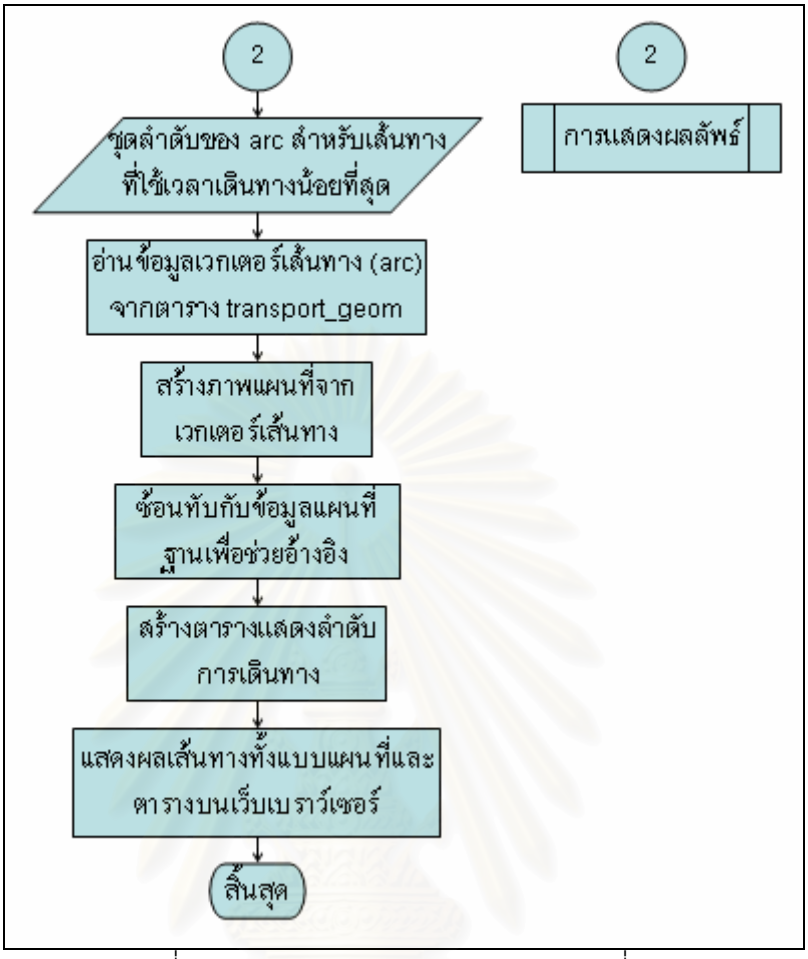
3.4 ซ้อนทับข้อมูลแผนที่เส้นทางกับแผนที่ที่ฐานสำหรับช่วยอ้างอิงตำแหน่ง (ถนน ทางน้ำ และขอบเขตการปกครอง)

3.5 จัดเตรียมข้อมูลเส้นทางผลลัพธ์เพื่อแสดงในรูปแบบตาราง โดยแสดงแยกเป็นช่วงๆ (arc) ในแต่ละช่วงแสดงรูปแบบการเดินทาง ระยะเวลาที่ใช้ ระยะทางของ arc เวลา/วันที่เดินทางถึง node ปลายทาง และหมายเหตุ ซึ่งในที่นี้ใช้แสดงหมายเลขสายรถประจำทาง

3.6 นำผลลัพธ์ทั้งในรูปตารางและแผนที่แสดงออกทางเว็บเบราว์เซอร์



รูปที่ 3-19 Flow-chart ของการทำงานส่วนที่สอง



รูปที่ 3-20 Flow-chart ของการทำงานส่วนที่สาม

รหัสโปรแกรมสำหรับโปรแกรมทั้งหมดแสดงอยู่ในภาคผนวก ข

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4
ผลการวิเคราะห์ข้อมูล

ผลการวิเคราะห์ข้อมูลพฤติกรรมของระบบขนส่งมวลชน (การออกแบบโครงสร้างข้อมูล)

ผลจากการวิเคราะห์และออกแบบโครงสร้างข้อมูล ผู้วิจัยมีความเห็นและสรุปเป็นประเด็นได้ดังนี้

1. การเดินเท้าเป็นรูปแบบการเดินทางที่ต้องใช้เพื่อเชื่อมหรือเปลี่ยนรูปแบบของระบบขนส่งมวลชน และควรมีการจัดทำเส้นทางเชื่อมต่อ (topology) และความสัมพันธ์ระหว่าง node เตรียมไว้ล่วงหน้าในฐานข้อมูล การจัดทำนี้อาจจะเป็นการจัดทำโดยคน หรือกรณีข้อมูลมีปริมาณมากอาจทำแบบอัตโนมัติโดยโปรแกรมก็ได้ ซึ่งจะต้องมีการพัฒนาโปรแกรมขึ้นมาเฉพาะ พร้อมทั้งต้องให้โปรแกรมสามารถตรวจสอบความถูกต้องของเส้นทางเชื่อมต่อ ที่สร้างขึ้นได้ด้วย เช่น การสร้างเส้นทางเชื่อมต่อหลายเส้นทางกรณีมีจุดเชื่อมต่อมากกว่า 1 จุด เส้นทางเชื่อมต่อต้องไม่มีการพาดผ่านแม่น้ำ หากต้องมีการข้ามถนนก็จะต้องตรวจสอบด้วยว่าเป็นถนนที่อนุญาตให้ผู้เดินเท้าเดินข้ามได้ ไม่มีแผงกั้น เป็นต้น

2. อัตราเร็วของพาหนะที่ไม่มีระบบทางวิ่งเป็นการเฉพาะ สามารถแปรผันได้มาก ขึ้นกับปริมาณการจราจร เช่น รถโดยสารประจำทาง ซึ่งต้องใช้ถนนร่วมกับรถโดยสารส่วนบุคคล

3. ทุกครั้งที่มีการเปลี่ยนรูปแบบ (ประเภท) ของระบบขนส่งมวลชน ก็จะเกิดระยะเวลารอคอยขึ้น ระยะเวลารอคอยนี้สำหรับระบบขนส่งมวลชนระบบรางจะมีช่วงค่าที่แน่นอน แต่สำหรับระบบขนส่งมวลชนซึ่งต้องใช้ถนนร่วมกับพาหนะอื่นหลายประเภทช่วงของค่าที่เป็นไปได้จะกว้างกว่ามาก

4. ข้อมูลที่จำเป็นหากแบ่งตามจุดประสงค์ในการประมวลผลจะแบ่งได้สองกลุ่ม คือ
   - กลุ่มที่หนึ่งเวกเตอร์เส้นทางที่เชื่อมระหว่าง vertex ซึ่งจัดเก็บในรูปแบบของข้อมูลเชิงตำแหน่ง (Linestring) เพื่อใช้แสดงกราฟิกของเส้นทางบนจอภาพ
   - กลุ่มที่สองข้อมูลความสัมพันธ์เชิงโครงข่าย (network topology) ของ vertex ทั้งหมด ซึ่งจัดเก็บในรูปแบบตารางในฐานข้อมูลเพื่อใช้สำหรับคำนวณและสืบค้นเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด

   แม้ข้อมูลทั้งสองกลุ่มจะถูกจัดเก็บแยกจากกัน แต่ก็มีความสัมพันธ์ต่อกันโดยตรง เช่น ความยาวของเส้นเวกเตอร์ที่เปลี่ยนแปลงไปจะกระทบต่อความยาวของ arc ที่บันทึกไว้ในข้อมูลความสัมพันธ์เชิงโครงข่ายด้วย

5. ระยะเวลาที่เกิดขึ้นในการเดินทาง อาจเป็นระยะเวลาโดยตรง เช่น ระยะเวลารอคอยยานพาหนะ ระยะเวลาจอดรอรับส่งผู้โดยสาร หรืออาจเป็นส่วนกลับของอัตราเร็วของยานพาหนะและระยะทาง

6. ระยะเวลารอคอยอาจมีความแตกต่างกันในแต่ละสายรถโดยสารประจำทาง ทั้งนี้เนื่องจากในแต่ละสายรถนั้นจะมีความแตกต่างกันหลายปัจจัย ได้แก่ ช่วงเวลาที่เริ่มต้นและสิ้นสุดการให้บริการ จำนวนรถวิ่งให้บริการ ระยะสายทาง ระยะ Headway และปริมาณการจราจรที่จะแตกต่างกันไปในแต่ละพื้นที่ แต่ละเส้นทาง และช่วงเวลา

7. ตัวแปรบางตัว ยังเป็นผลจากอนุพันธ์ของตัวแปรอื่นได้อีก เช่น
   - ระยะเวลารอคอยยานพาหนะ เป็นอนุพันธ์ของความถี่ในการปล่อยยานพาหนะ สายรถประจำทาง และสภาพการจราจร
   - อัตราเร็วของยานพาหนะเป็นอนุพันธ์ของสภาพการจราจร และชนิดของระบบขนส่งมวลชน

   นอกจากนี้ยังอาจเกิดจากเหตุการณ์เฉพาะบางอย่างซึ่งมีผลต่ออัตราเร็ว เช่น อุบัติเหตุ การซ่อมแซม ก่อสร้าง งานเทศกาลรื่นเริง เป็นต้น ในทางปฏิบัติอาจต้องมีระบบย่อยๆ คอยตรวจสอบและคำนวณค่าต่างๆ เพื่อป้อนให้กับระบบแนะนำเส้นทางอีกต่อหนึ่ง

8. ตำแหน่งของป้ายหยุดรถโดยสารประจำทางมีโอกาสเปลี่ยนแปลงได้บ่อยกว่าสถานีของระบบขนส่งขนส่งมวลชนประเภทอื่นมาก เนื่องจากเป็นการก่อสร้างแบบไม่ถาวร และมีการเปลี่ยนแปลงของถนนและการจัดการจราจรบ่อย

9. ในกรณีของรถโดยสารประจำทางนั้นการพิจารณาจำนวนครั้งการเปลี่ยนสายรถโดยสารประจำทางนั้นมีความสำคัญมาก เนื่องจากระยะเวลารอคอยรถประจำทางจะยาวนานกว่ายานพาหนะประเภทอื่น เส้นทางที่มีระยะทางเท่าๆ กันแม้จะมีอัตราเร็วเฉลี่ยของรถเท่ากัน แต่หากต้องเปลี่ยนสายรถมาก ก็อาจทำให้ระยะเวลาเดินทางต่างกันได้มาก

10. ในกรณีของรถโดยสารประจำทาง จำนวนป้ายที่ต้องหยุดจอดระหว่างทางก็มีผลต่อระยะเวลาการเดินทางเช่นกัน เช่น เส้นทางที่มีระยะทางเท่ากัน แต่มีจำนวนป้ายแตกต่างกันก็จะทำให้ระยะเวลาการเดินทางต่างกันมากน้อยขึ้นกับ ความแตกต่างของจำนวนป้าย นอกจากนี้ระยะเวลาที่ใช้ในการเข้าจอดและออกจากป้ายแต่ละป้ายจะยิ่งมากขึ้นหากสภาพการจราจรแออัดและมีการฝ่าฝืนกฎจราจร

ผลการออกแบบและพัฒนาโปรแกรมฯ

โปรแกรมที่พัฒนาขึ้นจะดำเนินตามขั้นตอนวิธีของ Dijkstra พร้อมทั้งผู้วิจัยได้ปรับแต่งเพิ่มปัจจัยต่างๆ เข้าไปในการพิจารณาด้วย ดังนี้

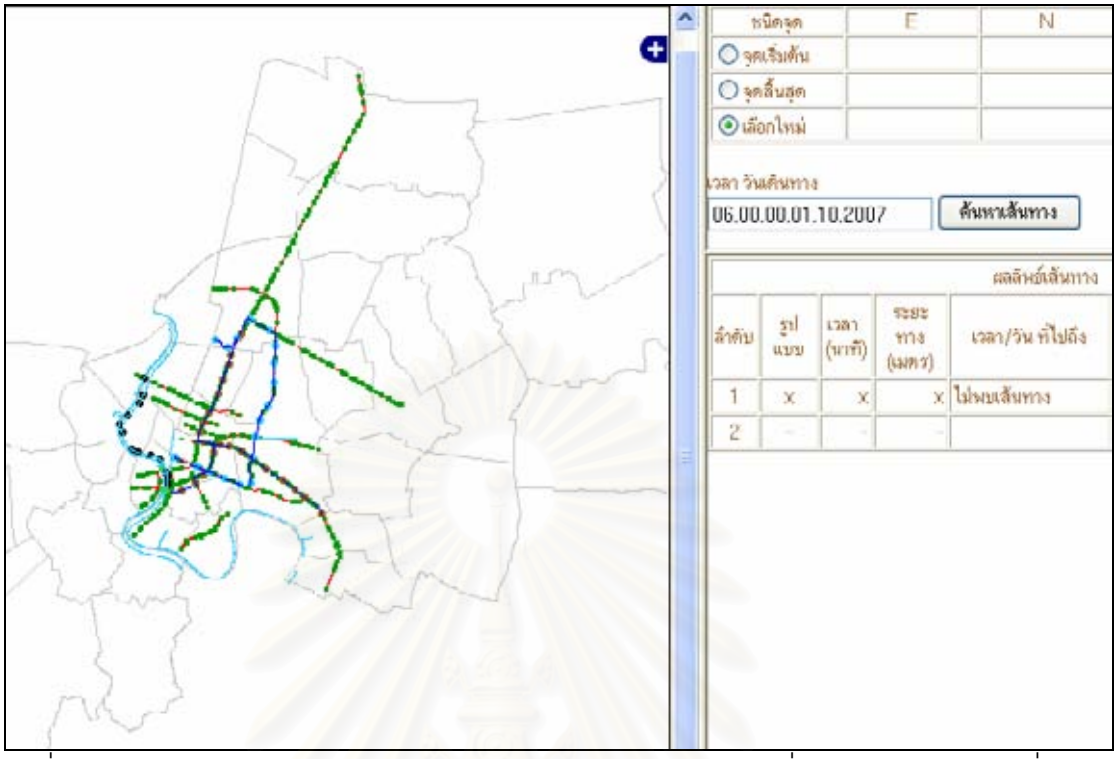
1. การพิจารณาเรื่องเวลาที่ระบบขนส่งมวลชนแต่ละระบบมีให้บริการหรือสามารถเดินทางผ่าน arc ใดๆ ที่กำลังพิจารณาได้ การทำเช่นนี้จะทำให้สามารถนำปัจจัยเรื่องตารางเวลาการให้บริการของระบบขนส่งมวลเข้ามาร่วมพิจารณาได้ด้วย
2. การพิจารณาเรื่องอัตราเร็วของระบบขนส่งมวลชนที่วิ่งผ่านแต่ละ arc ในโครงข่ายแยกออกจากกันอิสระ ทำให้สามารถปรับแก้ค่าอัตราเร็วให้เหมาะสมได้ก่อนทำการคำนวณ อีกนัยหนึ่งคือ สามารถรองรับอัตราเร็วที่เปลี่ยนแปลงตามสภาพการจราจรซึ่งขึ้นกับพื้นที่และช่วงเวลาได้*

<u>หมายเหตุ</u> *โครงสร้างข้อมูลเพียงถูกออกแบบให้รองรับกรณีดังกล่าวเท่านั้น ในงานวิจัยนี้ยังไม่มีระบบหรือโปรแกรมย่อยที่จะทำหน้าที่ปรับเปลี่ยนค่าต่างๆ แบบพลวัต เพียงแต่ค่าตัวแปรต่างๆ ถูกจัดเก็บไว้ในฐานข้อมูลและเปิดช่องทางให้เข้าไปปรับแก้ได้ด้วยภาษา SQL และซอฟต์แวร์ที่มีความสามารถเชื่อมต่อกับฐานข้อมูล

หน้าต่างโปรแกรมแบ่งออกเป็นสามส่วน ดังแสดงในรูปที่ 4-1 ส่วนที่หนึ่งแผนที่แสดงเส้นทางระบบขนส่งมวลชนและสถานี ทางด้านซ้ายมือ ส่วนที่สองแบบฟอร์มรับข้อมูลการเดินทาง ด้านมุมบนขวามือ และส่วนที่สามแสดงผลเส้นทาง ในตารางทางด้านขวามือ

ผู้ใช้สามารถคลิกเลือกจุดเริ่มต้นและจุดหมายปลายทางผ่านทางหน้าเว็บ ดังกล่าวได้จากบนแผนที่ด้านซ้ายมือ พิกัดของจุดที่คลิกจะปรากฏในตารางในแบบฟอร์มรับข้อมูล หากต้องการยกเลิกจุดเดิมและกำหนดจุดใหม่ให้คลิกที่ตัวเลือก “เลือกใหม่”

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4-1 ส่วนโต้ตอบกับผู้ใช้ของโปรแกรมคำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด

ตัวอย่างผลลัพธ์จากการทดลองสืบค้นเส้นทาง

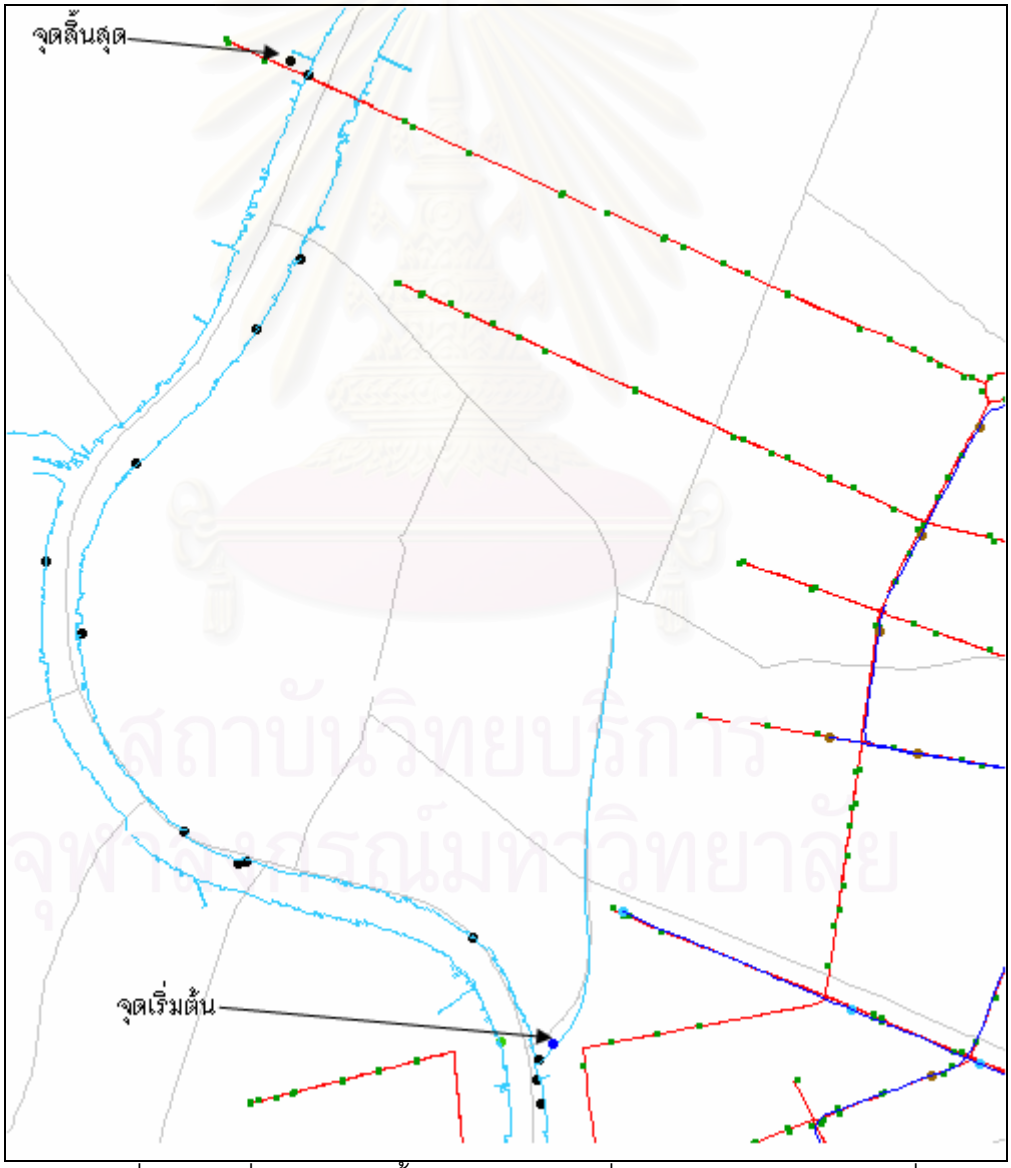
ในที่นี้ขอนำเสนอตัวอย่างการสืบค้นเส้นทางในสองลักษณะคือ

1. การเปรียบเทียบการเดินทางโดยทางเรือ ณ เวลาที่มีการเปิดให้บริการ (หลัง 8:00 น.) และยังไม่เปิดให้บริการ (ก่อน 8:00 น.) และเปรียบเทียบการเดินทางโดยรถโดยสารประจำทางในเส้นทางเดียวกัน
2. การเดินทางผสมรวมสองรูปแบบคือเรือข้ามฟาก และรถไฟฟ้าบีทีเอส

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1. การเปรียบเทียบการเดินทางโดยทางเรือ ณ เวลาที่มีการเปิดให้บริการ (8:00 น.) และยังไม่เปิดให้บริการ (ก่อน 8:00 น.) และเปรียบเทียบการเดินทางโดยรถโดยสารประจำทางในเส้นทางเดียวกัน

ในการเดินทางนี้ได้สมมติสถานการณ์ว่าผู้เดินทางต้องการเดินทางจากทางตอนใต้ของกรุงเทพมหานคร ไปยังตอนเหนือของกรุงเทพมหานคร ซึ่งมีเส้นทางให้เลือกสองเส้นทาง คือ โดยเรือด่วนตามแม่น้ำเจ้าพระยาและโดยรถโดยสารประจำทาง พร้อมทั้งมีเงื่อนเวลาเข้ามาเกี่ยวข้องด้วย คือ ช่วงก่อนและหลังเวลา 8:00 น. ตำแหน่งจุดเริ่มต้นและสิ้นสุดการเดินทางซึ่งผู้ใช้คลิกกำหนดบนแผนที่แสดงอยู่ในรูปที่ 4-2



รูปที่ 4-2 จุดเริ่มต้นและจุดสิ้นสุดการเดินทางซึ่งกำหนดโดยคลิกบนแผนที่

เมื่อกำหนดตำแหน่งจุดเริ่มต้นและสิ้นสุดแล้วผู้ใช้จะต้องกำหนดเวลาที่เริ่มต้นเดินทางด้วย ในที่นี้กำหนดเป็น 8:00 น. ซึ่งเป็นช่วงเวลาที่เรือเริ่มให้บริการ การระบุเวลาและวันที่ที่เริ่มต้นเดินทาง อยู่ในรูปแบบ “ชั่วโมง, นาที, วินาที, วัน, เดือน, ปี ค.ศ.” เมื่อกรอกเงื่อนไขต่างๆ ครบแล้ว จะปรากฏข้อมูลในแบบฟอร์ม ดังแสดงตัวอย่างในรูปที่ 4-3 แล้วคลิกที่ปุ่ม “ค้นหาเส้นทาง” เพื่อให้โปรแกรมเริ่มทำการคำนวณและสืบค้น

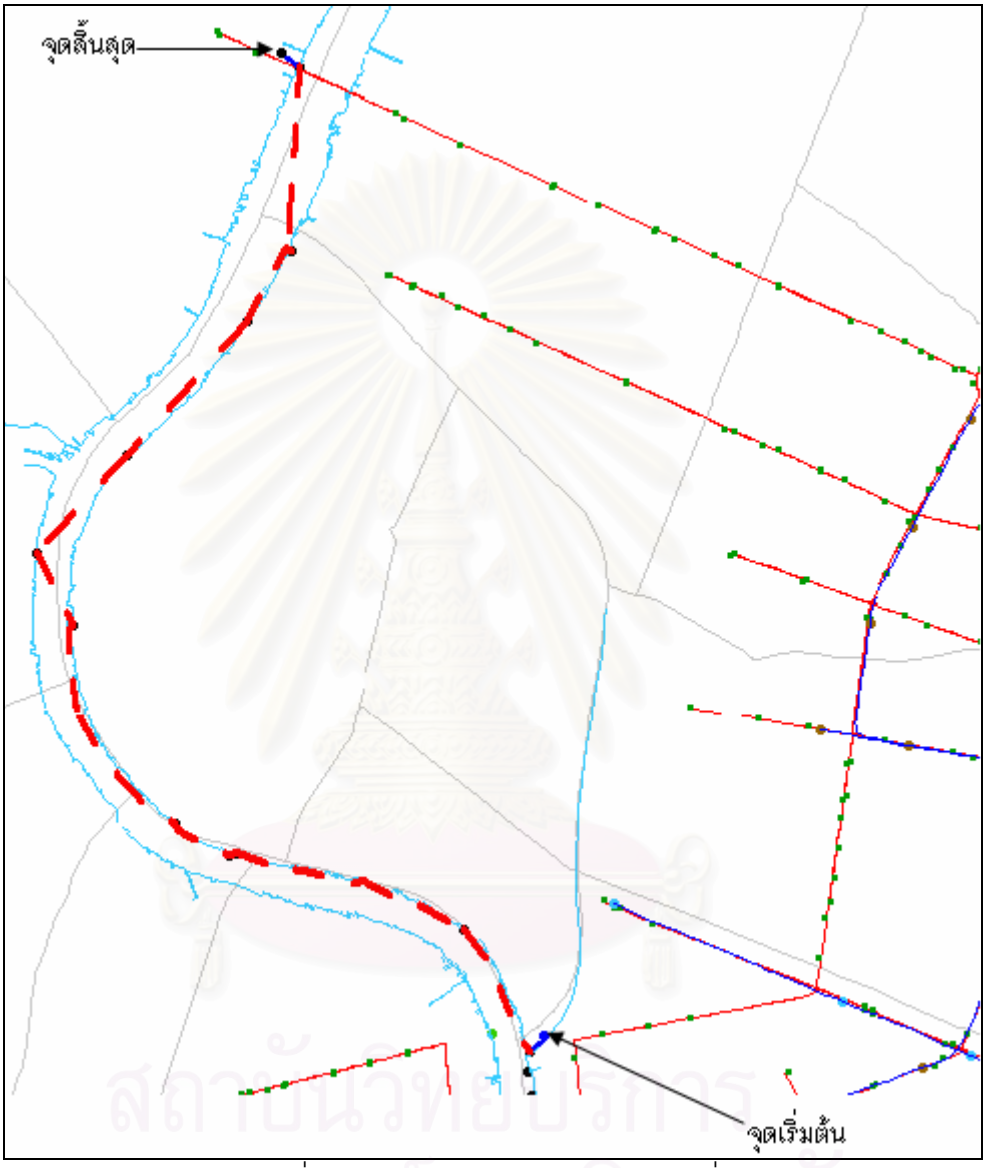
รูปที่ 4-3 แบบฟอร์มสำหรับรับข้อมูลเงื่อนไขการเดินทาง

ผลลัพธ์จากการประมวลผลสามารถแสดงในสองรูปแบบคือ แผนที่เส้นทาง ซึ่งแสดงเป็นเส้นประหนา ตัวอย่างแสดงในรูปที่ 4-4 เส้นประในรูปเป็นเส้นทางที่เกิดจากการเดินทางจากบริเวณท่าเรือทางด้านใต้ของกรุงเทพมหานครไปยังด้านเหนือของกรุงเทพมหานคร รูปขยายของเส้นทางดังกล่าวแสดงในรูปที่ 4-5 ซึ่งแสดงให้เห็นบริเวณจุดเริ่มต้นการเดินทาง นอกจากนี้จะสังเกตเห็นเส้นทึบหนาสั้นๆ เชื่อมระหว่างจุดเริ่มต้นและท่าเรือ ซึ่งเป็นเส้นทางที่สร้างขึ้นโดยอัตโนมัติ ขณะทำการคำนวณและค้นหาเส้นทาง เพื่อใช้เป็นเส้นทางและ arc แทนการเดินเท้าเข้าสู่ท่าเรือ

ผลลัพธ์อีกรูปแบบหนึ่ง คือ ตารางแสดงรายละเอียดข้อมูลจากการประมวลผล โดยเป็นการแสดงรายละเอียดการเดินทางในแต่ละช่วง ประกอบด้วยรูปแบบการเดินทาง เวลาที่เดินทาง ระยะทาง เวลา/วันที่เดินทางถึงปลายช่วง และหมายเหตุ ซึ่งใช้แสดง หมายเลขสายรถโดยสารประจำทาง (ถ้ามี) ตัวอย่างผลลัพธ์ข้อมูลแสดงในรูปที่ 4-6

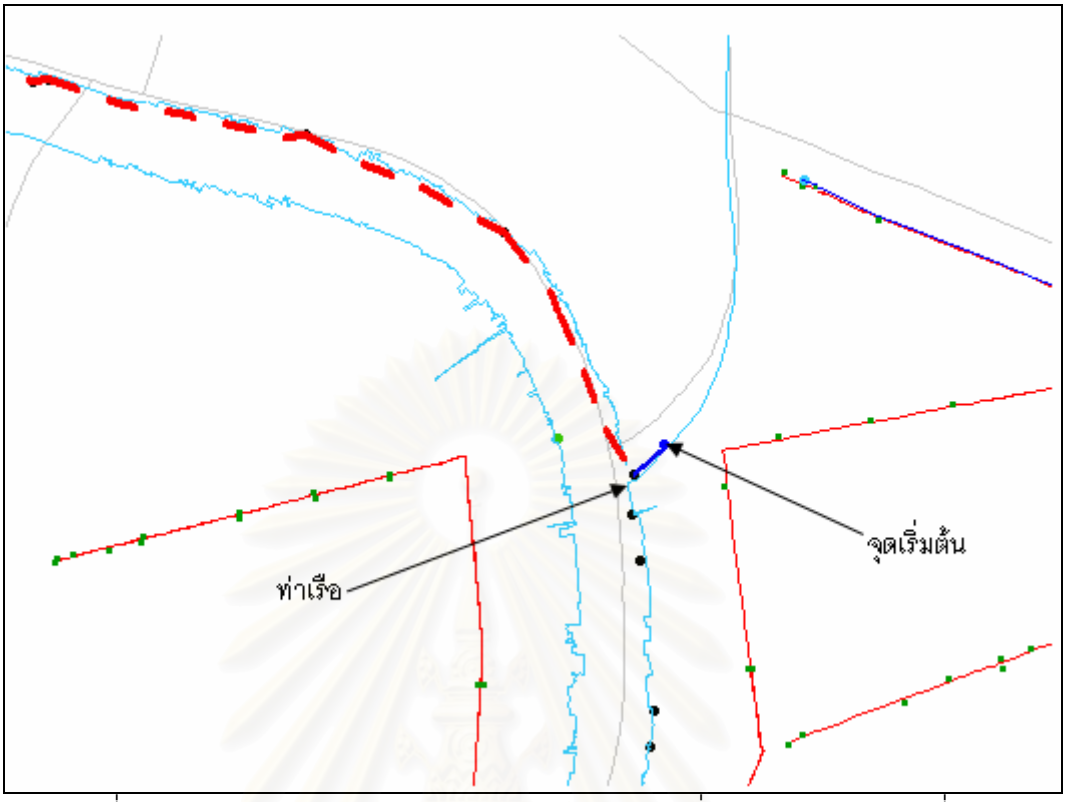
ผลลัพธ์ในตารางจะแสดงเป็น segment ย่อยของเส้นทางที่เชื่อมระหว่างสถานีขนส่งแต่ละคู่ เริ่มจากการเดินเท้าไปสู่ท่าเรือด้วยระยะทาง 115.6 เมตร ใช้เวลาเดินเท้า 1.4 นาที (แสดงในแถวแรกของตาราง) ไปรอเรือเพื่อขึ้นเรือ รวมกับระยะเวลาที่ใช้เดินทางช่วงท่าเรือคู่แรก ระยะเวลารอคอยเรือ 10 นาที (จากตาราง waiting_time ในฐานข้อมูล) รวมกับ ระยะเวลาเดินทางโดยเรือประมาณ 1.6 นาที (จากการคำนวณเทียบกับอัตราเร็วของเรือด่วนที่ 30 กิโลเมตรต่อชั่วโมงที่

ระยะทาง 814.3 เมตร) และระยะเวลาจอดรอรับส่งผู้โดยสาร 10 วินาที (0.2 นาที, สืบค้นจากตาราง dwell_time ในฐานข้อมูล)



รูปที่ 4-4 ตัวอย่างผลลัพธ์ (รูปแบบแผนที่)

สรุปรวมระยะเวลาเดินทางที่เห็นในแถวที่สองของตารางผลลัพธ์เส้นทาง คือ 11.8 นาที ซึ่งเป็นผลรวมที่เกิดจากระยะเวลารอคอย 10 นาที รวมกับระยะเวลาเดินทางโดยเรือ 1.6 นาที และระยะเวลาจอดรอรับส่งผู้โดยสาร 0.2 นาทีแล้ว



รูปที่ 4-5 รูปขยายผลลัพธ์เส้นทางตัดมาจากบริเวณจุดเริ่มต้นเส้นทางจากรูปที่ 4-4

ผลลัพธ์เส้นทาง					
ลำดับ	รูปแบบ	เวลา (นาที)	ระยะทาง (เมตร)	เวลา/วัน ที่ไปถึง	หมายเหตุ
1	walk	1.4	115.6	08:01:23:01:10:2007	-
2	boat	11.8	814.3	08:13:10:01:10:2007	-
3	boat	1.5	643	08:14:38:01:10:2007	-
4	boat	1.7	776.9	08:16:21:01:10:2007	-
5	boat	0.3	46.5	08:16:36:01:10:2007	-
6	boat	0.9	382.7	08:17:32:01:10:2007	-
7	boat	3	1408.4	08:20:31:01:10:2007	-
8	boat	1.1	468.8	08:21:38:01:10:2007	-
9	boat	1.7	779.2	08:23:21:01:10:2007	-
10	boat	2.3	1056.3	08:25:38:01:10:2007	-
11	boat	1.2	503	08:26:48:01:10:2007	-
12	boat	2.3	1090.8	08:29:09:01:10:2007	-
13	walk	1.2	102.6	08:30:23:01:10:2007	-

รูปที่ 4-6 ตัวอย่างผลลัพธ์การสืบค้นเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด (รูปแบบตาราง)

ผลลัพธ์ในตารางจะแสดงทีละช่วงของเส้นทางอย่างนี้เรื่อยไปจนกระทั่งถึงรายการที่ 12 และ 13 ซึ่งเป็นการเดินทางโดยเรือช่วงสุดท้ายและต่อด้วยการเดินเท้าเข้าสู่จุดสิ้นสุดการเดินทาง และเช่นเดียวกัน ช่วงการเดินทางช่วงสุดท้าย (ช่วงที่ 13) เป็นเส้นทางที่สร้างขึ้นโดยอัตโนมัติ เพื่อเชื่อมท่าเรือเข้ากับจุดสิ้นสุดการเดินทาง

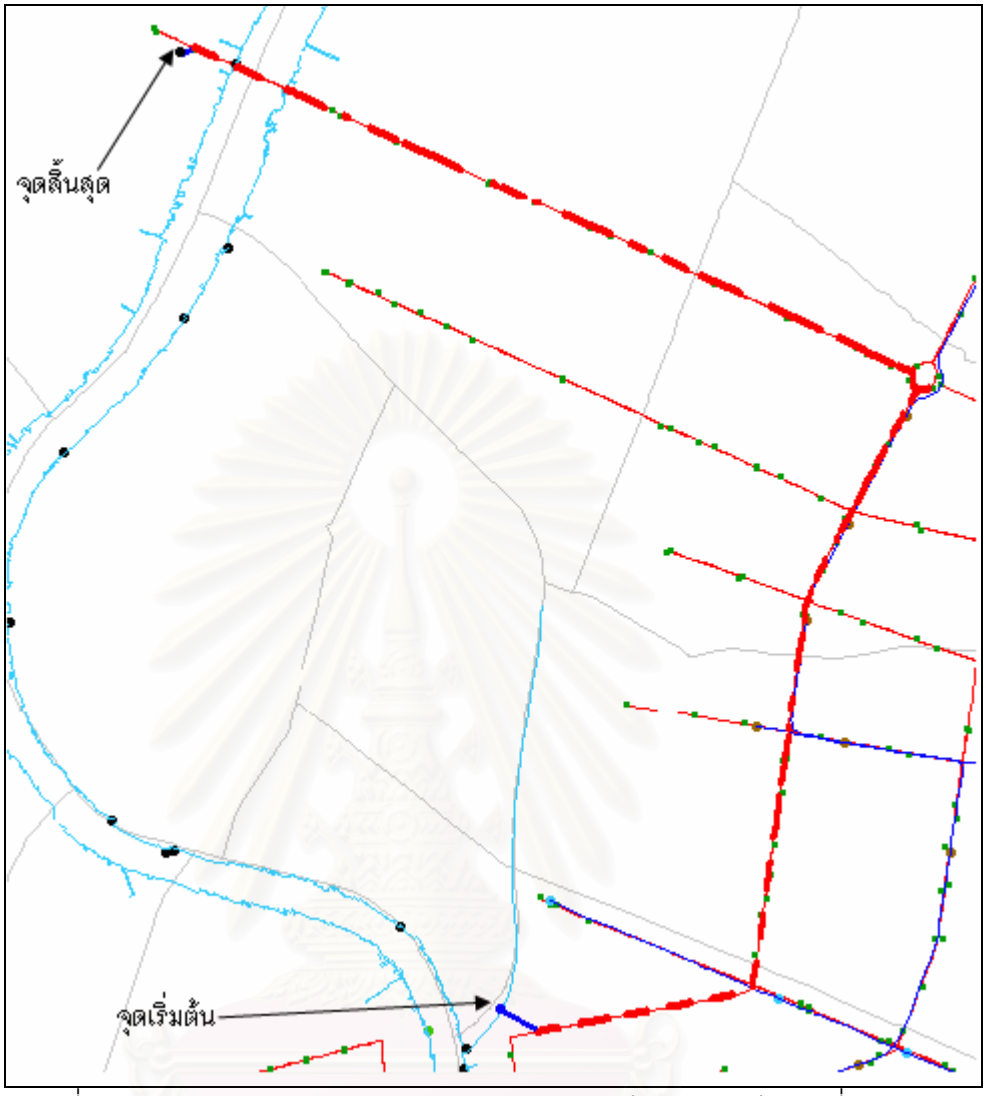
หากผู้ใช้กำหนดเวลาเริ่มต้นเดินทางก่อน 8:00 น. ตารางจะแสดงผลลัพธ์ว่า “ไม่พบเส้นทาง” ดังแสดงตัวอย่างในรูปที่ 4-7

ผลลัพธ์เส้นทาง					
ลำดับ	รูปแบบ	เวลา (นาที)	ระยะทาง (เมตร)	เวลา/วัน ที่ไปถึง	หมายเหตุ
1	x	x	x	ไม่พบเส้นทาง	x
2	-	-	-		

รูปที่ 4-7 ผลลัพธ์ของตารางกรณีไม่พบเส้นทางการเดินทางตามเงื่อนไขที่กำหนด

ในกรณีที่ผู้เดินทางเลือกจุดเริ่มต้นการเดินทางใกล้กับป้ายหยุดรถโดยสารประจำทางมากกว่าท่าเรือ โปรแกรมจะทำการสร้างเส้นทางเดินเท้าเชื่อมเข้ากับป้ายหยุดรถโดยสารประจำทางที่ใกล้ที่สุดแทนการเชื่อมเข้าสู่ท่าเรือ (เป็นข้อกำหนดในโปรแกรมให้เลือกสถานีหรือป้ายที่อยู่ใกล้จุดเริ่มต้นและสิ้นสุดการเดินทางมากที่สุด) เส้นทางใหม่ที่ได้จะเป็นการเดินทางโดยรถโดยสารประจำทาง ผลลัพธ์เส้นทางใหม่จะเป็นดังแสดงในรูปที่ 4-8

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

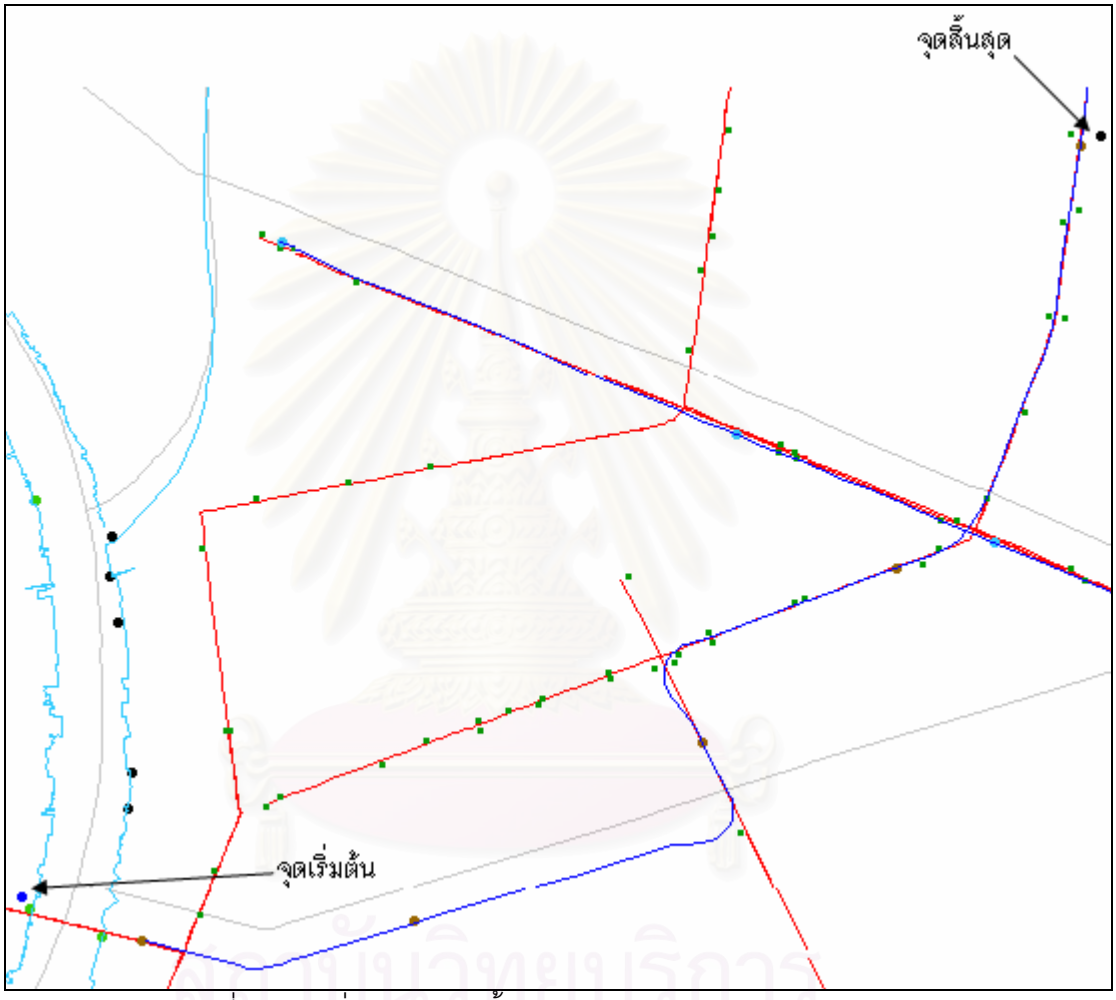


รูปที่ 4-8 เส้นทางเดินทางโดยรถโดยสารประจำทางเทียบกับกรณีในรูปที่ 4-4

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

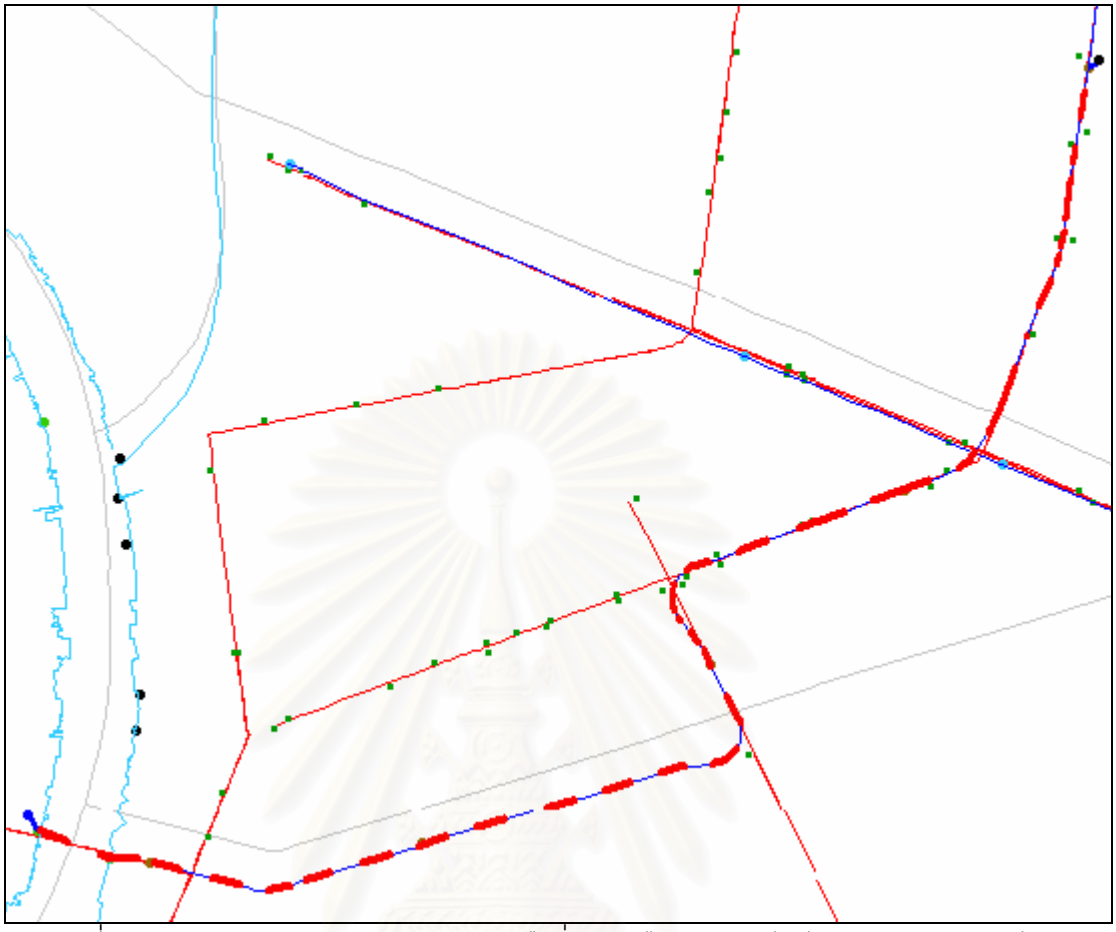
2. การเดินทางผสมรวมสองรูปแบบคือเรือข้ามฟาก และรถไฟฟ้าบีทีเอส

กรณีนี้เป็นการสมมติสถานการณ์ว่าผู้เดินทางต้องการเดินทางจากบริเวณท่าเรือฝั่งธนบุรีไปยังสถานีรถไฟฟ้าบริเวณถนนสีลม แสดงตำแหน่งจุดเริ่มต้นและจุดสิ้นสุดการเดินทางในรูปที่ 4-9 ส่วนเงื่อนไขเวลาการเริ่มต้นเดินทางกำหนดให้เป็น 8:00 น. แสดงอยู่ในรูปที่ 4-10



รูปที่ 4-9 จุดเริ่มต้นและจุดสิ้นสุดการเดินทาง คลิกกำหนดโดยผู้ใช้

รูปที่ 4-10 เงื่อนไขเวลาการเดินทางที่ระบุในแบบฟอร์ม



รูปที่ 4-11 ผลลัพธ์เส้นทางจากบริเวณท่าเรือฝั่งธนบุรีไปยังสถานีรถไฟฟ้าบริเวณถนนสีลม

ผลลัพธ์จากการคำนวณและสืบค้น แสดงอยู่ในรูปที่ 4-11 โดยแสดงเป็นเส้นประทึบ และรายละเอียดการเดินทางแสดงอยู่ในรูปที่ 4-12 เริ่มต้นจากการเดินเท้าไปที่ท่าเรือข้ามฟากใช้เวลา 0.6 นาที ระยะทาง 53.3 เมตร เพื่อรอและขึ้นเรือข้ามฟาก โดยระยะเวลาคอยคือ 6 นาที ระยะเวลาเดินทาง 0.9 นาที (คำนวณจากอัตราเร็วเรือที่ 15 กิโลเมตรต่อชั่วโมงและระยะทาง 223.9 เมตร) นอกจากนี้ยังมีระยะเวลาจอดรอรับส่งผู้โดยสารก่อนออกเรืออีก 5 นาที (ดึงจากค่าที่กำหนดไว้ในตาราง dwell_time ในฐานข้อมูล) รวมระยะเวลาที่ใช้ในส่วนของเรือข้ามฟากคือ 6+0.9+5=11.9 นาที หลังจากนั้นในแถวที่สามเป็นการเดินเท้าต่อเพื่อจะเข้าสู่สถานีรถไฟฟ้าบีทีเอส โดยใช้ระยะเวลา 1.4 นาที แล้วจึงรถรถไฟฟ้าบีทีเอสแล้วเริ่มเดินทางด้วยรถไฟฟ้าบีทีเอส สำหรับช่วงแรก 5.5 นาที เกิดจาก ระยะเวลาคอย 4 นาที ระยะเวลาจอดรอรับส่งผู้โดยสาร 10 วินาที และระยะเวลาเดิมทางส่วนแรก 1.4 นาที

ผลลัพธ์เส้นทาง					
ลำดับ	รูปแบบ	เวลา (นาที)	ระยะทาง (เมตร)	เวลา/วัน ที่ไปถึง	หมายเหตุ
1	walk	0.6	53.3	08:00:38:01:10:2007	-
2	ferry	11.9	223.9	08:12:32:01:10:2007	-
3	walk	1.4	117.9	08:13:56:01:10:2007	-
4	bts	5.5	802	08:19:29:01:10:2007	-
5	bts	2.3	1247	08:21:47:01:10:2007	-
6	bts	1.9	1005.3	08:23:41:01:10:2007	-
7	bts	2.6	1435.6	08:26:18:01:10:2007	-
8	walk	0.9	71.3	08:27:10:01:10:2007	-
-	-	-	-	-	-
-	-	-	-	-	-

รูปที่ 4-12 ผลลัพธ์ในรูปแบบตารางของเส้นทางจากบริเวณท่าเรือฝั่งธนบุรีไปยังสถานีรถไฟฟ้าบริเวณถนนสีลม

<u>สรุปผลลัพธ์จากการทดลองสืบค้นเส้นทาง</u>

สถานะการณ์สมมติทั้งสองกรณีสามารถแสดงให้เห็นการเดินทางสี่รูปแบบคือ รถโดยสารประจำทาง รถไฟฟ้าบีทีเอส เรือข้ามฟาก และเรือด่วนตามแม่น้ำเจ้าพระยา

ในแง่ของระยะเวลา ก็ครอบคลุมทั้งระยะเวลาเดินเท้า ระยะเวลาเดินทางโดยยานพาหนะ ระยะเวลารอคอยยานพาหนะ และระยะเวลาจอดรอรับส่งผู้โดยสาร

ในแง่ของเงื่อนไขด้านเวลาที่ระบุเมื่อสอบถามเส้นทาง คือ ได้แสดงให้เห็นว่ารองรับการตรวจสอบช่วงเวลาให้บริการของระบบขนส่งมวลชนก่อนทำการสืบค้นและแนะนำเส้นทางแก่ผู้ใช้ โดยหากตรวจพบว่าเวลาที่ผู้เดินทางเดินทางไปถึงยังจุดรับส่งผู้โดยสาร แล้วพบว่าอยู่นอกเหนือจากช่วงเวลาให้บริการของระบบขนส่งมวลชนชนิดนั้น ก็จะไม่นำระบบขนส่งมวลชนชนิดนั้นมาพิจารณา

บทที่ 5
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

สรุปผลการวิจัย

โครงสร้างข้อมูลและโปรแกรมที่พัฒนาขึ้นมีความสามารถดังนี้

1. สามารถสืบค้นและคัดเลือกหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุดได้ สำหรับขณะใดขณะหนึ่ง (ขึ้นกับเวลาที่เริ่มต้นเดินทาง)

2. สามารถคำนวณระยะเวลาที่ใช้ในการเดินทางได้ โดยสามารถคำนวนแยกย่อยในแต่ช่วงของการเดินทาง

3. รองรับการกำหนดจุดเริ่มต้นและจุดสิ้นสุดการเดินทางจากการให้ผู้ใช้คลิกกำหนดบนแผนที่ได้

4. สามารถสร้างเส้นทาง (arc) เชื่อมระหว่าง vertex แบบอัตโนมัติระหว่างจุดเริ่มต้นไปยังสถานีแรก และจากสถานีสุดท้ายไปยังจุดปลายทางได้ พร้อมทั้งคำนวณระยะเวลาในการเดินเท้า โดยตั้งสมมติฐานว่าเส้นทางการเดินเป็นเส้นตรง (จุดเริ่มต้นและจุดปลายทางคือจุดเริ่มต้นและสิ้นสุดการเดินทางที่เกิดจากการคลิกระบุตำแหน่งบนแผนที่โดยผู้ใช้)

5. สามารถรองรับการระบุวันเวลาที่เริ่มต้นเดินทางได้ เพื่อนำมาใช้ในการตรวจสอบกับช่วงเวลาการให้บริการของระบบขนส่งมวลชน หรือสามารถนำไปใช้เป็นปัจจัยสำหรับคำนวณค่าตัวแปรต่างๆ ที่เกี่ยวกับสภาพการจราจรได้ โดยการคำนวณนี้สามารถทำเป็นส่วนต่อขยายจากงานวิจัยนี้

6. เส้นทางที่เลือกขึ้นมาโดยโปรแกรมเป็นเส้นทางที่เกิดจากการเดินทางโดยหลายระบบขนส่งมวลชนผสมรวมกัน และยึดเงื่อนไขว่าใช้เวลาเดินทางน้อยที่สุด

7. เส้นทางที่โปรแกรมเลือกออกมาให้ยังมิได้คำนึงถึงค่าใช้จ่ายในการเดินทาง และให้คำตอบออกมาเพียงเส้นทางเดียว (ยังไม่มีเส้นทางตัวเลือก)

8. โครงสร้างข้อมูลได้รับการออกแบบให้ผู้ดูแลข้อมูลสามารถแก้ไขปรับเปลี่ยนค่าตัวแปรที่มีผลต่อระยะเวลาการเดินทางและการพิจารณาเลือกเส้นทางได้ ได้แก่
   8.1 อัตราเร็วของยานพาหนะที่วิ่งอยู่ในแต่ละ arc โดยสามารถปรับแก้แต่ละ arc แยกอิสระจากกัน รวมไปถึงอัตราเร็วในการเดินเท้าด้วย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

8.2 ระยะเวลารอคอยสำหรับยานพาหนะแต่ละประเภท

8.3 ช่วงเวลาให้บริการของระบบขนส่งมวลชนแต่ละประเภท และยังสามารถกำหนดช่วงเวลาให้บริการแยกเป็นอิสระจากกันในแต่ละ arc ด้วย สำหรับกรณีที่อาจมีการปิดการจราจรหรืองดให้บริการในบางเส้นทางบางช่วงเวลา

อภิปรายผลการวิจัย

แนวคิด โครงสร้างข้อมูลและโปรแกรมที่พัฒนาขึ้นนั้นยังมีจุดด้อยบางประการดังนี้

1. การปรับแก้และจัดเตรียมข้อมูลเวกเตอร์ของเส้นทางและความสัมพันธ์เชิงโครงข่ายยังเป็นกระบวนการที่แยกกันปฏิบัติอยู่ ดังนั้นจึงยังต้องอาศัยการตรวจสอบโดยผู้จัดทำข้อมูลค่อนข้างมาก การปรับแก้ที่เกิดขึ้นในข้อมูลเวกเตอร์เส้นทาง จะต้องคำนวณระยะทางใหม่แล้วนำไปปรับแก้ค่าระยะทางในข้อมูลความสัมพันธ์เชิงโครงข่ายด้วยจึงมีโอกาสที่อาจเกิดความไม่สอดคล้องกันได้

2. เส้นทางเดินเพื่อเปลี่ยนระหว่างระบบขนส่งมวลชนแต่ละรูปแบบที่ได้จัดทำไว้ในงานวิจัยนี้ส่วนใหญ่จะเป็นเพียงเส้นตรงหรือเส้นที่เลียบใกล้เคียงไปตามแนวถนนเท่านั้น ยังอาจแตกต่างจากเส้นทางเดินจริงไปมากหรือน้อยแล้วแต่กรณี เช่น กรณีการเปลี่ยนระหว่างท่าเรือและป้ายหยุดรถโดยสารประจำทางอาจใกล้เคียงเนื่องจากเป็นการเดินไปตามเส้นทางราบเป็นส่วนใหญ่และมักเดินไปตามแนวถนน ส่วนกรณีรถไฟฟ้าใต้ดิน ผู้เดินทางอาจต้องเดินเป็นระยะไกล 100-300 เมตร ก่อนที่จะออกจากสถานี ซึ่งระยะทางและเส้นทางดังกล่าวมิได้ทำการจัดเก็บหรือนำมาพิจารณา

3. เส้นเวกเตอร์ของแต่ละ arc อาจจะไม่แม่นยำ คือยังไม่บรรจบพอดีกับ vertex หากทำการซูมขยายขึ้นมาหลายๆ เท่า อาจเห็นความเหลื่อมกันได้ ทั้งนี้เนื่องจากยังใช้การแบ่งข้อมูลเป็นชั้นแยกจากกันอยู่ (สถานี ป้าย และเส้นทางเชื่อม) ทำให้ยังขาดการควบคุมความสัมพันธ์ระหว่างชั้นข้อมูลดังกล่าว

4. โปรแกรมที่พัฒนาขึ้นยังไม่ได้เสนอเส้นทางที่อาจเป็นตัวเลือกอื่นๆ ให้ซึ่งมีระยะเวลาเดินทางสั้นในลำดับรองๆ ลงไป ทำให้อาจขาดความยืดหยุ่นสำหรับผู้ใช้

5. ระยะเวลาในการประมวลผล (คำนวณและสืบค้นเส้นทางฯ)

เนื่องจาก Dijkstra's Algorithm เป็นอัลกอริทึมที่ต้องทำการสืบค้น node ทั้งหมดที่มีอยู่เพื่อพิจารณาว่า node นั้นมีการเชื่อมต่อไปยัง node ใดบ้าง หมายความว่ายิ่งข้อมูล (จำนวน node และ arc) เพิ่มมากขึ้นเท่าใด ปริมาณการคำนวณหรืออีกนัยหนึ่งคือระยะเวลาประมวลผลก็จะยิ่งเพิ่มมากขึ้นเป็นเงาตามตัว โดยเป็นกำลังสองของปริมาณข้อมูล แนวทางแก้ไขอาจทำได้หลายทาง เช่น

5.1 ทำการคำนวณเส้นทางที่สั้นที่สุดสำหรับทุกคู่การเดินทางเตรียมไว้ล่วงหน้าทั้งหมดแล้วจัดเก็บไว้ เพื่อรอการสืบค้น วิธีนี้จะเสียเวลาคำนวณเพียงครั้งเดียวในช่วงแรก แต่มีความเป็นไปได้ที่จะต้องเตรียมคำนวณไว้ สูงมาก โดยเฉพาะอย่างยิ่งเมื่อต้องคำนวณเตรียมไว้ทั้งหมดล่วงหน้าสำหรับค่าตัวแปรต่างๆ ทุกช่วงเวลา อีกด้วย และยังอาจไม่เหมาะกับการสืบค้นแบบตามเวลาจริง ซึ่งอาจมีเหตุการณ์ที่ไม่สามารถคาดไว้ล่วงหน้า เช่น การปิดถนน หรืออุบัติเหตุ เป็นต้น

5.2 การพยายามลดทอนจำนวน node ลงให้มากที่สุด เหลือเฉพาะเท่าที่จำเป็น เช่น กรณีรถโดยสารประจำทางซึ่งในวิทยานิพนธ์กำหนดให้หนึ่งป้าย เป็นหนึ่ง node อาจจะลดจำนวนป้ายลงให้เหลือเฉพาะป้ายที่จะใช้เป็นจุดเชื่อมต่อกับระบบขนส่งมวลชนประเภทอื่น หลายๆ ป้ายที่ต่อเนื่องกันและไม่มีการเปลี่ยนแปลงสายรถ ก็สามารถยุบรวมให้เหลือป้ายเดียว วิธีนี้จะทำให้สามารถลดจำนวน node ลงไปได้มาก แต่ก็มีข้อเสียคือ arc จะขยายความยาวออกไปมากจนกำหนดอัตราเร็วเฉลี่ยของยานพาหนะที่สิ่งบนบน arc นั้นได้ยาก หรือทำได้แต่มีความคลาดเคลื่อนมาก

6. การสร้างเส้นทางเชื่อมระหว่างป้ายหยุดรถโดยสารประจำทางสำหรับกรณีผู้เดินทางเปลี่ยนสายรถ

ในงานวิจัยนี้ใช้การสร้างเส้นทางเชื่อมดังกล่าวล่วงหน้าด้วยมือ โดยหากต้องการสร้างโดยอัตโนมัตินั้นก็สามารถทำได้ แต่ก็จะมีข้อดี-ข้อด้อยต่างๆ ดังนี้

6.1 การตรวจสอบความถูกต้องของการสร้างเส้นทางเชื่อม เช่น เส้นทางเชื่อมต่อต้องไม่มีการพาดผ่านแม่น้ำ หรือหากต้องมีการข้ามถนนก็จะต้องตรวจสอบด้วยว่าเป็นถนนที่อนุญาตให้ผู้เดินเท้าเดินข้ามได้ และไม่มีแผงกั้น เป็นต้น ซึ่งการกำหนดเงื่อนไขเหล่านี้สามารถพิจารณาได้ง่ายกว่ามากหากทำโดยมนุษย์

6.2 หากใช้วิธีการสืบค้นป้ายรถโดยสารประจำทางที่อยู่ใกล้กันแล้วสร้าง arc เชื่อมต่อกันแบบอัตโนมัติ จะเกิดข้อจำกัดเรื่องจำนวน arc จะเพิ่มขึ้นอย่างมากเนื่องจากใน

ความเป็นจริงผู้เดินทางสามารถลงรถแล้วเปลี่ยนต่อสายรถที่ป้ายใดก็ได้ ซึ่งอาจแก้โดยกำหนดให้เกิดขึ้นที่เฉพาะบางบริเวณเท่านั้น เช่น บริเวณสามแยกหรือสี่แยก ซึ่งมักเป็นจุดที่เกิดการตัดหรือเปลี่ยนแปลงเส้นทางเดินรถ นอกจากการสังเกตจากค่าพิกัดตำแหน่งของป้ายแล้ว ยังอาจใช้การเปรียบเทียบจากสายรถโดยสารประจำทางที่ผ่านป้ายนั้นๆ ช่วยด้วย เช่น ป้ายสองป้ายที่มีสายรถผ่านแตกต่างกัน (แม้เพียงสายเดียว) ก็สามารถตั้งสมมติฐานได้ว่าสองป้ายนั้นๆ เป็นป้ายที่อาจเกิดการเปลี่ยนสายรถได้

6.3 ในกรณีที่มีข้อมูลปริมาณมาก การสร้างเส้นทางเชื่อมต่อโดยอัตโนมัติย่อมประหยัดแรงงานและเวลามากกว่า

6.4 เทคนิคการสร้างเส้นทางเชื่อมอัตโนมัติสามารถช่วยลดระยะเวลาในการจัดเตรียมข้อมูลลง คือไม่ต้องทำการ digitize เส้น arc ใหม่ เมื่อมีการเปลี่ยนแปลงตำแหน่งป้ายรถโดยสารประจำทาง แต่อาจมีข้อด้อยคือเป็นการเพิ่มปริมาณการประมวลผลแก่ระบบมากขึ้น โดยเฉพาะเมื่อข้อมูลมีปริมาณมาก

แนวคิดการสร้างเส้นทางเชื่อมแบบอัตโนมัติทั้งหมดอาจมีข้อจำกัดอยู่บ้าง ในการปฏิบัติจริงอาจจะใช้วิธีผสมผสานกันทั้งทำโดยมนุษย์และโปรแกรม โดยควรพิจารณาความเหมาะสมในกรณีต่างๆ ว่ากรณีใดเหมาะจะสร้างโดยอัตโนมัติ และกรณีใดเหมาะจะสร้างโดยมนุษย์ นอกจากนี้การสร้างโปรแกรมที่ช่วยตรวจสอบความถูกต้องของข้อมูลที่สร้างขึ้นโดยมนุษย์ก็เป็นการผสมผสานการทำงานอีกแนวทางหนึ่งที่น่าสนใจ

7. ระยะเวลารอคอยของรถโดยสารประจำทางที่แตกต่างกันในแต่ละสาย

เนื่องจากรถโดยสารประจำทางแต่ละสายนั้นมีความแตกต่างกันในแง่ของจำนวนรถที่วิ่งให้บริการ และสภาพการจราจร จึงทำให้ระยะเวลารอคอยสำหรับรถโดยสารประจำทางมักแปรสามารถแยกตามสายรถฯ ด้วย

แนวความคิดการกำหนดหรือสร้างฟังก์ชั่นขึ้นมาสำหรับกำหนดค่าระยะเวลารอคอยให้ละเอียดและมีความจำเพาะสำหรับรถฯ แต่ละสายด้วยถือเป็นแนวคิดที่น่าสนใจ แนวทางอย่างหนึ่งที่เป็นไปได้คือการจัดเก็บสถิติระยะเวลารอคอยของรถฯ แต่ละสายแยกตามช่วงเวลาไว้ และเมื่อต้องการใช้งานก็ทำการสืบค้นในตารางที่จัดเก็บค่าดังกล่าว

8. การนำปัจจัยเรื่องเวลาและวันที่ที่เดินทางเข้ามาร่วมพิจารณาในการค้นหาเส้นทางฯ

การนำปัจจัยเรื่องเวลาและวันที่ที่เดินทางเข้ามาร่วมพิจารณาด้วยนี้ ทำให้สามารถนำปัจจัยเหล่านี้รวมกับปัจจัยเรื่องข้อมูลเชิงพื้นที่ (ตำแหน่งของเส้นทาง) ไปใช้ร่วมกันในการคำนวณค่าอื่นๆ ด้านการจราจรเพื่อนำมาป้อนให้กับโปรแกรมได้ จึงเป็นการเปิดช่องทางที่สามารถพัฒนาให้ผลการคำนวณระยะเวลาการเดินทางมีความใกล้เคียงสภาพความเป็นจริงเพิ่มขึ้นอีกมาก

ข้อเสนอแนะ

1. ระบบเก็บรวบรวมข้อมูลจราจรแบบตามเวลาจริงและแบบจำลองทางสถิติเพื่อการทำนายสภาพการจราจร

หากมีระบบตรวจจับสภาพการจราจรตามเวลาจริง หรือเก็บรวบรวมสถิติข้อมูลจราจรเพื่อนำมาใช้ทำนายสภาพการจราจรล่วงหน้า โดยอาศัยแบบจำลองทางสถิติต่างๆ ก็จะช่วยให้โปรแกรมแนะนำเส้นทางฯ มีความยืดหยุ่นมากขึ้น เช่น ช่วยคำนวณอัตราเร็วที่เหมาะสมสอดคล้องกับสภาพการจราจรมาป้อนให้กับโปรแกรม

ทั้งนี้โครงสร้างข้อมูลที่ออกแบบไว้สามารถรับข้อมูลดังกล่าวมาเพื่อใช้ประกอบการคำนวณได้แบบพลวัตอยู่แล้ว ดังนั้นงานวิจัยหรือการพัฒนาเกี่ยวกับการรวบรวมและทำนายสภาพการจราจรจึงสามารถส่งเสริมกับงานวิจัยนี้ได้

อย่างไรก็ตามจากการศึกษาพบว่า ในปัจจุบันมิได้ขาดแคลนแบบจำลองทางสถิติหรืองานวิจัยที่จะใช้ทำนายสภาพการจราจร หากแต่ในทางปฏิบัติเมื่อจะทำให้เกิดการเชื่อมโยงข้อมูลกันจริงระหว่างหน่วยงานที่ทำหน้าที่จัดเก็บข้อมูลจราจร เพื่อเป็นแหล่งข้อมูลป้อนสู่แบบจำลองทางสถิติ จะต้องมีการตกลงวิธีแปลง อ้างอิง หรือส่งและสื่อสารข้อมูลที่เข้าใจตรงกัน ว่าข้อมูลดิบที่ส่งมาให้นั้นข้อมูลของส่วนย่อยของถนนใด จะต้องมีวิธีการอ้างอิงที่สามารถเข้าใจได้ตรงกัน เพื่อให้ระบบปลายทางสามารถนำไปปรับแปลงให้เข้ากับโครงสร้างข้อมูลที่จัดทำไว้แล้ว หรืออาจจะต้องมีการปรับเปลี่ยนโครงสร้างข้อมูลใหม่เพื่อความเหมาะสม

2. การอนุญาตให้ผู้ใช้เลือกได้ว่าต้องการระบบขนส่งมวลชนประเภทใดโดยเฉพาะ กำหนดเงื่อนไขอื่นๆ เพิ่มเติม

การพัฒนาขั้นต่อเนื่องอาจเป็นการอนุญาตให้ผู้ใช้สามารถระบุได้ว่าต้องการใช้ระบบขนส่งมวลชนประเภทใดเป็นพิเศษหรือไม่ มีข้อกำหนดอื่นๆ เพิ่มเติมหรือไม่ เช่น ค่าใช้จ่ายในการเดินทาง หรือจำนวนต่อในการเปลี่ยนรูปแบบการเดินทาง ก็จะทำให้โปรแกรมมีความยืดหยุ่นและตอบสนองความต้องการของผู้ใช้ได้มากขึ้น

3. การเพิ่มประเภทของระบบขนส่งมวลชนที่จะมาวิเคราะห์

การพัฒนาขั้นต่อเนื่องอีกประการหนึ่งที่น่าสนใจได้แก่ การเพิ่มเติมความสามารถของแบบจำลองข้อมูล ให้สามารถรองรับระบบขนส่งมวลชนประภทอื่นๆ ได้เพิ่มเติมอีก เช่น รถไฟ เครื่องบิน รถตู้ ก็จะช่วยให้แบบจำลองข้อมูลมีความใกล้เคียงกับสภาพความเป็นจริงเพิ่มขึ้นมาก

นอกจากนี้ยังอาจมีการเพิ่มทางเดินต่างๆ ที่เป็นเส้นทางลัดเพื่อเลี่ยงถนนหรือเส้นทางที่มีสภาพการจราจรแออัด หรือแม้แต่การใช้รถจักรยานยนต์รับจ้าง เพื่อผ่านไปตามเส้นทางลัดดังกล่าว

4. แผนที่เส้นทางเดินแบบละเอียด

การปรับปรุงให้แผนที่เส้นทางเดินมีความละเอียดมากขึ้นและตรงตามระยะทางจริงมากขึ้น ตลอดจนสามารถแสดงแผนที่รายละเอียดเพื่อแนะผู้เดินทางก็จะช่วยให้ผู้เดินทางสามารถวางแผนการเดินทางได้ดีขึ้น

5. การเปลี่ยนสายรถโดยสารประจำทางอาจกำหนดให้เปลี่ยน ณ ป้ายที่มีคุณลักษณะเฉพาะบางอย่าง

การเปลี่ยนสายรถโดยสารประจำทางอาจกำหนดให้เปลี่ยน ณ ป้ายที่มีคุณลักษณะเฉพาะบางอย่าง เช่น มีศาลารอพักผู้โดยสาร หรือ ป้ายที่มีขนาดใหญ่ (มีจำนวนสายรถวิ่งผ่านให้เลือกมาก) เพื่อความอำนวยความสะดวกแก่ผู้เดินทางมากขึ้น

6. การอนุญาตให้ผู้ใช้สามารถเลือกจุดเริ่มต้นและจุดสิ้นสุดการเดินทางได้โดยอิสระ

การอนุญาตให้ผู้ใช้สามารถเลือกจุดเริ่มต้นและจุดสิ้นสุดการเดินทางได้โดยอิสระนั้น จะต้องออกแบบและพัฒนาโปรแกรมให้สามารถสร้าง arc ชั่วคราวขึ้นเพื่อใช้เชื่อมจุดที่

ผู้ใช้คลิกกำหนดบนแผนที่เข้ากับ vertex ที่มีอยู่ในโครงข่าย เพื่อให้สามารถเริ่มต้นคำนวณได้

การทำเช่นนี้ทำให้ยังมีข้อด้อยคือหากมี vertex อยู่ในบริเวณนั้นหลายจุด vertex ที่ใกล้ที่สุดอาจไม่ใช่ vertex ที่เหมาะสมก็ได้ เนื่องจากเป็น vertex ที่ไม่ได้นำสู่ระบบขนส่งมวลชนที่ให้เวลาเดินทางสั้นที่สุด

แนวทางแก้ไขอาจทำได้โดยเลือก vertex ที่อยู่ในระยะเดินเท้าที่กำหนดมาทั้งหมด แล้วพิจารณาเลือก vertex แต่ละ vertex เป็นจุดเริ่มต้นทีละจุด (ไม่เลือกเฉพาะจุดที่อยู่ใกล้ที่สุดเพียงจุดเดียวมาใช้) แล้วคำนวณเปรียบเทียบระยะเวลาเดินทางไปเรื่อยๆ จนกว่าจะได้เส้นทางที่ใช้เวลาน้อยที่สุด

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

ภาษาไทย

การรถไฟฟ้าขนส่งมวลชนแห่งประเทศไทย. สาระสำคัญโครงการรถไฟฟ้ามหานคร [Online]. -. แหล่งที่มา: http://www.mrta.co.th/news/doc-news/MainPoint.pdf [10 กุมภาพันธ์ 2550]

ชรัด พิริยะวัฒน์. ความพึงพอใจของผู้เดินทางด้วยรถโดยสารประจำทาง และการยอมรับของผู้เดินทางต่อระบบขนส่งสาธารณะแบบก้าวหน้า ในกรุงเทพมหานคร, วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมโยธา บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2543

บริษัทระบบขนส่งมวลชนกรุงเทพ จำกัด (มหาชน). เว็บบอร์ดถามตอบ [Online]. 2550. แหล่งที่มา: http://www.bts.co.th/th/wbr_view.asp?id=42224 [10 กุมภาพันธ์ 2550]

บริษัท เรือด่วนเจ้าพระยา จำกัด. ตารางเวลาการเดินเรือ [Online]. 2549. แหล่งที่มา: http://www.chaophrayaboat.co.th/xboatschedule_new_t.htm [16 กุมภาพันธ์ 2550]

วศิน สินธุภิญโญ. โปรแกรมสำเร็จรูปสำหรับแนะนำเส้นทางเดินรถโดยสารประจำทาง สาขาวิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2538.

ศิริพร ชวนิช. การประยุกต์ใช้ระบบสารสนเทศภูมิศาสตร์ในการแนะนำเส้นทางรถประจำทางที่สะดวก สาขาวิชาระบบสารสนเทศปริภูมิทางวิศวกรรม จุฬาลงกรณ์มหาวิทยาลัย. วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมสำรวจ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2545.

องค์การขนส่งมวลชนกรุงเทพ. เส้นทางการเดินรถโดยสารประจำทาง [Online]. 2549. แหล่งที่มา: http://www.bmta.co.th/thaiversion/thai_index.htm [10 กุมภาพันธ์ 2550]

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาษาอังกฤษ

Dennis. A. Networking in the Internet Edge. USA: John Wiley & Sons, 2002.

Environmental Systems Research Institute (ESRI) Preparing Street Data for Use with the Network Dataset. J9484_Street_Data_w_Network_Dataset.pdf [Online]. 2005. Available from: http://files.esri-sweden.com/ArcGIS_Desktop/whitepapers/ [2007, April]

Environmental Systems Research Institute (ESRI) ArcGIS Transportation Data Model (UNETRANS). TransDataModel08.doc [Online]. 2003. Available from: http://support.esri.com/index.cfm?fa=downloads.dataModels.filteredGateway&dmid=14 [2007, March]

Gonzalez, R. C. and Woods, R. E. Digital Image Processing. 2nd ed. USA: Prentice-Hall, 2001.

Illingworth, V. and Pyle, I. Dictionary of Computing, New York: Oxford University Press, 1996

ISO/DIS 19134, Geographic Information - Location based services- Multimodal routing and navigation. International Organization for Standardization (ISO), 2005

ISO/FDIS 14825, Intelligent transport systems –Geographic Data File (GDF) - Overall data specification, International Organization for Standardization (ISO), 2003.

ISO/TC211 N1762, Geographic Information - Location based services- Tracking and navigation, International Organization for Standardization (ISO), 2005.

Kumar P., Reddy, D. and Singh, V. Intelligent transport system using GIS. GISdevelopment.net. Map India Conference 2003.

MetaCarta, Inc. OpenLyaers-2.3.zip [Online]. 2006. Available from: http://www.openlayers.org/ [2006, November 22]

Miller, H. J. and Shaw, S. Geographic Information Systems for Transformation Principles and Applications. Oxford University Press, 2001.

Open Geospatial Consortium. OpenGIS Simple Features Specification For SQL Revision 1.1 [Online]. 1999. Available from: http://www.opengeospatial.org/standards [2004]

Silberschatz, Korth and Sudarshan, Database System Concepts. 4th ed. McGraw-Hill Higher Education Singapore, 2002.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. Introduction to Algorithms. McGraw-Hill, 1990.

Veeranna, B., Jayachandra, K. and Gopi, S. Map India Conference 2002 GISdevelopment.net, 2002.

Weiss, M. A. Data Structure and Algorithm Analysis. 2nd ed. 390 Bridge Parkway Redwood City California 94065: The Benjamin/Cummings Publishing, 1994.

Worboys, M.F. GIS : A Computing Perspective. Great Britain: T.J. International, 1997

World Wide Web Consortium (W3C). Vector Markup Language (VML) [Online]. 1998. Available from: http://www.w3.org/TR/1998/NOTE-VML-19980513 [2004, May 20]

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

รหัสโปรแกรมสำหรับตรวจสอบข้อมูลโครงข่ายในตารางความสัมพันธ์

รหัสโปรแกรมสำหรับตรวจสอบข้อมูลความสัมพันธ์เชิงโครงข่ายของรถไฟฟ้าบีทีเอส

```
Sub Macro1()
    Dim CounterRow As Integer
    Dim CounterColumn As Integer
    For CounterRow = 2 To 44
        d1 = Worksheets("Sheet1").Cells(CounterRow, 10).Value
        d2 = Worksheets("Sheet1").Cells(CounterRow, 11).Value
        count_the_repeat = 1
            For CounterRow2 = 2 To 44
                twoway_row = 45
                count_the_repeat = 1
                d3 = Worksheets("Sheet1").Cells(CounterRow2, 10).Value
                d4 = Worksheets("Sheet1").Cells(CounterRow2, 11).Value
                If d1 = d4 And d2 = d3 And CounterRow <> CounterRow2 Then
                    Worksheets("Sheet1").Cells(CounterRow, 12).Activate
                    ActiveCell.Value = "two way:" & CounterRow2
                    Worksheets("Sheet1").Cells(CounterRow2, 12).Activate
                    ActiveCell.Value = "two way:" & CounterRow
                End If
                If d1 = d3 And d2 = d4 And CounterRow <> CounterRow2 Then
                    count_the_repeat = count_the_repeat + 1
                    Worksheets("Sheet1").Cells(CounterRow2, 4).Activate
                    ActiveCell.Interior.ColorIndex = 9
                    Worksheets("Sheet1").Cells(CounterRow2, 13).Activate
                    ActiveCell.Value = "duplicate"
                End If
```

```
        Next CounterRow2
    Next CounterRow
End Sub
'=========================================================
```

รหัสโปรแกรมสำหรับตรวจสอบข้อมูลความสัมพันธ์เชิงโครงข่ายของรถไฟฟ้าใต้ดิน

```
Sub Macro1()
    Dim CounterRow As Integer
    Dim CounterColumn As Integer
    For CounterRow = 2 To 35
        a1 = Worksheets("MRT").Cells(CounterRow, 9).Value
        a2 = Worksheets("MRT").Cells(CounterRow, 10).Value
        d1 = Worksheets("MRT").Cells(CounterRow, 9).Value
        d2 = Worksheets("MRT").Cells(CounterRow, 10).Value
        count_the_repeat = 1
            For CounterRow2 = 2 To 35
                twoway_row = 36
                count_the_repeat = 1
                d3 = Worksheets("MRT").Cells(CounterRow2, 9).Value
                d4 = Worksheets("MRT").Cells(CounterRow2, 10).Value
                If d1 = d4 And d2 = d3 And CounterRow <> CounterRow2 Then
                    Worksheets("MRT").Cells(CounterRow, 1).Activate
                    ActiveCell.Interior.ColorIndex = 11
                    'Worksheets("MRT").Cells(CounterRow2, 1).Activate
                    'ActiveCell.Interior.ColorIndex = 9
                End If
                a3 = Worksheets("MRT").Cells(CounterRow2, 9).Value
                a4 = Worksheets("MRT").Cells(CounterRow2, 10).Value
                If a1 = a3 And a2 = a4 And CounterRow <> CounterRow2 Then
                    count_the_repeat = count_the_repeat + 1
```

```
            twoway_row = CounterRow2
        End If
        If count_the_repeat > 1 Then
            Worksheets("MRT").Cells(twoway_row, 4).Activate
            ActiveCell.Interior.ColorIndex = 9
        End If
    Next CounterRow2
Next CounterRow
'=============================================================
```

รหัสโปรแกรมสำหรับตรวจสอบข้อมูลความสัมพันธ์เชิงโครงข่ายของเส้นทางเดินเท้าเชื่อมระหว่างระบบขนส่งมวลชนแต่ละแบบ และเส้นทางเดินเรือ

```
Sub Macro1()
    even_status = 1
    Dim CounterRow As Integer
    Dim CounterColumn As Integer
    twoway_status = 0
    For CounterRow = 2 To 315
        d1 = Worksheets("crosstrans_ex_all_100207").Cells(CounterRow, 4).Value
        d2 = Worksheets("crosstrans_ex_all_100207").Cells(CounterRow, 9).Value
        For CounterRow_2 = 2 To 315
            d3 = Worksheets("crosstrans_ex_all_100207").Cells(CounterRow_2, 9).Value
            d4 = Worksheets("crosstrans_ex_all_100207").Cells(CounterRow_2, 4).Value
            c1 = Worksheets("crosstrans_ex_all_100207").Cells(CounterRow_2, 12).Value
            c2 = Worksheets("crosstrans_ex_all_100207").Cells(CounterRow, 12).Value
            e1 = Worksheets("crosstrans_ex_all_100207").Cells(CounterRow, 11).Value
            e2 = Worksheets("crosstrans_ex_all_100207").Cells(CounterRow_2, 11).Value

            If d1 = d3 And d2 = d4 And CounterRow <> CounterRow_2 Then
                twoway_status = 1
                Worksheets("crosstrans_ex_all_100207").Cells(CounterRow_2, 4).Activate
```

```
                ActiveCell.Interior.ColorIndex = 38
                Worksheets("crosstrans_ex_all_100207").Cells(CounterRow_2, 9).Activate
                ActiveCell.Interior.ColorIndex = 38
                Worksheets("crosstrans_ex_all_100207").Cells(CounterRow, 4).Activate
                ActiveCell.Interior.ColorIndex = 38
                Worksheets("crosstrans_ex_all_100207").Cells(CounterRow, 9).Activate
                ActiveCell.Interior.ColorIndex = 38
                If c1 <> c2 Then
                    Worksheets("crosstrans_ex_all_100207").Cells(CounterRow, 14).Activate
                     ActiveCell.Value = "not = " & CounterRow_2
                     Worksheets("crosstrans_ex_all_100207").Cells(CounterRow_2,
14).Activate
                     ActiveCell.Value = "not = " & CounterRow
                End If
                Exit For
            End If
        Next CounterRow_2
    Next CounterRow
End Sub
'=========================================================
```

รหัสโปรแกรมสำหรับตรวจสอบข้อมูลความสัมพันธ์เชิงโครงข่ายของรถโดยสารประจำทาง

```
Sub Macro1()
    Dim CounterRow As Integer
    Dim CounterColumn As Integer
    For CounterRow = 2 To 529
        a1 = Worksheets("road_segment_buslink_new2").Cells(CounterRow, 4).Value
        a2 = Worksheets("road_segment_buslink_new2").Cells(CounterRow, 8).Value
        c1 = Worksheets("road_segment_buslink_new2").Cells(CounterRow, 9).Value
        c2 = Worksheets("road_segment_buslink_new2").Cells(CounterRow + 1, 9).Value
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
    d1 = Worksheets("road_segment_buslink_new2").Cells(CounterRow, 4).Value
    d2 = Worksheets("road_segment_buslink_new2").Cells(CounterRow, 8).Value
    count_the_repeat = 1

      If c1 = c2 Then
          Worksheets("road_segment_buslink_new2").Cells(CounterRow, 9).Activate
          ActiveCell.Interior.ColorIndex = 7
      End If
      For CounterRow2 = 2 To 529
        twoway_row = 531
        count_the_repeat = 1
        d3 = Worksheets("road_segment_buslink_new2").Cells(CounterRow2, 4).Value
        d4 = Worksheets("road_segment_buslink_new2").Cells(CounterRow2, 8).Value
        If d1 = d4 And d2 = d3 Then
          Worksheets("road_segment_buslink_new2").Cells(CounterRow, 1).Activate
          ActiveCell.Interior.ColorIndex = 9
        End If
        a3 = Worksheets("road_segment_buslink_new2").Cells(CounterRow2, 4).Value
        a4 = Worksheets("road_segment_buslink_new2").Cells(CounterRow2, 8).Value
        If a1 = a3 And a2 = a4 And CounterRow <> CounterRow2 Then
          count_the_repeat = count_the_repeat + 1
          twoway_row = CounterRow2
        End If
        If count_the_repeat > 1 Then
            Worksheets("road_segment_buslink_new2").Cells(twoway_row,
4).Activate
            ActiveCell.Interior.ColorIndex = 8
        End If
      Next CounterRow2
  Next CounterRow
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
                If count_the_repeat = 1 Then
                    Worksheets("road_segment_buslink_new2").Cells(twoway_row,
4).Activate
                    ActiveCell.Value = "No two way bus"
                End If
End Sub
```


สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

รหัสโปรแกรมสำหรับคำนวณและค้นหาเส้นทางที่ใช้เวลาเดินทางน้อยที่สุด

```
<?php
function Dijkstra($start_node,$input_from_map,$origin_id,$destination_id, $origin_cost,
$des_cost,$time_constraint, $starting_time){
        if ($time_constraint==true && $starting_time==''){
                echo "You enable time constraint function, but no starting time
input.<br>";
                echo "Please verify your input.<br>";
                echo "Program terminated because of not complete input.";
                exit;
        }
        if ($input_from_map==true && ($destination_id =='' || $origin_id='')){

                echo "$start_node,$input_from_map,$origin_id,$destination_id,
$origin_cost, $des_cost<br>";
                echo "On the fly origin and destination is set to active, but no origin id
and des. id specified<br>";
                echo "Please check input paramaters for Dijkstra module."; exit;
        }
        //===============Database configuration================
        include './db_config.php';
        //=======================================================
        //==============Constant Variables====================
        //==============Variable for graph analysis===============
        $temp_ori_1='000001';                   $temp_des_1='000002';
        if ($starting_time!=''){
                $temp=explode(',',$starting_time);
                $default_date=",".$temp[3].",".$temp[4].",".$temp[5];
        }else{                  $default_date=',01,10,2007';            }
```

```
$Known_Node=array();
$Previous_Node=array();
$Distance_Link=array();
$Trans_Mode=array();
$Cumm_Cost=array();
$Current_Node=array();
$Graph=array();
$Node_List=array();
$Temp_Node_Cons=array();
$m=0;
$Infinity_val=9999999999;
$time_tole=20;
$walking_speed=1.389;
$min_date=',01,01,2000';
$max_date=',01,01,2020';
$Temp_Node_Cons[0]=$start_node;
$Number_of_Cons=sizeof($Temp_Node_Cons)-1;
$dbconn= pg_connect("host=$Host port=$Port dbname=$DBName user=$User
password=$Password");
$SQLCmd="SELECT $source_col, $target_col, $cost_col, $time_start,
$time_end, $distance, $transport_mode, $bus_num, $speed_col";
$SQLCmd=$SQLCmd." FROM $edge_table ORDER BY $source_col;";
$Result=pg_exec($dbconn, $SQLCmd);
$Number_of_edges=pg_num_rows($Result)-1;
$Graph=pg_fetch_all($Result);
$SQLCmd="SELECT $node_column FROM $node_table;";
$Result=pg_exec($dbconn, $SQLCmd);
$Number_of_nodes=pg_num_rows($Result)-1;
$Node_List = pg_fetch_all($Result);
pg_free_result($Result);
```

```
	for ($i=0;$i<=$Number_of_nodes;$i++){
		$Cumm_Cost[$i]=$Infinity_val;
		$Current_Node[$i]=$Node_List[$i][$node_column];
		$Known_Node[$i]=0;
		$Previous_Node[$i]=-1;
		$Distance_Link[$i]=0;
		$Trans_Mode[$i]='-';
	}
	switch ($input_from_map){
		case false:
			for ($i=0;$i<=$Number_of_nodes;$i++){
				if ($Current_Node[$i]==$start_node){
					$Cumm_Cost[$i]=convert_time($time_constraint,
$starting_time ,'date',',');
					$Previous_Node[$i]=0;		break;
				}
			}
			break;
		case true:
			$Cumm_Cost[$Number_of_nodes+1]=$Infinity_val;
			$Current_Node[$Number_of_nodes+1]="$temp_ori_1";
			$Known_Node[$Number_of_nodes+1]=0;
			$Previous_Node[$Number_of_nodes+1]=-1;//"$start_node";
			$Distance_Link[$Number_of_nodes+1]=$origin_cost;
			$Trans_Mode[$Number_of_nodes+1]='walk';
			$Cumm_Cost[$Number_of_nodes+2]=$Infinity_val;
			$Current_Node[$Number_of_nodes+2]="$temp_des_1";
			$Known_Node[$Number_of_nodes+2]=0;
			$Previous_Node[$Number_of_nodes+2]=-1;
			$Distance_Link[$Number_of_nodes+2]= $des_cost;
			$Trans_Mode[$Number_of_nodes+2]='walk';
```

```
if ($Current_Node[$Number_of_nodes+1]==$temp_ori_1){
        $Cumm_Cost[$Number_of_nodes+1]=convert_time($time_constraint,
$starting_time ,'date',',');
        $Previous_Node[$Number_of_nodes+1]=0;
}
$Graph[$Number_of_edges+1][$source_col]="$temp_ori_1";
$Graph[$Number_of_edges+1][$cost_col]=$origin_cost/$walking_speed;
$Graph[$Number_of_edges+1][$target_col]="$origin_id";
$Graph[$Number_of_edges+1][$time_start]="00,00";//.$default_date;
$Graph[$Number_of_edges+1][$time_end]="24,00";//.$default_date;
$Graph[$Number_of_edges+1][$distance]=$origin_cost;
$Graph[$Number_of_edges+1][$transport_mode]="walk";//.$default_date;
$Graph[$Number_of_edges+1][$waiting_time]=0;
$Graph[$Number_of_edges+1][$speed_col]=1.388;
$Graph[$Number_of_edges+2][$source_col]="$destination_id";
$Graph[$Number_of_edges+2][$cost_col]=$des_cost/$walking_speed;
$Graph[$Number_of_edges+2][$target_col]="$temp_des_1";
$Graph[$Number_of_edges+2][$time_start]="00,00";//.$default_date;
$Graph[$Number_of_edges+2][$time_end]="24,00";//.$default_date;
$Graph[$Number_of_edges+2][$distance]=$des_cost;
$Graph[$Number_of_edges+2][$transport_mode]='walk';
$Graph[$Number_of_edges+2][$waiting_time]=0;
$Graph[$Number_of_edges+2][$speed_col]=1.388;
$Number_of_nodes=$Number_of_nodes+2;
$Number_of_edges=$Number_of_edges+2;
break;
}
for ($j=0;$j<=$Number_of_Cons;$j++){
for ($i=0;$i<=$Number_of_edges;$i++){
        if ($Graph[$i][$source_col]==$Temp_Node_Cons[$j]) {
                for ($k=0;$k<=$Number_of_nodes;$k++){
```

```
                    if ($Current_Node[$k]==$Temp_Node_Cons[$j]){
                        $Cumm=$Cumm_Cost[$k];     break;          }
                    }
                    for ($k=0;$k<=$Number_of_nodes;$k++){
                        if (($Current_Node[$k]==$Graph[$i][$target_col]) &&
$Known_Node[$k]==0){
        $Graph[$i][$cost_col]=$Graph[$i][$distance_col]/$Graph[$i][$speed_col];
        if ($Cumm_Cost[$k]>$Cumm+$Graph[$i][$cost_col]){

        if (($Graph[$i][$time_start]=='00:00' && $Graph[$i][$time_end]=='24:00') ||
($Graph[$i][$time_start]=='00,00' && $Graph[$i][$time_end]=='24,00')){

        $a=$Graph[$i][$time_start].",00".$min_date;
        $b=$Graph[$i][$time_end].",00".$max_date;
}else{
        $temp=convert_time($time_constraint,$Cumm,'second',',');
        $temp=explode(',',$temp);
        $default_date=",".$temp[3].",".$temp[4].",".$temp[5];
        $a=$Graph[$i][$time_start].",00".$default_date;
        $b=$Graph[$i][$time_end].",00".$default_date;
}
        $a=str_replace(":", ",", $a);
        $b=str_replace(":", ",", $b);
        $c=convert_time($time_constraint,$Cumm+$Graph[$i][$cost_col],'second',',');
        $d=$time_tole;
if (check_time($a,$b,$c,$d)){
        $m=$m+1;
        $Temp_Node_Cons[$m]=$Graph[$i][$target_col];
        $Previous_Node[$k]=$Graph[$i][$source_col];
        $Cumm_Cost[$k]=$Cumm+$Graph[$i][$cost_col]+cal_waiting_time($Graph[$i][
$target_col],$Graph[$i][$source_col]);
```

```
            $Distance_Link[$k]=$Graph[$i][$distance];
            $Trans_Mode[$k]=$Graph[$i][$transport_mode];
            $BusNum[$k]=$Graph[$i][$bus_num];
            }}}}}}
            $Number_of_Cons=sizeof($Temp_Node_Cons)-1;
            }
return
array($Previous_Node,$Current_Node,$Cumm_Cost,$Distance_Link,$Trans_Mode,$Bus
Num);
}

function Retrieve_node_order($start,$end,$Matrix_result){
            $route='';                                  $cost=0;
            $width_of_matrix=sizeof($Matrix_result[0])-1;
             for ($i=0;$i<=$width_of_matrix;$i++){
                        if ($Matrix_result[1][$i]==$end){
cost='.$Matrix_result[2][$i].'<br>'.$route;
                                    if ($route==''){
            $route=$Matrix_result[0][$i].$Matrix_result[1][$i];
            $cost=$Matrix_result[2][$i];
            $dis_of_link=$Matrix_result[3][$i];
            $mode_of_link=$Matrix_result[4][$i];
            $bus_num_of_link=$Matrix_result[5][$i];
}else {
            $route=$Matrix_result[0][$i].$Matrix_result[1][$i].','.$route;
            $cost=$Matrix_result[2][$i].','.$cost;
            $dis_of_link=$Matrix_result[3][$i].','.$dis_of_link;
            $mode_of_link=$Matrix_result[4][$i].','.$mode_of_link;
            $bus_num_of_link=$Matrix_result[5][$i].'/'.$bus_num_of_link;
            }
```

```
		$end=$Matrix_result[0][$i];
		if ($Matrix_result[0][$i]==$start){		break;		}
			$i=-1;
		}
 }
 if ($cost==9999999999 || $route==-1000002){			$route='No route to the
specified destination.'; $cost='Infinity';}
		 return Array($route,$cost,$dis_of_link,$mode_of_link,$bus_num_of_link);
}

function Check_bus_num($bus_stop_1,$bus_stop_2){
		$bus_stop_1=explode(',',$bus_stop_1);
		$bus_stop_2=explode(',',$bus_stop_2);
		array_walk($bus_stop_1, 'trim');
		array_walk($bus_stop_2, 'trim');
		$result = array_intersect($bus_stop_1, $bus_stop_2);
		echo count($result)."<br><br>";
		print_r($result);
}

function check_time($t_start,$t_end,$time_inq,$time_tole){
if (!$time_tole){		$time_tole=0;		}
		$t_start=explode(',',$t_start);
		$t_end=explode(',',$t_end);
		$time_inq=explode(',',$time_inq);
		if ($t_start[5]>2500) {		$t_start[5]=$t_start[5]-543;	}
		if ($t_end[5]>2500) {		$t_end[5]=$t_end[5]-543;	}
		if ($time_inq[5]>2500) {	$time_inq[5]=$time_inq[5]-543;		}
		$tim_start=mktime($t_start[0], $t_start[1], $t_start[2], $t_start[3], $t_start[4],
$t_start[5]);
```

```
        $tim_end=mktime($t_end[0], $t_end[1], $t_end[2], $t_end[3], $t_end[4],
$t_end[5])-$time_tole;
        $tim_inq=mktime($time_inq[0], $time_inq[1], $time_inq[2], $time_inq[3],
$time_inq[4], $time_inq[5]);
"start=".$tim_start."<br>inq=".$tim_inq."<br>end=".$tim_end."<br>";
if ($tim_inq>=$tim_start && $tim_inq<=$tim_end){                    return true;
        }else {                 return false;                   }
}

function convert_time($time_constraint, $input,$input_type,$separator){
//===return second or time (hh:mm:ss)
        if (!$separator || $separator==''){              $separator=',';}
        if ($time_constraint){
                switch ($input_type){
                        case 'date':
                                $input=explode(',',$input);
                                if ($input[5]>2500) {           $input[5]=$input[5]-543;
        }
                                $result=mktime($input[0], $input[1], $input[2], $input[3],
$input[4], $input[5]);
                                break;
                        case 'second':
        $pattern="H".$separator."i".$separator."s".$separator."m".$separator."d".$separat
or."Y";
                                $result=date($pattern,$input);
                                break;
                }
        }else {                 $result=$input;                 }
        return $result;
}
```

```
function cal_waiting_time($id_target,$id_source){
        $id_source=substr($id_source, 0, 3);
        $id_target=substr($id_target, 0, 3);
        // =======any mode to walking
        if ($id_target==0)      {       return 0;       exit;                   }
        //========any mode (except ferry) to ferry
        if ($id_target==888 && $id_target!=$id_source){     return 360;     exit;   }
        //========any mode (except boat) to express boat
        if ($id_target==999 && $id_target!=$id_source){     return 600;     exit;   }
        //========any mode to bts
        if (($id_target==36 || $id_target==37)  && ($id_source!=36 && $id_source!=37))
        {return 240;            exit;           }
        //========any mode (except bus) to bus
        if (($id_target>38 && $id_target<888) && ($id_source<38 || $id_source>=888))
        {return 540;            exit;           }
        //========any mode (except mrt) to mrt
        if (($id_target<3 && $id_target >0) && ($id_source>3 || $id_source==0))
        {       return 240;             exit;           }
        if (($id_target>38 && $id_target<888) && ($id_source>38 && $id_source<888))
        {return 0;              exit;           }
        if ($id_target==888 && $id_source==888) {
        return 0;               exit;           }
        if ($id_target==999 && $id_source==999){
        return 0;               exit;           }
        if (($id_target==36 || $id_target==37)  && ($id_source==36 || $id_source==37))
        {return 0;              exit;           }
        if (($id_target<3 && $id_target >0) && ($id_source<3 && $id_source>0)) {
        return 0;       exit;           }
        return 5000;
}
?>
```

ภาคผนวก ค

ตารางการเดินรถและให้บริการรถไฟฟ้าบีทีเอส

อัตราเร็วในการขับเคลื่อนรถไฟฟ้าบีทีเอสจะไม่เกิน 80 กิโลเมตร/ชั่วโมง ซึ่งระดับความเร็วในการขับเคลื่อนที่ใช้เพื่อความปลอดภัยจะคงอยู่ที่ประมาณ 35 กิโลเมตร/ชั่วโมง

สำหรับในช่วงชั่วโมงเร่งด่วนจะเพิ่มความถี่และจะแตกต่างกันไปตามวันและสายรถไฟฟ้า ดังแสดงในตารางที่ ค-1

ตาราง ค-1 รายละเอียดอัตราการปล่อยรถไฟฟ้า

วัน	เวลา	เส้นทางรถไฟฟ้า (สาย)	ช่วงห่างการปล่อยรถ
จันทร์ - พฤหัสบดี	06.00 น. - 07.00 น.	สุขุมวิทและสีลม	ไม่เกิน 5 นาที
	07.00 น. - 08.30 น.	สุขุมวิท	ไม่เกิน 2.33 นาที
	07.00 น. - 08.30 น.	สีลม	ไม่เกิน 3.21 นาที
	08.30 น. - 09.30 น.	สุขุมวิท	ไม่เกิน 4.00 นาที
	08.30 น. - 09.30 น.	สีลม	ไม่เกิน 5 นาที
	09.30 น. -15.30 น.	สุขุมวิทและสีลม	ไม่เกิน 5.55 นาที
	15.30 น. - 16.30 น.	สุขุมวิท	ไม่เกิน 5 นาที
	15.30 น. - 16.30 น.	สีลม	ไม่เกิน 5.55 นาที
	16.30 น. - 17.30 น.	สุขุมวิทและสีลม	ไม่เกิน 3.40 นาที
	17.30 น. - 20.00 น.	สุขุมวิท	ไม่เกิน 3.40 นาที
	17.30 น. - 20.00 น.	สีลม	ไม่เกิน 4.20 นาที
	20.00 น. - 21.00 น.	สุขุมวิท	ไม่เกิน 5 นาที
	20.00 น. - 21.00 น.	สีลม	ไม่เกิน 5.30 นาที
	21.00 น. - 22.00 น.	สุขุมวิทและสีลม	ไม่เกิน 6 นาที
	22.00 น. - 24.00 น.	สุขุมวิทและสีลม	ไม่เกิน 8 นาที
ศุกร์	06.00 น. - 07.00 น.	สุขุมวิทและสีลม	ไม่เกิน 5 นาที
	07.00 น. - 08.30 น.	สุขุมวิท	ไม่เกิน 2.28 นาที
	07.00 น. - 08.30 น.	สีลม	ไม่เกิน 3.21 นาที
	08.30 น. - 09.30 น.	สุขุมวิท	ไม่เกิน 4 นาที
	08.30 น. - 09.30 น.	สีลม	ไม่เกิน 5 นาที
	09.30 น. - 15.30 น.	สุขุมวิทและสีลม	ไม่เกิน 5.55 นาที

ตาราง ค-1 (ต่อ)

วัน	เวลา	เส้นทางรถไฟฟ้า (สาย)	ช่วงห่างการปล่อยรถ
ศุกร์	15.30 น. - 16.30 น.	สุขุมวิท	ไม่เกิน 5 นาที
	15.30 น. - 16.30 น.	สีลม	ไม่เกิน 5.55 นาที
	16.30 น. - 17.30 น.	สุขุมวิทและสีลม	ไม่เกิน 3.40 นาที
	17.30 น. - 20.00 น.	สุขุมวิท	ไม่เกิน 3.40 นาที
	17.30 น. - 20.00 น.	สีลม	ไม่เกิน 4.20 นาที
	20.00 น. - 21.00 น.	สุขุมวิท	ไม่เกิน 5 นาที
	20.00 น. - 21.00 น.	สีลม	ไม่เกิน 5.30 นาที
	21.00 น. - 22.00 น.	สุขุมวิทและสีลม	ไม่เกิน 6 นาที
	22.00 น. - 24.00 น.	สุขุมวิทและสีลม	ไม่เกิน 8 นาที
เสาร์	06.00 น. - 11.00 น.	สุขุมวิทและสีลม	ไม่เกิน 5.55 นาที
	11.00 น. - 19.00 น.	สุขุมวิทและสีลม	ไม่เกิน 5 นาที
	19.00 น. - 21.00 น.	สุขุมวิทและสีลม	ไม่เกิน 5.55 นาที
	21.00 น. - 22.00 น.	สุขุมวิทและสีลม	ไม่เกิน 7 นาที
	22.00 น. - 24.00 น.	สุขุมวิทและสีลม	ไม่เกิน 8 นาที
อาทิตย์และ วันหยุดนักขัตฤกษ์	06.00 น. - 11.00 น.	สุขุมวิทและสีลม	ไม่เกิน 5.55 นาที
	11.00 น. - 19.00 น.	สุขุมวิท	ไม่เกิน 5 นาที
	11.00 น. - 19.00 น.	สีลม	ไม่เกิน 5.55 นาที
	19.00 น. - 21.00 น.	สุขุมวิทและสีลม	ไม่เกิน 5.55 นาที
	21.00 น. - 22.00 น.	สุขุมวิทและสีลม	ไม่เกิน 7 นาที
	22.00 น. - 24.00 น.	สุขุมวิทและสีลม	ไม่เกิน 8 นาที

แหล่งที่มา: http://www.bts.co.th/th/wbr_view.asp?id=42224

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ง

ตารางเวลาให้บริการเรือด่วนเจ้าพระยา (ปรับปรุงใหม่ 14 มิถุนายน 2549)

• ตารางเวลาเรือปรับปรุงใหม่ 14 มิถุนายน 2549:

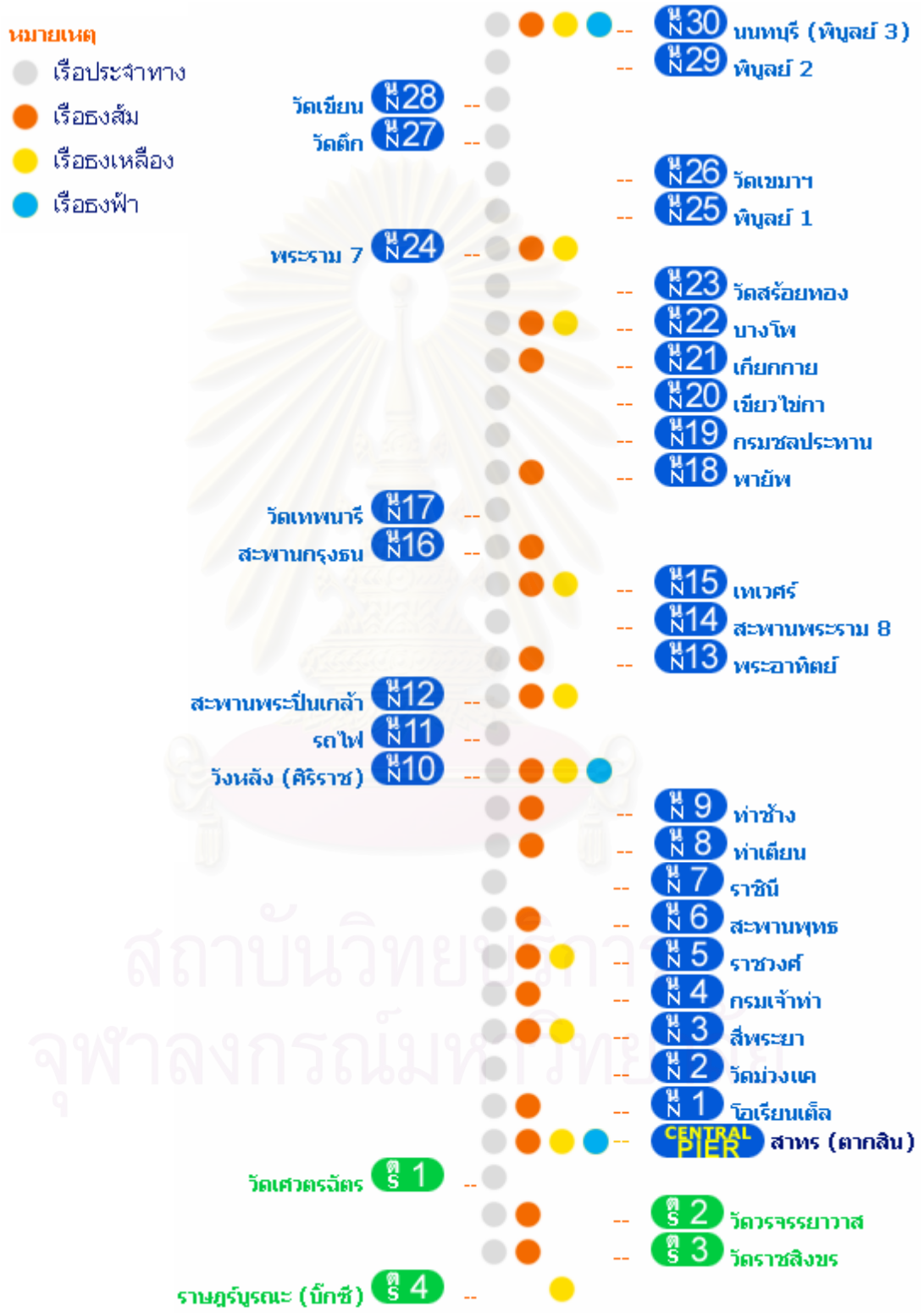
ไทย อังกฤษ หน้าแรก

เรือ	ออกจาก	เวลา	ออกทุกๆ / นาที
เรือด่วนประจำทาง นนทบุรี - วัดราชสิงขร	N30 S3 จ. - ศ.	06.00 - 08.40 15.00 - 18.00	20 20
เรือด่วนพิเศษ (ธงส้ม) นนทบุรี - วัดราชสิงขร	N30 จ. - ศ.	05.50 - 06.30 06.30 - 08.00 08.00 - 09.00 09.00 - 18.40	10 5 15 20
	N30 ส.	06.00 - 07.00 07.00 - 09.00 09.00 - 18.40	20 15 20
	N30 อา.	06.00 - 18.40	20
	S3 จ. - ศ.	06.00 - 08.00 08.00 - 09.00 09.00 - 16.00 16.00 - 19.00	10 15 20 10
	S3 ส.	06.00 - 16.00 16.00 - 17.00 17.00 - 18.00 18.00 - 18.40	20 15 12 20
	S3 อา.	06.00 - 18.40	20
เรือด่วนพิเศษ (ธงเหลือง) นนทบุรี - สาทร นนทบุรี - ราษฎร์บูรณะ	N30 จ. - ศ.	06.10 - 06.30, 08.00 - 08.40 06.30 - 08.00 16.30 - 18.20	10 5 20
	S4 จ. - ศ.	06.45 - 08.05	10
	CENTRAL PIER จ. - ศ.	15.45 - 16.00 16.00 - 19.00	15 10
	LAST BOAT	19.30	
เรือด่วนพิเศษ (ธงฟ้า) นนทบุรี - สาทร	N30 จ. - ศ.	07.00 - 07.45	15
	CENTRAL PIER จ. - ศ.	17.05 - 18.25	30

แหล่งที่มา: http://www.chaophrayaboat.co.th/xboatschedule_new_t.htm

ภาคผนวก จ

ผังแสดงเส้นทางและท่าเรือสำหรับบริการเรือด่วนเจ้าพระยา



แหล่งที่มา: http://www.chaophrayaboat.co.th/routemap_fare_t.htm#

ภาคผนวก ฉ

รหัสโปรแกรมสำหรับสร้างตารางและนำเข้าข้อมูลสู่ตาราง

transport_node และ transport_station

รหัสโปรแกรมสำหรับสร้างตารางและนำเข้าข้อมูลสู่ตาราง transport_node

```
<?php
$HostAddress="127.0.0.1";
$DBLoginName="postgres";
$DBPassword="sunday17";
$DBName="Dijkstra";
$DBPort="5432";
$transport_node="transport_node";
transport_node
$vertex='node_id';
$unique_id="unique_id";
$transport_edges="transport_edges";
$unique_source="source";
$unique_target="target";
$temp_source=$unique_source;
$temp_target =$unique_target;
$DBConnectID = pg_connect("host=$HostAddress port=$DBPort dbname=$DBName
user=$DBLoginName password=$DBPassword") or die ("<br><br><center>Sorry, we
cannot connect  database at this time.<br>Please inform system administrator to fix this
problem.<br>monsak_s@hotmail.com<center><br><br>Error message is : "
.pg_errormessage($DBConnect));
$SQL="CREATE TABLE $transport_node
(
  node_id  character varying(6)
)
WITHOUT OIDS;
```

```
ALTER TABLE $transport_node OWNER TO postgres;";
pg_exec($DBConnectID,$SQL);

$SQLCmd="SELECT DISTINCT $temp_source as $vertex FROM $transport_edges;";
$QueryResult3=pg_exec($DBConnectID,$SQLCmd);
$source_row=pg_num_rows($QueryResult3);
        for ($j=0;$j<$source_row;$j++){
                $temp=pg_fetch_row($QueryResult3,$j);
                $SQLCmd="INSERT INTO $transport_node ($vertex) VALUES
('$temp[0]');";
                echo "$SQLCmd<br>";
                $InsertResult=pg_exec($DBConnectID,$SQLCmd);
        }
                $SQLCmd="SELECT DISTINCT $temp_target FROM $transport_edges;";
                $QueryResult=pg_exec($DBConnectID,$SQLCmd);
                if ($QueryResult==false){                print $ErrorMessage;    exit;
        }
$NumberOfRows=pg_num_rows($QueryResult);
for ($j=0;$j<$NumberOfRows;$j++){
        $target_id=pg_fetch_row($QueryResult,$j);
        $SQLCmd="SELECT $vertex FROM $transport_node WHERE
$vertex='$target_id[0]';";
        $QueryResult2=pg_exec($DBConnectID,$SQLCmd);
        if (pg_num_rows($QueryResult2)==0){
        $SQLCmd="INSERT INTO $transport_node ($vertex) VALUES ('$target_id[0]');";
        echo "$SQLCmd<br>";
        $InsertResult=pg_exec($DBConnectID,$SQLCmd); }
}
echo "Generating unique ID in $transport_node is successful<br>";
?>
//============================================================
```

รหัสโปรแกรมสำหรับสร้างตารางและนำเข้าข้อมูลสู่ตาราง transport_station

```
<?php
$HostAddress="127.0.0.1";
$DBLoginName="postgres";
$DBPassword="sunday17";
$DBName="Dijkstra";
$DBPort="5432";
$source_tab_1="bus_stop";
$source_tab_2='bts_station';
$source_tab_3="mrt_station";
$source_tab_4="ferry_station";
$source_tab_5="boat_station";
$target_tab="transport_station";
$column_target_tab="station_id, name_e, name_t, trans_type, remark, the_geom";
$rec_count=0;
//=========================================================
 $DBConnectID = pg_connect("host=$HostAddress port=$DBPort dbname=$DBName
 user=$DBLoginName password=$DBPassword") or die ("<br><br><center>Sorry, we
cannot connect  database at this time.<br>Please inform system administrator to fix this
      problem.<br>monsak_s@hotmail.com<center><br><br>Error message is : "
                                .pg_errormessage($DBConnect));
$SQL="CREATE TABLE $target_tab    (
gid serial NOT NULL,
statioin_id character varying(6),
name_e character varying(60),
name_t character varying(60),
trans_type character varying(5),
remark text,
the_geom geometry,
CONSTRAINT ".$target_tab."_pkey PRIMARY KEY (gid)
```

```
            )
WITHOUT OIDS;
ALTER TABLE $target_tab OWNER TO postgres;";
pg_exec($DBConnectID,$SQL);
$SQLCmd="SELECT bsid, location, bus_num_ne, the_geom FROM $source_tab_1;";
$QueryResult3=pg_exec($DBConnectID,$SQLCmd);
$source_row=pg_num_rows($QueryResult3);
$rec_count=$rec_count+$source_row;
for ($j=0;$j<$source_row;$j++){
$temp=pg_fetch_row($QueryResult3,$j);
$SQLCmd="INSERT INTO  $target_tab ($column_target_tab) VALUES";
$SQLCmd=$SQLCmd."('$temp[0]','-', '$temp[1]' , 'Bus', '$temp[2]', '$temp[3]');";
$InsertResult=pg_exec($DBConnectID,$SQLCmd);
}
$SQLCmd="SELECT bts_st_id, name_e, name_t, the_geom FROM $source_tab_2;";
$QueryResult3=pg_exec($DBConnectID,$SQLCmd);
$source_row=pg_num_rows($QueryResult3);
$rec_count=$rec_count+$source_row;
for ($j=0;$j<$source_row;$j++){
$temp=pg_fetch_row($QueryResult3,$j);
$SQLCmd="INSERT INTO  $target_tab ($column_target_tab) VALUES";
$SQLCmd=$SQLCmd."('$temp[0]','$temp[1]','$temp[2]', 'BTS', '-', '$temp[3]');";
$InsertResult=pg_exec($DBConnectID,$SQLCmd);
                                                    }
$SQLCmd="SELECT mrt_st_id, name_e, name_t, the_geom FROM $source_tab_3;";
$QueryResult3=pg_exec($DBConnectID,$SQLCmd);
$source_row=pg_num_rows($QueryResult3);
$rec_count=$rec_count+$source_row;
for ($j=0;$j<$source_row;$j++){
$temp=pg_fetch_row($QueryResult3,$j);
$SQLCmd="INSERT INTO  $target_tab ($column_target_tab) VALUES";
```

```
$SQLCmd=$SQLCmd."('$temp[0]','$temp[1]','$temp[2]','MRT', '-', '$temp[3]');";
$InsertResult=pg_exec($DBConnectID,$SQLCmd); }
$SQLCmd="SELECT route_id, boat_id, name, the_geom FROM $source_tab_4;";
$QueryResult3=pg_exec($DBConnectID,$SQLCmd);
$source_row=pg_num_rows($QueryResult3);
$rec_count=$rec_count+$source_row;
for ($j=0;$j<$source_row;$j++){
$temp=pg_fetch_row($QueryResult3,$j);
$SQLCmd="INSERT INTO  $target_tab ($column_target_tab) VALUES";
$temp_station_id=$temp[0].$temp[1];
$SQLCmd=$SQLCmd."('$temp_station_id','-','$temp[2]','Ferry','-','$temp[3]');";
$InsertResult=pg_exec($DBConnectID,$SQLCmd); }
$SQLCmd="SELECT stationid, name, the_geom FROM $source_tab_5;";
$QueryResult3=pg_exec($DBConnectID,$SQLCmd);
$source_row=pg_num_rows($QueryResult3);
$rec_count=$rec_count+$source_row;
for ($j=0;$j<$source_row;$j++){
$temp=pg_fetch_row($QueryResult3,$j);
$SQLCmd="INSERT INTO  $target_tab ($column_target_tab) VALUES";
$SQLCmd=$SQLCmd."('$temp[0]','-','$temp[1]','Boat','-','$temp[2]');";
$InsertResult=pg_exec($DBConnectID,$SQLCmd); }
echo "Merging process successful, $rec_count stations are inserted into $target_tab.";
?>
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ช

รหัสโปรแกรมสำหรับรวมข้อมูล arc จากแต่ละชั้นข้อมูลลงในตาราง transport_geom

```
<?php
$HostAddress="127.0.0.1";
$DBLoginName="postgres";
$DBPassword="sunday17";
$DBName="Dijkstra";
$DBPort="5432";
$source_tab_1="bts_segment_edit";
$source_tab_2='cross_trans';
$source_tab_3="mrt_segment_edit";
$source_tab_4="bus_segment";
$target_tab="transport_geom";
$rec_count=0;
$DBConnectID = pg_connect("host=$HostAddress port=$DBPort dbname=$DBName
user=$DBLoginName password=$DBPassword") or die ("<br><br><center>Sorry, we
cannot connect  database at this time.<br>Please inform system administrator to fix this
problem.<br>monsak_s@hotmail.com<center><br><br>Error message is : "
.pg_errormessage($DBConnect));
$SQL="CREATE TABLE $target_tab
(
gid serial NOT NULL,
name_e character varying(30),
name_t character varying(30),
trans_type character varying(5),
segid character varying(12),
distance real,
the_geom geometry,
CONSTRAINT transport_geom_pkey PRIMARY KEY (segid)
)
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
WITHOUT OIDS;
ALTER TABLE transport_geom OWNER TO postgres;";
pg_exec($DBConnectID,$SQL);
$SQLCmd="SELECT name_e, name_t, seg_id, length, the_geom FROM
$source_tab_1;";
$QueryResult3=pg_exec($DBConnectID,$SQLCmd);
$source_row=pg_num_rows($QueryResult3);
$rec_count=$rec_count+$source_row;
for ($j=0;$j<$source_row;$j++){
        $temp=pg_fetch_row($QueryResult3,$j);
        $SQLCmd="INSERT INTO  $target_tab (name_e, name_t, trans_type, segid,
distance, the_geom) VALUES";
        $SQLCmd=$SQLCmd."('$temp[0]','$temp[1]','BTS','$temp[2]',$temp[3],'$temp[4
]');";
        echo "$temp[2]<br>";
        $InsertResult=pg_exec($DBConnectID,$SQLCmd);
        }
        $SQLCmd="SELECT remark, seg_id, length, the_geom FROM $source_tab_2;";
        $QueryResult3=pg_exec($DBConnectID,$SQLCmd);
        $source_row=pg_num_rows($QueryResult3);
        $rec_count=$rec_count+$source_row;
for ($j=0;$j<$source_row;$j++){
        $temp=pg_fetch_row($QueryResult3,$j);
        $SQLCmd="INSERT INTO  $target_tab (name_e, name_t, trans_type, segid,
distance, the_geom) VALUES";
        $SQLCmd=$SQLCmd."('-','$temp[0]','Walk','$temp[1]',$temp[2],'$temp[3]');";
        echo "$temp[1]<br>";
        $InsertResult=pg_exec($DBConnectID,$SQLCmd);
}
        $SQLCmd="SELECT name_e, name_t, seg_id, length, the_geom FROM
$source_tab_3;";
```

```
$QueryResult3=pg_exec($DBConnectID,$SQLCmd);
$source_row=pg_num_rows($QueryResult3);
$rec_count=$rec_count+$source_row;
for ($j=0;$j<$source_row;$j++){
        $temp=pg_fetch_row($QueryResult3,$j);
        $SQLCmd="INSERT INTO  $target_tab (name_e, name_t, trans_type, segid,
distance, the_geom) VALUES";
        $SQLCmd=$SQLCmd."('$temp[0]','$temp[1]','MRT','$temp[2]',$temp[3],'$temp[
4]');";
        echo "$temp[2]<br>";
        $InsertResult=pg_exec($DBConnectID,$SQLCmd);
}
$SQLCmd="SELECT rdlnnamt, segid, length, the_geom FROM $source_tab_4;";
$QueryResult3=pg_exec($DBConnectID,$SQLCmd);
$source_row=pg_num_rows($QueryResult3);
$rec_count=$rec_count+$source_row;
for ($j=0;$j<$source_row;$j++){
$temp=pg_fetch_row($QueryResult3,$j);
        $SQLCmd="INSERT INTO  $target_tab (name_e, name_t, trans_type, segid,
distance, the_geom) VALUES";
        $SQLCmd=$SQLCmd."('-','$temp[0]','Bus','$temp[1]',$temp[2],'$temp[3]');";
        echo "$temp[1]<br>";
        $InsertResult=pg_exec($DBConnectID,$SQLCmd);
}
echo "Merging process successful, $rec_count are inserted into $target_tab.";
?>
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ซ

การปรับปรุงแก้ไขข้อมูลเชิงเรขาคณิต ตำแหน่งที่ตั้งสถานี (vertex) หรือลักษณะรูปร่างของเส้น (arc)

1. การปรับแก้ตำแหน่งที่ตั้งสถานี (vertex)

การปรับแก้จะต้องทราบรหัสประจำสถานีหรือ vertex ที่ต้องการจะปรับแก้ตำแหน่ง แล้วใช้คำสั่ง SQL ในการปรับแก้ดังตัวอย่างด้านล่าง เป็นการเลื่อนตำแหน่งของ vertex รหัส “053003” ไปยังจุดใหม่ที่มีพิกัดเป็น E=663505.084705868 และ N=1517417.46572873 ในระบบพิกัดที่มี spatial reference หมายเลข 32647

```
UPDATE transport_station SET the_geom =
GeomFromText('POINT(663505.084705868 1517417.46572873)', 32647) WHERE
station_id ='053003'
```

2. การเพิ่มตำแหน่งที่ตั้งสถานี (vertex)

หากต้องการเพิ่ม vertex ใหม่เข้าในระบบจะต้องใช้คำสั่ง INSERT ดังตัวอย่างด้านล่าง

```
INSERT INTO transport_station (station_id, name_e, name_t, trans_type,
remark, the_geom) VALUES('053003', 'new station', 'สถานีใหม่', 'bus', '33,99',
GeomFromText('POINT(663505.084705868 1517417.46572873)',32647))
```

3. การลบตำแหน่งที่ตั้งสถานี (vertex)

หากต้องการลบ vertex ออกจากระบบจะต้องใช้คำสั่ง DELETE ดังตัวอย่างด้านล่างพร้อมทั้งระบุเงื่อนไขหรือคุณสมบัติของ vertex ที่ต้องการจะลบ เช่นกรณีเป็นการลบ vertex ที่มีค่า station_id='053003'

```
DELETE FROM transport_station WHERE station_id='053003'
```

4. การปรับแก้ลักษณะรูปร่างของเส้น arc ในตาราง transport_geom

ก่อนทำการเพิ่มเส้นจะต้องทราบค่าพิกัดของเส้น arc ใหม่ตลอดทั้งเส้นเสียก่อน แล้วจึงทำการจัดชุดค่าพิกัดเหล่านั้นให้อยู่ในรูป LINESTRING(x y, x y, x y, …) โดย x y คือ ค่าพิกัดของจุดที่ประกอบกันเป็นเส้น คั่นระหว่างจุดด้วยเครื่องหมาย “,” และเขียนเรียงกันไปตามลำดับ นอกจากนี้ยังต้องทราบรหัสของ arc ที่ต้องการจะปรับแก้ด้วย ในที่นี้คือ segid

เมื่อจัดชุดค่าพิกัดแล้วสามารถใช้คำสั่ง SQL นำเข้าเส้น arc ในตาราง transport_geom ได้ดังตัวอย่างแสดงด้านล่าง

```
UPDATE transport_geom SET the_geom=
GeomFromText('LINESTRING(671010.897388183 1517496.62926804,
671034.899966318 1517479.40204564, 671513.038409228 1517140.86769232,
671723.449416842 1516988.33680803)', 32647) WHERE segid=’147002147001’;
```

5. การเพิ่มเส้น arc ในตาราง transport_geom

ก่อนทำการเพิ่มเส้นจะต้องทราบค่าพิกัดของเส้น arc ตลอดทั้งเส้นเสียก่อน แล้วจึงทำการจัดชุดค่าพิกัดเหล่านั้นให้อยู่ในรูป LINESTRING(x y, x y, x y, …) โดย x y คือ ค่าพิกัดของจุดที่ประกอบกันเป็นเส้น คั่นระหว่างจุดด้วยเครื่องหมาย “,” และเขียนเรียงกันไปตามลำดับ เมื่อจัดชุดค่าพิกัดแล้วสามารถใช้คำสั่ง SQL นำเข้าเส้น arc พร้อมค่าคุณสมบัติอื่นๆ ได้ ดังตัวอย่างแสดงด้านล่าง

```
INSERT INTO transport_geom (name_e, name_t, trans_type, segid, distance,
the_geom) VALUES ('new vertex', 'จุด vertex ใหม่', 'Bus', '147002147001',
534.40002, GeomFromText('LINESTRING(671010.897388183 1517496.62926804,
671034.899966318 1517479.40204564, 671513.038409228 1517140.86769232,
671723.449416842 1516988.33680803)',32647);
```

ในการเพิ่มเส้น arc นี้ยังมีอีกขั้นตอนหนึ่งซึ่งจะต้องกระทำควบคู่กันไปด้วยคือเพิ่มรายการข้อมูลความสัมพันธ์เชิงโครงข่ายของ arc ที่เพิ่มเข้าในตาราง transport_geom รายละเอียดของขั้นตอนดังกล่าวระบุไว้ใน หัวข้อ “การปรับปรุงแก้ไขข้อมูลความสัมพันธ์เชิงโครงข่าย”

6. การลบเส้น arc ในตาราง transport_geom

หากต้องการลบ arc ออกจากระบบจะต้องใช้คำสั่ง DELETE ดังตัวอย่างด้านล่างพร้อมทั้งระบุเงื่อนไขหรือคุณสมบัติของ arc ที่ต้องการจะลบ เช่นกรณีเป็นการลบ arc ที่มีค่า segid='147002147001'

```
DELETE FROM transport_geom WHERE segid='147002147001'
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ฌ

การปรับปรุงแก้ไขข้อมูลความสัมพันธ์เชิงโครงข่าย

1. การแก้ไขข้อมูลความสัมพันธ์เชิงโครงข่าย

การแก้ไขข้อมูลในตารางความสัมพันธ์เชิงโครงข่ายในตาราง transport_edges แบ่งออกเป็นสองกรณีคือ กรณีที่จะต้องพิจารณาหรือตรวจสอบความสัมพันธ์ใหม่ที่สร้างขึ้น และกรณีที่ไม่ต้องตรวจสอบรายการของทั้งสองกรณีแสดงอยู่ในตาราง 3-8

รายละเอียดการตรวจสอบความสัมพันธ์เชิงโครงข่ายได้กล่าวไว้แล้วในหัวข้อ “การจัดเตรียมและปรับแก้ข้อมูล” หัวข้อ 1 ถึง 3

ตัวอย่างที่แสดงด้านล่างเป็นคำสั่งสำหรับการแก้ไขค่าอัตราเร็วของยานพาหนะที่วิ่งผ่าน arc ซึ่งมีรหัสของ source vertex =’147020’ และ target vertex = ‘147021’ โดยกำหนดอัตราเร็วใหม่เป็น 33 กิโลเมตรต่อชั่วโมง

```
UPDATE transport_edges SET speed=33 WHERE source=’147020’ AND target= ‘147021’
```

2. การลบข้อมูลความสัมพันธ์เชิงโครงข่าย

การลบข้อมูลความสัมพันธ์เชิงโครงข่ายซึ่งจัดเก็บอยู่ในตาราง transport_edges สามารถกระทำได้โดยไม่จำเป็นต้องตรวจสอบความถูกต้องก่อน แต่เนื่องจากตาราง transport_edges มีความสัมพันธ์แบบ 1 ต่อ 1 กับตาราง transport_geom ตามที่แสดงอยู่ใน E-R Diagram ในรูปที่ 3-14

ดังนั้นเมื่อลบรายการข้อมูลในตาราง transport_edges แล้วจึงควรลบรายการข้อมูลในตาราง transport_geom ที่สัมพันธ์กันออกไปด้วย

ตัวอย่างคำสั่ง SQL สำหรับการลบข้อมูลดังกล่าวในสองตาราง สำหรับกรณีต้องการลบรายการข้อมูลที่มีค่ารหัส source vertex =’147020’ และ target vertex = ‘147021’ ได้แสดงไว้ด้านล่าง

```
DELETE FROM transport_edges WHERE source=’147020’ AND target= ‘147021’;
DELETE FROM transport_geom WHERE segid=’147020147021’;
```

3. การเพิ่มข้อมูลความสัมพันธ์เชิงโครงข่าย

การเพิ่มข้อมูลความสัมพันธ์เชิงโครงข่ายซึ่งจัดเก็บอยู่ในตาราง transport_edges จะต้องตรวจสอบความถูกต้องของความสัมพันธ์เชิงโครงข่ายก่อนทำการเพิ่มข้อมูล และเนื่องจากตาราง transport_edges มีความสัมพันธ์แบบ 1 ต่อ 1 กับตาราง transport_geom ตามที่แสดงอยู่ใน E-R Diagram ในรูปที่ 3-14

ดังนั้นเมื่อเพิ่มรายการข้อมูลในตาราง transport_edges แล้วจึงต้องเพิ่มรายการข้อมูลในตาราง transport_geom ที่สัมพันธ์กันเข้าไปด้วย มิฉะนั้นจะไม่มีชุดพิกัดของเส้น arc สำหรับความสัมพันธ์คู่ที่เพิ่มเข้าในตาราง transport_edges และมีผลให้การแสดงผลเส้นทางผิดพลาด

ตัวอย่างคำสั่ง SQL สำหรับการเพิ่มข้อมูลดังกล่าวในตาราง transport_edges ได้แสดงไว้ด้านล่าง

```
INSERT INTO transport_edges (source, target, distance, cost, transport_mode,
speed, time_start, time_end, bus_num_start, bus_num_end, bus_num) VALUES
('147020','147021',455.5,55,'bus',20,'05:00','23:00','23,34,18','23,34,5',' 23,34');
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายมนต์ศักดิ์ โซ่เจริญธรรม เกิดเมื่อวันที่ 28 พฤษภาคม 2519 กรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาบัณฑิตสาขาวิชาวิศวกรรมสำรวจ จากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ.2542

สำเร็จการศึกษาระดับมหาบัณฑิตสาขาวิชาเทคโนโลยีสารสนเทศ จากคณะเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เมื่อ พ.ศ. 2547 และสาขาวิชาวิศวกรรมสำรวจ จากภาควิชาวิศวกรรมสำรวจ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2550 ตามลำดับ

มีประสบการณ์ในงานวิจัยทางด้านระบบสารสนเทศภูมิศาสตร์ เทคโนโลยีฐานข้อมูล โปรแกรมประยุกต์บนเว็บ และเทคโนโลยีอินเทอร์เน็ต ปัจจุบัน (พศ. 2550) ปฏิบัติงานเป็นนักวิจัยในสังกัดศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ กระทรวงวิทยาศาสตร์และเทคโนโลยี

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย