



รายงานวิจัยฉบับสมบูรณ์

โครงการแบบจำลองเชิงวัตถุ 3 มิติมาตราส่วนจริง
สำหรับงานออกแบบอาคารและบริหารโครงการก่อสร้าง

โดย ผศ.ดร.ศุภาวุฒิ มัลลียกฤษณะชลี

กุมภาพันธ์ 2554

รายงานวิจัยฉบับสมบูรณ์

โครงการแบบจำลองเชิงวัตถุ 3 มิติมาตราส่วนจริง
สำหรับงานออกแบบอาคารและบริหารโครงการก่อสร้าง

ผศ.ดร.ศุภาวดี มัลลย์กฤษณะชลี
ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์
มหาวิทยาลัยเกษตรศาสตร์

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย

(ความเห็นในรายงานนี้เป็นของผู้วิจัย สกว.ไม่จำเป็นต้องเห็นด้วยเสมอไป)

กิตติกรรมประกาศ

โครงการวิจัยนี้ได้รับทุนสนับสนุนจากสำนักงานคณะกรรมการการอุดมศึกษาและสำนักงานกองทุนสนับสนุนการวิจัย ผู้วิจัยขอขอบคุณหน่วยงานทั้งสองที่เล็งเห็นถึงความสำคัญของงานวิจัยนี้และให้การสนับสนุนเป็นอย่างดีมาโดยตลอด

ขอกราบขอบพระคุณ รศ.ดร.हरषा วัฒนานุกิจ ที่กรุณาให้แนวคิดในการทำวิจัยและคอยให้ความช่วยเหลือและให้คำแนะนำแก่ผู้วิจัยมาโดยตลอด ซึ่งช่วยให้ผู้วิจัยสามารถแก้ปัญหาและดำเนินงานแต่ละขั้นตอนลุล่วงไปได้ด้วยดี ผู้วิจัยขอขอบคุณในความกรุณาของท่านมา ณ ที่นี้เป็นอย่างสูง

ขอขอบคุณนายวรวิทย์ แสงแก้วที่ให้การสนับสนุนการดำเนินงานวิจัยนี้ในทุกเรื่องมาโดยตลอด โครงการวิจัยนี้คงไม่สามารถสำเร็จลุล่วงได้ด้วยดีถ้าปราศจากการสนับสนุนของนายวรวิทย์ แสงแก้ว

ขอขอบคุณนิสิตปริญญาโทสาขาบริหารงานก่อสร้าง ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ที่ได้ให้ความช่วยเหลือในงานวิจัยนี้

ท้ายที่สุด ขอขอบคุณครอบครัวของผู้วิจัยที่ให้กำลังใจ สนับสนุน และส่งเสริมการปฏิบัติงานเพื่อการวิจัยนี้

ศุภวุฒิ มัลลัษณะชลี

28 กุมภาพันธ์ 2554

บทคัดย่อ

รหัสโครงการ: MRG5080034

ชื่อโครงการ: แบบจำลองเชิงวัตถุ 3 มิติมาตราส่วนจริงสำหรับงานออกแบบอาคารและบริหาร
โครงการก่อสร้าง

ชื่อนักวิจัย: ผศ.ดร.ศุภวุฒิ มัลลย์กฤษณะชลี
ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

E-mail Address: fengshm@ku.ac.th

ระยะเวลาโครงการ: 1 ธันวาคม 2549 – 28 กุมภาพันธ์ 2554

การนำภาษาจาวามาใช้ร่วมกับโปรแกรมฐานข้อมูล PostgreSQL/PostGIS เป็นทางเลือกใหม่ในการพัฒนาระบบสารสนเทศสำหรับการจัดการข้อมูลแบบจำลองอาคารเชิงวัตถุ 3 มิติ ซึ่งจะช่วยให้การสร้างการวิเคราะห์ และการแสดงผลแบบจำลองรวมถึงการจัดการข้อมูลเชิงเรขาคณิตของแบบจำลองอาคารเชิงวัตถุ 3 มิติเป็นไปอย่างมีประสิทธิภาพ งานวิจัยนี้นำเสนอองค์ความรู้ในการพัฒนาระบบสารสนเทศสำหรับการจัดการข้อมูลแบบจำลองอาคารเชิงวัตถุ 3 มิติ โดยใช้ภาษาจาวาร่วมกับโปรแกรมฐานข้อมูล PostgreSQL/PostGIS โดยผู้วิจัยได้ทำการศึกษา วิเคราะห์ ออกแบบ และพัฒนาต้นแบบระบบ (Prototype System) โดยใช้เทคโนโลยีโอเพนซอร์ส (Open Source) ที่สามารถนำมาใช้งานได้ฟรีและมีลิขสิทธิ์ถูกต้องตามกฎหมาย ผู้วิจัยได้จัดทำแบบจำลองอาคารต้นแบบเพื่อแสดงให้เห็นถึงศักยภาพและแนวทางในการใช้ภาษาจาวาร่วมกับโปรแกรมฐานข้อมูล PostgreSQL/PostGIS ในการสร้างการวิเคราะห์ และแสดงผลแบบจำลองอาคารเชิงวัตถุ 3 มิติ

อย่างไรก็ดีต้นแบบระบบที่พัฒนาขึ้นยังมีข้อจำกัดในการจัดการข้อมูลเชิงเรขาคณิตของเหล็กเสริมคอนกรีตที่มีจำนวนมากซึ่งทำให้ข้อมูลมีขนาดใหญ่และยากต่อการประมวลผล ต้นแบบระบบที่พัฒนาขึ้นยังมีข้อจำกัดในการจัดการข้อมูลเชิงเรขาคณิตของระบบไฟฟ้าและเครื่องกลซึ่งมีองค์ประกอบที่มีความซับซ้อนสูง นอกจากนั้นต้นแบบระบบที่พัฒนาขึ้นยังขาดซึ่งฟังก์ชันสนับสนุนงานออกแบบและบริหารโครงการซึ่งควรได้รับการวิจัยและพัฒนาเพิ่มเติมเพื่อเพิ่มแรงจูงใจในการใช้งานระบบและเพื่อให้ระบบสามารถสนับสนุนการทำงานของผู้ประกอบการก่อสร้างได้อย่างครบถ้วน

คำหลัก: แบบจำลอง 3 มิติ, บริหารงานก่อสร้าง, ฐานข้อมูลงานก่อสร้าง, ฐานข้อมูลอาคาร, สารสนเทศ
ด้านงานก่อสร้าง

Abstract

Project Code: MRG5080034

Project Title: True-Scale Three-Dimensional Object-Oriented Model for Building Design and Construction Management

Investigator: Asst. Prof. Suphawut Malaikrisanachalee
Department of Civil Engineering, Faculty of Engineering, Kasetsart University

E-mail Address: fengshm@ku.ac.th

Project Period: December 1, 2006 – February 28, 2011

Integrating Java with PostgreSQL/PostGIS database offers a promising development platform for an object-oriented 3D building information system where an object-oriented 3D building model can be proficiently constructed, analyzed and displayed while geometry data of the 3D building model are efficiently managed. This research presents the prototype system integration design to methodically demonstrate how the system can be developed. This research also meticulously presents the logical and physical data models in designing the object-oriented 3D building database. In addition, a reinforced concrete building was used as an implementation case study to validate the proposed prototype system as well as to demonstrate its potentials. Not only the proposed prototype system offers technological advantages, it is also an open-source solution that is free to acquire, customize and distribute.

In order to fully exploit the proposed prototype system, multiple implementation issues are still required further research including how to overcome insurmountable data processing time when the model contains a large number of reinforcing steels and how to handle the complexity of mechanical and electrical systems' geometry data. In addition, more functions to support building design and construction project management should be developed to better fulfill functional needs of construction companies as well as to add motives for the companies to use the system.

Keywords: BIM, 3D model, construction database, project management, information technology in construction

สารบัญ

สารบัญ.....	i
สารบัญรูป.....	ii
สารบัญตาราง	vi
1. บทนำ.....	1
2. วัตถุประสงค์.....	3
3. วิธีดำเนินการวิจัย	3
4. ผลการวิจัย	6
4.1 การวิเคราะห์ความต้องการของระบบฐานข้อมูลแบบจำลองเชิงวัตถุ 3 มิติ.....	6
4.2 การออกแบบระบบสารสนเทศแบบจำลองเชิงวัตถุ 3 มิติ	12
4.3 การออกแบบฐานข้อมูลแบบจำลองเชิงวัตถุ 3 มิติ	18
4.4 การพัฒนาต้นแบบระบบสารสนเทศแบบจำลองเชิงวัตถุ 3 มิติ	24
5. สรุปและวิจารณ์ผลการวิจัย.....	50
6. เอกสารอ้างอิง.....	51
7. ผลผลิตจากโครงการวิจัย.....	53
ภาคผนวก ก การติดตั้งโปรแกรมสนับสนุนต้นแบบระบบ	54
ภาคผนวก ข คู่มือการใช้งานต้นแบบระบบ	73
ภาคผนวก ค ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติ.....	92

สารบัญรูป

รูปที่ 4.1 การตัวอย่างการประยุกต์ใช้งาน BIM ในงานวิศวกรรมโยธา (NIBS 2007).....	7
รูปที่ 4.2 ภาพรวมลำดับขั้นการเชื่อมโยงข้อมูล BIM (NIBS 2007)	12
รูปที่ 4.3 ภาพรวมโครงสร้างต้นแบบระบบ (Prototype System)	13
รูปที่ 4.4 แบบจำลองเชิงตรรก (Logical Data Model) ของอาคารคอนกรีตเสริมเหล็ก	19
รูปที่ 4.5 แบบจำลองเชิงสัมพันธ์ (Physical Data Model) ของอาคารคอนกรีตเสริมเหล็ก	21
รูปที่ 4.6 ขั้นตอนการสร้างแบบจำลองเชิงวัตถุ 3 มิติโดยภาษาจาวา	24
รูปที่ 4.7 Graphic User Interface (GUI) ของโปรแกรม	27
รูปที่ 4.8 ขั้นตอนการสร้างแบบจำลองอาคารเชิงวัตถุ 3 มิติ	28
รูปที่ 4.9 ขั้นตอนการแสดงรูปด้านของอาคาร.....	29
รูปที่ 4.10 ขั้นตอนการสร้างรูปตัดทางขวางของอาคาร	30
รูปที่ 4.11 แบบจำลอง 3 มิติของอาคารขนาด 3 ชั้น	31
รูปที่ 4.12 ตำแหน่งจุดอ้างอิงของเสาเข็ม	33
รูปที่ 4.13 ตำแหน่งจุดอ้างอิงของฐานราก	33
รูปที่ 4.14 ตำแหน่งจุดอ้างอิงของเสา	34
รูปที่ 4.15 ตำแหน่งจุดอ้างอิงของคาน.....	34
รูปที่ 4.16 ตำแหน่งจุดอ้างอิงของพื้น	34
รูปที่ 4.17 ตำแหน่งจุดอ้างอิงของผนัง.....	35
รูปที่ 4.18 ตำแหน่งจุดอ้างอิงของรั้ว.....	35
รูปที่ 4.19 ตำแหน่งจุดอ้างอิงของประตู.....	35
รูปที่ 4.20 ตำแหน่งจุดอ้างอิงของหน้าต่าง	36
รูปที่ 4.21 ตำแหน่งจุดอ้างอิงของบันได	36
รูปที่ 4.22 ตำแหน่งจุดอ้างอิงของเพดาน.....	36
รูปที่ 4.23 ลักษณะมุมมองแบบแรเงา (Shading).....	37
รูปที่ 4.24 ลักษณะมุมมองแบบเส้นโครง (Wireframe).....	37
รูปที่ 4.25 ลักษณะมุมมองแบบโปร่งแสง (Transparency).....	37

รูปที่ 4.26 การกำหนดวัสดุพื้นผิวและมาตราส่วนของวัสดุพื้นผิวของชิ้นส่วน	38
รูปที่ 4.27 หน้าต่างโปรแกรมประยุกต์สำหรับการกำหนดการแสดงผลรูปด้านของอาคาร	39
รูปที่ 4.28 รูปด้านหน้าของอาคาร	39
รูปที่ 4.29 รูปด้านหลังของอาคาร	40
รูปที่ 4.30 รูปด้านซ้ายของอาคาร	40
รูปที่ 4.31 รูปด้านขวาของอาคาร	41
รูปที่ 4.32 รูปด้านบนของอาคาร	41
รูปที่ 4.33 หน้าต่างโปรแกรมประยุกต์สำหรับกำหนดการสร้างรูปตัดทางขวางของอาคาร	42
รูปที่ 4.34 รูปตัดทางขวางตามแกน X ของอาคาร	42
รูปที่ 4.35 รูปตัดทางขวางตามแกน Y ของอาคาร	43
รูปที่ 4.36 รูปตัดทางขวางตามแกน Z ของอาคาร	43
รูปที่ 4.37 รูปแปลนฐานรากของอาคาร	44
รูปที่ 4.38 รูปแปลนพื้นชั้น 1 ของอาคาร	44
รูปที่ 4.39 รูปแปลนพื้นชั้น 2 ของอาคาร	45
รูปที่ 4.40 รูปแปลนพื้นชั้น 3 ของอาคาร	45
รูปที่ 4.41 รูปแปลนพื้นชั้นดาดฟ้าของอาคาร	46
รูปที่ 4.42 อาคาร 3 มิติที่สร้างขึ้นโดยใช้ AutoCAD 3D Solid Model	48
รูปที่ 4.43 การเปรียบเทียบแกนอ้างอิงระบบพิกัด 3 มิติ	48
รูปที่ 4.44 การแสดงผล AutoCAD 3D Solid Model ที่ขาดรายละเอียดเชิงวัตถุบน	49
รูปที่ 4.45 การแปลง AutoCAD 3D Solid Model เป็นแบบจำลองเชิงวัตถุ	49
รูปที่ ก.1 ไอคอนของโปรแกรม jdk-1_5_0_08-windows-i586-p	55
รูปที่ ก.2 ขั้นตอนที่ 2 ในการติดตั้ง jdk-1_5_0_08-windows-i586-p	55
รูปที่ ก.3 ขั้นตอนที่ 3 ในการติดตั้ง jdk-1_5_0_08-windows-i586-p	56
รูปที่ ก.4 ขั้นตอนที่ 4 ในการติดตั้ง jdk-1_5_0_08-windows-i586-p	56
รูปที่ ก.5 ขั้นตอนที่ 5 ในการติดตั้ง jdk-1_5_0_08-windows-i586-p	57
รูปที่ ก.6 ขั้นตอนที่ 6 ในการติดตั้ง jdk-1_5_0_08-windows-i586-p	57

รูปที่ ก.7 ขั้นสุดท้ายในการติดตั้ง jdk-1_5_0_08-windows-i586-p	58
รูปที่ ก.8 ไอคอนของโปรแกรม java3d-1_3_1-windows-i586-opengl-sdk.....	58
รูปที่ ก.9 ขั้นตอนที่ 2 ในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk.....	59
รูปที่ ก.10 ขั้นตอนที่ 3 ในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk.....	59
รูปที่ ก.11 ขั้นตอนที่ 4 ในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk.....	60
รูปที่ ก.12 ขั้นตอนที่ 5 ในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk.....	60
รูปที่ ก.13 ขั้นตอนที่ 6 ในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk.....	61
รูปที่ ก.14 ขั้นตอนที่สุดท้ายในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk.....	61
รูปที่ ก.15 ไอคอนของโปรแกรม postgresql-8.4.6-1-windows	62
รูปที่ ก.16 ไอคอนของโปรแกรม postgresql-8.4.6-1-windows	62
รูปที่ ก.17 ขั้นตอนที่ 1 ในการติดตั้งโปรแกรม postgresql-8.4.6-1	63
รูปที่ ก.18 ขั้นตอนที่ 2 ในการติดตั้งโปรแกรม postgresql-8.4.6-1	63
รูปที่ ก.19 ขั้นตอนที่ 3 ในการติดตั้งโปรแกรม postgresql-8.4.6-1	64
รูปที่ ก.20 ขั้นตอนที่ 4 ในการติดตั้งโปรแกรม postgresql-8.4.6-1	64
รูปที่ ก.21 ขั้นตอนที่ 5 ในการติดตั้งโปรแกรม postgresql-8.4.6-1	65
รูปที่ ก.22 ขั้นตอนที่ 6 ในการติดตั้งโปรแกรม postgresql-8.4.6-1	65
รูปที่ ก.23 ขั้นตอนที่ 7 ในการติดตั้งโปรแกรม postgresql-8.4.6-1	66
รูปที่ ก.24 ขั้นตอนที่ 8 ในการติดตั้งโปรแกรม postgresql-8.4.6-1	66
รูปที่ ก.25 ขั้นตอนที่ 1 ในการติดตั้งโปรแกรม PostGIS 1.5.....	67
รูปที่ ก.26 ขั้นตอนที่ 2 ในการติดตั้งโปรแกรม PostGIS 1.5.....	67
รูปที่ ก.27 ขั้นตอนที่ 3 ในการติดตั้งโปรแกรม PostGIS 1.5.....	68
รูปที่ ก.28 ขั้นตอนที่ 4 ในการติดตั้งโปรแกรม PostGIS 1.5.....	68
รูปที่ ก.29 ขั้นตอนที่ 5 ในการติดตั้งโปรแกรม PostGIS 1.5.....	69
รูปที่ ก.30 ขั้นตอนที่ 6 ในการติดตั้งโปรแกรม PostGIS 1.5.....	69
รูปที่ ก.31 ขั้นตอนที่ 7 ในการติดตั้งโปรแกรม PostGIS 1.5.....	70
รูปที่ ก.32 ขั้นตอนที่ 8 ในการติดตั้งโปรแกรม PostGIS 1.5.....	70

รูปที่ ก.33 ขั้นตอนที่ 9 ในการติดตั้งโปรแกรม PostGIS 1.5.....	71
รูปที่ ก.34 ขั้นตอนที่ 10 ในการติดตั้งโปรแกรม PostGIS 1.5.....	71
รูปที่ ก.35 ขั้นตอนที่ 11 ในการติดตั้งโปรแกรม PostGIS 1.5.....	72
รูปที่ ข.1 แสดงส่วนประกอบของหน้าจอ	74
รูปที่ ข.2 ระบบพิกัดในระนาบ 3 มิติของภาษาจาวา	75
รูปที่ ข.3 หน้าต่างย่อยของการสร้างชิ้นส่วน.....	76
รูปที่ ข.4 หน้าต่างย่อยสำหรับกำหนดขนาดและตำแหน่งของชิ้นส่วน.....	77
รูปที่ ข.5 การสร้างแบบจำลอง 3 มิติของอาคารจากข้อมูลเดิมในฐานข้อมูล	78
รูปที่ ข.6 หน้าต่างสำหรับกำหนดคุณสมบัติของชิ้นส่วน.....	79
รูปที่ ข.7 หน้าต่างสำหรับกำหนดสีของชิ้นส่วน.....	80
รูปที่ ข.8 การกำหนดมุมมองแบบเส้นโครง ของชิ้นส่วน.....	81
รูปที่ ข.9 การกำหนดมุมมองแบบเส้นโครงและกำหนด Cull เป็นแบบ Cull Front.....	81
รูปที่ ข.10 การกำหนดมุมมองแบบเส้นโครงและกำหนด Cull เป็นแบบ Cull Back.....	82
รูปที่ ข.11 การกำหนดไม่ให้แสดงผลชิ้นส่วน.....	82
รูปที่ ข.12 การกำหนดมุมมองโปรังใส.....	83
รูปที่ ข.13 การกำหนดวัสดุพื้นผิวของชิ้นส่วน	84
รูปที่ ข.14 หน้าต่างสำหรับกำหนด Mode การแสดงผลวัสดุพื้นผิวของชิ้นส่วน.....	85
รูปที่ ข.15 แสดงการกำหนดความละเอียดของวัสดุพื้นผิว.....	86
รูปที่ ข.16 หน้าต่างสำหรับการแสดงผลรูปด้านของอาคาร.....	87
รูปที่ ข.17 รูปด้านซ้ายของอาคาร	87
รูปที่ ข.18 หน้าต่างสำหรับกำหนดค่าเพื่อสร้างรูปตัดทางขวางของอาคาร	88
รูปที่ ข.19 รูปตัดทางขวางตามแกน X ของอาคาร.....	89
รูปที่ ข.20 การใช้เมาส์ Zoom เข้าไปยังตำแหน่งที่ต้องการพิจารณา	90
รูปที่ ข.21 การกำหนดสีพื้นหลังของระนาบ 3 มิติให้เป็นสีดำ.....	91

สารบัญตาราง

ตารางที่ 4.1 กลุ่มบุคคลที่เกี่ยวข้องในการนำข้อมูล BIM ไปใช้งาน (NIBS 2007)	8
ตารางที่ 4.2 ประเภทและมิติของข้อมูลเชิงเรขาคณิตที่รองรับโดยโปรแกรม PostGIS	14
ตารางที่ 4.3 มาตรฐานรูปแบบข้อมูลเชิงเรขาคณิตกำหนดโดย OGC.....	15
ตารางที่ 4.4 ตัวอย่างการจัดเก็บข้อมูลในรูปแบบ WKT และ WKB	15
ตารางที่ 4.5 การเชื่อมโยงโปรแกรม PostgreSQL กับโปรแกรมภายนอก	16
ตารางที่ 4.6 องค์ประกอบในการสร้างแบบจำลองเชิงวัตถุ 3 มิติโดยภาษาจาวา	17
ตารางที่ 4.7 สรุปรายละเอียด Class ของโปรแกรมภาษาจาวา	26
ตารางที่ 4.8 สรุปข้อมูลที่จำเป็นสำหรับการสร้างชิ้นส่วน 3 มิติ	32
ตารางที่ 4.9 ผลรวมปริมาตรและพื้นที่ผิวของชิ้นส่วนโครงสร้างอาคาร	47

1. บทนำ

อุตสาหกรรมก่อสร้างในปัจจุบันมีการแข่งขันสูง บริษัทผู้ประกอบการจำเป็นต้องสรรหาวิธีการและเทคโนโลยีสมัยใหม่เข้ามาช่วยเพิ่มประสิทธิภาพในการทำงานเพื่อให้สามารถแข่งขันได้ในตลาดปัจจุบัน การนำเทคโนโลยีสารสนเทศมาช่วยในการสร้างแบบจำลองเชิงวัตถุ 3 มิติแสดงรายละเอียดขององค์อาคารในการออกแบบและวางแผนก่อสร้างเป็นอีกทางเลือกหนึ่งที่จะช่วยเพิ่มประสิทธิภาพในการทำงานทั้งยังช่วยลดระยะเวลาและต้นทุนในการออกแบบและก่อสร้างซึ่งจะเป็นผลดีต่อทั้งบริษัทผู้ประกอบการและต่ออุตสาหกรรมก่อสร้างโดยรวมของประเทศไทย ทั้งยังจะช่วยลดความเสียหายทางด้านเทคโนโลยีของบริษัทผู้ประกอบการชาวไทยต่อบริษัทผู้ประกอบการขนาดใหญ่จากต่างประเทศ ที่ในปัจจุบันมีการประยุกต์ใช้เทคโนโลยีสารสนเทศในการออกแบบ วางแผนและบริหารโครงการก่อสร้างอย่างแพร่หลาย

การนำแบบจำลองเชิงวัตถุ 3 มิติมาช่วยในการออกแบบอาคารและสิ่งก่อสร้างที่มีความซับซ้อนของโครงสร้างสูงจะช่วยให้การออกแบบเป็นอย่างรวดเร็วและมีความถูกต้องมากขึ้น โดยสถาปนิกและวิศวกรผู้ออกแบบสามารถตรวจสอบลักษณะรูปร่างและฟังก์ชันการใช้งานของอาคารจากมุมมองต่างๆ ได้อย่างชัดเจนในระหว่างการออกแบบทั้งนี้ช่วยลดข้อผิดพลาดในการออกแบบและปัญหาการเปลี่ยนแปลงแบบในระหว่างการก่อสร้างซึ่งเป็นสาเหตุหนึ่งที่ทำให้โครงการก่อสร้างล่าช้าและเกินงบประมาณที่กำหนดไว้ แบบจำลองเชิงวัตถุ 3 มิติยังช่วยให้การสื่อสารภายในโครงการระหว่างเจ้าของงาน สถาปนิกและวิศวกรผู้ออกแบบรวมถึงบริษัทรับเหมาก่อสร้างเป็นไปอย่างมีประสิทธิภาพ ทำให้ผลงานที่ออกมาสอดคล้องกับความต้องการที่แท้จริงของเจ้าของงานและยังช่วยลดข้อขัดแย้งภายในโครงการซึ่งเป็นอีกสาเหตุหนึ่งที่ทำให้การดำเนินการของโครงการเป็นไปอย่างล่าช้า

แบบจำลองเชิงวัตถุ 3 มิติยังช่วยลดเวลาการเขียนแบบ ถอดแบบและประมาณราคาโดยแบบรูป 2 มิติแสดงรูปด้านข้าง (Front and Side Views) รูปด้านบน (Plan View) และภาพตัด (Section) ของอาคาร รวมทั้งแบบแสดงรายละเอียดที่ใช้ประกอบการก่อสร้าง (Shop Drawing) รายการวัสดุก่อสร้าง และปริมาณที่ใช้ สามารถจัดทำขึ้นได้โดยอัตโนมัติจากแบบจำลองเชิงวัตถุ 3 มิติ ทำให้ช่วยลดต้นทุนและเวลาในการทำงาน ทั้งยังช่วยป้องกันข้อผิดพลาดที่อาจเกิดขึ้นในแต่ละขั้นตอนโดยเฉพาะอย่างยิ่งในการเขียนภาพตัดและการประมาณราคาซึ่งเป็นงานที่มีความซับซ้อนและต้องการรายละเอียดและความถูกต้องสูง

นอกเหนือจากประโยชน์ที่กล่าวไว้ข้างต้นแล้ว แบบจำลองเชิงวัตถุ 3 มิติยังช่วยเพิ่มความยืดหยุ่นในการนำเทคโนโลยีสารสนเทศอื่นๆ มาประยุกต์ใช้ในอุตสาหกรรมก่อสร้างเนื่องจากแบบจำลองเชิงวัตถุ 3 มิติจะถูกจัดเก็บในรูปแบบของแฟ้มข้อมูลหรือฐานข้อมูลดิจิทัลอย่างมีระบบ การนำข้อมูลจากแบบจำลองเชิงวัตถุ 3 มิติมาประยุกต์ใช้หรือเชื่อมโยงกับเทคโนโลยีสารสนเทศอื่นๆ อาทิเช่น การใช้โปรแกรมคอมพิวเตอร์วางแผนการตัดเหล็กเสริมคอนกรีตให้เหลือเศษเหล็กน้อยที่สุด การใช้โปรแกรมคอมพิวเตอร์ในการวิเคราะห์โครงสร้างของอาคาร การใช้โปรแกรมคอมพิวเตอร์ในการวางแผนและติดตามความก้าวหน้าของโครงการก่อสร้าง หรือแม้แต่การนำแบบจำลองเชิงวัตถุ 3 มิติมาจัดทำสื่อการสอนแสดงขั้นตอนการก่อสร้าง จึงเป็นไปได้โดยสะดวก

ปัจจุบันมีโปรแกรมสำเร็จรูปหลายโปรแกรมที่สามารถใช้สร้างแบบจำลองอาคารเชิงวัตถุ 3 มิติ อาทิเช่น โปรแกรมออโตแคด (AutoCAD) โปรแกรมออโตเดสก์รีวิท (Autodesk Revit) โปรแกรมอาร์คิแคด (ArchiCAD) โปรแกรมโซลิดเวิร์ค (SolidWorks) และโปรแกรมในกลุ่มเวอร์ชวลเรียลลิตี (Virtual Reality) โปรแกรมเหล่านี้หลายโปรแกรมไม่ได้ถูกพัฒนาขึ้นเพื่อใช้งานในสาขาวิศวกรรมโยธาทำให้เกิดข้อจำกัดด้านเทคนิคในการนำโปรแกรมสำเร็จรูปเหล่านี้มาประยุกต์ใช้ในการออกแบบและวางแผนก่อสร้าง อาทิเช่น โปรแกรมในกลุ่มเวอร์ชวลเรียลลิตีถูกพัฒนาขึ้นเพื่อเน้นการสร้างผลงานในรูปแบบแอนิเมชัน (Animation) สำหรับงานภาพยนตร์และสื่อบันเทิงต่างๆ จะขาดซึ่งฟังก์ชันในการวัดและกำหนดขนาด รวมถึงฟังก์ชันในการแสดงผลในรูปแบบแฟ้มข้อมูลหรือตารางซึ่งจะเป็นอุปสรรคในการนำแบบจำลอง 3 มิติที่สร้างขึ้นมาใช้คำนวณในการถอดแบบและประเมินราคารวมทั้งการเชื่อมโยงแบบจำลอง 3 มิติกับเทคโนโลยีสารสนเทศอื่นๆ จึงเป็นไปได้ยาก

ข้อจำกัดด้านเทคนิคอีกประการหนึ่งคือ แบบจำลอง 3 มิติที่ถูกสร้างขึ้นโดยโปรแกรมสำเร็จรูปหลายโปรแกรมไม่ใช่แบบจำลองเชิงวัตถุ อาทิเช่น แบบจำลอง 3 มิติที่ถูกสร้างขึ้นโดยโปรแกรมออโตแคดเป็นเพียงแค่การนำเส้นหรือรูปทรงทางเรขาคณิตต่างๆ มาประกอบกันให้ผลงานแสดงออกมาในลักษณะ 3 มิติ โดยไม่มีการกำหนดคุณสมบัติของแต่ละชิ้นส่วนรวมถึงความสัมพันธ์ระหว่างชิ้นส่วน ทั้งนี้จะส่งผลให้เกิดอุปสรรคในกระบวนการสร้างภาพตัด การถอดแบบและการประเมินราคา

เนื่องจากโปรแกรมสำเร็จรูปเหล่านี้ถูกพัฒนาขึ้นโดยบริษัทผู้ผลิตชาวต่างชาติ โปรแกรมเหล่านี้จึงถูกพัฒนามาบนพื้นฐานความต้องการของผู้ใช้และมาตรฐานการออกแบบและก่อสร้างของต่างประเทศซึ่งมีความแตกต่างกับความต้องการของผู้ใช้และมาตรฐานในประเทศไทย อุปสรรคสำคัญอีกประการหนึ่งในการนำโปรแกรมสำเร็จรูปเหล่านี้มาใช้งานคือ ค่าลิขสิทธิ์ของโปรแกรมที่มีราคาสูง ตัวอย่างเช่น โปรแกรมออโตเดสก์รีวิทซึ่งเป็นโปรแกรมที่ถูกนำมาใช้สร้างแบบจำลองอาคารเชิงวัตถุ 3 มิติเพื่อช่วยในการออกแบบและวางแผนก่อสร้างอย่างแพร่หลายในต่างประเทศมีราคาสูงกว่า 250,000 บาท ต่อ 1 ลิขสิทธิ์การใช้งาน (<http://www.pesik.com>) ทั้งนี้ยังไม่รวมค่าใช้จ่ายอื่นๆ ที่บริษัทผู้ประกอบการอาจต้องจ่ายเพิ่มเติม อาทิเช่น ค่าใช้จ่ายในการอบรมการใช้โปรแกรม ค่าใช้จ่ายในการซ่อมแซมและบำรุงรักษาโปรแกรม ทั้งนี้เป็นผลให้ผู้ประกอบการชาวไทยโดยเฉพาะผู้ประกอบการขนาดกลางและย่อยซึ่งมีต้นทุนการดำเนินการจำกัดไม่สามารถจัดซื้อโปรแกรมเหล่านี้มาใช้งานได้

จากเหตุผลและความจำเป็นดังกล่าวข้างต้นจึงมีความจำเป็นที่จะต้องมีการพัฒนาทางเลือกใหม่ในการสร้างแบบจำลองเชิงวัตถุ 3 มิติสำหรับงานออกแบบอาคารและบริหารโครงการก่อสร้างสำหรับอุตสาหกรรมก่อสร้างในประเทศไทย โดยทางเลือกใหม่ที่พัฒนาขึ้นควรคำนึงถึงมาตรฐานการก่อสร้างและความต้องการของผู้ใช้ในประเทศไทยเป็นหลักและควรเป็นอิสระจากโปรแกรมสำเร็จรูปที่นำเข้าจากต่างประเทศเพื่อหลีกเลี่ยงปัญหาลิขสิทธิ์การใช้งานและลดการขาดดุลการค้าให้กับประเทศ

ภาษาจาวา (Java Program) เป็นภาษาสาขารณมิติสำหรับเขียนโปรแกรมคอมพิวเตอร์เชิงวัตถุ (Object-Oriented Program) ที่ไม่ขึ้นกับแพลตฟอร์มและระบบปฏิบัติการใดๆ จากการสำรวจเบื้องต้นพบว่าภาษาจาวาเป็นภาษาที่มีความเหมาะสมในการนำมาใช้เป็นแพลตฟอร์ม (Platform) ในการพัฒนาโปรแกรมคอมพิวเตอร์เชิงวัตถุ 3 มิติสำหรับงานออกแบบอาคารและบริหารโครงการก่อสร้าง เนื่องจาก

ภาษาจาวามีไลบรารี (Library) มาตรฐานที่บรรจุฟังก์ชันการทำงานที่สนับสนุนการขึ้นรูปและการวิเคราะห์เชิง 3 มิติซึ่งช่วยอำนวยความสะดวกในการพัฒนาโปรแกรมคอมพิวเตอร์ที่ต้องการแสดงภาพหรือการวิเคราะห์ในลักษณะ 3 มิติ จุดเด่นประการหนึ่งของภาษาจาวาคือ สามารถทำงานผ่านระบบเครือข่ายและระบบอินเทอร์เน็ต (Internet) โดยผู้ใช้สามารถเรียกใช้งานโปรแกรมคอมพิวเตอร์ผ่านทาง Web Browser โดยไม่จำเป็นต้องเสียค่าใช้จ่ายในการจัดซื้อหรือติดตั้งโปรแกรมใดๆ

อย่างไรก็ดีประเทศไทยยังขาดซึ่งงานวิจัยและองค์ความรู้ในการนำภาษาจาวามาใช้ในการพัฒนาโปรแกรมคอมพิวเตอร์เชิงวัตถุ 3 มิติสำหรับงานออกแบบอาคารและบริหารโครงการก่อสร้าง การขาดซึ่งงานวิจัยและองค์ความรู้ในสาขานี้ก่อให้เกิดอุปสรรคในการพัฒนาโปรแกรมคอมพิวเตอร์ประยุกต์ขึ้นใช้งานได้เอง ทั้งนี้ส่งผลให้ผู้ประกอบการอุตสาหกรรมก่อสร้างในประเทศไทยจำเป็นต้องพึ่งโปรแกรมสำเร็จรูปที่ต้องนำเข้าจากต่างประเทศที่มีราคาแพงโดยปราศจากทางเลือก

2. วัตถุประสงค์

เพื่อพัฒนาองค์ความรู้และวิธีการสร้างแบบจำลองเชิงวัตถุ 3 มิติมาตราส่วนจริงสำหรับงานออกแบบอาคารและบริหารโครงการก่อสร้างด้วยภาษาจาวา จุดประสงค์ของการวิจัยโดยละเอียดมีดังนี้

1. วิเคราะห์และออกแบบฐานข้อมูลสำหรับจัดเก็บข้อมูลและสนับสนุนการขึ้นรูปและการวิเคราะห์เชิงวัตถุ 3 มิติขององค์อาคาร
2. พัฒนาชุดคำสั่ง (Algorithm) และซอร์สโคด (Source Code) ด้วยภาษาจาวาสำหรับฟังก์ชันการทำงานพื้นฐานดังต่อไปนี้
 - 2.1 การสร้างภาพแบบจำลองเชิงวัตถุ 3 มิติมาตราส่วนจริงด้วยภาษาจาวา
 - 2.2 การสร้างแบบรูป 2 มิติขององค์อาคารจากมุมมองต่างๆ ของแบบจำลองเชิงวัตถุ 3 มิติ
 - 2.3 การสร้างภาพตัดขององค์อาคารจากแนวการตัดต่างๆ ของแบบจำลองเชิงวัตถุ 3 มิติ
 - 2.4 การคำนวณปริมาณวัสดุก่อสร้างจากแบบจำลองเชิงวัตถุ 3 มิติ

3. วิธีดำเนินการวิจัย

3.1 วิเคราะห์ความต้องการพื้นฐาน (Functional Requirements)

การวิเคราะห์ความต้องการพื้นฐานเป็นขั้นตอนสำคัญในการพัฒนาแบบจำลองเชิงวัตถุ 3 มิติเพื่อให้แบบจำลองสามารถตอบสนองความต้องการของผู้ใช้อย่างครบถ้วน การวิเคราะห์ความต้องการพื้นฐานสามารถทำได้โดยการศึกษาจากระบบหรือโปรแกรมสำเร็จรูปที่มีการใช้งานในปัจจุบัน การศึกษาจากเอกสารหรืองานวิจัยในอดีต และการสัมภาษณ์ผู้ที่เกี่ยวข้องจากทุกฝ่าย

3.2 พัฒนาแบบจำลองเชิงตรรก (Logical Data Model) ของระบบฐานข้อมูล

การพัฒนาแบบจำลองเชิงตรรกเป็นการออกแบบระบบฐานข้อมูลเบื้องต้นโดยพิจารณาจากความต้องการพื้นฐานของแบบจำลองเชิงวัตถุ 3 มิติ จุดประสงค์ของการพัฒนาแบบจำลองเชิงตรรกเพื่อป้องกันข้อผิดพลาดและความซ้ำซ้อนของข้อมูลในระบบฐานข้อมูลเพื่อเพิ่มประสิทธิภาพและลดต้นทุนในการพัฒนาและบำรุงรักษาระบบฐานข้อมูล

3.3 พัฒนาแบบจำลองเชิงสัมพันธ์ (Physical Data Model) ของระบบฐานข้อมูล

การพัฒนาแบบจำลองเชิงสัมพันธ์เป็นการออกแบบระบบฐานข้อมูลโดยละเอียด โดยมีการวางแผนการจัดเก็บข้อมูลแบบตาราง ระบุรายละเอียดข้อมูลที่จะถูกบันทึกในตาราง และความสัมพันธ์เชิงตรรกะระหว่างตาราง รวมถึงการตรวจสอบ Normalization ซึ่งเป็นขั้นตอนสำคัญในการออกแบบระบบฐานข้อมูล แบบจำลองเชิงสัมพันธ์เปรียบเสมือนพิมพ์เขียวของระบบฐานข้อมูลที่ใช้อ้างอิงในการพัฒนาระบบฐานข้อมูล

3.4 พัฒนาชุดคำสั่ง (Algorithm)

ชุดคำสั่งแสดงขั้นตอนการทำงานของโปรแกรม ชุดคำสั่งหลักที่จะถูกพัฒนา ได้แก่

- ชุดคำสั่งในการสร้างแบบจำลองเชิงวัตถุ 3 มิติมาตราส่วนจริง
- ชุดคำสั่งในการสร้างแบบรูป 2 มิติจากแบบจำลองเชิงวัตถุ 3 มิติ
- ชุดคำสั่งในการสร้างภาพตัดจากแบบจำลองเชิงวัตถุ 3 มิติ
- ชุดคำสั่งในการถอดแบบและประมาณราคาจากแบบจำลองเชิงวัตถุ 3 มิติ

หนึ่งชุดคำสั่งที่พัฒนาขึ้นควรจะสามารถทำงานสอดคล้องกับไลบรารี (Library) มาตรฐานของภาษาจาวาเพื่อให้การทำงานของโปรแกรมเป็นไปอย่างถูกต้องและปราศจากความบกพร่อง

3.5 สร้างระบบฐานข้อมูลและเขียนซอร์สโค้ด (Source Code)

ขั้นตอนนี้เป็นการสร้างระบบฐานข้อมูลจากแบบจำลองเชิงสัมพันธ์และการเขียนซอร์สโค้ด (Source Code) ด้วยภาษาจาวาสำหรับการประมวลผลชุดคำสั่งและแสดงภาพแบบจำลองเชิงวัตถุ 3 มิติ การทำงานในขั้นตอนนี้รวมถึงการออกแบบโครงสร้างทางสถาปัตยกรรมในการเชื่อมโยงและส่งผ่านข้อมูลระหว่างฐานข้อมูลกับภาษาจาวาและโปรแกรมอื่นๆ ที่อาจเกี่ยวข้อง

3.6 ทดสอบและปรับแก้ซอร์สโค้ด

ขั้นตอนสุดท้ายเป็นการทดสอบและปรับแก้โปรแกรมเพื่อป้องกันข้อผิดพลาดที่อาจเกิดขึ้นในระหว่างการออกแบบและการเขียนโปรแกรม

หนึ่งโครงการวิจัยนี้ไม่ได้มุ่งเป้าในการพัฒนาโปรแกรมคอมพิวเตอร์ประยุกต์สำหรับสร้างแบบจำลองเชิงวัตถุ 3 มิติเนื่องจากการพัฒนาโปรแกรมคอมพิวเตอร์ประยุกต์ให้สามารถใช้งานได้จริงจะต้องใช้เวลาและงบประมาณจำนวนมากในการพัฒนาหน้าจอ (Interface) และฟังก์ชันการทำงานต่างๆ ที่จะช่วยอำนวยความสะดวกแก่ผู้ใช้ จุดมุ่งหมายของโครงการนี้คือการพัฒนาองค์ความรู้และฟังก์ชันการทำงานพื้นฐานที่จำเป็นสำหรับการพัฒนาแบบจำลองเชิงวัตถุ 3 มิติมาตราส่วนจริงสำหรับงานออกแบบอาคารและบริหารโครงการก่อสร้าง โดยองค์ความรู้ที่ได้จากงานวิจัยนี้จะสามารถนำไปใช้ในการพัฒนาโปรแกรมคอมพิวเตอร์ประยุกต์ให้สามารถใช้งานได้จริงในงานวิจัยต่อไป

4. ผลการวิจัย

4.1 การวิเคราะห์ความต้องการของระบบฐานข้อมูลแบบจำลองเชิงวัตถุ 3 มิติ

การนำแบบจำลองเชิงวัตถุ 3 มิติมาใช้สนับสนุนงานก่อสร้างเป็นที่นิยมมากในต่างประเทศ ตัวอย่างของอาคารที่ใช้แบบจำลอง 3 มิติมาใช้ในการออกแบบและบริหารโครงการ เช่น อาคารยูเรก้า ซึ่งเป็นอาคารขนาดใหญ่ที่มีความสูงถึง 300 เมตร จำนวน 92 ชั้น งบประมาณของโครงการ 354 ดอลลาร์สหรัฐ ตั้งอยู่ที่เมืองเมลเบิร์น ประเทศออสเตรเลีย ได้เริ่มทำการก่อสร้างเมื่อปี ค.ศ. 2004 โดยเจ้าของโครงการต้องการอาคารที่มีลักษณะที่ทันสมัย มีรูปร่างที่สวยงามเป็นธรรมชาติและดูเด่นแปลกตกว่าอาคารในลักษณะเก่า ๆ ขนาดที่ใหญ่โตของอาคารเป็นอุปสรรคในการทำงาน เพราะฉะนั้นเครื่องมือที่จะนำมาช่วยในการออกแบบจึงเป็นสิ่งสำคัญอย่างยิ่งในการที่จะช่วยให้การทำงานและออกแบบเป็นไปอย่างรวดเร็วและตรงตามความต้องการ

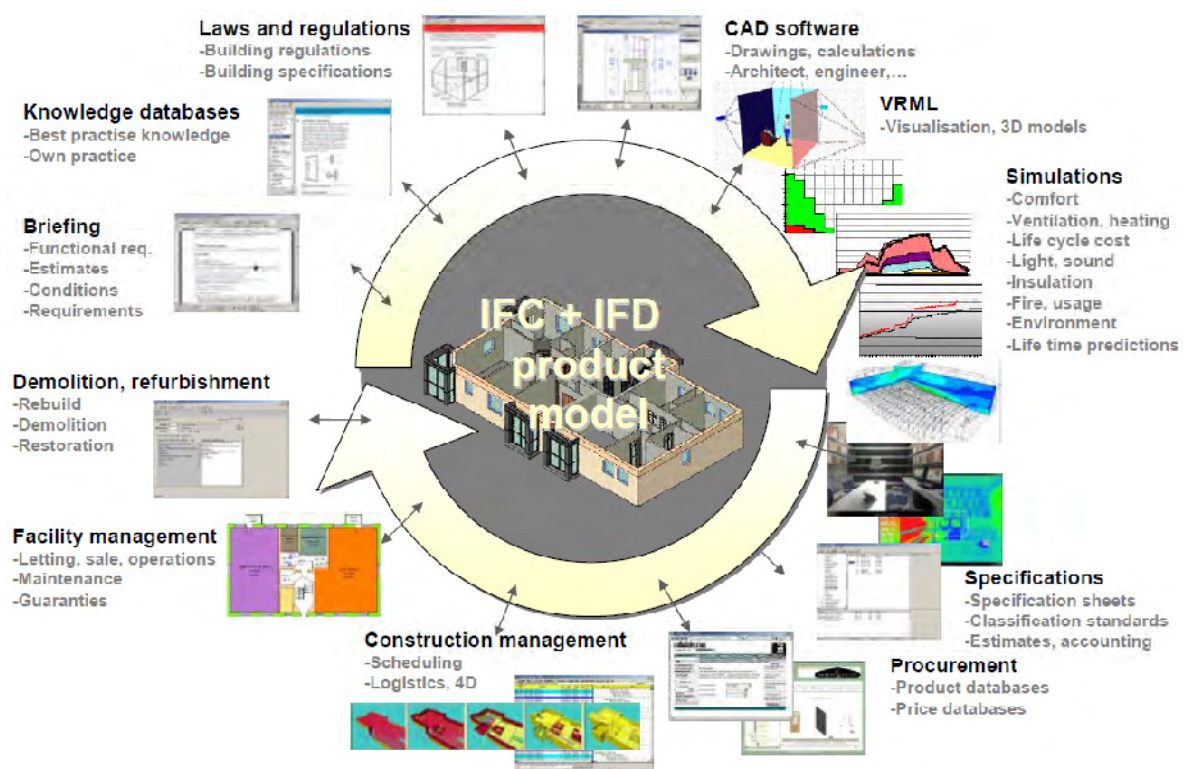
ทีมงานผู้ออกแบบได้เลือกใช้วิธีการออกแบบโดยการสร้างแบบจำลองเชิงวัตถุ 3 มิติของอาคารแทนที่จะใช้ 2D CAD ในแบบเดิมซึ่งมีข้อจำกัดหลายอย่าง เช่น เรื่องของการใช้ระยะเวลาการทำงานที่ยาวนาน ความยุ่งยากและน่าเบื่อหน่ายในการทำงาน ความถูกต้องของผลงานน้อย อีกทั้งถ่ายทอดแนวความคิดของผู้ออกแบบออกมาได้ไม่ชัดเจนและยากลำบาก การออกแบบโดยใช้แบบจำลองเชิงวัตถุ 3 มิติของอาคารจึงเป็นทางเลือกที่นำมาใช้ในการออกแบบอาคารหลังนี้ด้วยการออกแบบในลักษณะที่เป็นรูปธรรมสามารถเห็นรูปลักษณะของอาคารที่ออกแบบอย่างชัดเจน อีกทั้งการให้แสงเงาตามธรรมชาติอย่างสมจริงสามารถเห็นถึงผลกระทบทางสิ่งแวดล้อมทางอาคารส่งผลให้ผลงานที่ออกมามีคุณภาพดีและลดระยะเวลาในการทำงานลงได้มาก

ขั้นตอนในการออกแบบเริ่มจากการกำหนดแนวความคิดของอาคารให้ตรงตามความต้องการของเจ้าของ หลังจากนั้นจึงเริ่มสร้างรูปร่างลักษณะภายนอกของอาคารและสภาพแวดล้อมบริเวณรอบ ทำการให้แสงเงาตามธรรมชาติของอาคารในลักษณะต่าง ๆ ทำให้สามารถศึกษาวิเคราะห์ถึงผลกระทบทางสิ่งแวดล้อม ที่จะเกิดเนื่องจากขนาดที่ใหญ่โตของอาคาร จากนั้นจึงสร้างรายละเอียดภายในของอาคารแต่ละชั้น ในส่วนของการตกแต่งภายในสามารถสร้างวัสดุอุปกรณ์ที่จะใช้ในการตกแต่ง และสามารถจัดวางตำแหน่งของเฟอร์นิเจอร์รวมถึงอุปกรณ์ตกแต่งอื่น ๆ ได้อย่างเหมาะสม ด้วยการออกแบบและสร้างแบบจำลองของอาคารในลักษณะสมจริงสมจัง คล้ายกับว่าสามารถจับต้องได้จริง ซึ่งส่งผลให้การออกแบบสามารถสร้างผลงานออกมาได้อย่างมีคุณภาพและใช้ระยะเวลาในการทำงานน้อยลง

เอกสารในการก่อสร้างประมาณ 1,000 แผ่น ของกระดาษขนาด A1 สามารถสร้างออกมาจากแบบจำลองเชิงวัตถุ 3 มิติของอาคาร ซึ่งประกอบด้วย รายการประกอบ รูปด้าน รูปตัด และแบบแสดงรายละเอียดทั้งหมดของอาคาร ซึ่งสามารถพิมพ์ออกมาโดยอัตโนมัติ โดยน้ำหนักของเส้น การสร้างเส้นประ การระบายสี และรายละเอียดต่าง ๆ ของเนื้องานมีคุณภาพดี ไม่ต่างไปจากการสร้างงานด้วย 2d CAD อีกทั้งการเปลี่ยนแปลงแก้ไขแบบ สามารถทำได้ง่ายกว่า เนื่องจากเมื่อมีการแก้ไขส่วนใดส่วนหนึ่งของอาคาร จะส่งผลให้ส่วนอื่นที่เกี่ยวข้องแก้ไขตามไปโดยอัตโนมัติ อีกทั้งจำนวนเอกสารผลิตพลาดยังมีจำนวนน้อยมากอีกด้วย

การสร้างส่วนของอาคารที่ลักษณะคล้ายกันหรือซ้ำ ๆ เช่น ห้องพักหรือห้องน้ำ ได้มีการสร้างไฟล์แยกส่วนไว้ต่างหากแล้วจึงนำไปเชื่อมโยงกับส่วนของอาคารที่ใช้งาน การทำงานในลักษณะนี้ สามารถลดขนาดไฟล์ข้อมูลของอาคารลงได้มาก โดยไฟล์ข้อมูลของอาคารยูเรกามีขนาดทั้งหมดเพียง 300 MB สิ่งเหล่านี้ สามารถอธิบายได้ถึงความแตกต่างระหว่างการเลือกใช้การออกแบบอาคารโดยการสร้างแบบจำลองเชิงวัตถุ 3 มิติ แทนที่ใช้ 2D CAD ในระบบเดิมได้เป็นอย่างดี

แบบจำลองเชิงวัตถุ 3 มิตินับเป็นพื้นฐานสำคัญของเทคโนโลยี Building Information Modeling (BIM) ซึ่งเป็นวิธีจัดการงานก่อสร้างสมัยใหม่ โดยการใช้แบบจำลองเชิงวัตถุ 3 มิติสำหรับจัดเก็บและนำเสนอข้อมูลองค์ประกอบอาคาร (Building Element) สำหรับใช้สนับสนุนการวิเคราะห์ ออกแบบวางแผน และประมาณราคาโครงการ รวมถึงการติดตามความก้าวหน้างานก่อสร้างและการใช้งานอาคารตลอดวงจรโครงการ (Project Life Cycle) รูปที่ 4.1 แสดงการตัวอย่างการนำเทคโนโลยี BIM มาประยุกต์ใช้ในงานวิศวกรรมโยธาซึ่งแสดงถึงศักยภาพในการนำแบบจำลองเชิงวัตถุ 3 มิติมาใช้สนับสนุนการทำงานได้อย่างหลากหลาย ตารางที่ 4.1 แสดงกลุ่มบุคคล (Stakeholders) ที่เกี่ยวข้องในการนำเทคโนโลยี BIM มาใช้งาน



รูปที่ 4.1 การตัวอย่างการประยุกต์ใช้งาน BIM ในงานวิศวกรรมโยธา (NIBS 2007)

ตารางที่ 4.1 กลุ่มบุคคลที่เกี่ยวข้องในการนำข้อมูล BIM ไปใช้งาน (NIBS 2007)

กลุ่มบุคคล	การนำข้อมูลไปใช้
Owners	High level summary information about their facilities.
Planners	Existing information about physical site(s) and corporate program needs.
Realtors	Information about a site or facility to support purchase or sale.
Appraisers	Information about the facility to support valuation.
Mortgage Bankers	Information about demographics, corporations, and viability.
Designers	Planning and site information.
Engineers	Electronic model from which to import into design and analysis software.
Cost and Quantity Estimators	Electronic model to obtain accurate quantities.
Specifiers	Intelligent objects from which to specify and link to later phases.
Attorneys	More accurate legal descriptions to defend or on which to base litigation.
Construction Contractors	Intelligent objects for bidding and ordering and a place to store gained information.
Sub-Contractors	Clearer communication and same support for contractors.
Fabricators	Can use intelligent model for numerical controls for fabrication.
Code Officials	Code checking software can process model faster and more accurately.
Facility Managers	Provides product, warranty, and maintenance information.
Maintenance and Sustainment	Easily identify products for repair parts or replacement.
Renovation and Restoration	Minimizes unforeseen conditions and the resulting cost.
Disposal and Recycling	Better knowledge of what is recyclable.
Scoping, Testing, and Simulation	Electronically build facility and eliminate conflicts.
Safety and Occupational Health	Knowledge of what materials are in use and MSDS.
Environmental and NEPA	Improved information for environmental impact analysis.
Plant Operations	3D visualization of processes.

กลุ่มบุคคล	การนำข้อมูลไปใช้
Energy and LEED	Optimized energy analysis more easily accomplished allows for more review of alternatives such as impact of building rotation or relocation on site.
Space and Security	Intelligent objects in 3D provide better understanding of vulnerabilities.
Network Managers	3D physical network plan is invaluable for troubleshooting.
CIOs.	Basis for better business decisions and information about existing infrastructure.
Risk Management	Better understanding of potential risks and how to avoid or minimize.
Occupant Support	Visualization of facility for way finding (building users often cannot read floor plans).
First Responders	Minimize loss of life and property with timely and accurate information.

แบบจำลองเชิงวัตถุ 3 มิติประกอบด้วยข้อมูล 2 ส่วนคือ ข้อมูลเชิงเรขาคณิต (Geometry Data) และข้อมูลคุณลักษณะ (Attribute Data) โดยข้อมูลเชิงเรขาคณิตประกอบด้วยจุด (point) เส้น (line) รูปหลายเหลี่ยม (polygon) และรูปทรง 3 มิติ (3D shape) สำหรับแสดงตำแหน่ง (location) รูปร่าง (shape) และขนาด (size) ขององค์ประกอบอาคารแต่ละชั้น ส่วนข้อมูลคุณลักษณะเป็นข้อมูลสารสนเทศแสดงรายละเอียดคุณลักษณะและข้อมูลอื่นๆ ที่เกี่ยวข้องกับองค์ประกอบอาคารแต่ละชั้น เพื่อให้แบบจำลองเชิงวัตถุ 3 มิติที่จัดทำขึ้นสามารถสนับสนุนการพัฒนาเทคโนโลยี BIM ในอนาคตได้อย่างมีประสิทธิภาพ ระบบฐานข้อมูลที่พัฒนาขึ้นสำหรับจัดเก็บข้อมูลแบบจำลองเชิงวัตถุ 3 มิติควรตอบสนองความต้องการและเงื่อนไขการใช้งานดังต่อไปนี้

- **รองรับการจัดการข้อมูลเชิงเรขาคณิตแบบ 3 มิติที่มีความซับซ้อนสูง**
เนื่องจากแบบจำลองเชิงวัตถุที่จัดทำขึ้นเป็นแบบจำลองดิจิทัลแบบ 3 มิติ ดังนั้นระบบฐานข้อมูลที่นำมาใช้งานจะต้องรองรับการจัดเก็บข้อมูลแบบ 3 มิติได้เป็นอย่างดีเพื่อให้สามารถจัดเก็บข้อมูลเชิงเรขาคณิต (Geometry Data) ได้อย่างถูกต้องและครบถ้วน นอกจากนั้นข้อมูลเชิงเรขาคณิตของแบบจำลองเชิงวัตถุ 3 มิติจะมีขนาดใหญ่และมีความซับซ้อนสูงเนื่องจากอาคารแต่ละหลังจะประกอบด้วยองค์ประกอบอาคาร (Building Element) ซึ่งมีขนาดและรูปร่างที่หลากหลาย อาทิเช่น เสา คาน พื้น ผนัง ฐานราก ประตู หน้าต่าง หลังคา ระบบไฟฟ้า ระบบปรับอากาศ และระบบสุขาภิบาล โดยเฉพาะในกรณีที่เป็นอาคารคอนกรีตเสริมเหล็กและต้องการจัดเก็บข้อมูลรายละเอียดเหล็กเสริมอาคารที่มีจำนวนมากจะส่งผลให้ข้อมูลเชิงเรขาคณิตของแบบจำลองเชิงวัตถุ 3 มิติมีขนาดใหญ่และยากต่อการบริหารจัดการ ดังนั้นระบบฐานข้อมูลที่นำมาใช้งานในการจัดเก็บข้อมูล

แบบจำลองเชิงวัตถุ 3 มิติจะต้องมีความยืดหยุ่นในการจัดการข้อมูลเชิงเรขาคณิตสูงและสามารถบริหารจัดการข้อมูลเชิงเรขาคณิตขนาดใหญ่ได้อย่างมีประสิทธิภาพ

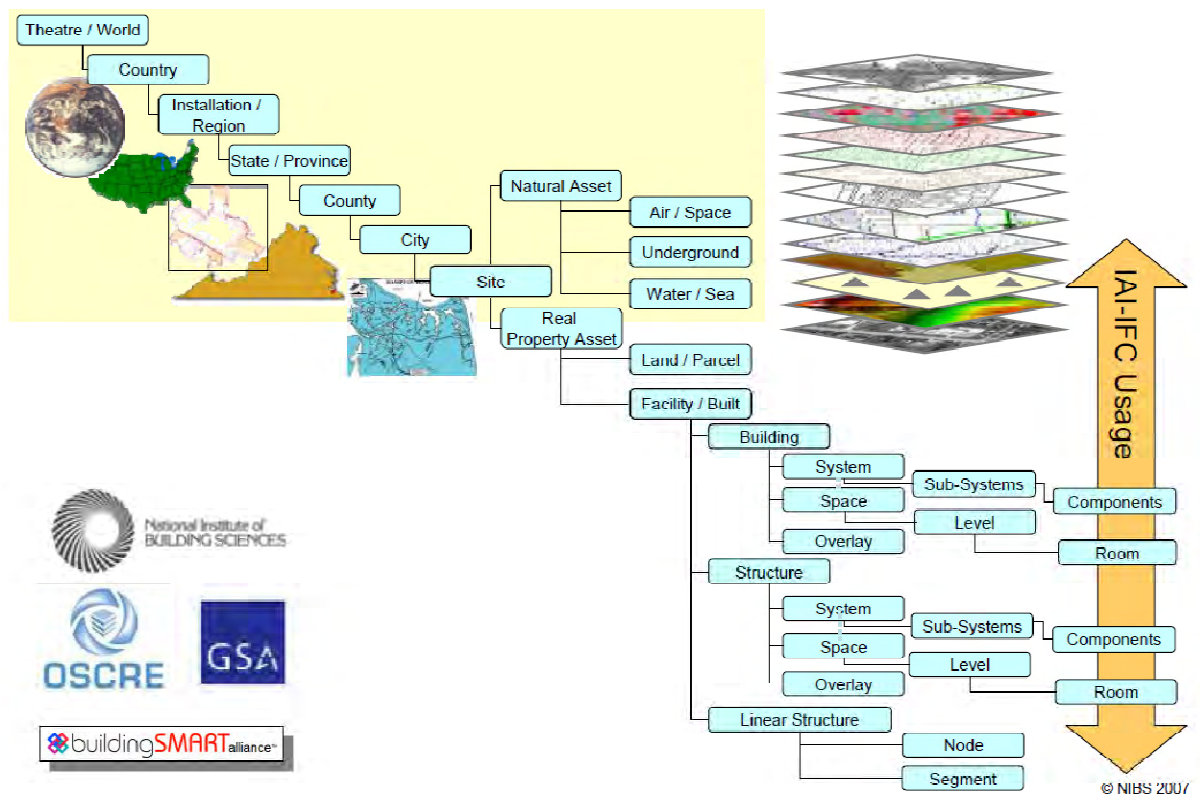
- **สามารถจัดเก็บข้อมูลเชิงเรขาคณิตและข้อมูลคุณลักษณะในฐานข้อมูลเดียวกัน**
ในอดีตข้อมูลเชิงเรขาคณิตและข้อมูลคุณลักษณะมักจะถูกจัดเก็บแยกเป็นอิสระจากกัน โดยข้อมูลคุณลักษณะจะถูกจัดเก็บในระบบฐานข้อมูลส่วนข้อมูลเชิงเรขาคณิตจะถูกจัดเก็บในแฟ้มข้อมูล (File-based system) แยกเป็นอิสระจากฐานข้อมูลคุณลักษณะ โดยข้อมูลคุณลักษณะและข้อมูลเชิงเรขาคณิตจะถูกเชื่อมโยงกันผ่านทางรหัสประจำตัว (Identification key) ที่จัดทำขึ้น การบริหารจัดการข้อมูลในรูปแบบนี้ทำให้การแก้ไข การสืบค้น และการวิเคราะห์ข้อมูลขาดความยืดหยุ่นและไม่มีประสิทธิภาพ ดังนั้นระบบฐานข้อมูลที่พัฒนาขึ้นควรรองรับการบริหารจัดการข้อมูลเชิงเรขาคณิตและข้อมูลคุณลักษณะร่วมกันในฐานข้อมูลเดียวกัน (Unified Storage) เพื่อเพิ่มประสิทธิภาพในการแก้ไข การสืบค้น และการวิเคราะห์ข้อมูลแบบจำลองเชิงวัตถุ 3 มิติ
- **สนับสนุนการวิเคราะห์เชิงพื้นที่ (Spatial Analysis)**
ระบบฐานข้อมูลสำหรับจัดเก็บข้อมูลแบบจำลองเชิงวัตถุ 3 มิติควรสนับสนุนการวิเคราะห์เชิงพื้นที่ เช่น การคำนวณความยาว (Length) การวัดระยะทาง (Distance) การคำนวณขนาดพื้นที่ (Area) และพื้นที่ผิว (Surface Area) การคำนวณปริมาตร (Volume) การรวมข้อมูล (Union) การคำนวณการตัดกัน (Intersection) และการคำนวณความแตกต่าง (Difference) ของข้อมูลเชิงเรขาคณิต เป็นต้น ทั้งนี้เพื่อเพิ่มความยืดหยุ่นและอำนวยความสะดวกในการนำแบบจำลองเชิงวัตถุ 3 มิติมาใช้สนับสนุนงานออกแบบอาคารและบริหารโครงการก่อสร้างรวมถึงสนับสนุนฟังก์ชันการทำงานพื้นฐานของเทคโนโลยี BIM
- **รองรับการสร้างดัชนีข้อมูลเชิงเรขาคณิต (Spatial Index)**
การสร้างดัชนีฐานข้อมูล (Database Index) เป็นเทคนิคพื้นฐานที่สำคัญในการสร้างตัวนำทางในการสืบค้นข้อมูลให้เร็วขึ้น ทั้งนี้จะช่วยเพิ่มประสิทธิภาพในการสืบค้นข้อมูล โดยเฉพาะสำหรับฐานข้อมูลที่มีขนาดใหญ่ เนื่องจากข้อมูลเชิงเรขาคณิตของแบบจำลองเชิงวัตถุ 3 มิติเป็นข้อมูลที่มีขนาดใหญ่และมีความซับซ้อนสูงดังที่ได้อธิบายไว้ในเบื้องต้น ดังนั้นการสร้างดัชนีของข้อมูลเชิงเรขาคณิต (Spatial Index) จึงเป็นสิ่งสำคัญต่อการสืบค้น (Spatial Query) และวิเคราะห์ข้อมูลเชิงพื้นที่ (Spatial Analysis) ของแบบจำลองเชิงวัตถุ 3 มิติให้เป็นไปอย่างรวดเร็วและมีประสิทธิภาพ

- **มีความยืดหยุ่นในการเชื่อมโยงกับโปรแกรมภายนอก (External Application)**

เนื่องจากแบบจำลองเชิงวัตถุ 3 มิติที่สร้างขึ้นจะถูกนำมาใช้งานอย่างหลากหลายและจะต้องถูกเชื่อมโยงกับโปรแกรมภายนอก (External Application) จำนวนมากดังแสดงในรูปที่ 3.1 ดังนั้นระบบฐานข้อมูลสำหรับจัดเก็บข้อมูลแบบจำลองเชิงวัตถุ 3 มิติจึงควรมีความยืดหยุ่นในการเชื่อมโยงกับโปรแกรมภายนอก เพื่อเพิ่มประสิทธิภาพในการนำแบบจำลองเชิงวัตถุ 3 มิติไปประยุกต์ใช้ร่วมกับโปรแกรมภายนอกต่างๆ
- **รองรับการเข้าถึงข้อมูลในเวลาเดียวกัน (Concurrency Access)**

แบบจำลองเชิงวัตถุ 3 มิติที่สร้างขึ้นควรรองรับการบริหารโครงการในระดับองค์กร (Enterprise Project Management) กล่าวคือข้อมูลแบบจำลองเชิงวัตถุ 3 มิติที่สร้างขึ้นควรจัดเก็บในฐานข้อมูลกลางเพื่อให้บุคลากรทุกฝ่ายที่เกี่ยวข้องสามารถที่จะนำเข้าแก้ไข และสืบค้นข้อมูลได้ตลอดเวลา ทั้งนี้เพื่อให้บุคลากรทุกฝ่ายใช้ฐานข้อมูลเดียวกันในการวิเคราะห์ ออกแบบ วางแผน ประเมินราคา และการติดตามความก้าวหน้างานก่อสร้าง เพื่อลดความขัดแย้งในการทำงานและปรับปรุงการสื่อสารภายในโครงการให้ เป็นไปอย่างมีประสิทธิภาพ ดังนั้นระบบฐานข้อมูลแบบจำลองเชิงวัตถุ 3 มิติควรรองรับการเรียกใช้ข้อมูลจากผู้ใช้หลายคนในเวลาเดียวกัน (Concurrency Access) เพื่อไม่ให้เกิดข้อจำกัดในการเรียกใช้ฐานข้อมูลและเพื่อให้แบบจำลองเชิงวัตถุ 3 มิติที่สร้างขึ้นสามารถตอบโจทย์การใช้งานได้อย่างสมบูรณ์แบบ
- **รองรับการเชื่อมโยงกับระบบสารสนเทศภูมิศาสตร์ (GIS)**

การประยุกต์ใช้งานเทคโนโลยี BIM ในอนาคตจะเชื่อมโยงข้อมูล BIM เข้ากับเทคโนโลยีสารสนเทศภูมิศาสตร์ (Geographic Information System - GIS) ดังแสดงในรูปที่ 4.2 ดังนั้นเพื่อรองรับการขยายตัวของเทคโนโลยี BIM ในอนาคตระบบฐานข้อมูลแบบจำลองเชิงวัตถุ 3 มิติควรรองรับการเชื่อมโยงกับระบบสารสนเทศภูมิศาสตร์ (GIS) เพื่อเพิ่มความยืดหยุ่นในการพัฒนาเทคโนโลยีต่อยอดในอนาคต



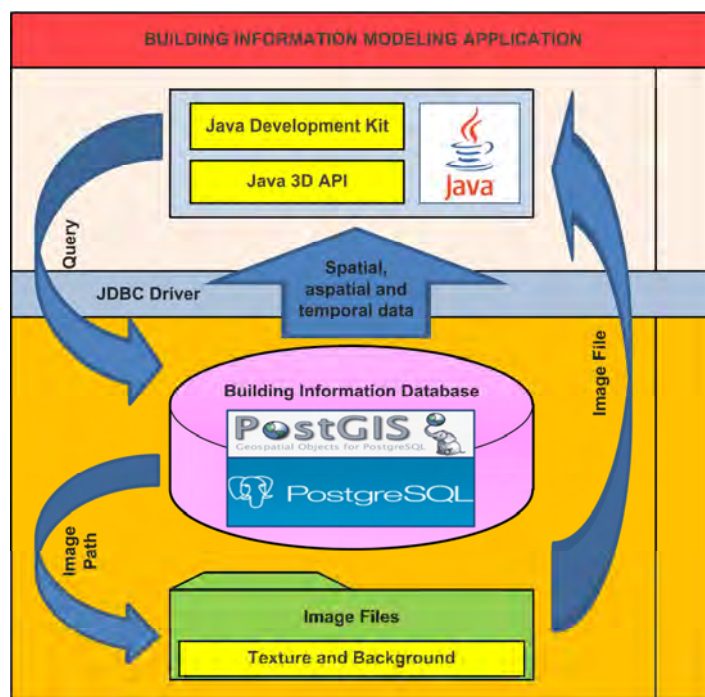
รูปที่ 4.2 ภาพรวมลำดับชั้นการเชื่อมโยงข้อมูล BIM (NIBS 2007)

- เป็นระบบฐานข้อมูลแบบโอเพนซอร์ส (Open Source)

ในปัจจุบันทั้งหน่วยงานภาครัฐและภาคเอกชนของประเทศไทยสูญเสียค่าใช้จ่ายจำนวนมากในการจัดซื้อซอฟต์แวร์สำเร็จรูปจากต่างประเทศ เพื่อลดการขาดดุลการค้าของประเทศระบบฐานข้อมูลที่น่าสนใจนำมาใช้งานควรเป็นระบบฐานข้อมูลแบบโอเพนซอร์ส (Open Source) ที่สามารถนำมาใช้งานได้ฟรีและมีลิขสิทธิ์ถูกต้องตามกฎหมาย ทั้งนี้เพื่อเป็นการส่งเสริมการใช้ซอฟต์แวร์แบบโอเพนซอร์สของประเทศและลดการขาดดุลการค้าของประเทศไทย

4.2 การออกแบบระบบสารสนเทศแบบจำลองเชิงวัตถุ 3 มิติ

การออกแบบระบบเป็นการกำหนดซอฟต์แวร์ที่จำเป็นสำหรับการพัฒนาระบบและวิธีการในการสื่อสารและเชื่อมโยงซอฟต์แวร์ทั้งหมดเข้าด้วยกัน เพื่อให้ซอฟต์แวร์ทั้งหมดสามารถทำงานร่วมกันอย่างเป็นเอกภาพและสามารถตอบสนองความต้องการในการใช้งานได้อย่างมีประสิทธิภาพ รูปที่ 4.3 แสดงภาพรวมต้นแบบระบบ (Prototype System) สำหรับสนับสนุนการสร้างแบบจำลอง การจัดการข้อมูลแบบจำลอง และการวิเคราะห์ข้อมูลแบบจำลองเชิงวัตถุ 3 มิติสำหรับงานออกแบบอาคารและบริหารโครงการก่อสร้างด้วยภาษาจาวา โดยมีรายละเอียดการออกแบบระบบต้นแบบดังต่อไปนี้



รูปที่ 4.3 ภาพรวมโครงสร้างต้นแบบระบบ (Prototype System)

ผู้วิจัยเลือกใช้โปรแกรม PostgreSQL (version 8.4) สำหรับบริหารจัดการฐานข้อมูลแบบจำลองเชิงวัตถุ 3 มิติ เนื่องจากโปรแกรม PostgreSQL เป็นโปรแกรมโอเพนซอร์ส (Open Source) ที่ได้รับการยอมรับในระดับสากลสำหรับจัดเก็บและบริหารจัดการฐานข้อมูลสารสนเทศ (MIS) ขนาดใหญ่ โดยผู้ใช้งานสามารถนำโปรแกรม PostgreSQL มาใช้งานได้อย่างถูกต้องตามกฎหมาย ทั้งนี้เพื่อเป็นการส่งเสริมการใช้ซอฟต์แวร์แบบโอเพนซอร์สของประเทศไทยและลดการขาดดุลการค้าของประเทศในการซื้อซอฟต์แวร์สำเร็จรูปมาใช้งาน

โปรแกรม PostGIS เป็นโปรแกรมเสริมด้านการบริหารจัดการข้อมูลเชิงพื้นที่ (Geospatial Extension) ที่ช่วยเพิ่มความสามารถให้โปรแกรม PostgreSQL สามารถจัดเก็บข้อมูลเชิงเรขาคณิตของแบบจำลองเชิงวัตถุ 3 มิติได้อย่างมีประสิทธิภาพ ตารางที่ 4.2 สรุปประเภทและมิติของข้อมูลเชิงเรขาคณิตที่รองรับโดยโปรแกรม PostGIS จากตารางจะพบว่าโปรแกรม PostGIS มีความยืดหยุ่นในการบริหารจัดการข้อมูลเชิงเรขาคณิตสูง โดยโปรแกรม PostGIS รองรับการจัดเก็บข้อมูลทั้งในรูปแบบจุด (Point) เส้น (Line) รูปหลายเหลี่ยม (Polygon) และข้อมูลผสม (Geometry Collection) ซึ่งเป็นการผสมกันระหว่างข้อมูลจุด เส้น และรูปหลายเหลี่ยม นอกจากนั้นโปรแกรม PostGIS ยังรองรับการจัดเก็บข้อมูลเชิงเรขาคณิตได้ตั้งแต่ 2 ถึง 4 มิติทำให้มีความยืดหยุ่นในการบริหารจัดการข้อมูลเชิงเรขาคณิตของแบบจำลองเชิงวัตถุ 3 มิติ

ตารางที่ 4.2 ประเภทและมิติของข้อมูลเชิงเรขาคณิตที่รองรับโดยโปรแกรม PostGIS

Geometry Type	Geometry Dimension		
	<u>2D</u> (x,y)	<u>3D</u> (x,y,z) or (x,y,m ¹)	<u>4D</u> (x,y,z,m ¹)
Point	SF ²	ESF ³	ESF ³
Line	SF ²	ESF ³	ESF ³
Polygon	SF ²	ESF ³	ESF ³
Multi Point	SF ²	ESF ³	ESF ³
Multi Line	SF ²	ESF ³	ESF ³
Multi Polygon	SF ²	ESF ³	ESF ³
Geometry Collection	SF ²	ESF ³	ESF ³
Curve	SQL-MM ⁴	SQL-MM ⁴	SQL-MM ⁴
Compound Curve	SQL-MM ⁴	SQL-MM ⁴	SQL-MM ⁴
Multi Curve	SQL-MM ⁴	SQL-MM ⁴	SQL-MM ⁴
Curve Polygon	SQL-MM ⁴	SQL-MM ⁴	SQL-MM ⁴
Multi Surface	SQL-MM ⁴	SQL-MM ⁴	SQL-MM ⁴

¹ Measure value such as linear referencing measurement or Unix time

² Simple Features for SQL specification defined by Open Geospatial Consortium

³ Extended Simple Features for 3D and 4D spatial data representation

⁴ SQL Multimedia specification for circularly interpolated curves

โปรแกรม PostGIS สามารถจัดเก็บข้อมูลเชิงเรขาคณิตได้ทั้งในรูปแบบ Well-Known Text (WKT) และ Well-Known Binary (WKB) ซึ่งเป็นมาตรฐานการจัดเก็บข้อมูลเชิงเรขาคณิตกำหนดโดย Open Geospatial Consortium (OGC) ดังตัวอย่างในตารางที่ 4.3 และ 4.4 ข้อดีของการจัดเก็บข้อมูลในรูปแบบ WKT คือเป็นข้อมูลที่อยู่ในรูปแบบ Text ที่คนสามารถดูแล้วเข้าใจได้ (Human Understandable Format) ส่วนข้อดีของการจัดเก็บข้อมูลในรูปแบบ WKB คือการเพิ่มประสิทธิภาพและลดระยะเวลาที่ใช้ในการประมวลผลข้อมูลเชิงเรขาคณิตโดยการจัดเก็บข้อมูลแบบ Binary ซึ่งสามารถประมวลผลได้เร็วกว่าข้อมูลแบบ Text อย่างไรก็ตามโปรแกรม PostGIS มีฟังก์ชันสนับสนุนการแปลงรูปแบบข้อมูลระหว่าง WKT และ WKB ได้โดยอัตโนมัติ

ตารางที่ 4.3 มาตรฐานรูปแบบข้อมูลเชิงเรขาคณิตกำหนดโดย OGC

ชนิดข้อมูล	ตัวอย่างมาตรฐานรูปแบบข้อมูลแบบ Well-Known Text (WKT)
จุด	POINT(0 0)
เส้น	LINESTRING(0 0,1 1,1 2)
รูปหลายเหลี่ยม	POLYGON((0 0,4 0,4 0 4,0 0),(1 1, 2 1, 2 2, 1 2,1 1))
จุด	MULTIPOINT(0 0,1 2)
เส้น	MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))
รูปหลายเหลี่ยม	MULTIPOLYGON(((0 0,4 0,4 0 4,0 0),(1 1,2 1,2 2,1 2,1 1)), ((-1 -1,-1 -2,-2 -2,-2 -1,-1 -1)))
ผสม	GEOMETRYCOLLECTION(POINT(2 3),LINESTRING((2 3,3 4)))

ตารางที่ 4.4 ตัวอย่างการจัดเก็บข้อมูลในรูปแบบ WKT และ WKB

ID	Well Known Text (WKT)	Well Known Binary (WKB)
1	POINT(755969.000099042 1697901.00259591)	010100000049FB0C00021227412720AA006DE83941
2	POINT(614796.194341468 1507924.11494889)	010100000090B9806318C322415A4A6D1D54023741
3	POINT(604554.727964299 1485653.75740339)	010100000093BCB774157322413730E5C155AB3641
4	POINT(604314.534428091 1483691.15899972)	0101000000088FA01135712241B534B428ABA33641
5	POINT(605785.969181435 1479796.46920045)	0101000000918C38F0B37C224138851D7874943641
6	POINT(604949.459484681 1479308.24647954)	0101000000789341EB2A7622417448193F8C923641
7	POINT(612113.13821114 1484124.2045045)	01010000004C9CC34622AE22412F685A345CA53641
8	POINT(611414.450722513 1483407.56015117)	0101000000EA19C5E6ACA822411C11668F8FA23641
9	POINT(604630.520348351 1490209.05008389)	010100000060196B0AAD732241344CD20C21BD3641
10	POINT(606666.859038227 1481141.23925095)	0101000000C6DBD3B795832241F08C3F3DB5993641

ข้อดีอีกประการหนึ่งของโปรแกรม PostGIS คือความสามารถในการสร้างดัชนีข้อมูลเชิงพื้นที่ (Spatial Index) แบบ Generalized Search Trees (GiST) ซึ่งจะช่วยลดระยะเวลาการค้นหา (Computational Search Time) และการวิเคราะห์ข้อมูลเชิงพื้นที่ (Spatial Analysis) โดยเฉพาะในฐานข้อมูลแบบจำลองเชิงวัตถุ 3 มิติที่มีขนาดใหญ่

โปรแกรม PostGIS มีฟังก์ชันการวิเคราะห์เชิงพื้นที่ (Spatial Analysis) ที่หลากหลายสำหรับใช้สนับสนุนงานบริหารโครงการก่อสร้าง อาทิเช่น การคำนวณความยาว การคำนวณระยะทาง การคำนวณ

ความยาวของเส้นรอบรูป การคำนวณจุดศูนย์กลางของวัตถุ และการคำนวณขนาดพื้นที่ของวัตถุ ซึ่งเป็นประโยชน์อย่างมากสำหรับการคำนวณปริมาณวัสดุและการประมาณราคา นอกจากนี้โปรแกรม PostGIS ยังมีฟังก์ชันการวิเคราะห์เชิงพื้นที่ที่เป็นประโยชน์ต่อการตรวจสอบความขัดแย้งในแบบก่อสร้าง (Design Conflicts) และความสัมพันธ์ระหว่างองค์ประกอบอาคาร อาทิเช่น union, intersection, equal, touch, contain และ buffer เป็นต้น

การใช้งานโปรแกรม PostgreSQL ร่วมกับโปรแกรม PostGIS ทำให้สามารถจัดเก็บข้อมูลสารสนเทศ (MIS) และข้อมูลเชิงเรขาคณิตทั้งหมดในฐานข้อมูลเดียวกัน ทำให้มีเพียงฐานข้อมูลเดียว (Unified Storage) ซึ่งง่ายต่อการจัดการและบำรุงรักษา นอกจากนี้ยังช่วยเพิ่มความน่าเชื่อถือและเสถียรภาพของฐานข้อมูล (Transactional Integrity) ในกรณีที่ระบบมีผู้ใช้หลายคนกำลังสืบค้นและแก้ไขข้อมูลเดียวกันในเวลาพร้อมกัน (Multiple Edits at Concurrent Time)

การใช้งานโปรแกรม PostgreSQL ร่วมกับโปรแกรม PostGIS ยังทำให้สามารถเชื่อมโยงฐานข้อมูลแบบจำลองเชิงวัตถุ 3 มิติกับโปรแกรมสารสนเทศภูมิศาสตร์ (GIS) เช่น โปรแกรม Mapserver, GeoServer, ArcGIS, Quantum GIS และ uDig ซึ่งเป็นการเพิ่มความยืดหยุ่นในการพัฒนาเทคโนโลยีต่อยอดในการเชื่อมโยงแบบจำลองเชิงวัตถุ 3 มิติกับระบบสารสนเทศภูมิศาสตร์ในอนาคต นอกจากนี้โปรแกรม PostgreSQL ยังมีความยืดหยุ่นในการเชื่อมต่อการสื่อสารกับโปรแกรมภายนอก ดังแสดงในตารางที่ 4.5

ตารางที่ 4.5 การเชื่อมโยงโปรแกรม PostgreSQL กับโปรแกรมภายนอก (Client Interface)

Name	Language	Comments
DBD::Pg	Perl	Perl DBI driver
JDBC	JDBC	Type 4 JDBC driver
Libpqxx	C++	New-style C++ interface
Npgsql	.NET	.NET data provider
ODBCng	ODBC	An alternative ODBC driver
Pgtclng	Tcl	-
psqlODBC	ODBC	The most commonly-used ODBC driver
Psycopg	Python	DB API 2.0-compliant

ผู้วิจัยใช้ภาษาจาวา (Java Programming Language) ในการสร้างแบบจำลองเชิงวัตถุ 3 มิติ เนื่องจากเป็นภาษาเชิงวัตถุ (Object Oriented) ที่ใช้งานอย่างแพร่หลายทั่วโลกและมีความยืดหยุ่นสูงสำหรับการพัฒนาโปรแกรมประยุกต์ ภาษาจาวาเหมาะสมในการนำมาใช้พัฒนาแบบจำลองอาคาร

เชิงวัตถุ 3 มิติสำหรับงานออกแบบอาคารและบริหารโครงการก่อสร้าง เนื่องจากโปรแกรมภาษาจาวามี คลาส (Class) สนับสนุนการขึ้นรูปและการวิเคราะห์เชิง 3 มิติซึ่งจะช่วยลดความซับซ้อนในการสร้าง การแสดงภาพ และการวิเคราะห์แบบจำลองอาคารเชิงวัตถุ 3 มิติ ภาษาจาวายังสามารถทำงานบน ระบบปฏิบัติการทุกประเภทโดยไม่มีข้อจำกัดและยังมีชุดคำสั่ง (Libraries) ในการสร้างหน้าจอ (User Interface) สำหรับติดต่อกับผู้ใช้ได้ด้วยความสะดวก

ในการพัฒนาโปรแกรมประยุกต์โดยภาษาจาวาเพื่อใช้ในการสร้างและแสดงผลแบบจำลองเชิงวัตถุ 3 มิติใช้ชุดคำสั่งในการพัฒนาด้วยกัน 5 ส่วนคือ Java Development Kit (JDK), java 3D API (javax.media.j3d), java 3D utility (com.sun.j3d.utils), java 3D vector math (javax.vecmath) และ java abstract window toolkit (AWT) (java.awt) โดย Java Development Kit (JDK) เป็นชุดโปรแกรม พื้นฐานสำหรับการทำงานของภาษาจาวาซึ่งมีลักษณะเป็นโปรแกรมเชิงวัตถุ (Object-Oriented Program) คือ มีแนวความคิดมองสิ่งต่างๆ เป็นวัตถุ โดย Class ที่ถูกสร้างขึ้นมาแล้วสามารถนำไปใช้กับ โปรแกรมอื่นได้โดยไม่ต้องเขียนขึ้นใหม่ซึ่งช่วยลดระยะเวลาและความซับซ้อนของการเขียนโปรแกรม ได้อย่างมาก ในส่วนของ java 3D API (javax.media.j3d), java 3D utility (com.sun.j3d.utils), java 3D vector math (javax.vecmath) และ java abstract window toolkit (AWT) (java.awt) เป็นโปรแกรม เสริมที่ช่วยให้สามารถสร้างวัตถุและวิเคราะห์เชิง 3 มิติได้ โปรแกรม Java3D API นั้นสามารถใช้ได้กับ Java Development Kit ตั้งแต่เวอร์ชัน 1.2 เป็นต้นไป ตารางที่ 4.6 สรุปหน้าที่ขององค์ประกอบแต่ละส่วน ในการสร้างและแสดงผลแบบจำลองเชิงวัตถุ 3 มิติ

ตารางที่ 4.6 องค์ประกอบในการสร้างแบบจำลองเชิงวัตถุ 3 มิติโดยภาษาจาวา

องค์ประกอบ	ฟังก์ชันการทำงาน
Java development kit	องค์ประกอบพื้นฐานในการพัฒนาโปรแกรมประยุกต์โดยภาษาจาวา
Java 3D API (javax.media.j3d)	องค์ประกอบพื้นฐานในการสร้างและแสดงผลวัตถุ 3 มิติ
Java 3D utility (com.sun.j3d.utils)	อำนวยความสะดวกในการสร้างและแสดงผลวัตถุ 3 มิติ
Java 3D vector math (javax.vecmath)	สนับสนุนฟังก์ชันการคำนวณในการสร้างและแสดงผลวัตถุ 3 มิติ
Java abstract window toolkit (java.awt)	สำหรับพัฒนาหน้าจอติดต่อผู้ใช้ในการสร้างและแสดงผลวัตถุ 3 มิติ

ในส่วนของรูปภาพที่จัดเก็บอยู่ในแฟ้มข้อมูลภายนอกในรูปที่ 4.3 ใช้สำหรับการแต่งภาพพื้นหลัง (Background) และภาพผิววัตถุ (Texture) ให้แบบจำลองเชิงวัตถุ 3 มิติที่สร้างขึ้นดูเสมือนจริง ภาษาจาวารองรับไฟล์ภาพทุกชนิดและไม่มีข้อกำหนดความละเอียด (Resolution) ขั้นต่ำของภาพ อย่างไรก็ตาม ภาพที่มีความละเอียดสูงจะทำให้ภาพผิววัตถุดูเสมือนจริงมากขึ้นแต่จะทำให้ใช้เวลาในการประมวลผล

แบบจำลองที่มากขึ้น ในงานวิจัยนี้ผู้วิจัยเลือกใช้ไฟล์ภาพนิ่งในสกุล jpg เพื่อดึงข้อมูลภาพนิ่งไปใช้ในการกำหนดภาพพื้นหลังของระนาบ 3 มิติและกำหนดลักษณะพื้นผิวของชิ้นส่วนโครงสร้างอาคาร การเก็บไฟล์ภาพนิ่งไว้ภายนอกฐานข้อมูลเพื่อลดขนาดการจัดเก็บข้อมูลและจะช่วยให้ฐานข้อมูลสามารถทำงานได้รวดเร็วและมีประสิทธิภาพมากขึ้น

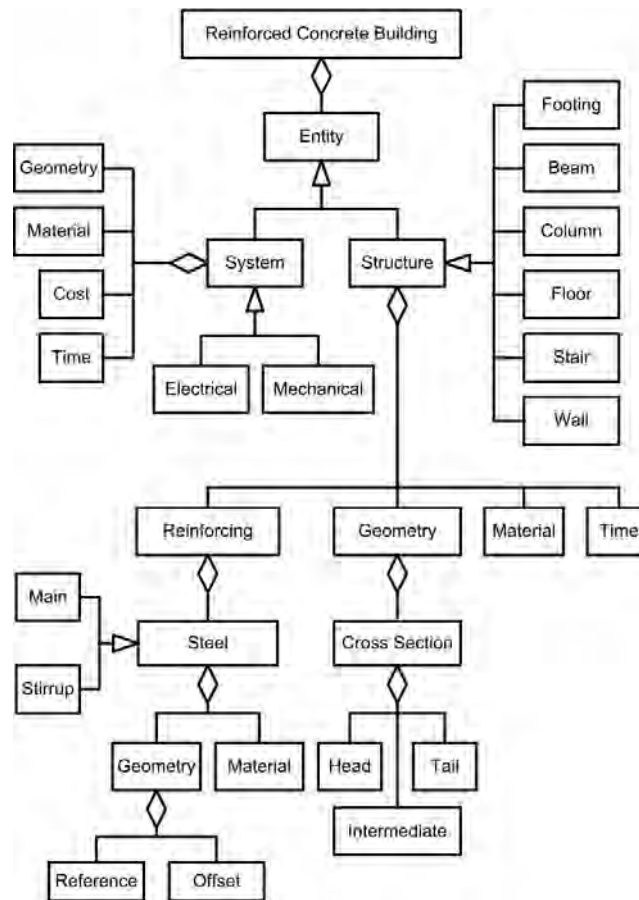
ในการเชื่อมต่อการสื่อสารระหว่างฐานข้อมูลในโปรแกรม PostgreSQL กับโปรแกรมประยุกต์ที่พัฒนาขึ้นโดยใช้ภาษาจาวาใช้การเชื่อมต่อผ่าน Java Database Connectivity (JDBC) ซึ่งเป็นวิธีการมาตรฐานในการเชื่อมต่อโปรแกรมประยุกต์ภาษาจาวากับโปรแกรมฐานข้อมูลทั่วไป

4.3 การออกแบบฐานข้อมูลแบบจำลองเชิงวัตถุ 3 มิติ

แบบจำลองเชิงตรรก (Logical Data Model) เป็นแผนภาพแสดงความสัมพันธ์ระหว่างเอนทิตี (Entity) หรือกลุ่มข้อมูลด้วยกันเพื่อแสดงให้เห็นความสัมพันธ์ของข้อมูลแต่ละส่วน รูปที่ 4.4 แสดง Unified Modeling Language (UML) Class Diagram ของแบบจำลองเชิงตรรกของฐานข้อมูลแบบจำลองเชิงวัตถุ 3 มิติสำหรับอาคารคอนกรีตเสริมเหล็ก จากรูปจะเห็นว่าอาคารคอนกรีตเสริมเหล็กประกอบด้วย Entity ย่อยๆ โดย Entity แต่ละหน่วยเป็นตัวแทนขององค์ประกอบอาคาร (Building Element) แต่ละชั้น Entity สามารถแบ่งได้ 2 ประเภทคือ Structure และ System โดย Structure เป็นตัวแทนขององค์ประกอบอาคารในส่วนงานโครงสร้าง อาทิเช่น ฐานราก คาน พื้น เสา ผนัง และบันได โดย Structure แต่ละหน่วยจะประกอบด้วยข้อมูล Geometry, Reinforcing, Material และ Time ซึ่งมีรายละเอียดการจัดเก็บข้อมูลดังต่อไปนี้

- Geometry เป็นข้อมูลเชิงเรขาคณิตแสดงรูปร่าง ตำแหน่ง และขนาดของ Structure แต่ละชั้น โดยข้อมูล Geometry ของ Structure แต่ละชั้นประกอบด้วยข้อมูลเชิงเรขาคณิตของหน้าตัด (Cross Section) ที่ส่วนหัวและส่วนท้ายของ Structure แต่ในกรณีที่ Structure มีหน้าตัดที่มีรูปร่างไม่ต่อเนื่องจะต้องจัดเก็บข้อมูลเชิงเรขาคณิตของหน้าตัดในส่วนของ Intermediate Section สำหรับแสดงการเปลี่ยนแปลงรูปร่างของหน้าตัด
- Reinforcing เป็นข้อมูลเชิงเรขาคณิตแสดงรายละเอียดการเสริมเหล็กของ Structure แต่ละหน่วย โดย Reinforcing สามารถแบ่งได้ 2 ประเภทคือ Main ซึ่งเป็นตัวแทนของเหล็กเสริมหลัก และ Stirrup ซึ่งเป็นตัวแทนของเหล็กปลอกรับแรงเฉือน โดย Reinforcing แต่ละหน่วยจะประกอบด้วยข้อมูล Geometry สำหรับจัดเก็บข้อมูลรูปร่างและตำแหน่งของเหล็กเสริมและ Material สำหรับจัดเก็บข้อมูลคุณลักษณะและราคาต่อหน่วยของเหล็กเสริม หนึ่งในการจัดเก็บข้อมูล Geometry ของเหล็กเสริม ไม่มีความจำเป็นต้องจัดเก็บข้อมูล Geometry ของเหล็กเสริมทุกเส้นซึ่งจะทำให้ฐานข้อมูลมีขนาดใหญ่มาก ดังนั้นเพื่อเป็นการลดขนาดของฐานข้อมูลจึงควรจัดเก็บ Geometry ของเหล็กเสริมเส้น Reference

สำหรับอ้างอิงเพียงเส้นเดียวพร้อมกับข้อมูล Offset ซึ่งเป็นการระบุจำนวน ระยะห่าง และ ทิศทางของเหล็กเส้นอื่นๆ ที่อยู่ในตำแหน่งคู่ขนานกับเหล็กเส้นเสริมเส้น Reference



รูปที่ 4.4 แบบจำลองเชิงตรรก (Logical Data Model) ของอาคารคอนกรีตเสริมเหล็ก

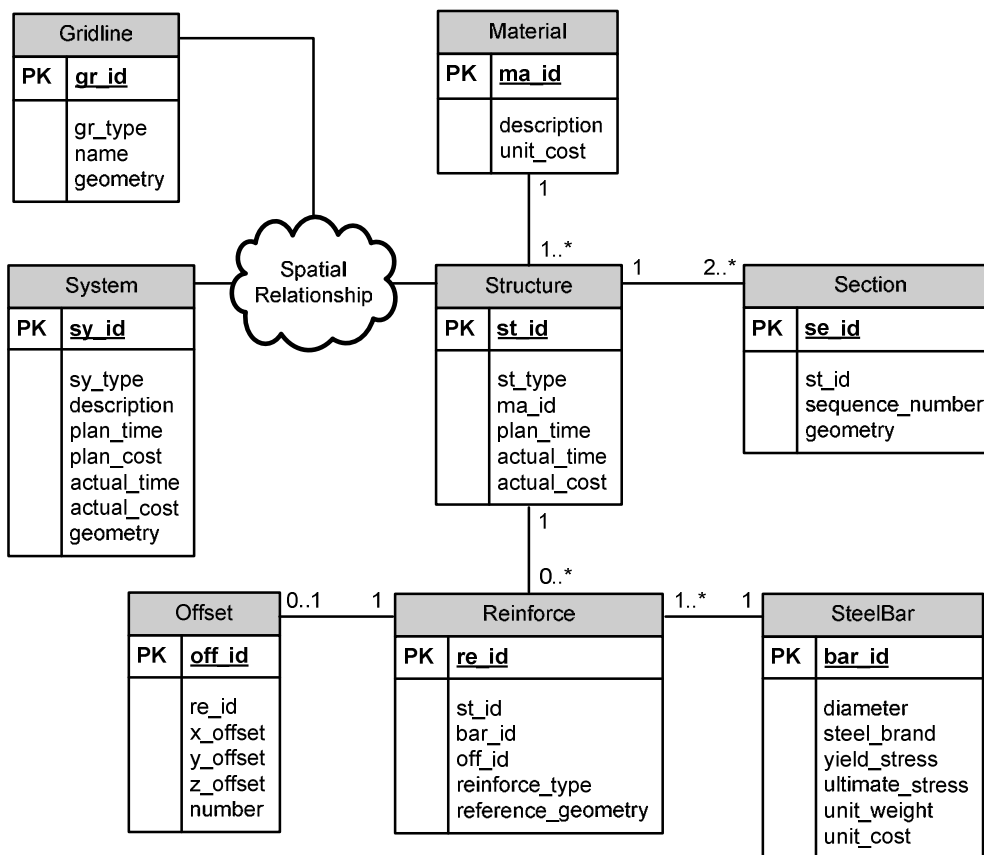
- Material เป็นข้อมูลคุณลักษณะของ Structure ซึ่งหมายถึงข้อมูลประเภทและความแข็งแรงของคอนกรีตที่ใช้ รวมถึงราคาต่อหน่วยและรายละเอียดอื่นๆ ของคอนกรีต อาทิ เช่น Slump Test เพื่อใช้ในการควบคุมและตรวจสอบคุณภาพงานก่อสร้าง เป็นต้น
- Time เป็นข้อมูลเชิงเวลาที่เกี่ยวข้องกับ Structure นั้นๆ อาทิเช่น กำหนดเวลาก่อสร้างของ Structure นั้นๆ หรือเวลาลอยตัวรวม (Total Float) เวลาลอยตัวอิสระ (Free Float) และเส้นตาย (Deadline) ในการดำเนินการดำเนินงานให้เสร็จสิ้น เป็นต้น

ในส่วนของ System ซึ่งเป็นตัวแทนขององค์ประกอบอาคารในส่วนงานระบบสามารถแบ่งได้ 2 ประเภทคือ Electrical ซึ่งเป็นตัวแทนของระบบไฟฟ้าและ Mechanical ซึ่งเป็นตัวแทนของระบบเครื่องกลและระบบท่อ โดย System มีองค์ประกอบ 4 ส่วนคือ Geometry, Material, Cost และ Time โดย Geometry เป็นตัวแทนของข้อมูลเชิงเรขาคณิตแสดงรูปร่าง ตำแหน่ง และขนาดของ System ส่วน Material, Cost และ Time แสดงข้อมูลเชิงคุณลักษณะ ข้อมูลเชิงราคา และข้อมูลเชิงเวลาของ System ตามลำดับ

แบบจำลองเชิงสัมพันธ์ (Physical Data Model) เป็นการออกแบบระบบฐานข้อมูลโดยละเอียด แบบจำลองเชิงสัมพันธ์นั้นเปรียบเสมือนพิมพ์เขียวของระบบฐานข้อมูลที่ใช้อ้างอิงในการพัฒนาระบบฐานข้อมูล โดยมีการวางแผนการจัดเก็บข้อมูลให้สามารถตอบสนองความต้องการของระบบได้อย่างครบถ้วน แบบจำลองเชิงสัมพันธ์ที่ดีจะช่วยป้องกันการเก็บข้อมูลซ้ำซ้อนทำให้ฐานข้อมูลมีขนาดเล็ก ทั้งยังช่วยเพิ่มความรวดเร็วในการค้นหาข้อมูลและปรับเปลี่ยนข้อมูลให้เป็นปัจจุบัน ทั้งนี้จะทำให้ระบบโดยรวมสามารถทำงานได้รวดเร็วและมีประสิทธิภาพสูงสุด รวมถึงต้องมีการตรวจสอบ Normalization เพื่อวิเคราะห์ความสัมพันธ์ระหว่างตาราง ซึ่งเป็นขั้นตอนสำคัญในการออกแบบระบบฐานข้อมูล

รูปที่ 4.5 แสดง UML Class Diagram ของแบบจำลองเชิงสัมพันธ์สำหรับฐานข้อมูลแบบจำลองเชิงวัตถุ 3 มิติของอาคารคอนกรีตเสริมเหล็กต้นแบบ โดยในการจัดทำฐานข้อมูลต้นแบบผู้วิจัยพบข้อสังเกตดังต่อไปนี้

- จากรูปที่ 4.5 พบว่าระหว่างตาราง Structure และ System ไม่มีความสัมพันธ์ที่ชัดเจนกันโดยตรง (Lack of Explicit Relationship) ดังนั้นในการหาความสัมพันธ์ระหว่างข้อมูลในตาราง Structure และ System ต้องทำการวิเคราะห์ผ่านความสัมพันธ์เชิงพื้นที่ (Spatial Relationship) ของข้อมูลเชิงเรขาคณิต
- เนื่องจากรูปทรงเรขาคณิตของชิ้นส่วน System มีความหลากหลายทั้งด้านรูปทรง ขนาด และมิติ ดังนั้นควรเลือกใช้ประเภทข้อมูลแบบ Geometry Collection สำหรับจัดเก็บข้อมูลเชิงเรขาคณิตของตาราง System เพื่อเพิ่มความยืดหยุ่นในการจัดเก็บข้อมูล
- โปรแกรม PostGIS สามารถรองรับการจัดเก็บข้อมูลรายละเอียดการเสริมเหล็ก เช่น ระยะทางและระยะงอในการเสริมเหล็กได้อย่างถูกต้อง ดังนั้นการคำนวณปริมาณเหล็กเสริมที่ใช้จะสามารถทำได้อย่างถูกต้อง
- การเลือกใช้ประเภทข้อมูลแบบ MultiPolygon สำหรับจัดเก็บข้อมูลเชิงเรขาคณิตของตาราง Section จะช่วยให้สามารถรองรับการจัดเก็บชิ้นส่วนโครงสร้างที่มีรูปร่างเป็นโพรงกลวง (Hollow) หรือรูปร่างแปลกๆ (Irregular Shape) ได้
- ราคาของชิ้นส่วนด้านโครงสร้าง (Structure) จะสามารถประเมินได้จากปริมาณวัสดุที่ใช้คูณกับราคาต่อหน่วยของวัสดุ ส่วนราคาของชิ้นส่วนด้านงานระบบควรให้ผู้ใช้อ้อนข้อมูลเข้ามาโดยตรงเพื่อลดความซับซ้อนในจัดเก็บและติดตามงบประมาณงานระบบ
- ตาราง Gridline ช่วยให้เราสามารถแปลงข้อมูลตำแหน่งในระนาบของระบบพิกัดฉาก (Cartesian coordinate system) ซึ่งคอมพิวเตอร์ใช้ในการประมวลผลกับระนาบกิดอ้างอิงของโครงการ (Construction Gridlines) ซึ่งคนสามารถเข้าใจได้เพื่อใช้ในการสื่อสารภายในโครงการ
- ควรสร้างดัชนีข้อมูลเชิงพื้นที่ (Spatial Index) แบบ Generalized Search Trees (GiST) สำหรับข้อมูลเชิงเรขาคณิตทั้งหมด เพื่อช่วยลดระยะเวลาการค้นหา (Computational Search Time) และการวิเคราะห์ข้อมูลเชิงพื้นที่ (Spatial Analysis) ของแบบจำลองเชิงวัตถุ 3 มิติ



รูปที่ 4.5 แบบจำลองเชิงสัมพันธ์ (Physical Data Model) ของอาคารคอนกรีตเสริมเหล็ก

พจนานุกรมฐานข้อมูล

ชื่อตาราง : Structure

คำอธิบาย : จัดเก็บข้อมูลคุณลักษณะและข้อมูลเชิงเวลาของชิ้นส่วนโครงสร้าง

ชื่อฟิลด์	ความหมาย	ประเภทข้อมูล	เพิ่มเติม
st_id	รหัสชิ้นส่วนโครงสร้าง	Serial	AUTO
st_type	ประเภทชิ้นส่วนโครงสร้าง	Character varying (150)	
ma_id	รหัสวัสดุ	Integer	
plan_time	แผนเวลาก่อสร้าง	Date	
actual_time	เวลาก่อสร้างจริง	Date	
actual cost	งบประมาณที่ใช้จริง	Double precision	

ตาราง : Section

คำอธิบาย : จัดเก็บข้อมูลเชิงเรขาคณิตของหน้าตัดของชิ้นส่วนโครงสร้าง

ชื่อฟิลด์	ความหมาย	ประเภทข้อมูล	เพิ่มเติม
se_id	รหัสหน้าตัด	Serial	AUTO
st_id	รหัสชิ้นส่วนโครงสร้าง	Integer	
sequence_number	ลำดับที่ของหน้าตัด	Integer	
geometry	ข้อมูลเชิงเรขาคณิต	Geometry (Multi Polygon)	

ชื่อตาราง : Reinforce

คำอธิบาย : จัดเก็บข้อมูลเชิงเรขาคณิตของเหล็กเสริมเส้นอ้างอิง

ชื่อฟิลด์	ความหมาย	ประเภทข้อมูล	เพิ่มเติม
re_id	รหัสเหล็กเสริมอ้างอิง	Serial	AUTO
st_id	รหัสชิ้นส่วนโครงสร้าง	Integer	
bar_id	รหัสรายละเอียดเหล็กเสริม	Integer	
off_id	รหัสเหล็กเสริมคู่ขนาน	Integer	
reinforce_type	ประเภทเหล็กเสริม	Integer	
reference_geometry	ข้อมูลเชิงเรขาคณิต	Geometry (Multi Line)	

ชื่อตาราง : Offset

คำอธิบาย : จัดเก็บข้อมูลรายละเอียดของเหล็กเสริมคู่ขนาน

ชื่อฟิลด์	ความหมาย	ประเภทข้อมูล	เพิ่มเติม
off_id	รหัสเหล็กเสริมคู่ขนาน	Serial	AUTO
re_id	รหัสเหล็กเสริมอ้างอิง	Integer	
x_offset	ระยะ offset ในแกน x	Double precision	
y_offset	ระยะ offset ในแกน y	Double precision	
z_offset	ระยะ offset ในแกน z	Double precision	
Number	จำนวนเหล็กเสริมคู่ขนาน	Integer	

ชื่อตาราง : SteelBar

คำอธิบาย : จัดเก็บข้อมูลคุณสมบัติของเหล็กเสริม

ชื่อฟิลด์	ความหมาย	ประเภทข้อมูล	เพิ่มเติม
bar_id	รหัสรายละเอียดเหล็กเสริม	Serial	AUTO
diameter	ขนาดเส้นผ่าศูนย์กลาง	Integer	
steel_brand	ยี่ห้อเหล็กเสริม	Character varying (150)	
yield_stress	ค่าความแข็งแรงที่จุด yield	Double precision	
ultimate_stress	ค่าความแข็งแรงที่จุด ultimate	Double precision	
unit_weight	น้ำหนักต่อหน่วย	Double precision	
unit_cost	ราคาต่อหน่วย	Double precision	

ชื่อตาราง : Material

คำอธิบาย : จัดเก็บข้อมูลคุณลักษณะของวัสดุก่อสร้าง

ชื่อฟิลด์	ความหมาย	ประเภทข้อมูล	เพิ่มเติม
ma_id	รหัสวัสดุ	Serial	AUTO
Description	รายละเอียดวัสดุ	Text	
unit_cost	ราคาต่อหน่วยของวัสดุ	Double precision	

ชื่อตาราง : System

คำอธิบาย : จัดเก็บข้อมูลงานระบบ

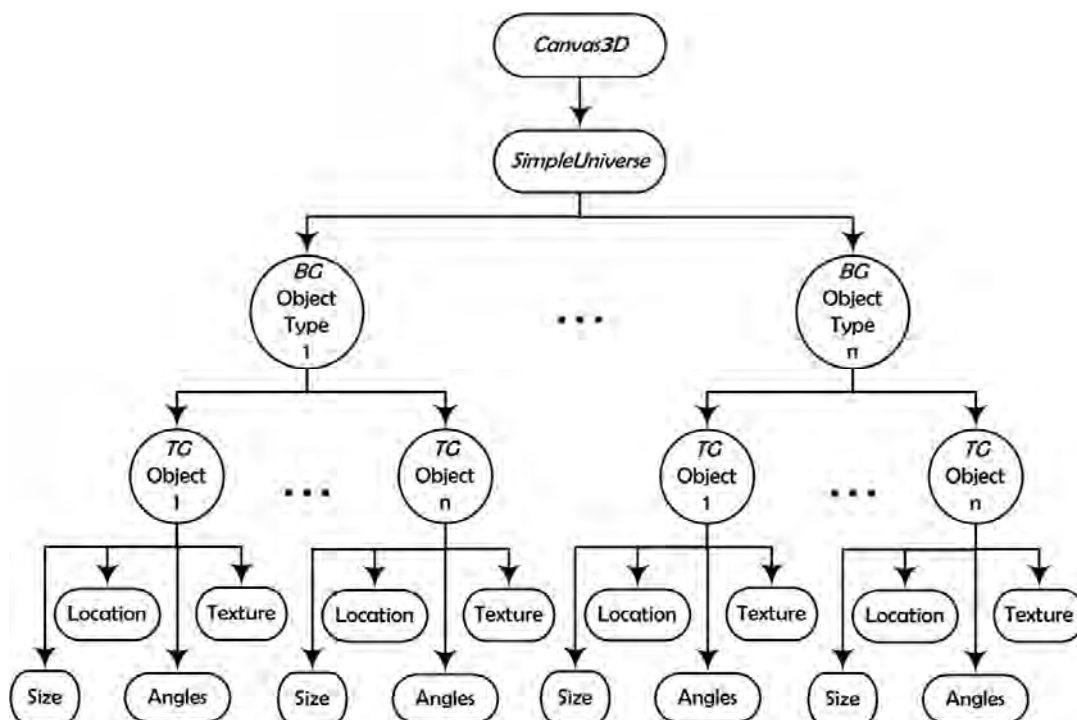
ชื่อฟิลด์	ความหมาย	ประเภทข้อมูล	เพิ่มเติม
sy_id	รหัสงานระบบ	Serial	AUTO
sy_type	ประเภทงานระบบ	Character varying (100)	
description	รายละเอียด	Character varying (250)	
plan_time	เวลาก่อสร้างตามแผน	Date	
plan_cost	ค่าใช้จ่ายตามแผน	Double precision	
actual_time	เวลาก่อสร้างจริง	Date	
actual_cost	ค่าใช้จ่ายจริง	Double precision	
geometry	ข้อมูลเชิงเรขาคณิต	Geometry (Geometry Collection)	

ชื่อตาราง : Gridline
 คำอธิบาย : จัดเก็บข้อมูล gridline อ้างอิงของโครงการ

ชื่อฟิลด์	ความหมาย	ประเภทข้อมูล	เพิ่มเติม
gr_id	รหัส gridline	Serial	AUTO
gr_type	ประเภท gridline	Character varying (150)	
Name	ชื่อ gridline	Character varying (150)	
geometry	ข้อมูลเชิงเรขาคณิต	Geometry (Line)	

4.4 การพัฒนาต้นแบบระบบสารสนเทศแบบจำลองเชิงวัตถุ 3 มิติ

รูปที่ 4.6 แสดงขั้นตอนการสร้างแบบจำลองเชิงวัตถุ 3 มิติโดยภาษาจาวา โดยเริ่มจากการสร้าง Canvas3D ใน AWT package เพื่อเป็นอินเทอร์เฟซ (Interface) สำหรับการแสดงผลวัตถุ 3 มิติโดยภาษาจาวา หลังจากนั้น SimpleUniverse ใน java utility package จะถูกเรียกใช้งานสำหรับสร้างระนาบแสดงผล 3 มิติผ่าน Scene Graph โดย SimpleUniverse จะถูกบรรจุอยู่ใน Canvas3D



รูปที่ 4.6 ขั้นตอนการสร้างแบบจำลองเชิงวัตถุ 3 มิติโดยภาษาจาวา

BranchGroup (BG) เป็นโหนดตั้งต้น (Root) ของ Scene Graph ซึ่งอยู่ภายใต้ SimpleUniverse ที่ถูกสร้างขึ้นในเบื้องต้น ผู้วิจัยสร้างแบบจำลองต้นแบบโดยใช้ Scene Graph 1 ชุดเป็นตัวแทนขององค์ประกอบอาคาร (Building Element) 1 ประเภทเพื่อแบ่งข้อมูลแบบจำลองเชิงวัตถุ 3 มิติออกเป็น

หลายชั้นข้อมูลตามประเภทองค์ประกอบของอาคาร ตัวอย่างเช่น แบบจำลองเชิงวัตถุของอาคาร 1 หลัง อาจประกอบด้วย Scene Graph 10 ชุด โดยแยกตามประเภทองค์ประกอบอาคาร อาทิเช่น เสาเข็ม ฐานราก คาน เสา พื้น ผนัง หลังคา ประตู หน้าต่าง และเหล็กเสริม เป็นต้น การแยก Scene Graph ตามประเภทองค์ประกอบอาคารจะเพิ่มความยืดหยุ่นในการเลือกแสดงผลข้อมูลองค์ประกอบอาคารแต่ละประเภทโดยการเปิดหรือปิด Scene Graph แต่ละชุด ในส่วนของ TransformGroup (TG) จะเป็นการควบคุมตำแหน่ง ขนาด มุมหมุน และภาพผิววัสดุขององค์ประกอบอาคารแต่ละหน่วย ตัวอย่างเช่น ถ้าอาคารหลังหนึ่งมีคานทั้งหมด 20 ตัว ดังนั้นจะต้องมี TransformGroup (TG) ทั้งหมด 20 หน่วยสำหรับควบคุมตำแหน่ง ขนาด มุมหมุน และภาพผิววัสดุของคานแต่ละตัว ในขณะที่ TransformGroup (TG) ทั้ง 20 หน่วยจะถูกอ้างอิงไปยัง BranchGroup (BG) อันเดียวกันเพื่อใช้ในการเลือกเปิดหรือปิดชั้นข้อมูลของคานในการแสดงผล

ก่อนดำเนินการพัฒนาต้นแบบระบบ (Prototype System) ด้วยภาษาจาวาสำหรับประมวลผลชุดคำสั่งสำหรับการสร้างและแสดงผลแบบจำลองเชิงวัตถุ 3 มิติ ผู้วิจัยได้ทำการติดตั้งโปรแกรมเพื่อสนับสนุนการพัฒนาต้นแบบระบบสารสนเทศแบบจำลองเชิงวัตถุ 3 มิติดังต่อไปนี้

- โปรแกรม Java Development Kit (JDK) รุ่น jdk-1_5_0_08-windows-i586-p เป็นชุดโปรแกรมพื้นฐานสำหรับการทำงานของภาษาจาวา
- โปรแกรม Java3D API รุ่น java3d-1_3_1-windows-i586-opengl-sdk เป็นโปรแกรมเสริมที่ช่วยให้สามารถสร้างและนำเสนอวัตถุเชิง 3 มิติ
- โปรแกรม PostgreSQL รุ่น 8.4 สำหรับบริหารจัดการฐานข้อมูลแบบจำลองอาคารเชิงวัตถุ 3 มิติ
- โปรแกรม PostGIS รุ่น 1.5 สำหรับเพิ่มความสามารถของฐานข้อมูล PostgreSQL ให้สามารถจัดการข้อมูลเชิงเรขาคณิตของแบบจำลองอาคารเชิงวัตถุ 3 มิติได้

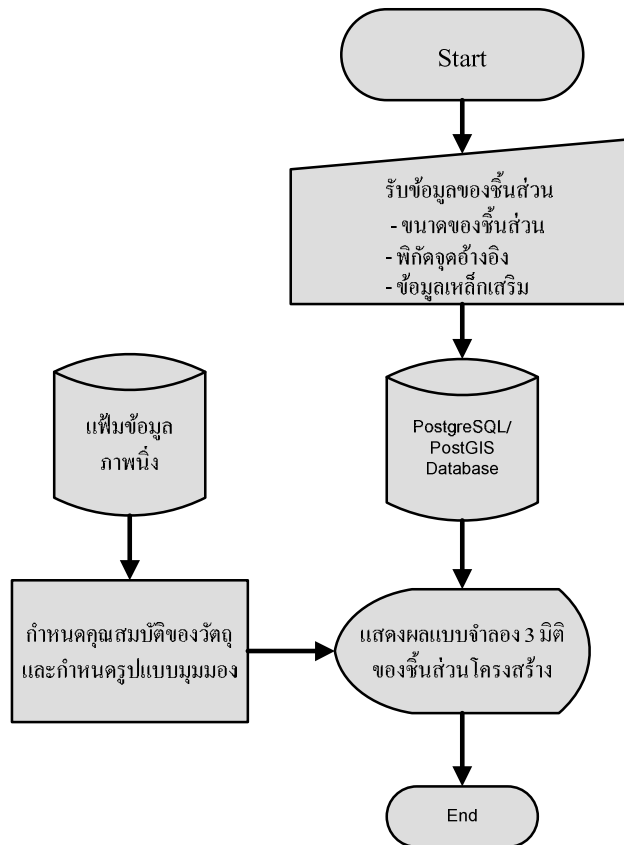
อนึ่งรายละเอียดขั้นตอนการติดตั้งโปรแกรม Java Development Kit (JDK), Java3D API, PostgreSQL และ PostGIS นำเสนอในภาคผนวก ก

ตารางที่ 4.7 สรุปรายละเอียด Class ของโปรแกรมภาษาจาวาที่ผู้วิจัยได้พัฒนาขึ้นเพื่อใช้สนับสนุนการสร้างและแสดงผลแบบจำลองอาคารเชิงวัตถุ 3 มิติและการพัฒนาหน้าจอ (Graphic User Interface - GUI) ของต้นแบบระบบ รูปที่ 4.7 แสดงภาพหน้าจอ (GUI) ของต้นแบบระบบที่พัฒนาขึ้นโดยภาษาจาวาเพื่อใช้สำหรับสร้างชิ้นส่วนอาคาร กำหนดคุณสมบัติของชิ้นส่วน และติดต่อสื่อสารกับผู้ใช้ในการสร้างและควบคุมการแสดงผลแบบจำลองอาคารเชิงวัตถุ 3 มิติ อนึ่งต้นแบบระบบที่ผู้วิจัยได้พัฒนาขึ้นมีจุดมุ่งหมายเพื่อแสดงความสามารถเชิงเทคนิคของภาษาจาวาในการสร้างและแสดงผลแบบจำลองอาคารเชิงวัตถุ 3 มิติ ไม่ได้มีจุดมุ่งหมายเพื่อนำไปใช้จริงในภาคอุตสาหกรรมเนื่องจากต้นแบบระบบที่ได้พัฒนาขึ้นยังขาดซึ่งความสะดวกในการใช้งานและฟังก์ชันการทำงานต่างๆ ที่จำเป็นสำหรับอำนวยความสะดวกแก่ผู้ใช้ อย่างไรก็ตามก็ต้องการความรู้ที่ได้จากการพัฒนาต้นแบบระบบในงานวิจัยนี้

สามารถนำไปใช้ในการพัฒนาต่อยอดเพื่อให้ภาคอุตสาหกรรมสามารถนำระบบไปใช้งานได้จริงในอนาคต

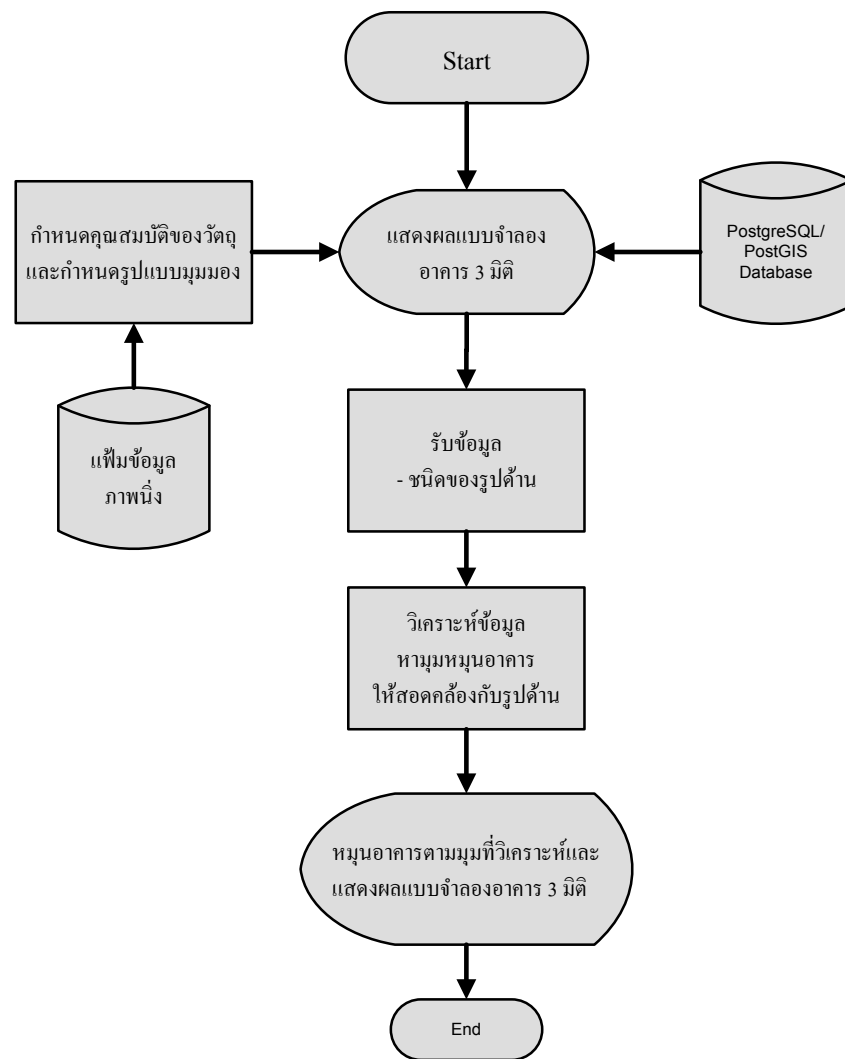
ตารางที่ 4.7 สรุปรายละเอียด Class ของโปรแกรมภาษาจาวา

Class name	Description
Cm3D	เป็น Class หลักสำหรับใช้เรียกใช้งานโปรแกรม ประกอบด้วย Method สำหรับติดต่อกับฐานข้อมูลและสร้างชิ้นส่วนอาคาร 3 มิติ
ClearDataBase	ลบข้อมูลทั้งหมดในฐานข้อมูล
Quit	ออกจากการทำงานของโปรแกรม
ButtonPositionControls	ควบคุมการทำงานของปุ่มสำหรับกำหนดตำแหน่งของอาคาร
PositionControls	กำหนดตำแหน่งของอาคาร
RotationControls	กำหนดทิศทางของอาคาร
SensorBehavior	คำนวณหาตำแหน่งปัจจุบันของอาคาร
IntEvent	กำหนดสีสำหรับใช้ในโปรแกรม
IntChooser	สร้าง Interface สำหรับกำหนดสี
OffScreenCanvas3D	กำหนดขนาดของระนาบ 3 มิติ
ColoringAttributesEditor	สร้าง Interface สำหรับกำหนดสีของชิ้นส่วน
Color3fEditor	สร้างการกำหนดสีของชิ้นส่วน
Color3fEvent	ควบคุมการกำหนดสีของชิ้นส่วน
PolygonAttributesEditor	สร้าง Interface สำหรับกำหนดการแสดงผลชิ้นส่วน
RenderingAttributesEditor	หน้าที่เปิดปิดการแสดงผลชิ้นส่วน
FloatLabelJSlider	สร้าง Interface ที่มีลักษณะเป็นแถบเลื่อน
FloatEvent	ควบคุมการทำงานของแถบเลื่อน
MaterialEditor	สร้าง Interface สำหรับกำหนดวัสดุของชิ้นส่วน
TransparencyAttributesEditor	สร้าง Interface สำหรับกำหนดความโปร่งแสงของชิ้นส่วน
Texture2DEditor	สร้าง Interface สำหรับกำหนด Mode ของวัสดุพื้นผิวของชิ้นส่วน
TextureAttributesEditor	สร้าง Interface สำหรับกำหนดวัสดุพื้นผิวของชิ้นส่วน
TexCoordGenerationEditor	สร้าง Interface สำหรับกำหนดความละเอียดของ



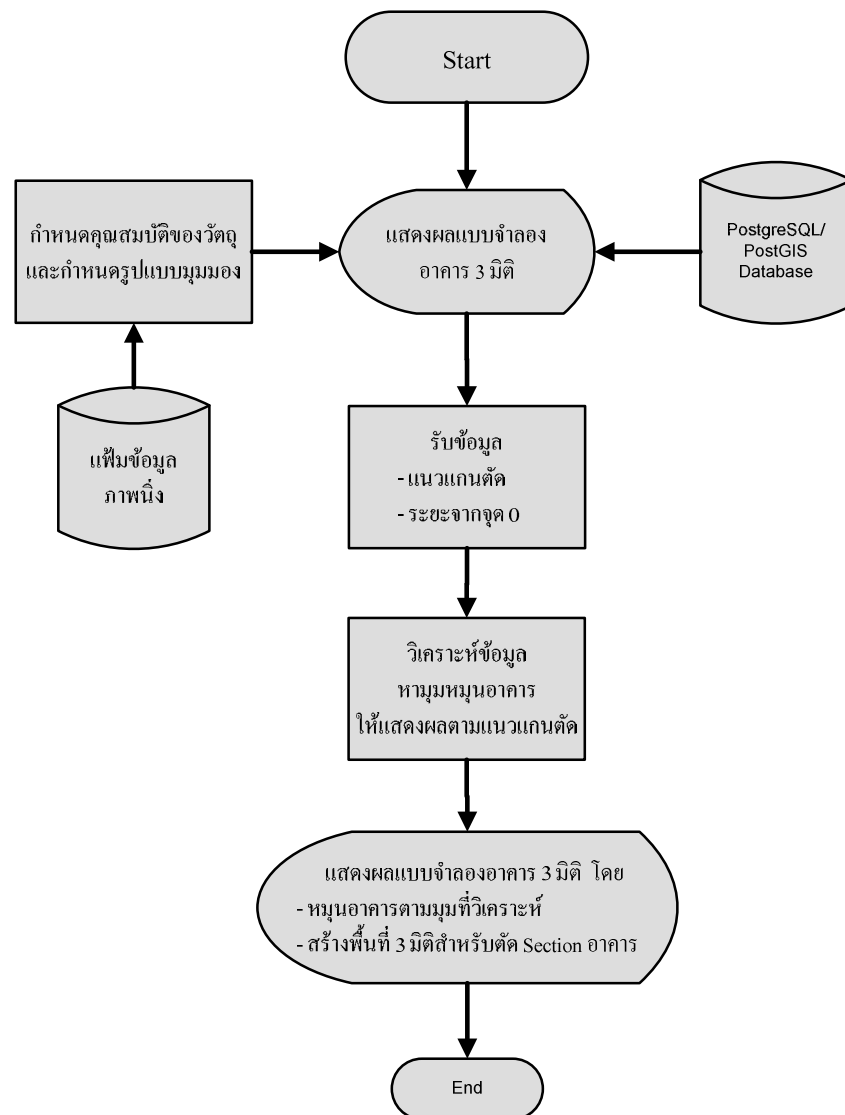
รูปที่ 4.8 ขั้นตอนการสร้างแบบจำลองอาคารเชิงวัตถุ 3 มิติ

ต้นแบบระบบที่ผู้วิจัยได้พัฒนาขึ้นมีความสามารถในการสร้างแบบรูป 2 มิติของรูปด้านหน้า รูปด้านหลัง รูปด้านซ้าย รูปด้านขวา และรูปด้านบนของอาคารได้โดยอัตโนมัติ รูปที่ 4.9 แสดงขั้นตอนการสร้างรูปด้านของอาคารโดยผู้ใช้งานกำหนดประเภทของรูปด้านที่ต้องการสร้าง หลังจากนั้นโปรแกรมจะทำการวิเคราะห์หามุมหมุนของอาคารที่สอดคล้องกับรูปด้านที่ผู้ใช้งานกำหนด แล้วทำการประมวลผลหมุนอาคารตามค่ามุมที่ได้ทำการวิเคราะห์ไว้ แล้วจึงแสดงผลรูปด้านของอาคารทางในระนาบ 3 มิติที่หน้าจอปฏิบัติการ



รูปที่ 4.9 ขั้นตอนการแสดงรูปด้านของอาคาร

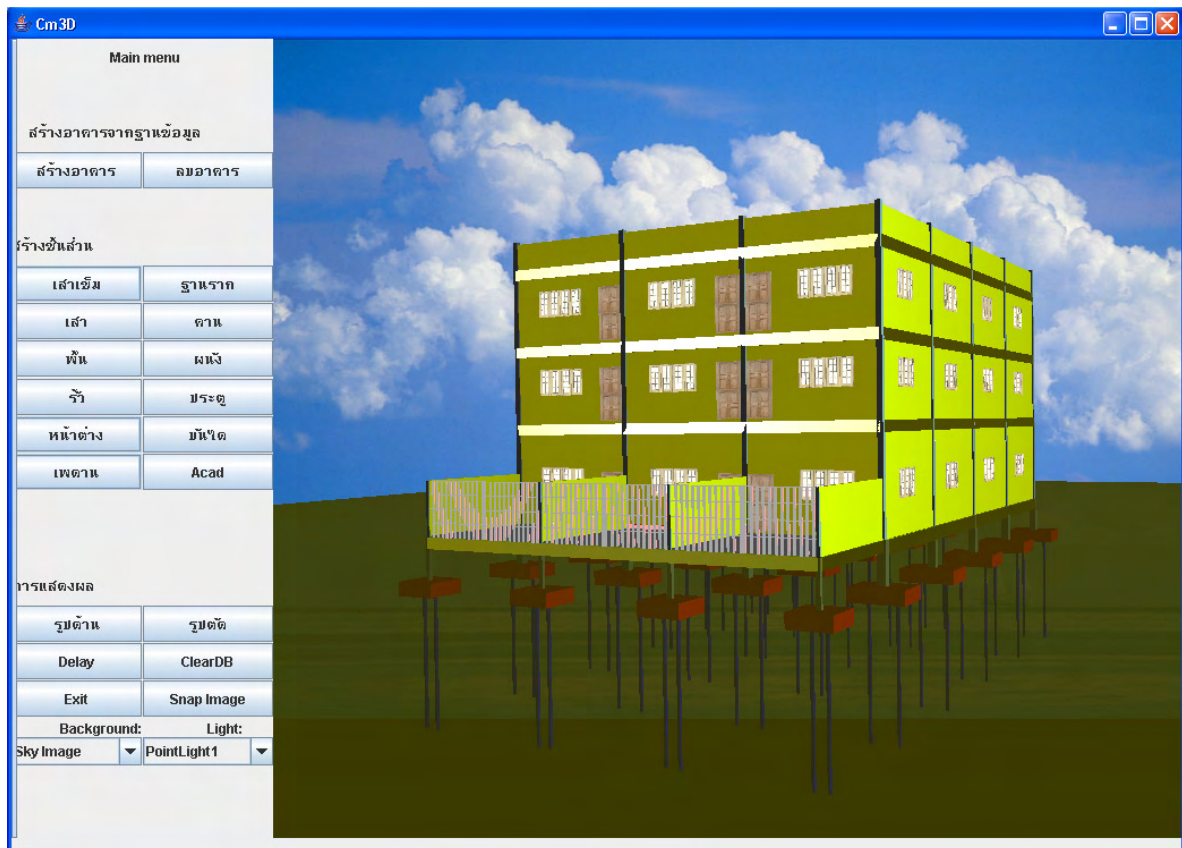
ต้นแบบระบบที่ผู้วิจัยได้พัฒนาขึ้นยังมีความสามารถในการสร้างรูปตัดตามแนวแกน X, Y และ Z ได้โดยอัตโนมัติ โดยมีขั้นตอนในการทำงานคล้ายกับการแสดงผลรูปด้านของอาคาร แต่แตกต่างกันตรงที่ต้องมีการสร้างพื้นที่ 3 มิติสำหรับตัดอาคารส่วนที่ไม่ต้องการแสดงผลออก รูปที่ 4.10 แสดงลำดับขั้นตอนในการสร้างรูปตัดทางขวางของอาคารโดยเมื่อเริ่มต้นการทำงานโปรแกรมจะดึงค่าข้อมูลต่างๆ จากฐานข้อมูลมาทำการประมวลผลเพื่อที่สร้างชิ้นส่วนอาคารและแสดงผลในระนาบ 3 มิติที่หน้าจอปฏิบัติการ จากนั้นโปรแกรมรับข้อมูลชนิดของรูปตัดทางขวางที่ผู้ใช้งานต้องการสร้างและระยะห่างของแนวตัดจากตำแหน่ง +0.00 ของแกนตัดจากผู้ใช้งาน แล้วโปรแกรมจะทำการวิเคราะห์หามุมหมุนของอาคารที่สอดคล้องกับชนิดของรูปตัดทางขวางที่ต้องการพิจารณา จากนั้นจะทำการประมวลผลหมุนอาคารตามค่ามุมที่ได้ทำการวิเคราะห์ไว้และสร้างพื้นที่ 3 มิติสำหรับตัดอาคารส่วนที่ไม่ต้องการแสดงผลออกตามระยะห่างของแนวตัดที่กำหนด แล้วจึงแสดงผลรูปตัดทางขวางของอาคารในระนาบ 3 มิติที่หน้าจอปฏิบัติการ



รูปที่ 4.10 ขั้นตอนการสร้างรูปตัดทางขวางของอาคาร

การสร้างรูปแปลนพื้นของอาคารมีขั้นตอนในการทำงานคล้ายกับการแสดงผลรูปตัดทางขวางของอาคาร แต่แตกต่างกันตรงที่การสร้างรูปแปลนพื้นของอาคารจะเป็นการสร้างรูปตัดตามแกน Y โดยผู้ใช้งานต้องกำหนดแนวให้อยู่ที่ระดับกึ่งกลางชั้นที่ต้องการพิจารณา หลังจากนั้นโปรแกรมจะประมวลผลแล้วจึงแสดงผลรูปแปลนพื้นของอาคารในระนาบ 3 มิติที่หน้าจอปฏิบัติการ

รูปที่ 4.11 แสดงตัวอย่างแบบจำลองอาคารเชิงวัตถุ 3 มิติของอาคารต้นแบบที่ผู้วิจัยสร้างขึ้น โดยอาคารดังกล่าวประกอบด้วยชิ้นส่วนของอาคาร ได้แก่ เสาเข็ม ฐานราก เสา คาน พื้น ผนัง รั้ว ประตู หน้าต่าง เพดานและบันได รายละเอียดการสร้างชิ้นส่วนของอาคารแต่ละชิ้นส่วนมีรายละเอียดดังต่อไปนี้



รูปที่ 4.11 แบบจำลอง 3 มิติของอาคารขนาด 3 ชั้น

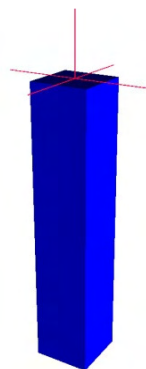
ในการสร้างชั้นส่วนของอาคารต้นแบบผู้ใช้จะต้องป้อนข้อมูล 2 ส่วนคือ ข้อมูลพิกัด x , y และ z ของจุดอ้างอิงและข้อมูลขนาดของชั้นส่วน โดยการป้อนข้อมูลลักษณะนี้จะช่วยลดความซับซ้อนในการป้อนข้อมูลของผู้ใช้ กล่าวคือ ผู้ใช้ไม่จำเป็นต้องป้อนข้อมูลพิกัด x , y และ z ของทุกจุดมุมของแต่ละชั้นส่วน (เช่น รูปทรงสี่เหลี่ยมมีจุดมุมทั้งหมด 8 จุด) ตารางที่ 4.8 สรุปข้อมูลที่ต้องป้อนผ่านหน้าจอปฏิบัติการในการสร้างชั้นส่วนแต่ละประเภท รูปที่ 4.12 ถึง 4.22 แสดงตำแหน่งอ้างอิงของชั้นส่วนอาคารแต่ละประเภท

ในการสร้างชั้นส่วนของอาคารเริ่มจากการรับข้อมูลรายละเอียดของชั้นส่วนจากผู้ใช้ผ่านหน้าจอปฏิบัติการแล้วนำข้อมูลที่ผู้ใช้ป้อนไปจัดเก็บไว้ในฐานข้อมูล หลังจากนั้นโปรแกรมจะดึงข้อมูลชั้นส่วนอาคารจากฐานข้อมูลมาทำการประมวลผลและแสดงผลในรูปแบบ 3 มิติ อาทิเช่น ในการสร้างชั้นส่วนเสาเข็มผู้ใช้ต้องทำการป้อนข้อมูลความกว้างและความยาวของหน้าตัดของเสาเข็ม และความยาวของเสาเข็ม ผ่านหน้าจอปฏิบัติการหลังจากนั้นโปรแกรมจะดึงข้อมูลของเสาเข็มจากฐานข้อมูลมาแสดงผลในรูปแบบ 3 มิติ

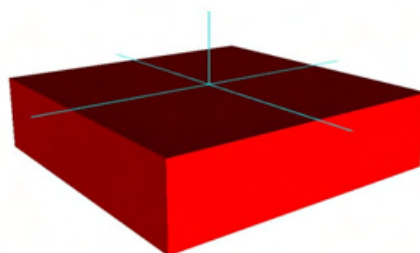
ตารางที่ 4.8 สรุปข้อมูลที่จำเป็นสำหรับการสร้างชั้นส่วน 3 มิติ

ชั้นส่วน	ข้อมูลที่จำเป็นในการสร้างชั้นส่วน
เสาเข็ม	- พิกัด (x,y,z) จุดอ้างอิงของเสาเข็ม - ความกว้างของหน้าตัดของเสาเข็ม - ความยาวของหน้าตัดของเสาเข็ม - ความยาวของเสาเข็ม
ฐานราก	- พิกัด (x,y,z) จุดอ้างอิงของฐานราก - ความกว้างของฐานราก - ความยาวของฐานราก - ความลึกของฐานราก
เสา	- พิกัด (x,y,z) จุดอ้างอิงของเสา - ความกว้างของหน้าตัดของเสา - ความยาวของหน้าตัดของเสา - ความยาวของเสา
คาน	- พิกัด (x,y,z) จุดอ้างอิงของคาน - ความกว้างของหน้าตัดคาน - ความลึกของหน้าตัดคาน - ความยาวของคาน
พื้น	- พิกัด (x,y,z) จุดอ้างอิงของพื้น - ความกว้างของพื้น - ความยาวของพื้น - ความหนาของพื้น
ผนัง	- พิกัด (x,y,z) จุดอ้างอิงของผนัง - ความกว้างของผนัง - ความยาวของผนัง - ความหนาของผนัง
รั้ว	- พิกัด (x,y,z) จุดอ้างอิงของรั้ว - ความกว้างของรั้ว - ความยาวของรั้ว - ความหนาของรั้ว
ประตู	- พิกัด (x,y,z) จุดอ้างอิงของประตู - ความกว้างของประตู

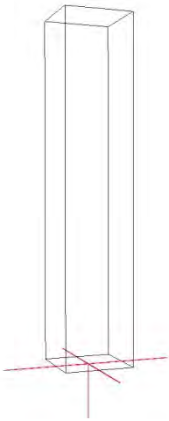
ชั้นส่วน	ข้อมูลที่จำเป็นในการสร้างชั้นส่วน
	- ความสูงของประตู - ความหนาของประตู
หน้าต่าง	- พิกัด (x,y,z) จุดอ้างอิงของหน้าต่าง - ความกว้างของหน้าต่าง - ความสูงของหน้าต่าง - ความหนาของหน้าต่าง
เพดาน	- พิกัด (x,y,z) จุดอ้างอิงของเพดาน - ความกว้างของเพดาน - ความยาวของเพดาน - ความหนาของเพดาน
บันได	- พิกัด (x,y,z) จุดอ้างอิงของบันได - ขนาดลูกตั้ง - ขนาดลูกนอน - จำนวนขั้นของบันได - ความกว้างของบันได - มุมหมุนของชั้นส่วนรอบแกน Y



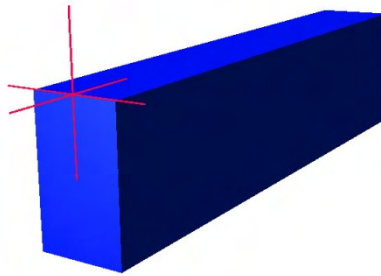
รูปที่ 4.12 ตำแหน่งจุดอ้างอิงของเสาเข็ม



รูปที่ 4.13 ตำแหน่งจุดอ้างอิงของฐานราก



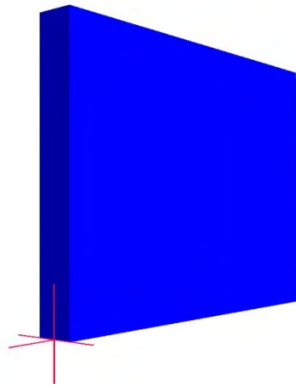
รูปที่ 4.14 ตำแหน่งจุดอ้างอิงของเสา



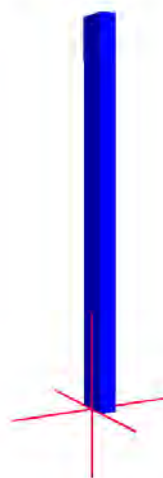
รูปที่ 4.15 ตำแหน่งจุดอ้างอิงของคาน



รูปที่ 4.16 ตำแหน่งจุดอ้างอิงของพื้น



รูปที่ 4.17 ตำแหน่งจุดอ้างอิงของผนัง



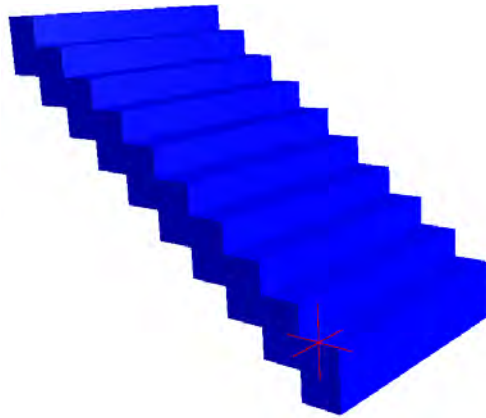
รูปที่ 4.18 ตำแหน่งจุดอ้างอิงของรื้อ



รูปที่ 4.19 ตำแหน่งจุดอ้างอิงของประตู



รูปที่ 4.20 ตำแหน่งจุดอ้างอิงของหน้าต่าง



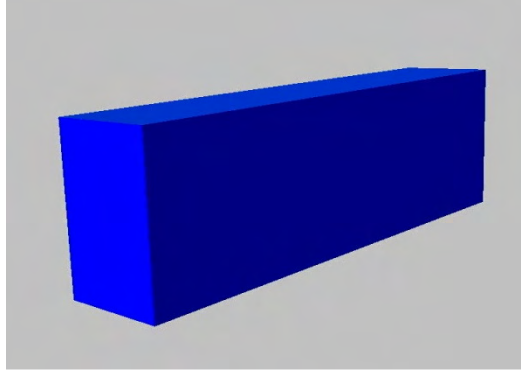
รูปที่ 4.21 ตำแหน่งจุดอ้างอิงของบันได



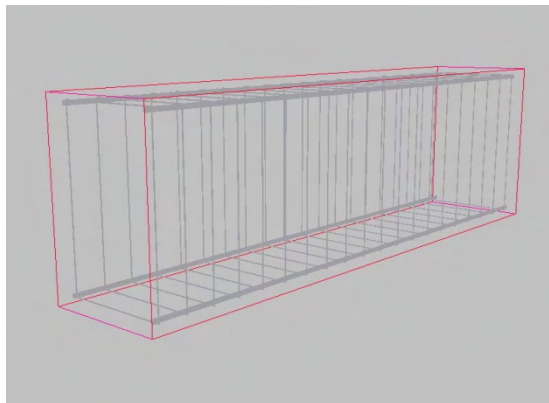
รูปที่ 4.22 ตำแหน่งจุดอ้างอิงของเพดาน

การกำหนดสภาพแวดล้อมของระนาบ 3 มิติ ทำได้โดยการรับข้อมูลภาพจากแฟ้มข้อมูลภาพหนึ่ง มากำหนดเป็นภาพพื้นหลังของระนาบ 3 มิติ เช่น การกำหนดเป็นภาพของท้องฟ้า ทั้งสามารถสร้างแบบจำลอง 3 มิติของพื้นดินได้ฐานรากแล้วกำหนดให้มีมุมมองแบบโปร่งแสงเพื่อให้สามารถมองเห็นลักษณะของชิ้นส่วนโครงสร้างใต้พื้นดิน

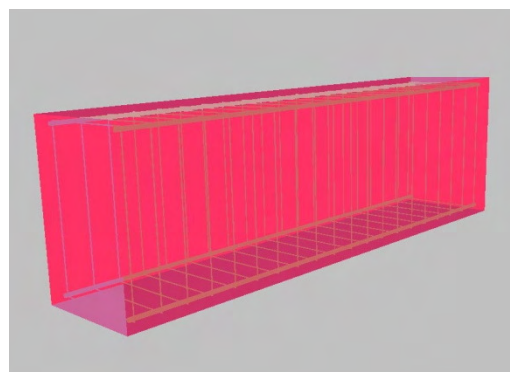
ต้นแบบระบบที่ผู้วิจัยพัฒนาขึ้นสนับสนุนการแสดงผลแบบจำลองอาคารเชิงวัตถุ 3 มิติได้ในหลายลักษณะมุมมอง อันประกอบด้วยมุมมองแบบแรเงา (Shading) มุมมองแบบเส้นโครง (Wireframe) และมุมมองแบบโปร่งแสง (Transparency) ดังแสดงในรูปที่ 4.23 ถึง 4.25



รูปที่ 4.23 ลักษณะมุมมองแบบแรเงา (Shading)

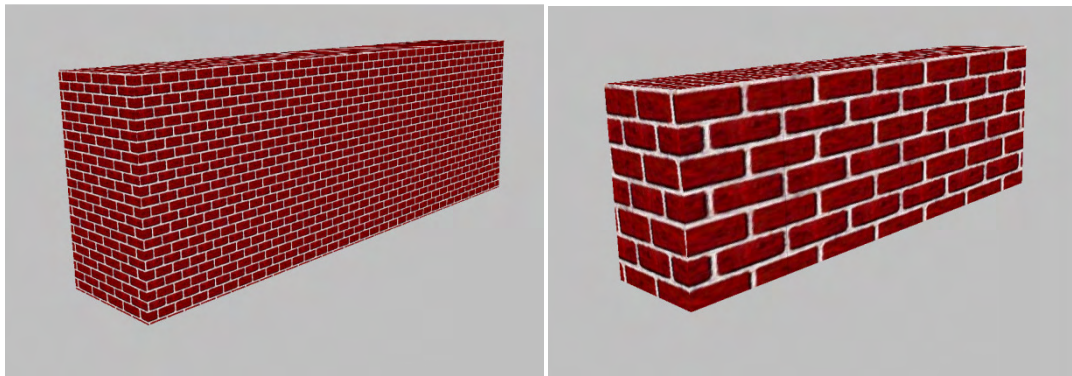


รูปที่ 4.24 ลักษณะมุมมองแบบเส้นโครง (Wireframe)



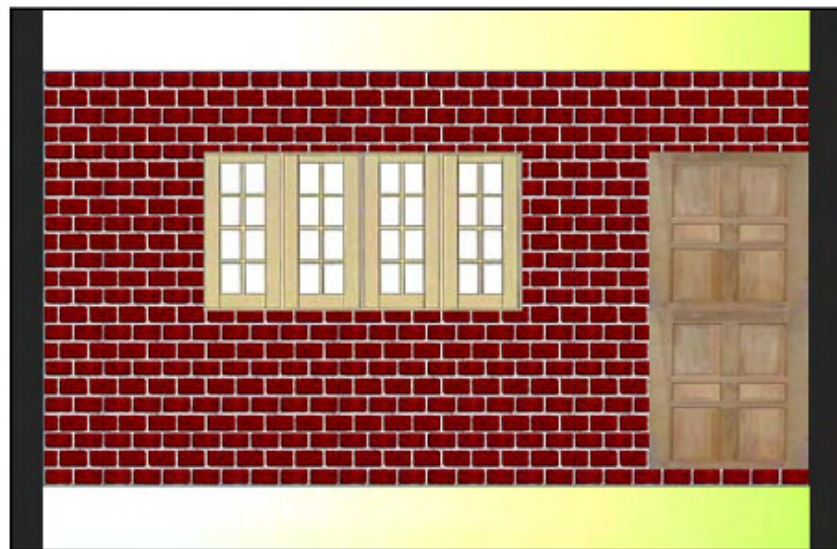
รูปที่ 4.25 ลักษณะมุมมองแบบโปร่งแสง (Transparency)

การกำหนดวัสดุพื้นผิวให้กับชิ้นส่วนเป็นการดึงภาพสกุล jpg จากแฟ้มข้อมูลภาพนิ่ง มาทำการวางทับ (Draping) ลงบนพื้นผิวของชิ้นส่วน 3 มิติที่สร้างขึ้น รูปที่ 4.26 แสดงลักษณะการกำหนดวัสดุพื้นผิวและมาตราส่วนของวัสดุพื้นผิวของชิ้นส่วน โดยรูปที่ 4.26 (a) แสดงการกำหนดมาตราส่วนของวัสดุพื้นผิวให้มีความละเอียดมาก รูปที่ 4.26 (b) แสดงการกำหนดมาตราส่วนของวัสดุพื้นผิวให้มีความละเอียดน้อย และรูปที่ 4.26 (c) แสดงแบบจำลอง 3 มิติของประตู หน้าต่าง ที่กำหนดลวดลายของบานประตูหน้าต่างบน ทำให้มีความสมจริงมากยิ่งขึ้น



(a)

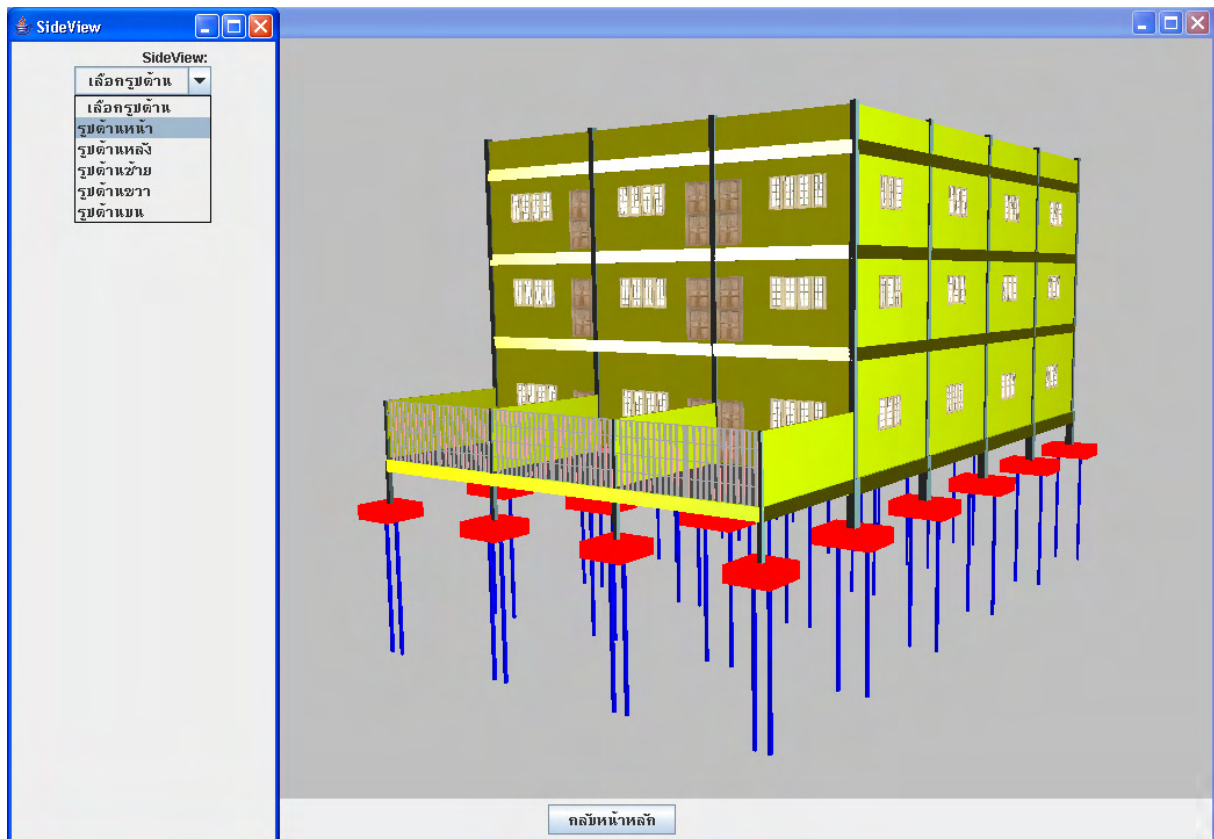
(b)



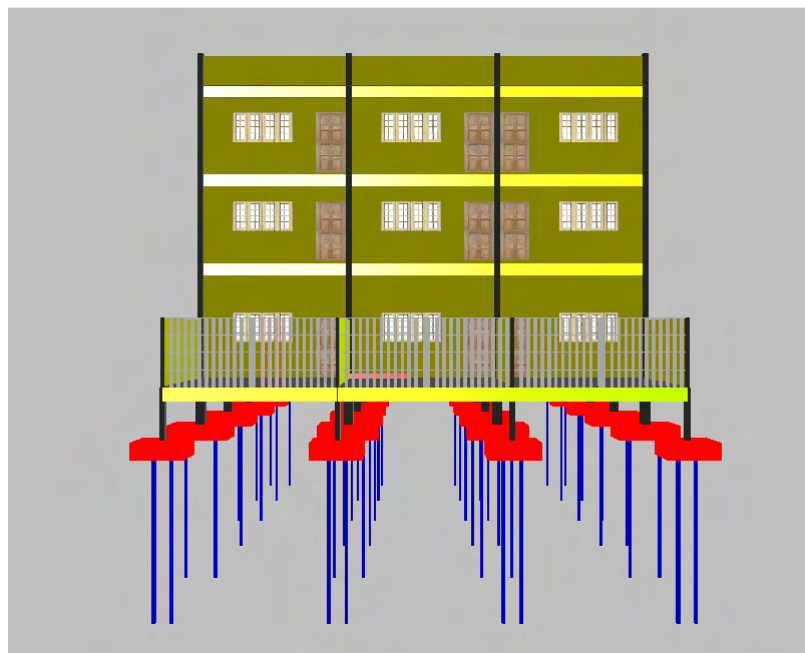
(c)

รูปที่ 4.26 การกำหนดวัสดุพื้นผิวและมาตราส่วนของวัสดุพื้นผิวของชิ้นส่วน

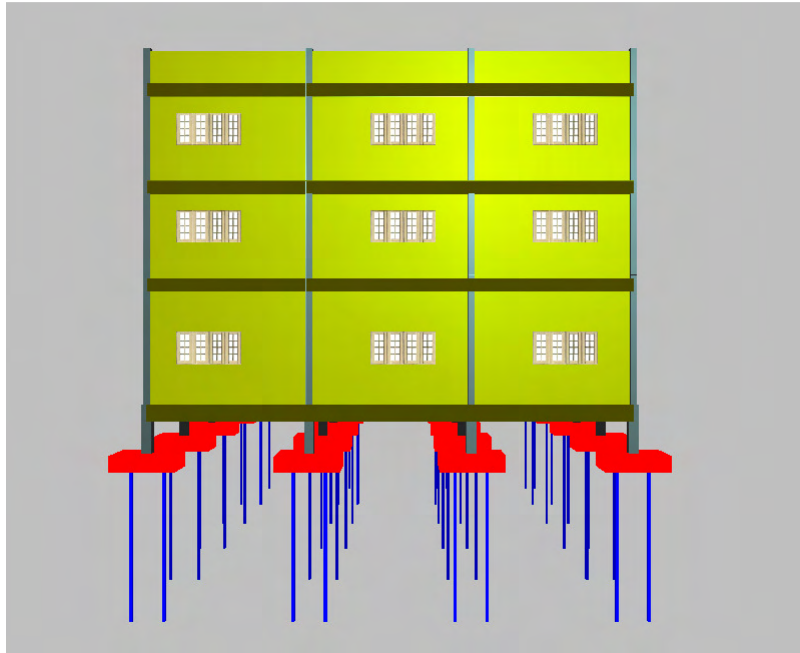
รูปที่ 4.27 ถึง 4.41 แสดงรูปด้าน รูปตัด และรูปแปลนของอาคารในมุมมองต่างๆ ที่สร้างขึ้นโดยอัตโนมัติจากแบบจำลองอาคารเชิงวัตถุ 3 มิติต้นแบบที่ผู้วิจัยได้จัดทำขึ้น หนึ่งใน การสร้างรูปตัดและรูปแปลนอาคาร ผู้ใช้งานสามารถเลือกแนวแกน (Section Plane) ในการสร้างรูปตัดและรูปแปลน รวมถึงสามารถกำหนดตำแหน่งของแนวตัดได้โดยการกำหนดค่าผ่านหน้าจอปฏิบัติการที่ผู้วิจัยได้พัฒนาขึ้น



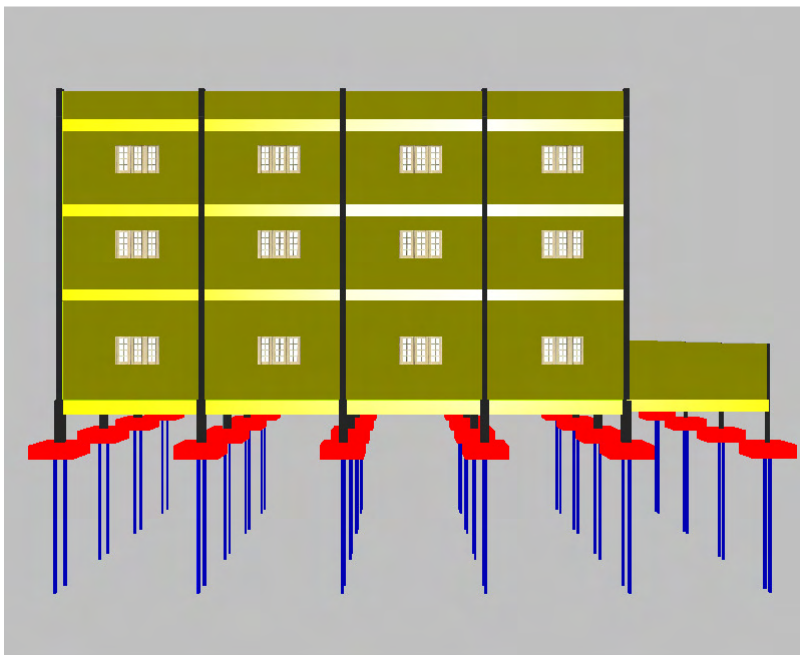
รูปที่ 4.27 หน้าต่างโปรแกรมประยุกต์สำหรับการกำหนดการแสดงผลรูปด้านของอาคาร



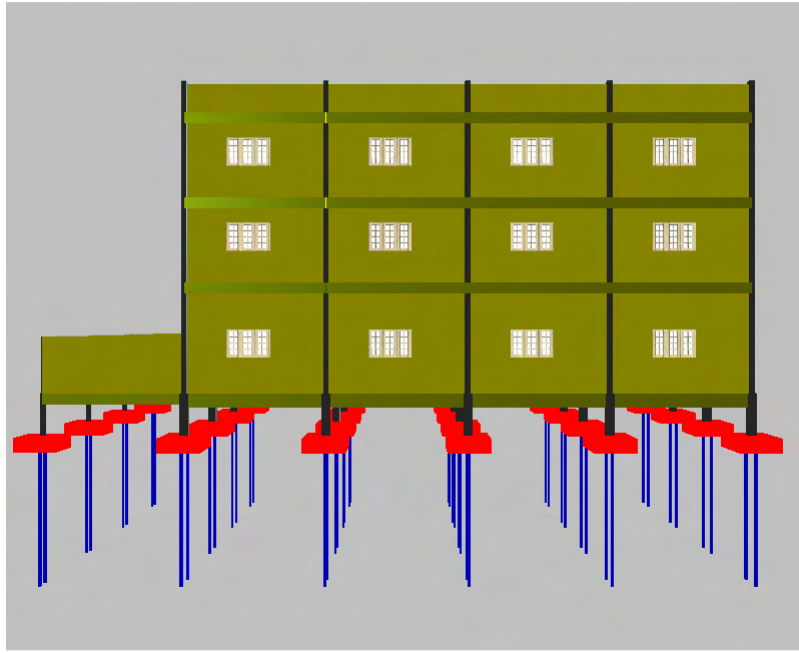
รูปที่ 4.28 รูปด้านหน้าของอาคาร



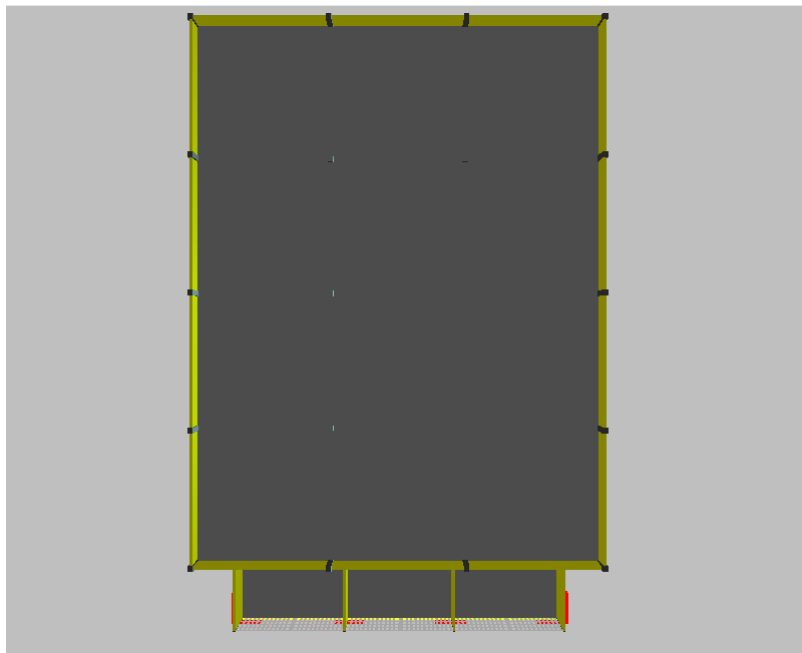
รูปที่ 4.29 รูปด้านหลังของอาคาร



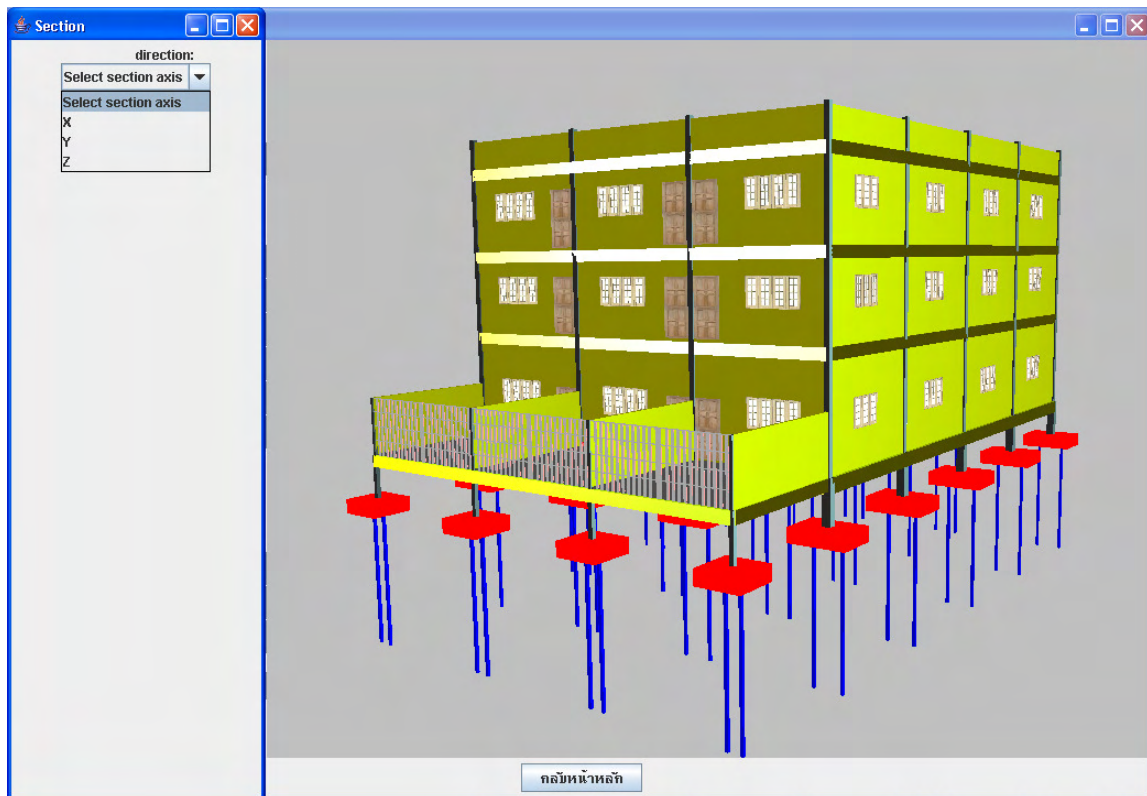
รูปที่ 4.30 รูปด้านซ้ายของอาคาร



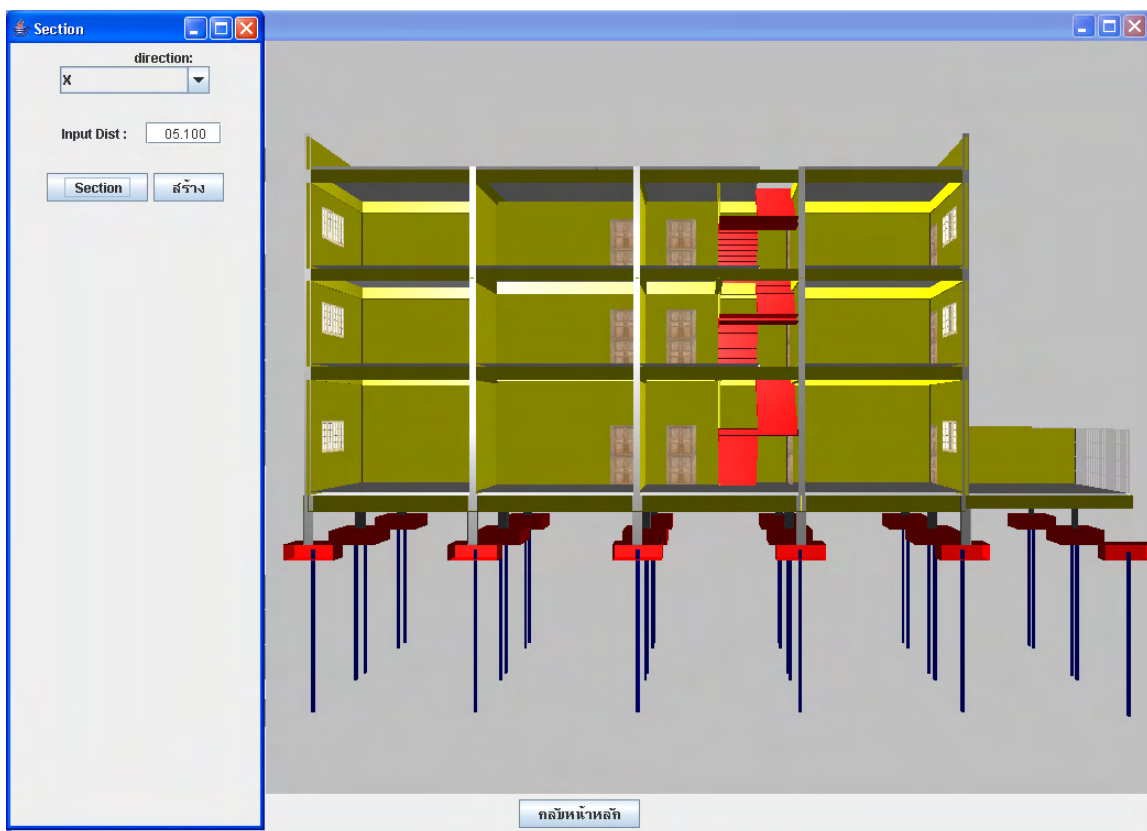
รูปที่ 4.31 รูปด้านขวาของอาคาร



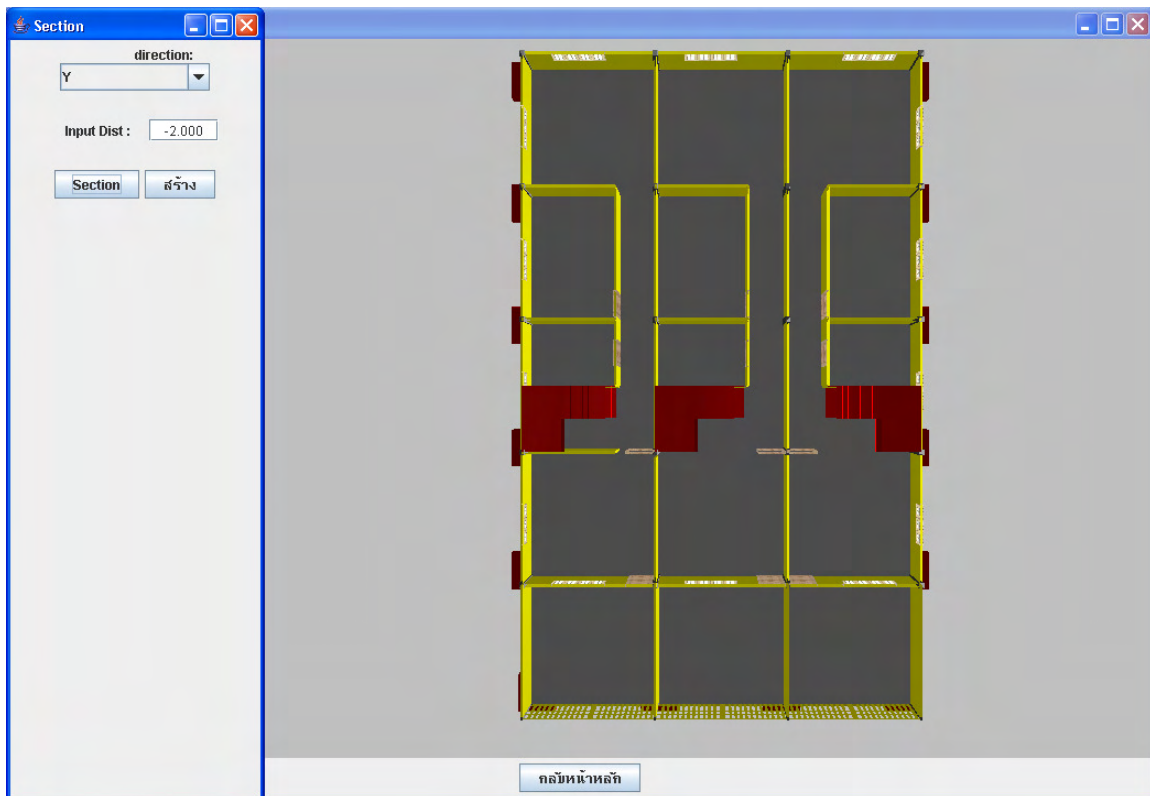
รูปที่ 4.32 รูปด้านบนของอาคาร



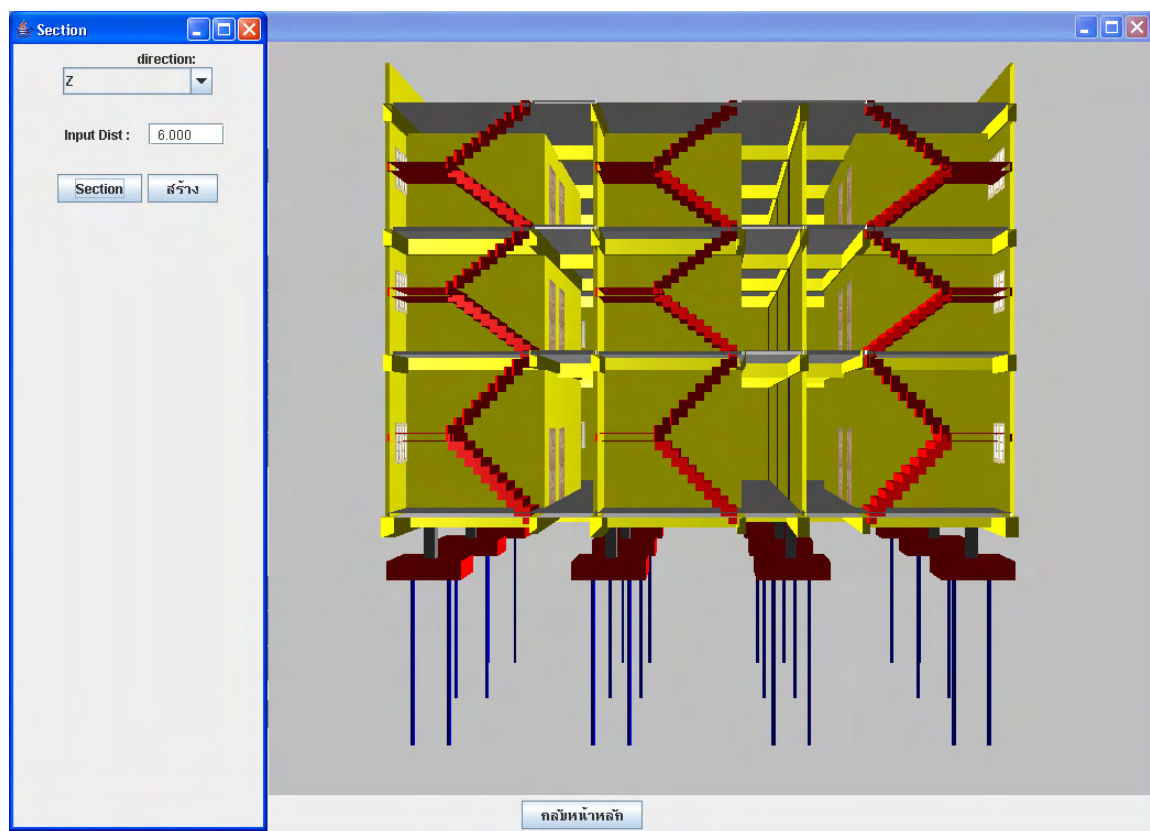
รูปที่ 4.33 หน้าต่างโปรแกรมประยุกต์สำหรับกำหนดการสร้างรูปตัดทางขวางของอาคาร



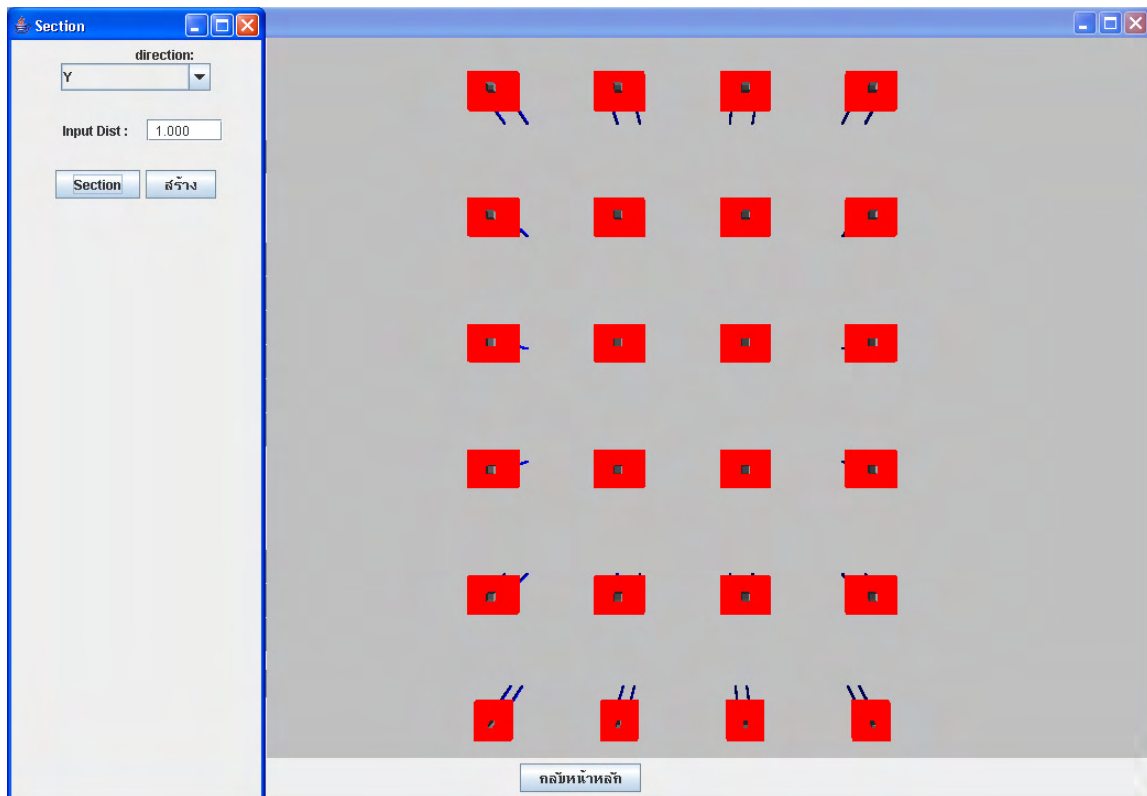
รูปที่ 4.34 รูปตัดทางขวางตามแกน X ของอาคาร



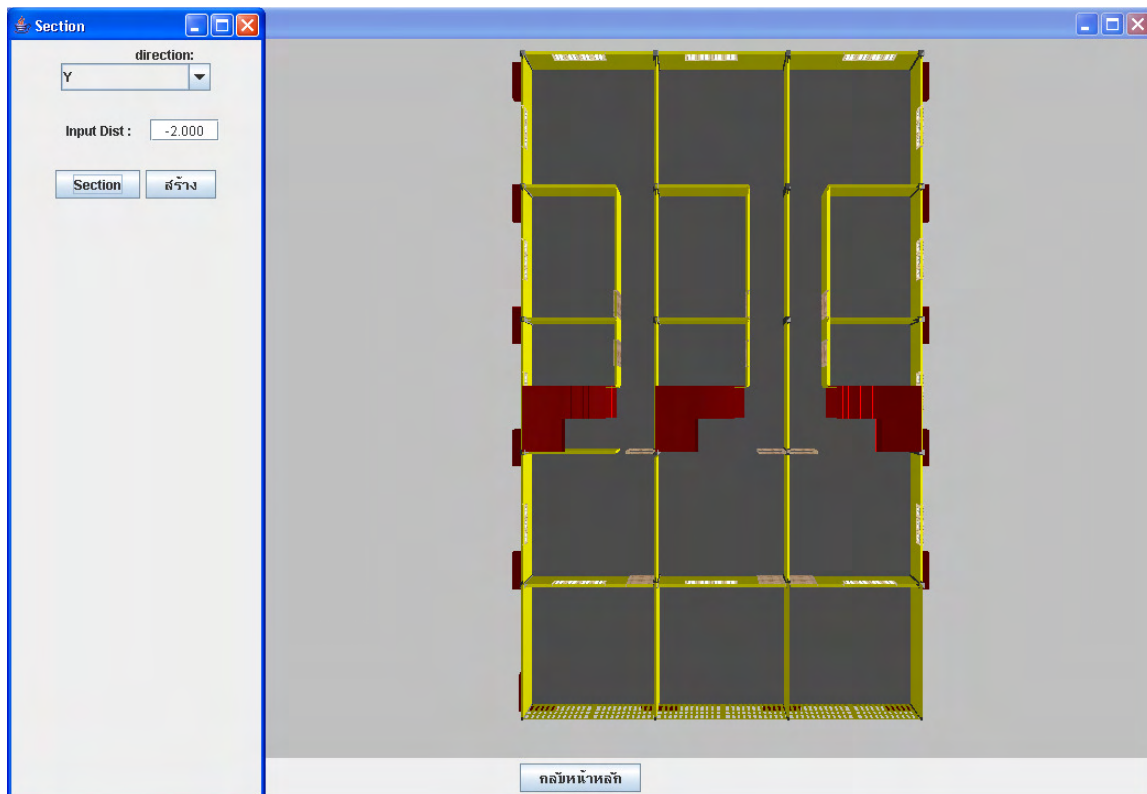
รูปที่ 4.35 รูปตัดทางขวางตามแกน Y ของอาคาร



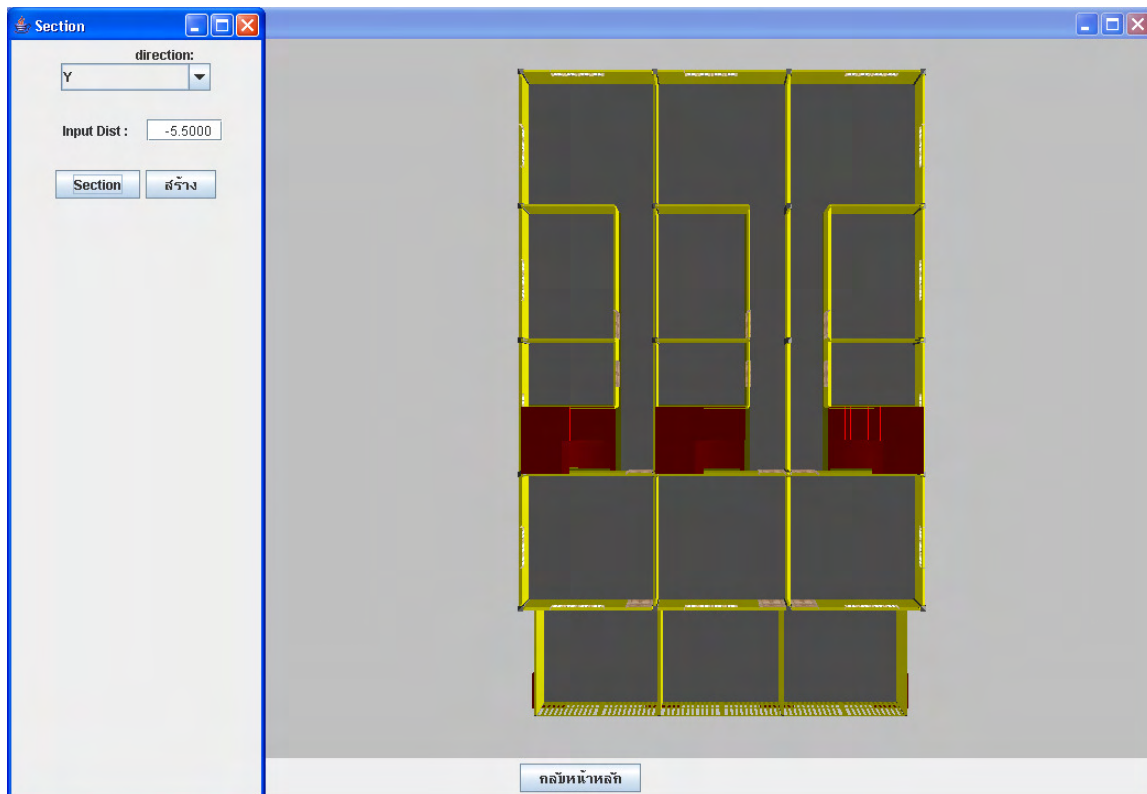
รูปที่ 4.36 รูปตัดทางขวางตามแกน Z ของอาคาร



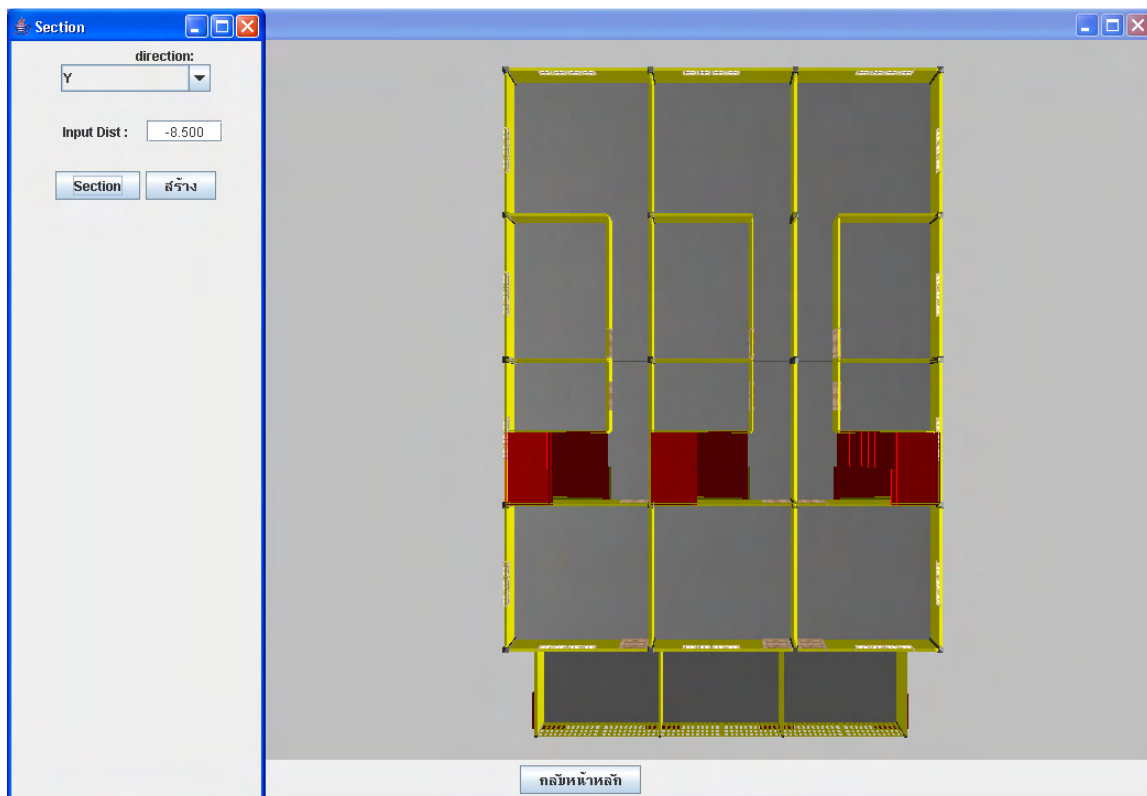
รูปที่ 4.37 รูปแปลนฐานรากของอาคาร



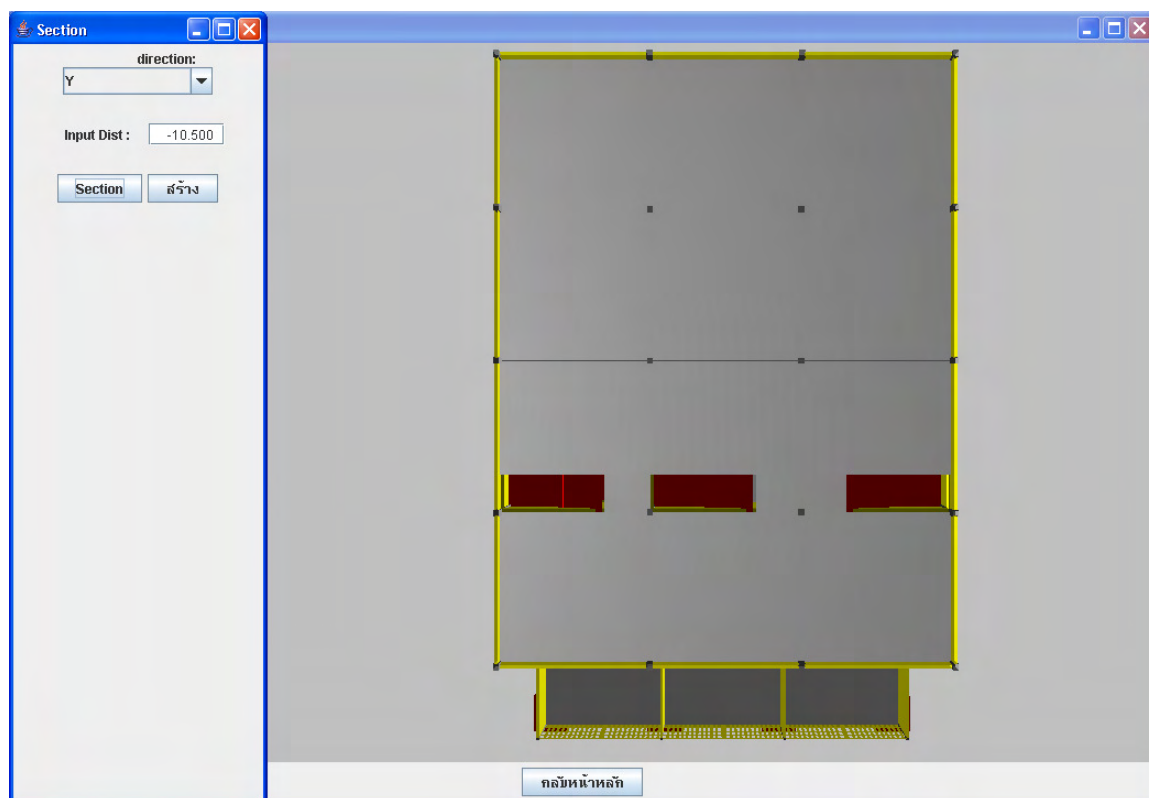
รูปที่ 4.38 รูปแปลนพื้นชั้น1 ของอาคาร



รูปที่ 4.39 รูปแปลนพื้นที่ 2 ของอาคาร



รูปที่ 4.40 รูปแปลนพื้นที่ 3 ของอาคาร



รูปที่ 4.41 รูปแปลนพื้นที่ชั้นดาดฟ้าของอาคาร

ผลจากการสร้างแบบจำลอง 3 มิติของอาคาร ข้อมูลพื้นฐานของชั้นส่วนอาคารแต่ละชั้นส่วนจะถูกจัดเก็บแยกตามชนิดของชั้นส่วนในฐานข้อมูล โดยข้อมูลเหล่านี้สามารถเรียกข้อมูลไปใช้ในการคำนวณปริมาตรของชั้นส่วน พื้นที่ผิวส่วนที่สัมผัสกับไม้แบบและพื้นที่ผิวส่วนที่ต้องทาสีของแต่ละชั้นส่วนได้โดยอัตโนมัติ ดังตารางที่ 4.9 แสดงผลรวมปริมาตรและพื้นที่ผิวของชั้นส่วนโครงสร้างอาคาร โดยข้อมูลในส่วนนี้สามารถนำไปพัฒนาต่อยอดได้ในเรื่องของ การประมาณราคา การออกแบบและวางแผนการใช้ไม้แบบ และการวางแผนการทาสีอาคาร เป็นต้น

ต้นแบบระบบที่พัฒนาขึ้นสามารถรองรับการวางแผนงานก่อสร้างแบบ 4 มิติ กล่าวคือ ต้นแบบระบบสามารถแสดงขั้นตอนการก่อสร้างอาคารในรูปแบบแอนิเมชัน (Animation) 3 มิติ โดยการแสดงผลชั้นส่วนอาคารตามลำดับแผนการทำงาน โดยผู้ใช้งานสามารถกำหนดวันที่จะทำการก่อสร้างชั้นส่วนแต่ละชั้นไปพร้อมกับการสร้างชั้นส่วนอาคารนั้นๆ หรือจะกำหนดวันที่ทำการก่อสร้างชั้นส่วนในตารางของฐานข้อมูลของโปรแกรม PostgreSQL โดยตรง นอกจากนี้ต้นแบบระบบยังมีศักยภาพสนับสนุนการวางแผนงานก่อสร้างแบบ 5 มิติ กล่าวคือ ผู้ใช้สามารถพิจารณามิติเวลาและค่าใช้จ่ายในงานก่อสร้างร่วมกับมิติ X, Y และ Z ของโครงการ

ตารางที่ 4.9 ผลรวมปริมาตรและพื้นที่ผิวของชิ้นส่วนโครงสร้างอาคาร

Type	Volume (m ³)	Formwork Area(m ²)	Color Area (m ²)
Pile	2.40	96.48	0
Footing	34.50	151.00	0
Column	11.65	221.62	216.60
Beam	74.12	834.50	834.50
Slab	125.44	1291.88	0
Wall	161.81	3236.16	3236.16
Fence	0.42	0	43.79
Door	8.64	0	169.92
Window	8.70	0	172.44
Stair	21.75	117.60	134.10
Ceiling	9.17	0	916.88

เพื่อเพิ่มความยืดหยุ่นในการนำต้นแบบระบบไปใช้งาน ผู้วิจัยได้ศึกษาแนวทางและความเป็นไปได้ในการเชื่อมโยงข้อมูลจากโปรแกรม AutoCAD 3D กับต้นแบบระบบที่พัฒนาขึ้น โดยการทดลองนำข้อมูลอาคาร 3 มิติ (รูปที่ 4.42) ในรูปแบบ AutoCAD 3D Solid Model นำเข้า (Import) สู่ฐานข้อมูล PostgreSQL ของต้นแบบระบบ จากผลการศึกษาพบว่าในการนำข้อมูล AutoCAD 3D Solid Model มาใช้งานร่วมกับต้นแบบระบบมีความจำเป็นต้องดำเนินการดัดแปลงข้อมูล AutoCAD 3D Solid Model ดังต่อไปนี้

1) การแปลงข้อมูลพิกัดตำแหน่ง

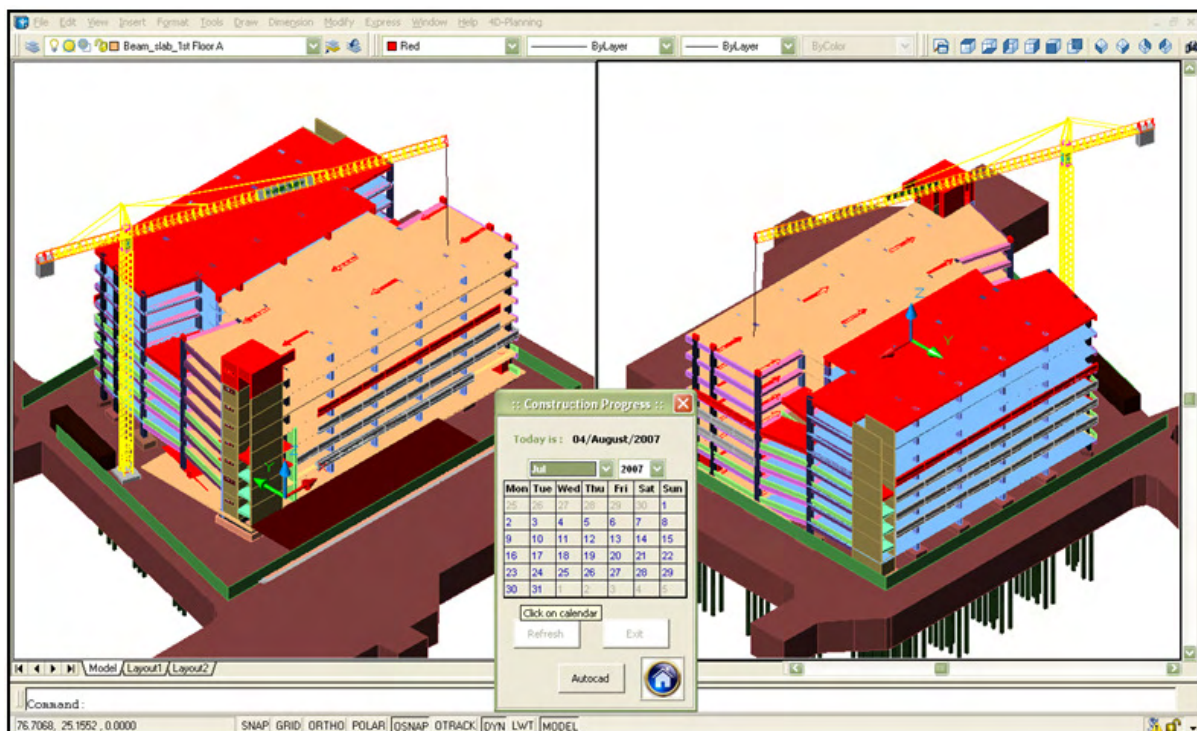
เนื่องจากโปรแกรม AutoCAD 3D กับภาษาจาวาใช้แกนอ้างอิงของระบบพิกัด 3 มิติที่แตกต่างกันดังแสดงในรูปที่ 4.43 ดังนั้นสิ่งแรกที่ต้องดำเนินการในการนำข้อมูล AutoCAD 3D Solid Model มาใช้งานร่วมกับต้นแบบระบบ คือการแปลงพิกัดแกน Z ของ AutoCAD 3D Solid Model เป็นพิกัดแกน Y ของภาษาจาวา โดยที่พิกัด Y ใน AutoCAD 3D Solid Model จะมีค่าเท่ากับพิกัด -Z ในภาษาจาวา

2) การกำหนดประเภทชิ้นส่วนของอาคาร

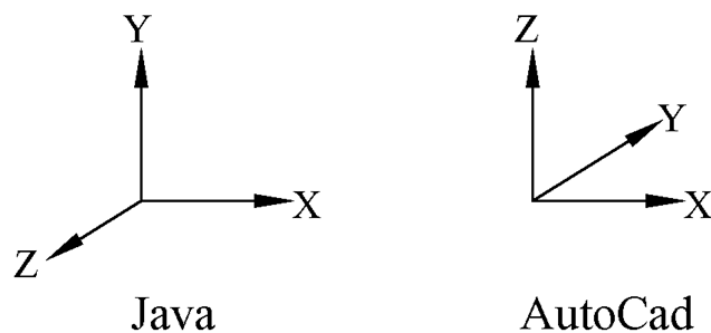
เนื่องจากโปรแกรม AutoCAD 3D เป็นเพียงโปรแกรมสำหรับขึ้นรูปแบบจำลอง 3 มิติ ดังนั้นข้อมูล AutoCAD 3D Solid Model จะถูกจัดเก็บในรูปแบบวัตถุเรขาคณิตทั่วไป (Simple Geometric Object) โดยไม่ระบุประเภทขององค์ประกอบอาคาร ทำให้ไม่สามารถนำไปใช้วิเคราะห์และประมวลผลเชิงวัตถุได้ รูปที่ 4.44 แสดงตัวอย่างการนำ AutoCAD 3D Solid Model ที่ขาดรายละเอียดเชิงวัตถุมาแสดงผลในต้นแบบระบบโดย

ทุกชั้นส่วนจะถูกเชื่อมโยงไปสู่ BranchGroup เดียวกัน การขาดรายละเอียดเชิงวัตถุจะส่งผลให้ผู้ใช้ไม่สามารถแยกองค์ประกอบอาคารได้ทำให้เกิดข้อจำกัดในการนำข้อมูลไปใช้วิเคราะห์และแสดงผลต่อไป

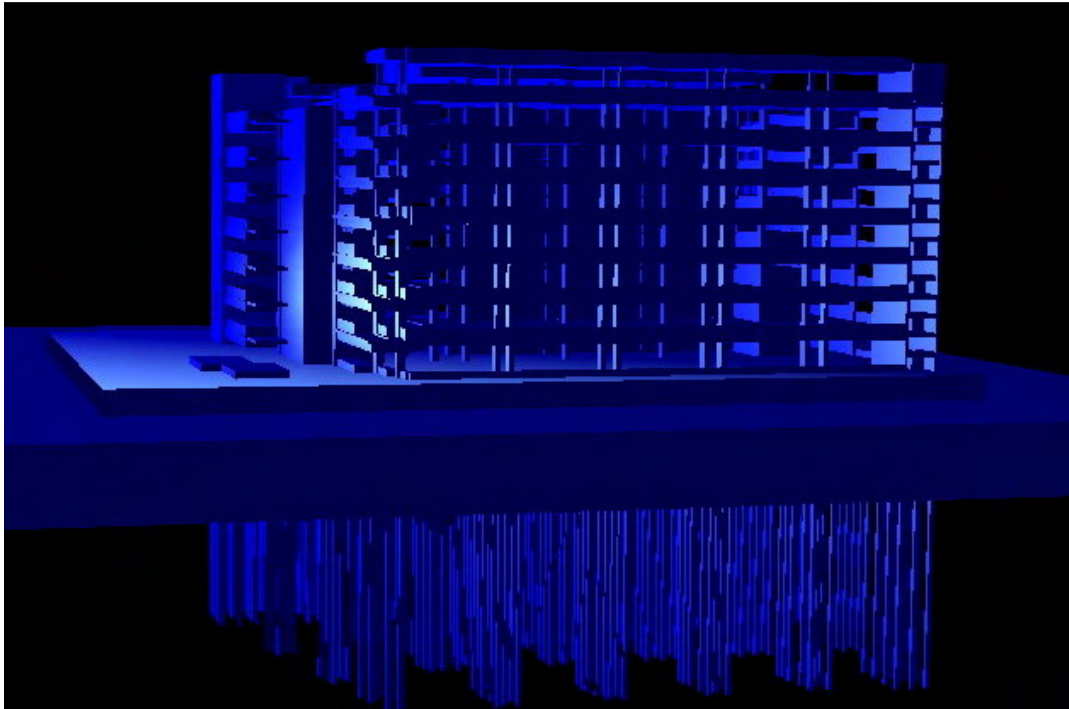
การแก้ปัญหาการขาดรายละเอียดเชิงวัตถุของ AutoCAD 3D Solid Model สามารถทำได้โดยการกำหนดสีมาตรฐานเฉพาะองค์ประกอบอาคารแต่ละประเภท โดยผู้ที่นำข้อมูล AutoCAD 3D Solid Model มาใช้งานจะสามารถตรวจสอบประเภทขององค์ประกอบอาคารได้โดยอัตโนมัติจากหมายเลขสีของวัตถุเรขาคณิตแต่ละชั้นของ AutoCAD 3D Solid Model



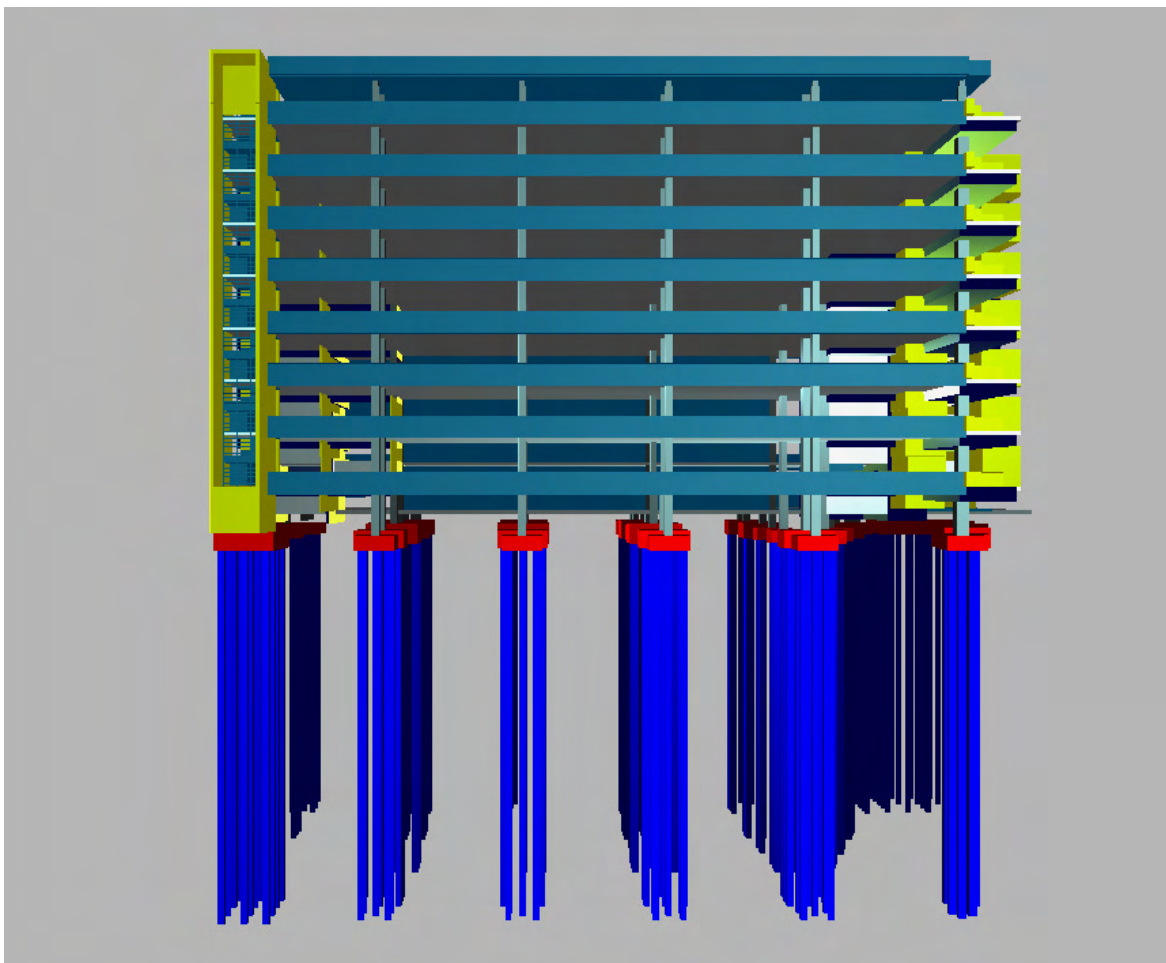
รูปที่ 4.42 อาคาร 3 มิติที่สร้างขึ้นโดยใช้ AutoCAD 3D Solid Model



รูปที่ 4.43 การเปรียบเทียบแกนอ้างอิงระบบพิกัด 3 มิติ



รูปที่ 4.44 การแสดงผล AutoCAD 3D Solid Model ที่ขาดรายละเอียดเชิงวัตถุบนต้นแบบระบบ



รูปที่ 4.45 การแปลง AutoCAD 3D Solid Model เป็นแบบจำลองเชิงวัตถุ

5. สรุปและวิจารณ์ผลการวิจัย

การนำภาษาจาวามาใช้ร่วมกับโปรแกรมฐานข้อมูล PostgreSQL/PostGIS เป็นทางเลือกใหม่ในการพัฒนาระบบสารสนเทศสำหรับการจัดการข้อมูลแบบจำลองอาคารเชิงวัตถุ 3 มิติ ซึ่งนอกจากจะช่วยลดการขาดดุลการค้าของประเทศในการจัดซื้อโปรแกรมสำเร็จรูปราคาแพงจากต่างประเทศแล้ว การนำภาษาจาวาซึ่งมีคลาส (Class) สนับสนุนการสร้าง การวิเคราะห์ และการแสดงผลวัตถุเชิง 3 มิติมาใช้ร่วมกับโปรแกรมฐานข้อมูล PostgreSQL/PostGIS ซึ่งเป็นโปรแกรมโอเพนซอร์ส (Open Source) ที่มีจุดเด่นในการบริหารจัดการข้อมูลเชิงพื้นที่ (Spatial Data) จะช่วยให้การสร้าง การวิเคราะห์ และการแสดงผลแบบจำลองรวมถึงการจัดการข้อมูลเชิงเรขาคณิตของแบบจำลองอาคารเชิงวัตถุ 3 มิติเป็นไปอย่างมีประสิทธิภาพ

งานวิจัยนี้ได้นำเสนอองค์ความรู้ในการพัฒนาระบบสารสนเทศสำหรับการจัดการข้อมูลแบบจำลองอาคารเชิงวัตถุ 3 มิติ โดยใช้ภาษาจาวาร่วมกับโปรแกรมฐานข้อมูล PostgreSQL/PostGIS โดยผู้วิจัยได้ทำการศึกษา วิเคราะห์ ออกแบบ และพัฒนาต้นแบบระบบ (Prototype System) รวมถึงได้จัดทำแบบจำลองอาคารต้นแบบเพื่อแสดงให้เห็นถึงศักยภาพและแนวทางในการใช้ภาษาจาวาร่วมกับโปรแกรมฐานข้อมูล PostgreSQL/PostGIS ในการสร้าง วิเคราะห์ และแสดงผลแบบจำลองอาคารเชิงวัตถุ 3 มิติ องค์ความรู้ที่ได้จากงานวิจัยนี้สามารถนำไปใช้ในการพัฒนาต่อยอดเพื่อให้ภาคอุตสาหกรรมก่อสร้างของประเทศไทยโดยเฉพาะสำหรับผู้ประกอบการขนาดกลางและขนาดเล็กสามารถนำระบบสารสนเทศสำหรับการจัดการข้อมูลแบบจำลองอาคารเชิงวัตถุ 3 มิติไปใช้งานจริงในอนาคต เพื่อเพิ่มประสิทธิภาพในการทำงานและช่วยลดระยะเวลาและต้นทุนในการออกแบบและก่อสร้างซึ่งเป็นผลดีต่อทั้งผู้ประกอบการและต่ออุตสาหกรรมก่อสร้างโดยรวมของประเทศไทย นอกจากนี้ยังเป็นการช่วยลดความเสียหายเปรียบทางด้านเทคโนโลยีของบริษัทผู้ประกอบการชาวไทยต่อบริษัทผู้ประกอบการขนาดใหญ่จากต่างประเทศที่ในปัจจุบันมีการประยุกต์ใช้เทคโนโลยีสารสนเทศในการออกแบบ วางแผน และบริหารโครงการก่อสร้างอย่างแพร่หลาย

อย่างไรก็ดีต้นแบบระบบที่ผู้วิจัยได้พัฒนาขึ้นยังมีข้อจำกัดในการจัดการข้อมูลเชิงเรขาคณิตของเหลี่ยมค้อนกริตที่มีจำนวนมากซึ่งทำให้ข้อมูลมีขนาดใหญ่และยากต่อการประมวลผล ทั้งนี้ส่งผลให้เป็นอุปสรรคต่อการคำนวณปริมาณเหลี่ยมค้อนกริตและการคำนวณค่าใช้จ่ายงานค้อนกริตเสริมเหล็ก ซึ่งข้อจำกัดในส่วนนี้ควรได้รับการศึกษาวิจัยต่อไปเพื่อให้ระบบสามารถสนับสนุนการสร้างแบบจำลองอาคารค้อนกริตเสริมเหล็กได้อย่างสมบูรณ์ นอกจากนั้นต้นแบบระบบที่ได้พัฒนาขึ้นยังมีข้อจำกัดในการสร้างและจัดการข้อมูลเชิงเรขาคณิตของงานระบบซึ่งมีความซับซ้อนและควรได้รับการศึกษาวิจัยต่อไป

ต้นแบบระบบที่พัฒนาขึ้นยังขาดซึ่งฟังก์ชันสำหรับอำนวยความสะดวกในการสร้างและแสดงผลแบบจำลองอาคาร 3 มิติ และฟังก์ชันสนับสนุนงานออกแบบและบริหารโครงการ อาทิเช่น การออกแบบโครงสร้าง การวางแผนงานก่อสร้าง การติดตามความก้าวหน้าและประเมินโครงการ การจัดการทรัพยากรโครงการ และการวางแผนการใช้งานแบบหล่อ เป็นต้น ซึ่งฟังก์ชันเหล่านี้ควรได้รับการวิจัยและพัฒนาต่อยอดในอนาคตเพื่อเพิ่มแรงจูงใจในการใช้งานระบบและเพื่อให้ระบบสามารถสนับสนุนการทำงานของผู้ประกอบการก่อสร้างได้อย่างมีประสิทธิภาพ

6. เอกสารอ้างอิง

- [1] Brown, M., Drummond, T., and Cipolla, R. (2000) "3D Model Acquisition by Tracking 2D Wireframes.", Proceedings, The Eleventh British Machine Vision Conference, University of Bristol, England, September 11-14, 2000
- [2] Eastman, C.M., Sacks, R., Lee, G. (2002) "Strategies for Realizing the Benefits of 3D Integrated Modeling of Buildings for the AEC Industry." Proceedings, International Symposium on Automation and Robotics in Construction, National Institute of Standards and Technology, Gaithersburg, Maryland, USA, September 23-25, 2002, p. 9-14
- [3] Griffis, F.H., Hogan, D. and Li, W. (1995) "An Analysis of the Impacts of Using Three-Dimensional Computer Models in the Management of Construction." Construction Industry Institute Report RR106-11
- [4] Griffis, F.H. and Sturts, C. (2000) "Three-Dimensional Computer Models and the Fully Integrated and Automated Project Process for the Management of Construction." Construction Industry Institute Report RR152-11
- [5] Khemlani, L. (2003) "Should We BIM? Pushing the State of the Art in AEC." Cadalyst, USA, Volume 6
- [6] Khemlani, L. (2004) "The Eureka Tower: A Case Study of Advanced BIM Implementation." AECbytes, USA, Volume 6
- [7] Moreau, K.A. and Back, W.E. (2000) "Improving the Design Process with Information Management." Automation in Construction, Volume 10, p. 127–140
- [8] Powers, M. B. (2005) "Contractors Scrambling To Master 3-D Modeling Technologies." McGraw-Hill Construction, Issue September/October 2005, <http://www.construction.com/>
- [9] Refrations Research (2009), "PostGIS 1.4.0 Manual", Available source: <http://postgis.refrations.net/>
- [10] Sampaio, A. Z., Henriques, P., and Studer, P. (2005) "Learning Construction Processes Using Virtual Reality Models.", Electronic Journal of Information Technology in Construction, Portugal, Volume 10, p. 141-151
- [11] Sangkaew, W., Malaikrisanachalee, S. and Vathananukit, H. (2008) "Java-based Three-Dimensional Object-Oriented Model for Building Information Modeling.", In Proceedings, 13th National Convention on Civil Engineering, Choburi, Thailand
- [12] Srisomboon, W. and Malaikrisanachalee, S. (2008) "Four-Dimensional Object-Oriented Construction Project Planning.", In Proceedings, 13th National Convention on Civil Engineering, Choburi, Thailand

- [13] National Institute of Building Sciences (2007) , “National Building Information Standard.”, version 1, United States
- [14] ชีววัฒน์ บุญศิริวนนท์ (2544) “VRML เทคนิคการสร้างกราฟิก 3 มิติบนอินเทอร์เน็ต.” บริษัทซีเอ็ดยูเคชั่น จำกัด(มหาชน). กรุงเทพฯ. พิมพ์ครั้งที่ 1
- [15] พาสทรี หล่อธีรพงศ์ และ กมลวัลย์ ลือประเสริฐ (2542) "Lean construction." โยธาสาร, เมษายน-มิถุนายน, ปีที่ 11 ฉบับที่ 2, pp. 11-16
- [16] ภาณุพงษ์ ปัตติสิงห์, (2548) “คู่มือการใช้โปรแกรม AutoCAD 14: 3D Modeling.” บริษัทซีเอ็ดยูเคชั่น จำกัด(มหาชน). กรุงเทพฯ, พิมพ์ครั้งที่ 1
- [17] วิชิต บัวแก้ว, ธีรภัทร หล่มบุญเรือง, และ พิสมัย พันธุ์อภัย (2547) “Solidworks 2001^{PLUS}.” บริษัท ซี วี แอล การพิมพ์ จำกัด, กรุงเทพฯ, พิมพ์ครั้งที่ 2
- [18] วิวัฒน์ อุดมปิติทรัพย์ (2548) “Autodesk Revit: เรียนรู้การสร้างงานออกแบบสถาปัตยกรรมอย่างมืออาชีพ.” Engineer and Architect Plus Innovation Company, กรุงเทพฯ, พิมพ์ครั้งที่ 1
- [19] สิทธิโชค ศิริวิวัฒน์ (2541) “การศึกษาการใช้โปรแกรมกราฟิก 3 มิติในการวางแผนงานก่อสร้างโดยอัตโนมัติ.” มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี, กรุงเทพฯ

7. ผลผลิตจากโครงการวิจัย

1. ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติ

S. Malaikrisanachalee and H. Vathananukij (In Press) “Integration of Java-based BIM with Spatial Database.” International Journal of Civil Engineering

หมายเหตุ

- คาดว่าจะตีพิมพ์ในเดือนมีนาคม 2011 (Vol. 9, No. 1, March 2011)
- วารสาร International Journal of Civil Engineering เป็นวารสารซึ่งอยู่ในฐานข้อมูล Thomson Reuters (Science Citation Index Expanded) และ SCOPUS

2. การนำผลงานวิจัยไปใช้ประโยชน์

ผู้วิจัยได้นำผลการวิจัยที่ได้ไปใช้ในการพัฒนาการเรียนการสอนในวิชา 01229531 (Computer-Aided Design and Information Management System) รวมถึงได้นำไปใช้ในงานวิจัยของนิสิตปริญญาโท สาขาบริหารงานก่อสร้าง ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

ภาคผนวก ก

การติดตั้งโปรแกรมสนับสนุนต้นแบบระบบ

ผู้วิจัยได้ทำการติดตั้งโปรแกรมเพื่อสนับสนุนการพัฒนาต้นแบบระบบสารสนเทศแบบจำลองเชิงวัตถุ 3 มิติดังต่อไปนี้

- โปรแกรม Java Development Kit (JDK) รุ่น jdk-1_5_0_08-windows-i586-p เป็นชุดโปรแกรมพื้นฐานสำหรับการทำงานของภาษาจาวา
- โปรแกรม Java3D API รุ่น java3d-1_3_1-windows-i586-opengl-sdk เป็นโปรแกรมเสริมที่ช่วยให้สามารถสร้างและนำเสนอวัตถุเชิง 3 มิติ
- โปรแกรม PostgreSQL รุ่น 8.4 สำหรับบริหารจัดการฐานข้อมูลแบบจำลองอาคารเชิงวัตถุ 3 มิติ
- โปรแกรม PostGIS รุ่น 1.5 สำหรับเพิ่มความสามารถของฐานข้อมูล PostgreSQL ให้สามารถจัดการข้อมูลเชิงเรขาคณิตของแบบจำลองอาคารเชิงวัตถุ 3 มิติได้

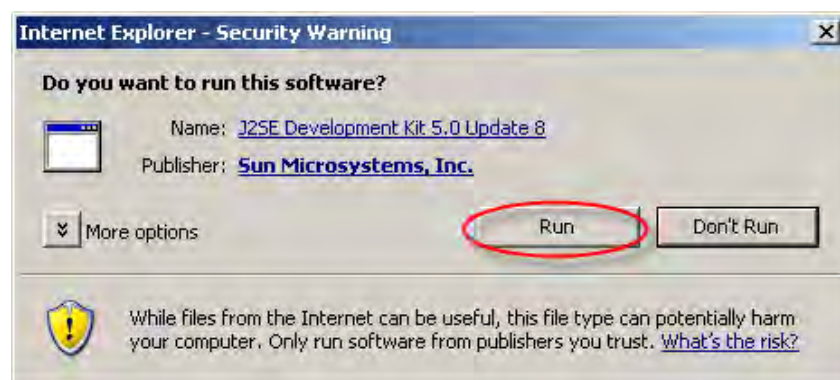
การติดตั้งโปรแกรม Java Development Kit

1) ดับเบิลคลิกที่ไอคอน jdk-1_5_0_08-windows-i586-p



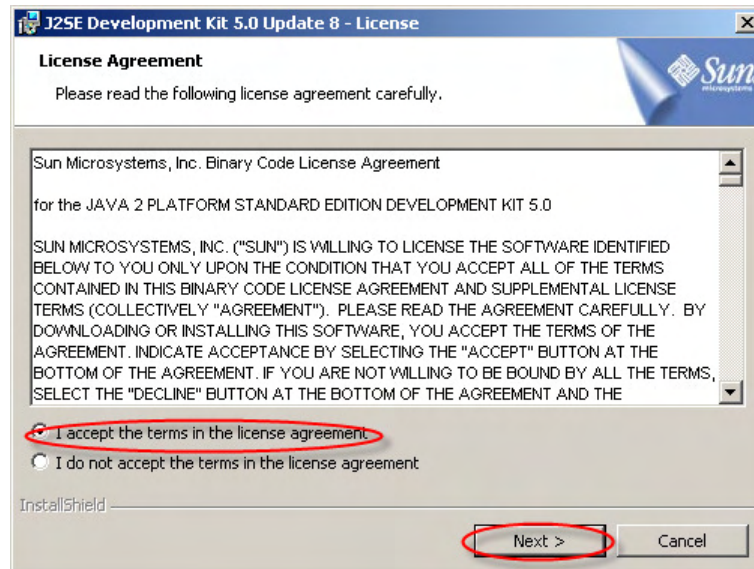
รูปที่ ก.1 ไอคอนของโปรแกรม jdk-1_5_0_08-windows-i586-p

2) เมื่อทำตามขั้นตอนที่ 1 เรียบร้อยแล้ว ขั้นตอนนี้ให้คลิกที่ปุ่ม “RUN” เพื่อเริ่มการติดตั้งโปรแกรม



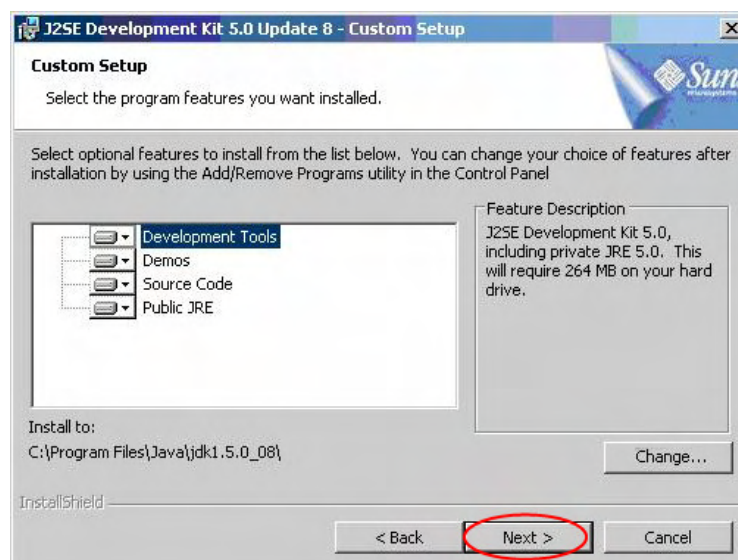
รูปที่ ก.2 ขั้นตอนที่ 2 ในการติดตั้ง jdk-1_5_0_08-windows-i586-p

3) เมื่อทำตามขั้นตอนที่ 2 เรียบร้อยแล้ว ขั้นตอนนี้เป็นการทำข้อตกลงก่อนใช้ jdk-1_5_0_08-windows-i586-p เมื่ออ่านข้อตกลงเรียบร้อยแล้วให้คลิกในช่อง "I accept the terms in the license agreement" แล้วคลิกที่ปุ่ม Next



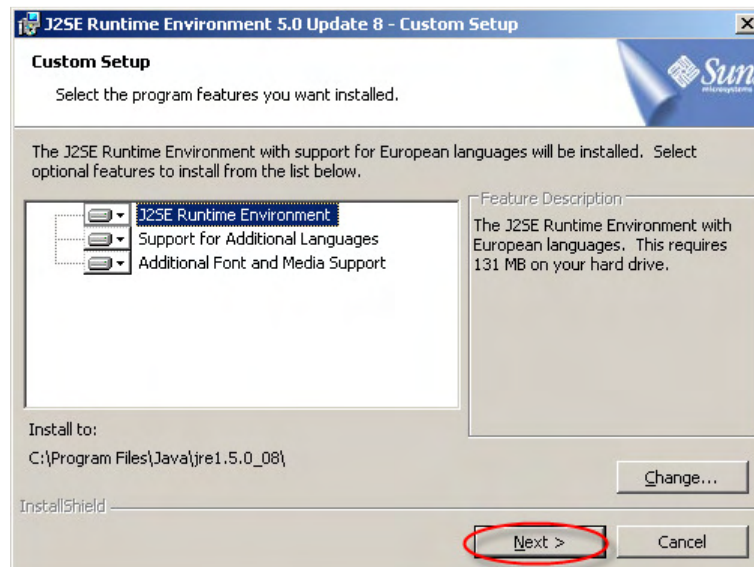
รูปที่ ก.3 ขั้นตอนที่ 3 ในการติดตั้ง jdk-1_5_0_08-windows-i586-p

4) เมื่อทำตามขั้นตอนที่ 3 เรียบร้อยแล้ว ขั้นตอนนี้เป็นการเลือกว่าต้องการติดตั้งเครื่องมือใดของ Java jdk บ้าง และเลือก directory ที่ต้องการจะติดตั้งโปรแกรมโดยคลิก ที่ปุ่ม Change และเมื่อทำการเลือกแล้วให้ คลิก ที่ปุ่ม Next



รูปที่ ก.4 ขั้นตอนที่ 4 ในการติดตั้ง jdk-1_5_0_08-windows-i586-p

5) เมื่อทำตามขั้นตอนที่ 4 เรียบร้อยแล้ว ขั้นตอนนี้เป็นการเลือกว่าต้องการติดตั้งเครื่องมือใดของ Java jre บ้าง และเลือก directory ที่ต้องการจะติดตั้งโปรแกรมโดยคลิก ที่ปุ่ม Change และเมื่อทำการเลือกแล้วให้คลิกที่ปุ่ม Next



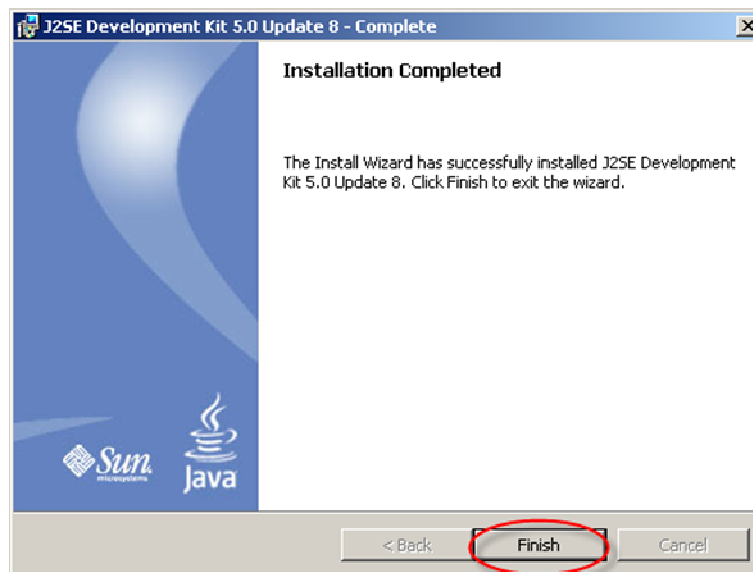
รูปที่ ก.5 ขั้นตอนที่ 5 ในการติดตั้ง jdk-1_5_0_08-windows-i586-p

6) เมื่อทำตามขั้นตอนที่ 5 เรียบร้อยแล้ว ขั้นตอนนี้ให้คลิกที่ปุ่ม Next เพื่อทำการติดตั้งโปรแกรม จากนั้นโปรแกรม jdk-1_5_0_08-windows-i586-p จะทำการติดตั้งโดยอัตโนมัติ



รูปที่ ก.6 ขั้นตอนที่ 6 ในการติดตั้ง jdk-1_5_0_08-windows-i586-p

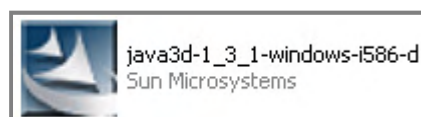
7) เมื่อทำตามขั้นตอนที่ 6 เรียบร้อยแล้ว ขั้นตอนสุดท้ายเป็นการบอกว่าผลการลงโปรแกรมสำเร็จหรือไม่ เมื่อสำเร็จแล้วให้คลิกที่ปุ่ม “Finish” เสร็จสิ้นการติดตั้ง



รูปที่ ก.7 ขั้นตอนสุดท้ายในการติดตั้ง jdk-1_5_0_08-windows-i586-p

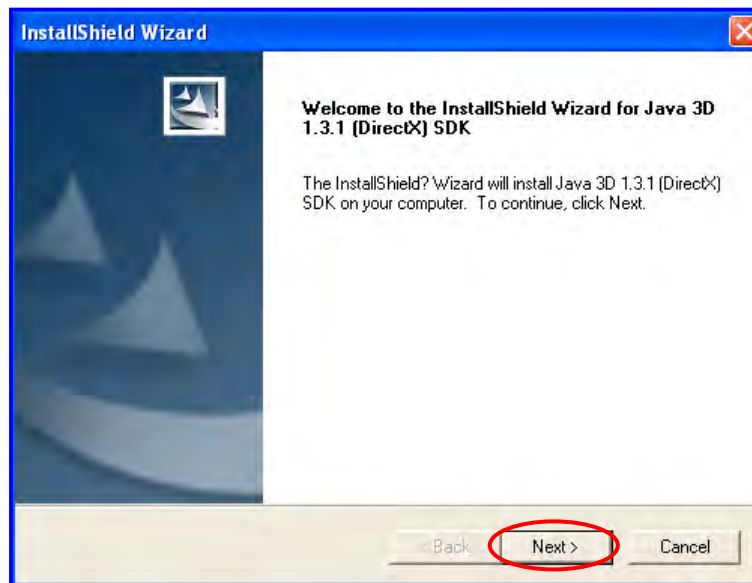
ขั้นตอนในการติดตั้ง JAVA3D-1_3_1-windows-i586-opengl-sdk

1) ดับเบิลคลิกที่ไอคอน java3d-1_3_1-windows-i586-opengl-sdk.exe p



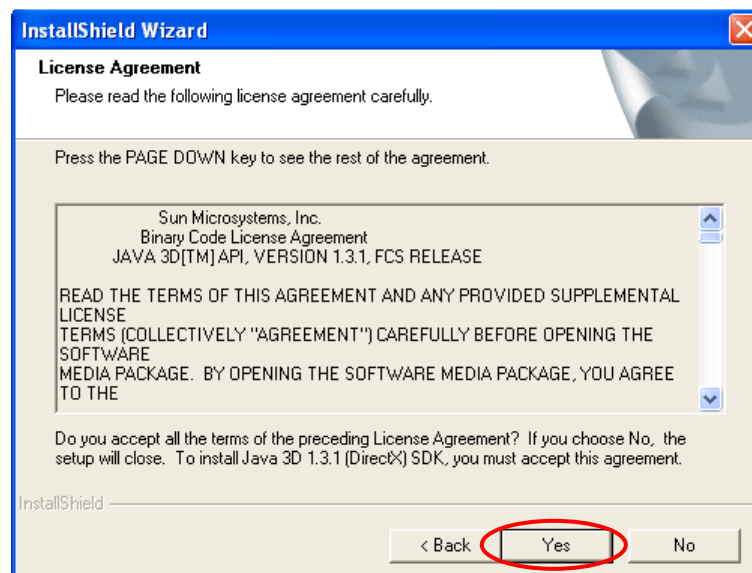
รูปที่ ก.8 ไอคอนของโปรแกรม java3d-1_3_1-windows-i586-opengl-sdk

2) เมื่อทำตามขั้นตอนที่ 1 เรียบร้อยแล้ว ขั้นตอนนี้ให้คลิกที่ปุ่ม “Next” เพื่อทำการ Run โปรแกรมติดตั้ง



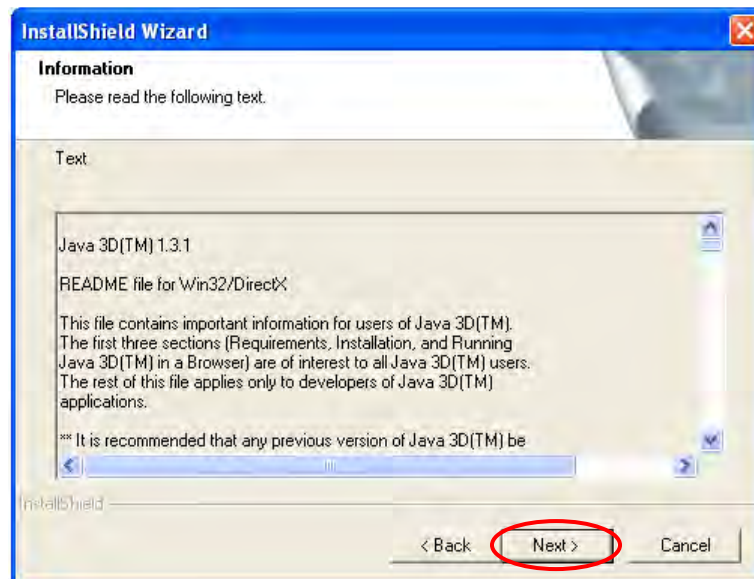
รูปที่ ก.9 ขั้นตอนที่ 2 ในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk

3) เมื่อทำตามขั้นตอนที่ 2 เรียบร้อยแล้ว ขั้นตอนนี้เป็นการทำข้อตกลงก่อนใช้ java3d-1_3_1-windows-i586-opengl-sdk.exe เมื่ออ่านข้อตกลงเรียบร้อยแล้วให้คลิกในช่อง “Yes”



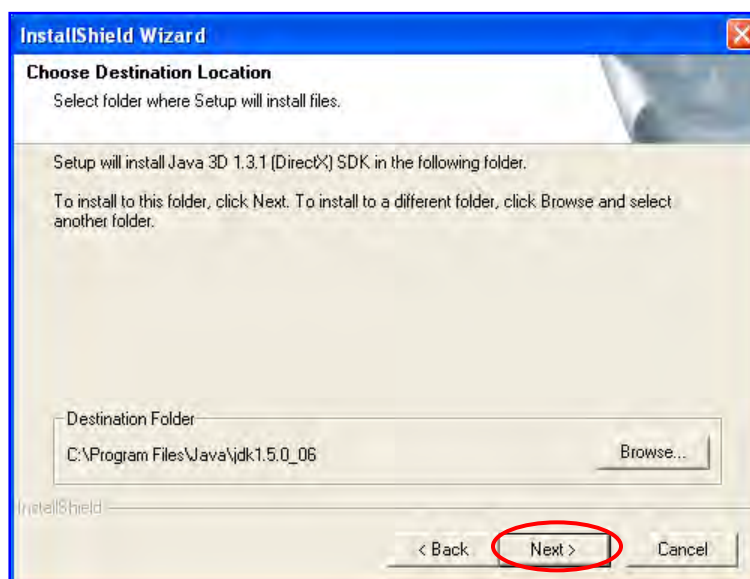
รูปที่ ก.10 ขั้นตอนที่ 3 ในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk

4) เมื่อทำตามขั้นตอนที่ 3 เรียบร้อยแล้ว ขั้นตอนนี้เป็นการทำข้อตกลงก่อนใช้งานเช่นเดียวกับขั้นตอนที่ 3 เมื่ออ่านข้อตกลงเรียบร้อยแล้วให้คลิกในช่อง “Next”



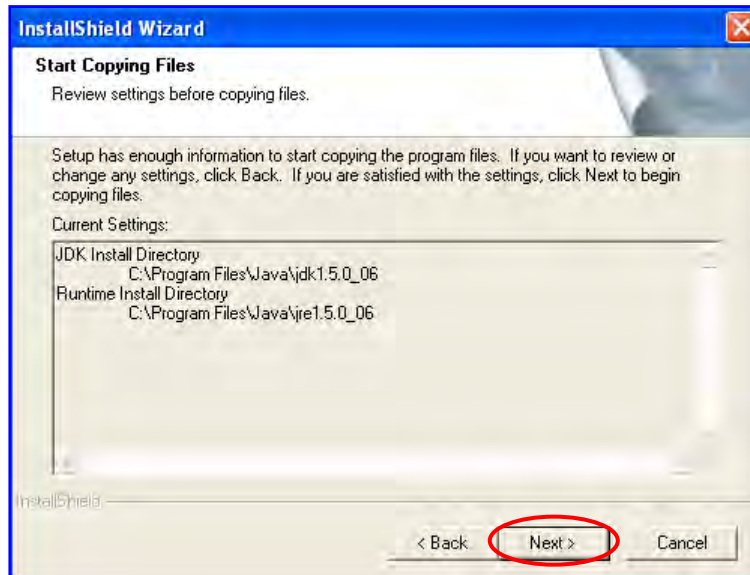
รูปที่ ก.11 ขั้นตอนที่ 4 ในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk

5) เมื่อทำตามขั้นตอนที่ 4 เรียบร้อยแล้ว ขั้นตอนนี้เป็นการเลือก directory ที่จะทำการลงโปรแกรม ให้เลือกไปยัง directory เดียวกันกับที่ได้ลงโปรแกรม jdk-1_5_0_08-windows-i586-p เพื่อให้สามารถทำงานร่วมกันได้โดย คลิก ที่ปุ่ม Browse เพื่อทำการเลือก directory แล้ว คลิก ที่ปุ่ม Next



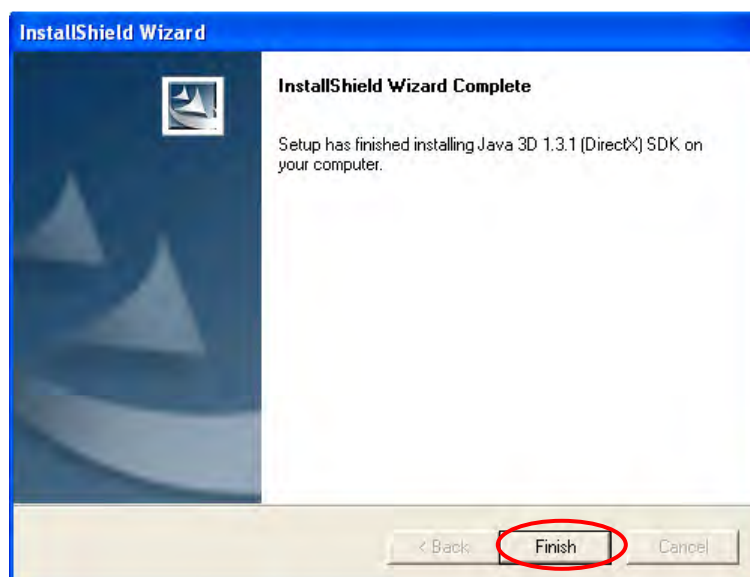
รูปที่ ก.12 ขั้นตอนที่ 5 ในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk

6) เมื่อทำตามขั้นตอนที่ 5 เรียบร้อยแล้ว เมื่อทำตามขั้นตอนที่ 4 เรียบร้อยแล้ว ขั้นตอนนี้ให้คลิกที่ปุ่ม “Next” เพื่อทำการติดตั้งโปรแกรมลงยัง directory ที่เลือกไว้ จากนั้นโปรแกรมจะทำการติดตั้งโดยอัตโนมัติ



รูปที่ ก.13 ขั้นตอนที่ 6 ในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk

7) เมื่อทำตามขั้นตอนที่ 6 เรียบร้อยแล้ว ขั้นตอนที่สุดท้ายเป็นการบอกว่าผลการลงโปรแกรมสำเร็จหรือไม่ เมื่อสำเร็จแล้วให้คลิกที่ปุ่ม “Finish” เสร็จสิ้นการติดตั้งโปรแกรม



รูปที่ ก.14 ขั้นตอนสุดท้ายในการติดตั้ง java3d-1_3_1-windows-i586-opengl-sdk

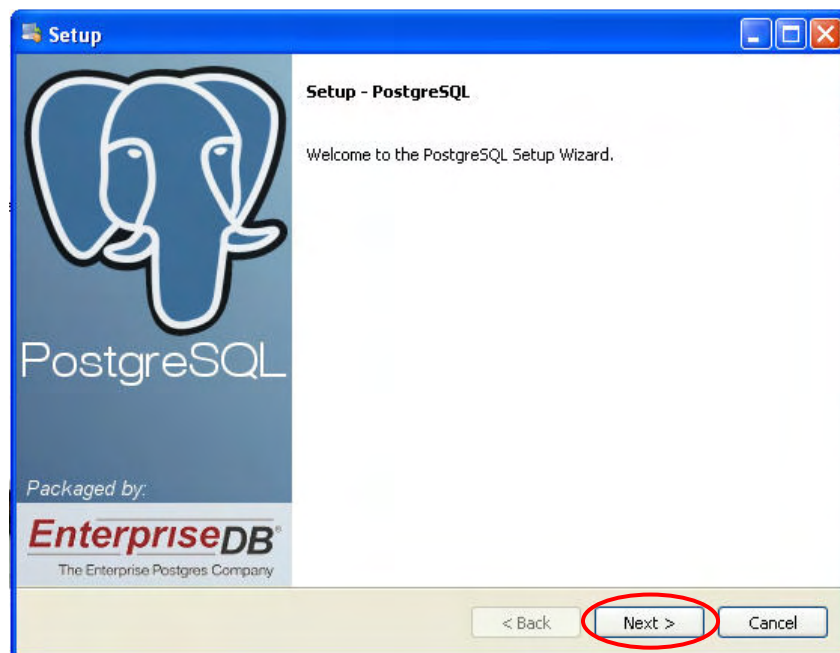
การติดตั้งโปรแกรม PostgreSQL

- 1) ดับเบิลคลิกที่ไอคอน postgresql-8.4.6-1-windows



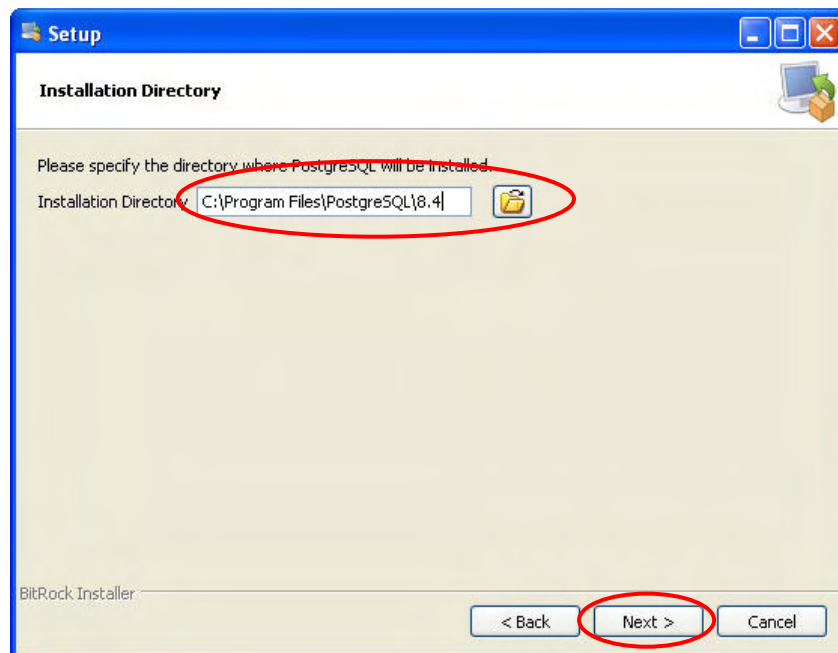
รูปที่ ก.15 ไอคอนของโปรแกรม postgresql-8.4.6-1-windows

- 3) เมื่อทำตามขั้นตอนที่ 1 เรียบร้อยแล้ว ขั้นตอนนี้ ให้คลิกที่ปุ่ม “Next” เพื่อทำการ Run โปรแกรมติดตั้ง



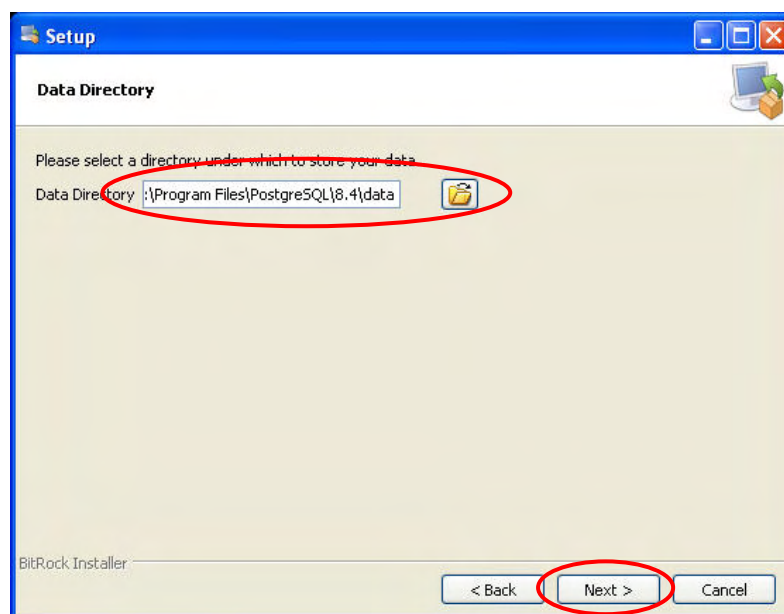
รูปที่ ก.16 ไอคอนของโปรแกรม postgresql-8.4.6-1-windows

3) เมื่อทำตามขั้นตอนที่ 2 เรียบร้อยแล้ว ขั้นตอนนี้เป็นการเลือกตำแหน่งที่ต้องการติดตั้งโปรแกรม PostgreSQL ให้ผู้ใช้งานเลือกที่ C:\Program Files\PostgreSQL\8.4 ดังรูปที่ ก.17 จากนั้นให้กดปุ่ม “Next”



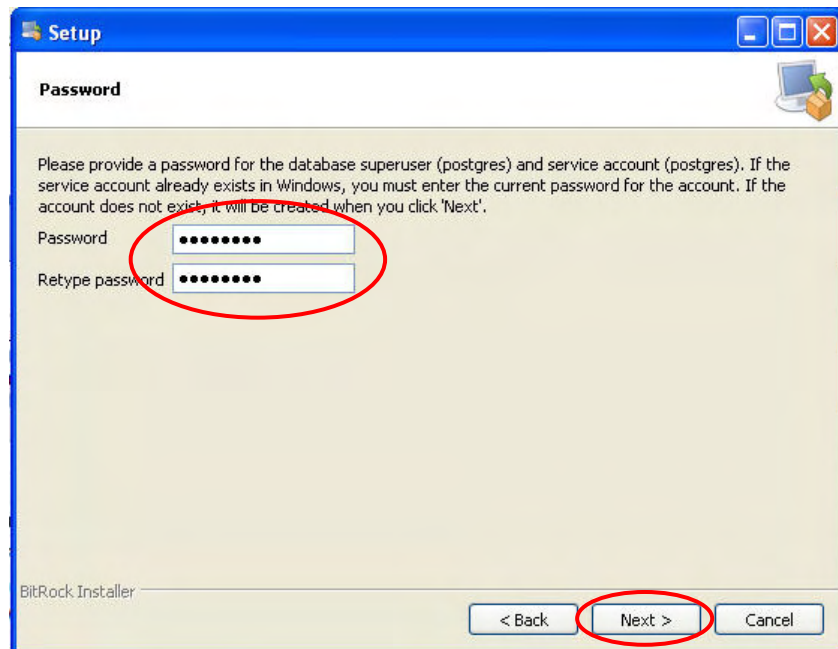
รูปที่ ก.17 ขั้นตอนที่ 1 ในการติดตั้งโปรแกรม postgresql-8.4.6-1

4) เมื่อทำตามขั้นตอนที่ 3 เรียบร้อยแล้ว ขั้นตอนนี้เป็นการเลือกตำแหน่งที่ต้องการใช้จัดเก็บข้อมูลของโปรแกรม PostgreSQL ให้ผู้ใช้งานเลือกที่ C:\Program Files\PostgreSQL\8.4\data ดังรูปที่ ก.18 จากนั้นให้กดปุ่ม “Next”



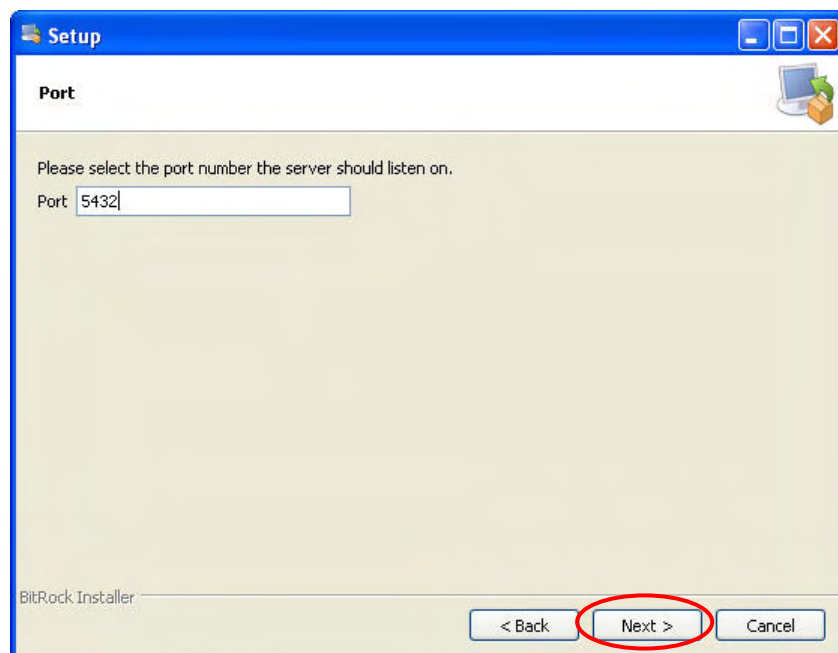
รูปที่ ก.18 ขั้นตอนที่ 2 ในการติดตั้งโปรแกรม postgresql-8.4.6-1

5) เมื่อทำตามขั้นตอนที่ 4 เรียบร้อยแล้ว ขั้นตอนต่อไปให้ผู้ใช้งานป้อนข้อมูล Password สำหรับระบบป้องกันความปลอดภัยของการใช้ฐานข้อมูล โดยให้ป้อนข้อมูล password และทำการยืนยัน password อีกครั้งหนึ่ง จากนั้นให้กดปุ่ม “Next”



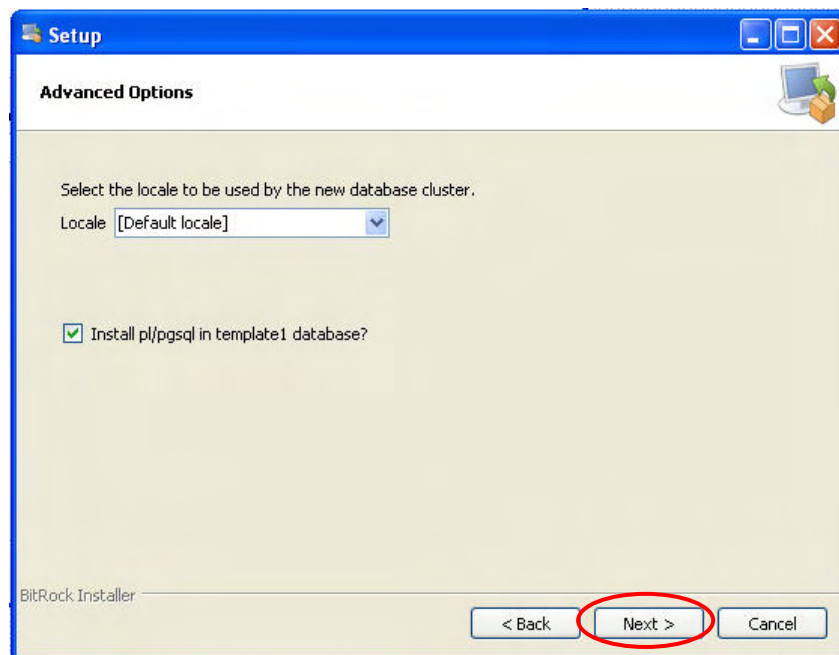
รูปที่ ก.19 ขั้นตอนที่ 3 ในการติดตั้งโปรแกรม postgresql-8.4.6-1

6) เมื่อทำตามขั้นตอนที่ 5 เรียบร้อยแล้ว ขั้นตอนต่อไปให้ผู้ใช้งานป้อน Port ของระบบที่ใช้งานให้ผู้ใช้งานป้อนข้อมูลเป็น “5432” จากนั้นให้กดปุ่ม “Next”



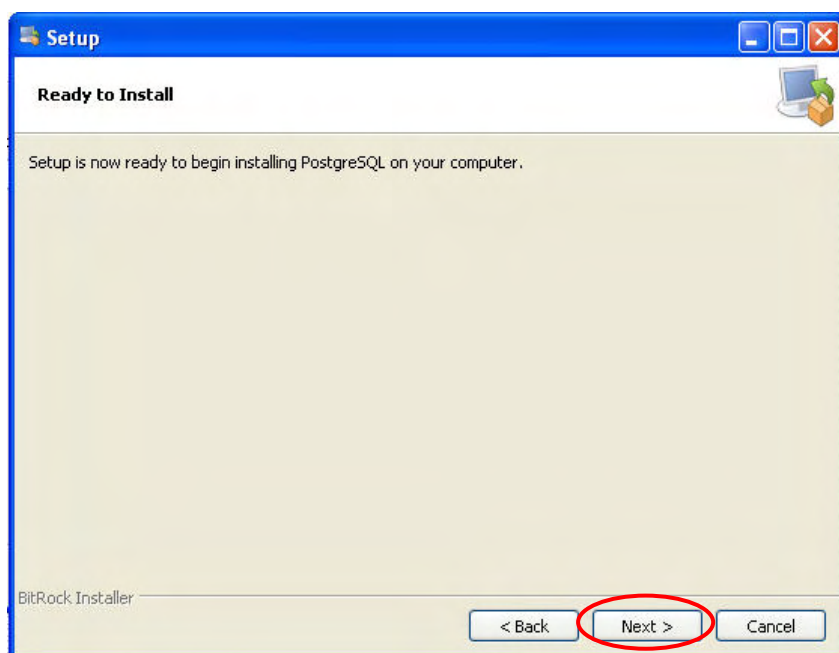
รูปที่ ก.20 ขั้นตอนที่ 4 ในการติดตั้งโปรแกรม postgresql-8.4.6-1

7) เมื่อทำตามขั้นตอนที่ 6 เรียบร้อยแล้ว ขั้นตอนต่อไปให้ผู้ใช้งานเลือกข้อมูลท้องถิ่นของผู้ใช้งาน และเลือกฟังก์ชันเสริมสำหรับฐานข้อมูล จากนั้นให้กดปุ่ม “Next”



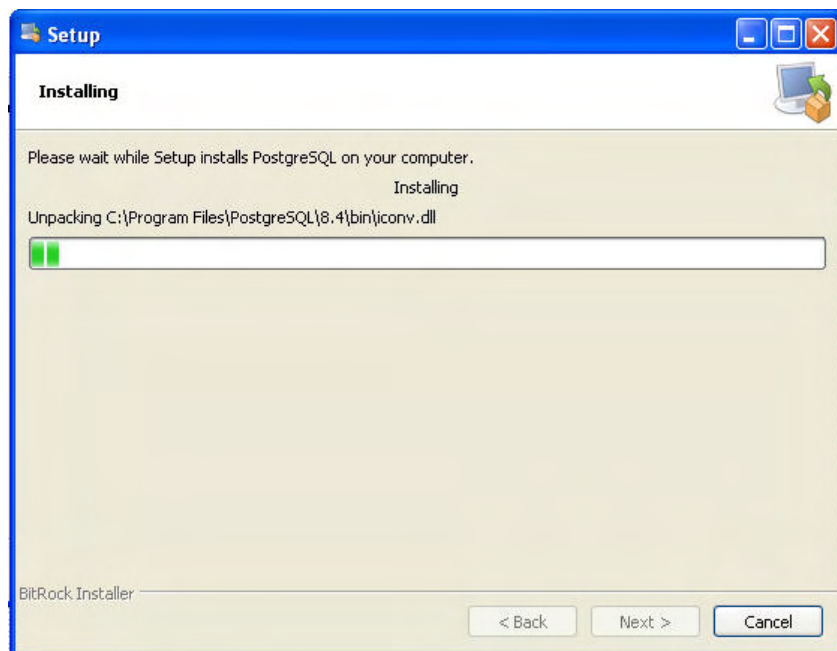
รูปที่ ก.21 ขั้นตอนที่ 5 ในการติดตั้งโปรแกรม postgresql-8.4.6-1

8) เมื่อทำตามขั้นตอนที่ 7 เรียบร้อยแล้ว ขั้นตอนต่อไปเป็นการยืนยันการติดตั้งโปรแกรม ให้กดปุ่ม “Next”



รูปที่ ก.22 ขั้นตอนที่ 6 ในการติดตั้งโปรแกรม postgresql-8.4.6-1

9) จากนั้นจะเริ่มติดตั้งโปรแกรม



รูปที่ ก.23 ขั้นตอนที่ 7 ในการติดตั้งโปรแกรม postgresql-8.4.6-1

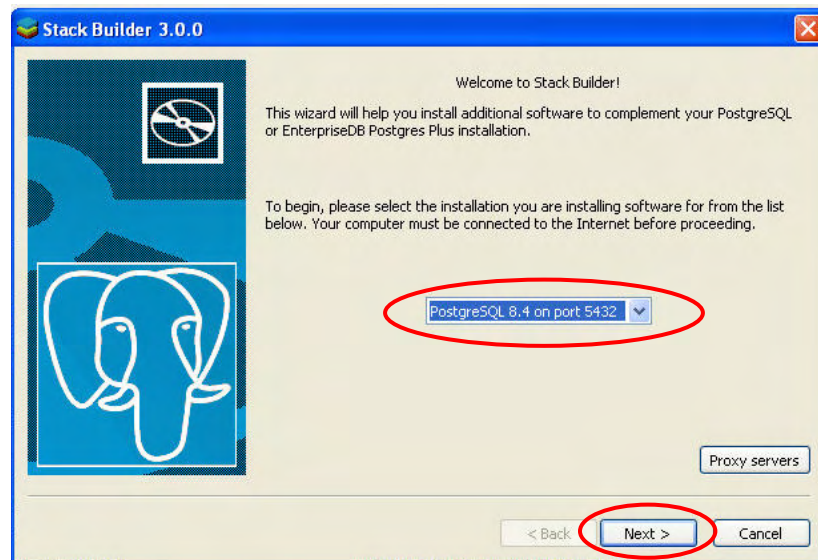
9) เมื่อติดตั้งโปรแกรมเรียบร้อยแล้ว ให้ผู้ใช้งานเลือกที่เช็คบ็อก ☒ Stack Builder เพื่อเริ่มต้นการทำงานของ Stack Builder สำหรับติดตั้งโปรแกรมเสริมของ PostgreSQL แล้วกดปุ่ม "Finish"



รูปที่ ก.24 ขั้นตอนที่ 8 ในการติดตั้งโปรแกรม postgresql-8.4.6-1

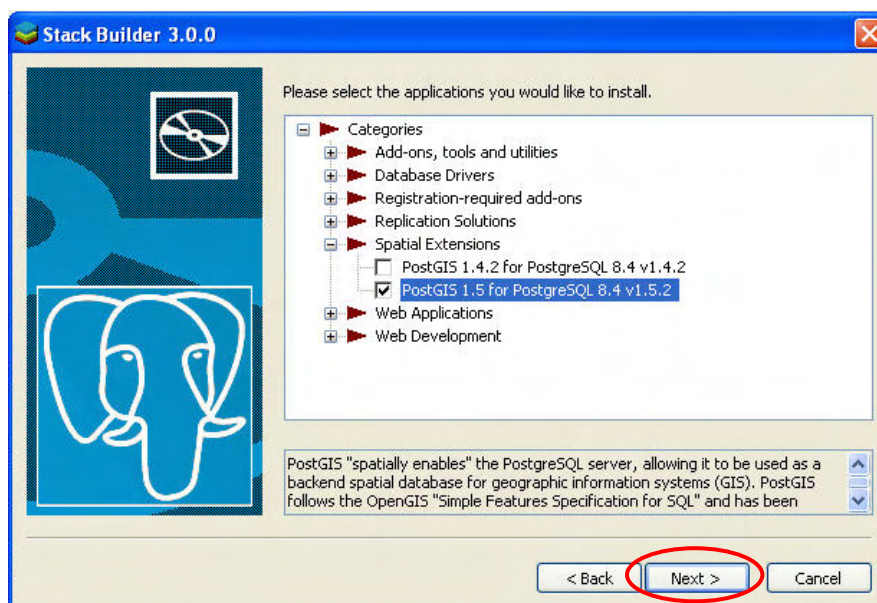
การติดตั้งโปรแกรม PostGIS

1) หลังจากผู้ใช้งาน เลือกเริ่มต้นการทำงานของ Stack Builder ในการติดตั้งโปรแกรม PostgreSQL จะเข้าสู่หน้าจอของ Stack Builder ขั้นตอนนี้ให้ผู้ใช้งานเลือกที่ PostgreSQL 8.4 on port 5432 จากนั้นกดปุ่ม “Next”



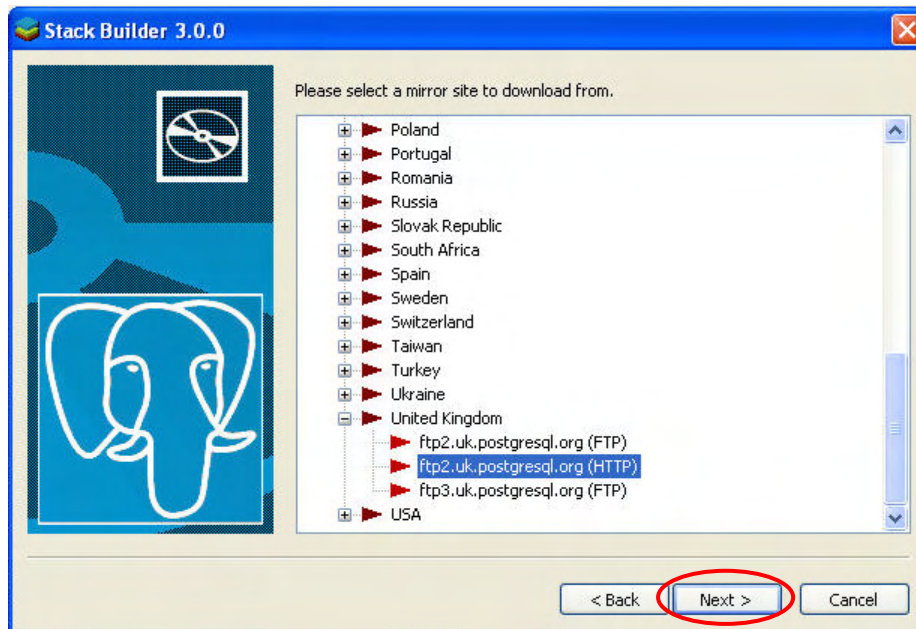
รูปที่ ก.25 ขั้นตอนที่ 1 ในการติดตั้งโปรแกรม PostGIS 1.5

2) เมื่อทำตามขั้นตอนที่ 1 เรียบร้อยแล้ว ขั้นตอนต่อไปให้ผู้ใช้งานเลือกที่ spatial Extensions แล้วเลือก PostGIS 1.5 for PostgreSQL 8.4 v 1.5.2 แล้วกดปุ่ม “Next”



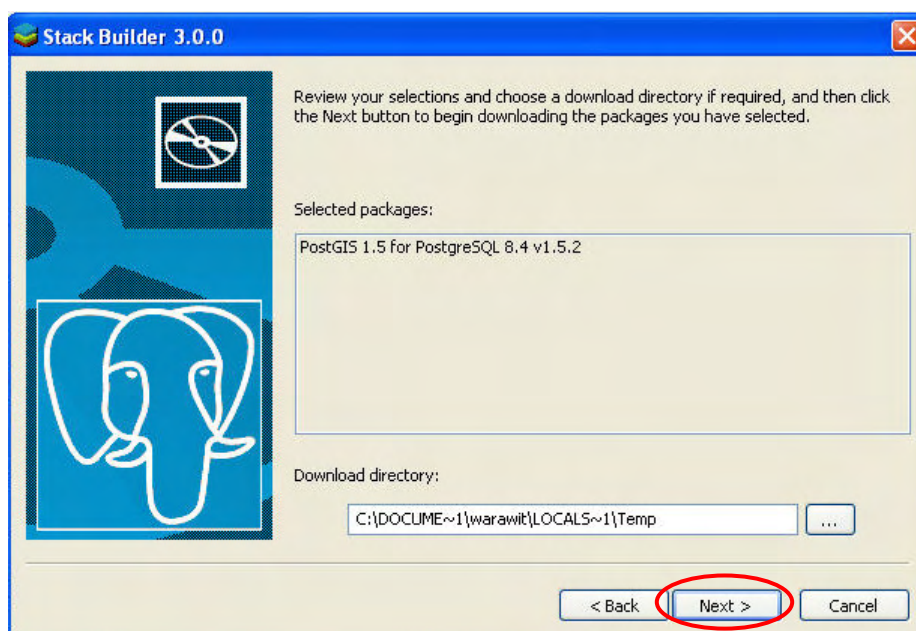
รูปที่ ก.26 ขั้นตอนที่ 2 ในการติดตั้งโปรแกรม PostGIS 1.5

3) เมื่อทำตามขั้นตอนที่ 2 เรียบร้อยแล้ว ขั้นตอนต่อไปให้ผู้ใช้งานเลือกท้องที่สำหรับ Download โปรแกรมสำหรับติดตั้งให้ผู้ใช้งานเลือกที่ United Kingdom จากนั้นเลือกที่ ftp2.uk.postgresql.org (HTTP) แล้วกดปุ่ม “Next”



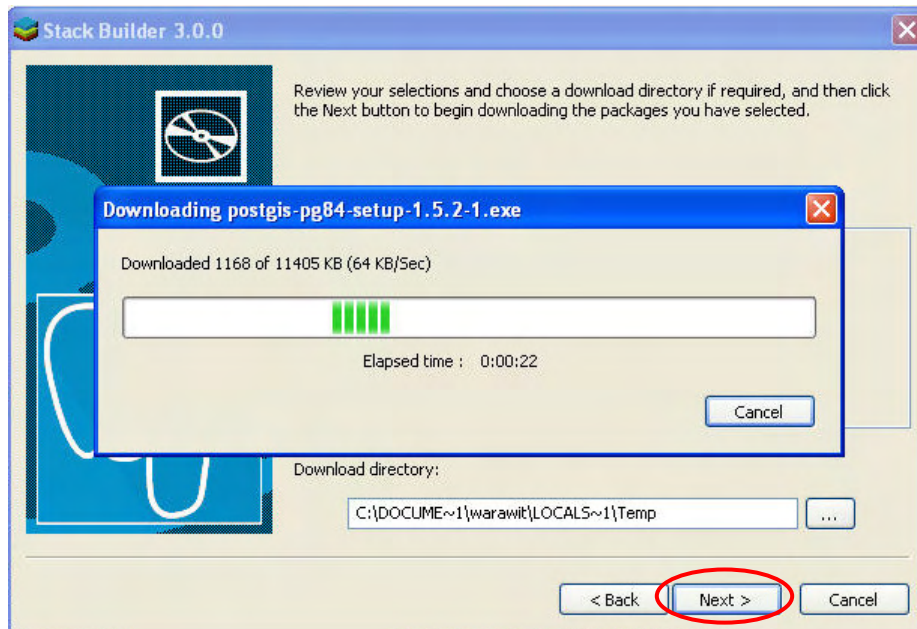
รูปที่ ก.27 ขั้นตอนที่ 3 ในการติดตั้งโปรแกรม PostGIS 1.5

4) เมื่อทำตามขั้นตอนที่ 3 เรียบร้อยแล้ว ขั้นตอนต่อไปให้ผู้ใช้งานเลือกไดเรกทอรีที่ต้องการจัดเก็บโปรแกรม แล้วกดปุ่ม “Next”



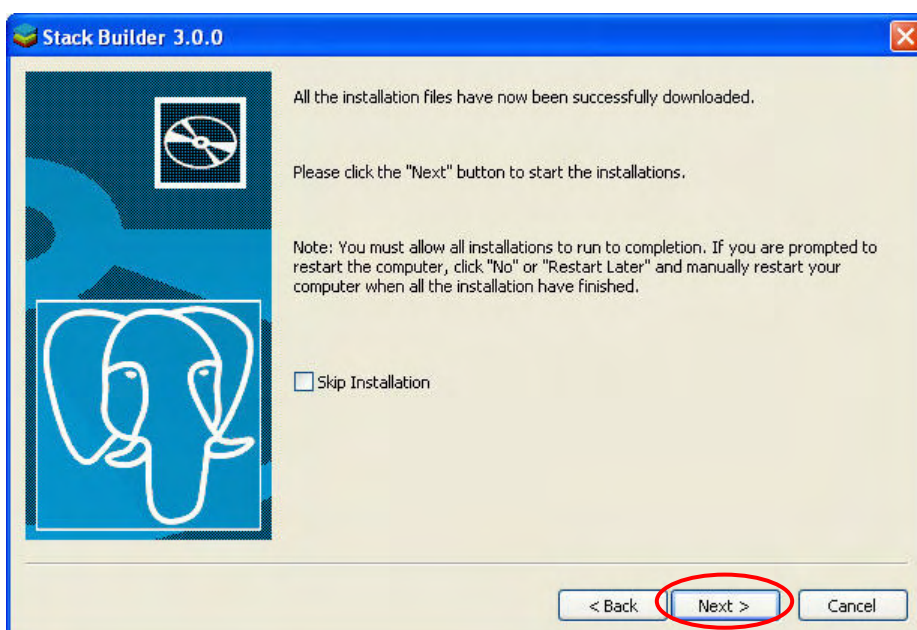
รูปที่ ก.28 ขั้นตอนที่ 4 ในการติดตั้งโปรแกรม PostGIS 1.5

5) เมื่อทำตามขั้นตอนที่ 4 เรียบร้อยแล้ว ขั้นตอนนี้ระบบจะทำการ Download โปรแกรมมาทำการจัดเก็บไว้ในที่ที่ผู้ใช้งานได้กำหนดไว้ ให้ผู้ใช้งานรอนจนกว่าระบบจะทำการ Download โปรแกรมเสร็จสมบูรณ์



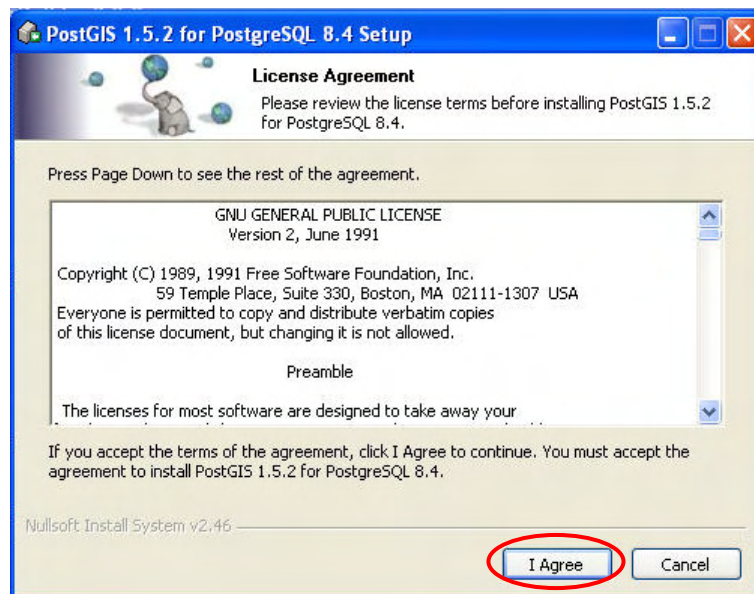
รูปที่ ก.29 ขั้นตอนที่ 5 ในการติดตั้งโปรแกรม PostGIS 1.5

6) เมื่อระบบจะทำการ Download โปรแกรมเสร็จสมบูรณ์แล้ว ให้กดปุ่ม “Next” เพื่อเริ่มติดตั้งโปรแกรม



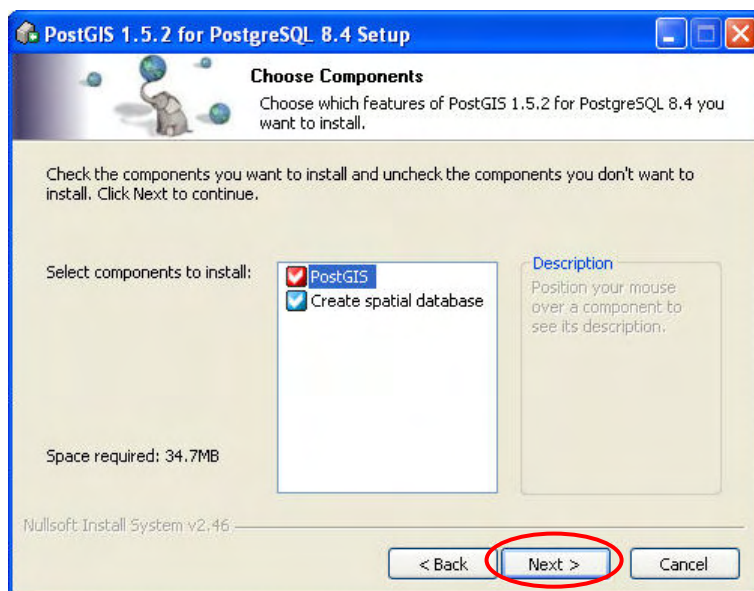
รูปที่ ก.30 ขั้นตอนที่ 6 ในการติดตั้งโปรแกรม PostGIS 1.5

7) เมื่อทำตามขั้นตอนที่ 6 เรียบร้อยแล้ว ขั้นตอนนี้ให้ผู้ใช้งานเลือกที่ปุ่ม “I Agree” เพื่อยอมรับข้อกำหนดของโปรแกรมที่จะติดตั้ง



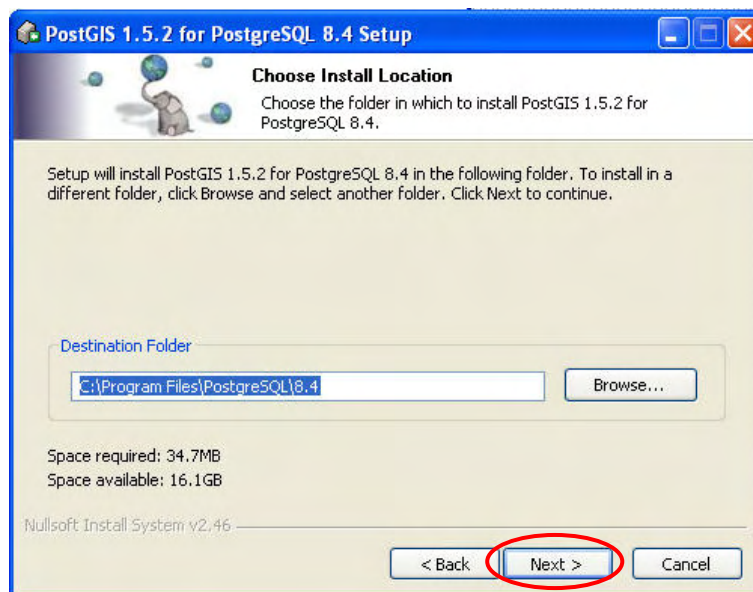
รูปที่ ก.31 ขั้นตอนที่ 7 ในการติดตั้งโปรแกรม PostGIS 1.5

8) เมื่อทำตามขั้นตอนที่ 7 เรียบร้อยแล้ว ขั้นตอนนี้ให้ผู้ใช้งานเลือกโปรแกรมที่ต้องการติดตั้ง ให้ผู้ใช้งานเลือกที่ PostGIS จากนั้นกดปุ่ม “Next”



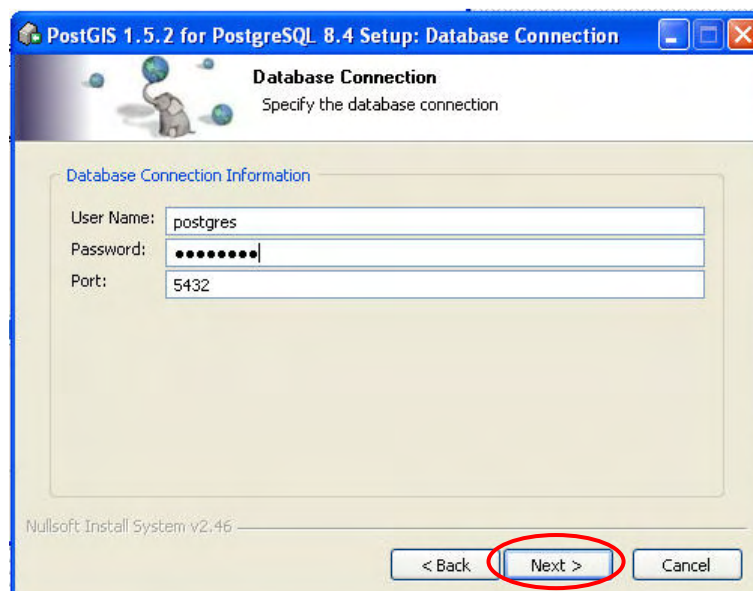
รูปที่ ก.32 ขั้นตอนที่ 8 ในการติดตั้งโปรแกรม PostGIS 1.5

9) เมื่อทำตามขั้นตอนที่ 8 เรียบร้อยแล้ว ขั้นตอนนี้เป็นการเลือกตำแหน่งที่ต้องการติดตั้งโปรแกรม PostGIS ให้ผู้ใช้งานเลือกที่ C:\Program Files\PostgreSQL\8.4 จากนั้นให้กดปุ่ม “Next”



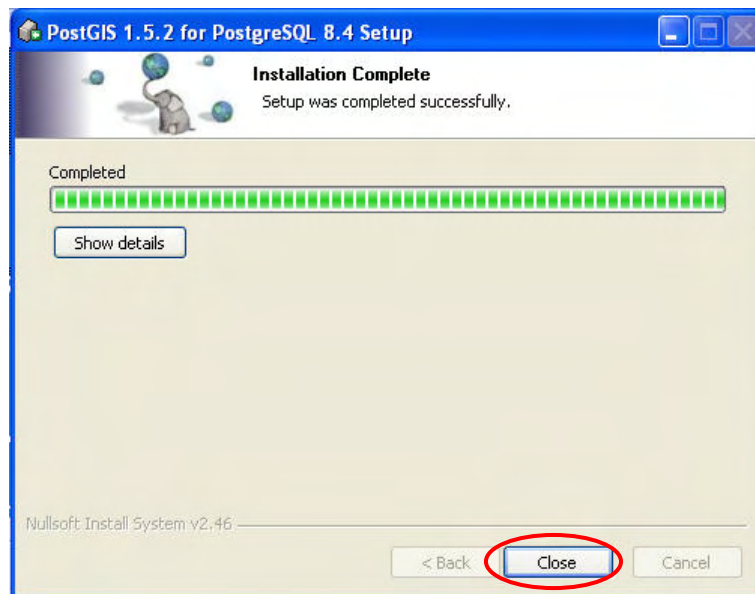
รูปที่ ก.33 ขั้นตอนที่ 9 ในการติดตั้งโปรแกรม PostGIS 1.5

10) เมื่อทำตามขั้นตอนที่ 9 เรียบร้อยแล้ว ขั้นตอนนี้ให้ผู้ใช้งานป้อนข้อมูล ชื่อผู้ใช้งาน รหัสผ่าน และ port สำหรับการเชื่อมโยงข้อมูล จากนั้นให้กดปุ่ม “Next”



รูปที่ ก.34 ขั้นตอนที่ 10 ในการติดตั้งโปรแกรม PostGIS 1.5

11) จากนั้นระบบจะทำการติดตั้งโปรแกรมตามไดเรกทอรีที่กำหนดไว้ เมื่อติดตั้งโปรแกรมเสร็จสมบูรณ์แล้ว ให้ผู้ใช้งานกดปุ่ม “close” เสร็จสิ้นการติดตั้งโปรแกรม



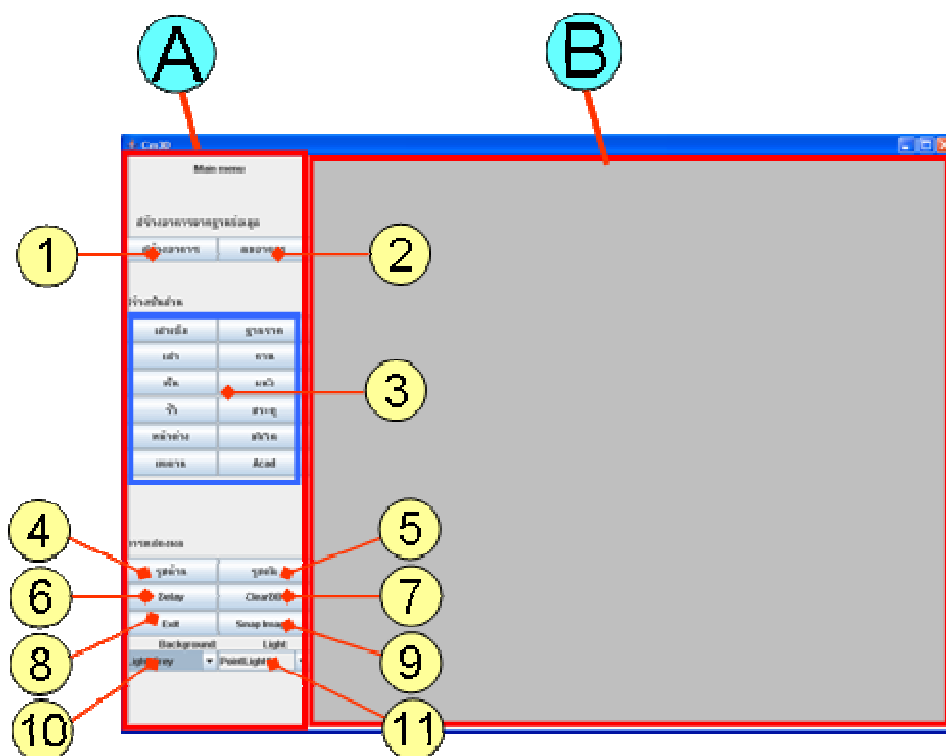
รูปที่ ก.35 ขั้นตอนที่ 11 ในการติดตั้งโปรแกรม PostGIS 1.5

ภาคผนวก ข

คู่มือการใช้งานต้นแบบระบบ

ภาพรวมหน้าจอต้นแบบระบบ

หน้าจอของต้นแบบระบบ (Prototype System) ประกอบด้วย 2 ส่วนหลัก ได้แก่ พื้นที่ส่วนที่ใช้กำหนดการใช้งานของโปรแกรม กับส่วนที่เป็นระนาบ 3 มิติของอาคารสำหรับแสดงผลวัตถุ 3 มิติที่สร้างขึ้น รูปที่ ข.1 แสดงส่วนประกอบหน้าจอของโปรแกรม ซึ่งมีรายละเอียดดังนี้



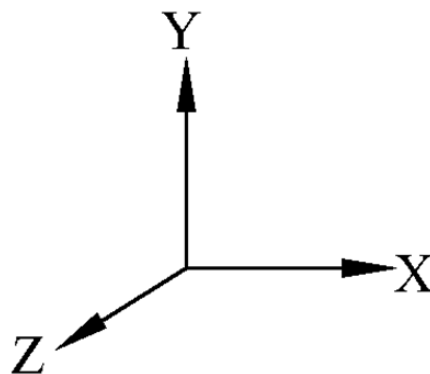
รูปที่ ข.1 แสดงส่วนประกอบของหน้าจอ

- พื้นที่ A เป็นพื้นที่สำหรับควบคุมการทำงานของโปรแกรม ซึ่งประกอบด้วย User interface ต่างๆ ที่ใช้ควบคุมการทำงานของโปรแกรม
- พื้นที่ B เป็นระนาบ 3 มิติสำหรับแสดงผลแบบจำลอง 3 มิติที่สร้างขึ้น
- ปุ่มหมายเลข 1 : ปุ่มสำหรับสร้างอาคารจากข้อมูลในฐานข้อมูลทั้งหมด
- ปุ่มหมายเลข 2 : ปุ่มสำหรับลบอาคารออกจากระนาบ 3 มิติ
- พื้นที่หมายเลข 3 : สำหรับเลือกชนิดของชิ้นส่วนอาคารที่ต้องการสร้างและกำหนดคุณสมบัติของชิ้นส่วน
- ปุ่มหมายเลข 4 : ปุ่มสำหรับกำหนดการแสดงรูปด้านของแบบจำลอง 3 มิติ
- ปุ่มหมายเลข 5 : ปุ่มสำหรับกำหนดการแสดงรูปตัดทางขวางของแบบจำลอง 3 มิติ
- ปุ่มหมายเลข 6 : ปุ่มสำหรับกำหนดการแสดงผลชิ้นส่วนแบบจำลอง 3 มิติ ตามลำดับแผนการก่อสร้างอาคาร

- ปุ่มหมายเลข 7 : ปุ่มสำหรับลบข้อมูลทั้งหมดในฐานข้อมูลเพื่อสร้างฐานข้อมูลใหม่
- ปุ่มหมายเลข 8 : ปุ่มสำหรับจบการทำงานของโปรแกรม
- ปุ่มหมายเลข 9 : ปุ่มสำหรับเก็บภาพแบบจำลอง 3 มิติที่แสดงบนระนาบ 3 มิติเป็นไฟล์ภาพสกุล jpg ไปจัดเก็บยังแฟ้มข้อมูลภาพหนึ่ง
- ปุ่มหมายเลข 10 : Combo Box สำหรับกำหนดสีและภาพพื้นหลังของระนาบ 3 มิติ
- ปุ่มหมายเลข 11 : Combo Box สำหรับกำหนดลักษณะการให้แสงเงา กับแบบจำลอง 3 มิติ

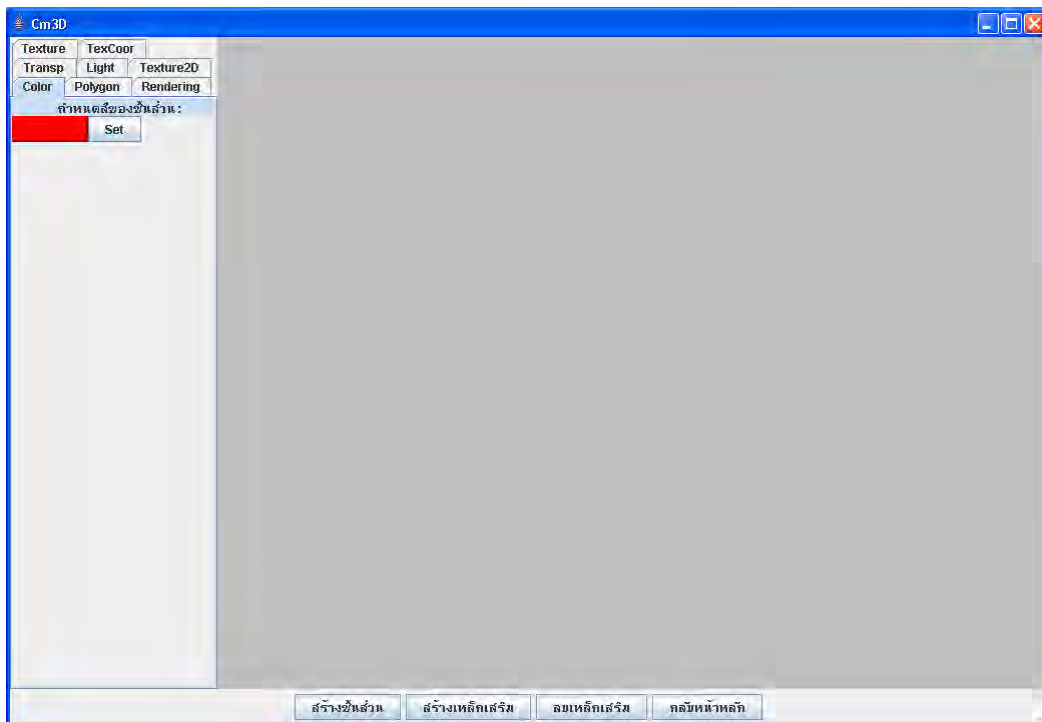
การสร้างชิ้นส่วนของอาคาร

ก่อนที่จะสร้างชิ้นส่วนของอาคารผู้ใช้งานต้องทำความเข้าใจในเรื่องของระบบพิกัด 3 มิติของโปรแกรมเสียก่อน โดยระบบพิกัด 3 มิติในการสร้างและกำหนดตำแหน่งของชิ้นส่วนอาคารในระนาบ 3 มิติ สำหรับโปรแกรมภาษาจาวานั้น มีลักษณะดังรูปที่ ข.2 แสดงระบบพิกัดในระนาบ 3 มิติของโปรแกรมภาษาจาวา โดยแกน X คือ แกนที่พุ่งจากจุดกำเนิดไปในแนวซ้ายขวา โดยค่าที่พุ่งไปทางขวาจะมีค่าเป็นบวก แกน Y คือ แกนที่พุ่งจากจุดกำเนิดไปในแนวด้านบน โดยค่าที่พุ่งขึ้นด้านบนจะมีค่าเป็นบวก และแกน Z คือ แกนที่ตั้งฉากกับแกน X และแกน Y โดยมีทิศทางพุ่งเข้าออกจากจอภาพ ค่าที่พุ่งเข้าหาจอภาพจะมีค่าเป็นบวก



รูปที่ ข.2 ระบบพิกัดในระนาบ 3 มิติของภาษาจาวา

การสร้างชิ้นส่วน 3 มิติของอาคาร เริ่มจากที่หน้าจอหลัก ให้เลือกคลิกที่ปุ่ม บริเวณ “หมายเลข 3” เพื่อเลือกชนิดของชิ้นส่วนตามที่ต้องการสร้าง หลังจากนั้นจะปรากฏหน้าต่างย่อยของการสร้างชิ้นส่วน ให้ผู้ใช้คลิกที่ปุ่ม “สร้างชิ้นส่วน” บริเวณด้านล่างของหน้าจอ รูปที่ ข.3 แสดงหน้าต่างย่อยของการสร้างชิ้นส่วน



รูปที่ ข.3 หน้าต่างย่อยของการสร้างชิ้นส่วน

หลังจากขั้นตอนเบื้องต้นจะปรากฏหน้าต่างสำหรับรับข้อมูลชิ้นส่วน รูปที่ ข.4 แสดงหน้าต่างย่อยสำหรับกำหนดขนาดและตำแหน่งของชิ้นส่วน โดยผู้ใช้งานต้องกำหนดข้อมูลของชิ้นส่วน ได้แก่

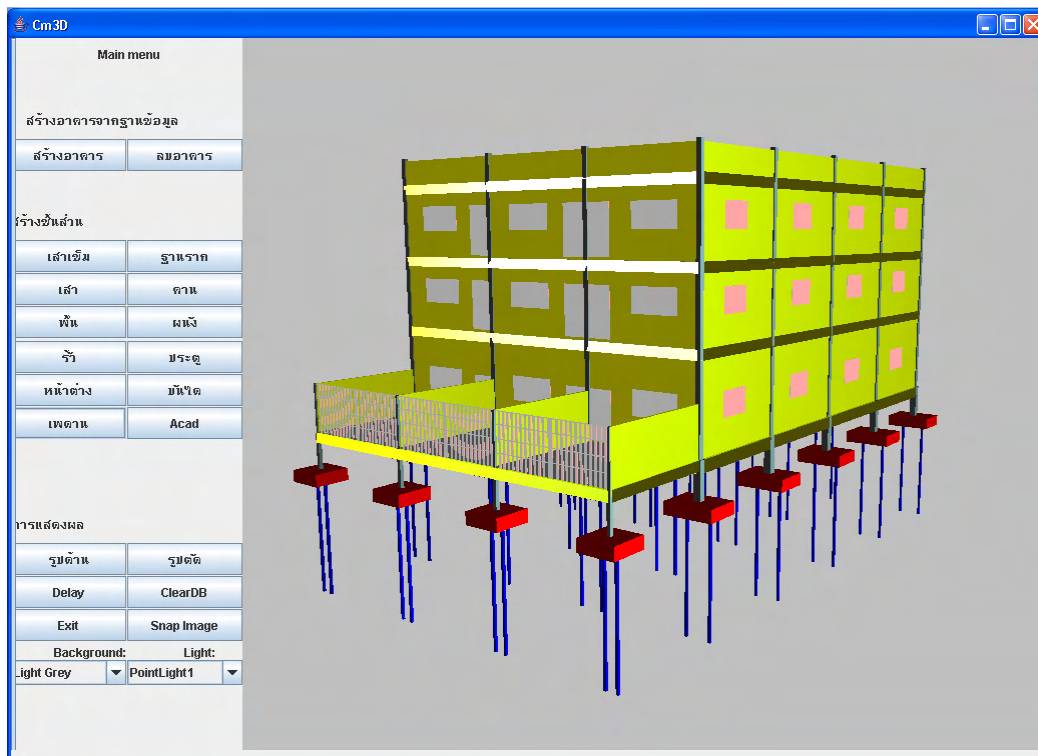
- ข้อมูลหมายเลขของชิ้นส่วน
- ข้อมูลขนาดของชิ้นส่วน ได้แก่ ข้อมูลขนาดความกว้างของหน้าตัด ขนาดความยาวของหน้าตัดและความยาวของเสา
- ข้อมูลพิกัดจุดอ้างอิงของชิ้นส่วน ได้แก่ ข้อมูลพิกัดตามแนวแกน X พิกัดตามแนวแกน Y และข้อมูลพิกัดตามแนวแกน Z ของตำแหน่งจุดอ้างอิง
- ข้อมูลเหล็กเสริมของชิ้นส่วน ได้แก่ ข้อมูลชนิดของเหล็กเสริมหลัก จำนวนของเหล็กเสริมหลัก ขนาดเส้นผ่านศูนย์กลางของเหล็กเสริมหลัก ชนิดของเหล็กปลอกกระยะห่างของเหล็กปลอกและขนาดเส้นผ่านศูนย์กลางของเหล็กปลอก
- ข้อมูลวันที่ทำการก่อสร้างชิ้นส่วนตามแผนการทำงาน

เมื่อกำหนดข้อมูลชิ้นส่วนครบถ้วนแล้วให้คลิกที่ปุ่ม “สร้าง” โปรแกรมจะทำการประมวลผลและสร้างชิ้นส่วน 3 มิติที่กำหนดบนระนาบ 3 มิติ

รูปที่ ข.4 หน้าต่างย่อยสำหรับกำหนดขนาดและตำแหน่งของชิ้นส่วน

การสร้างอาคารจากข้อมูลเดิมในฐานข้อมูลและการลบอาคาร

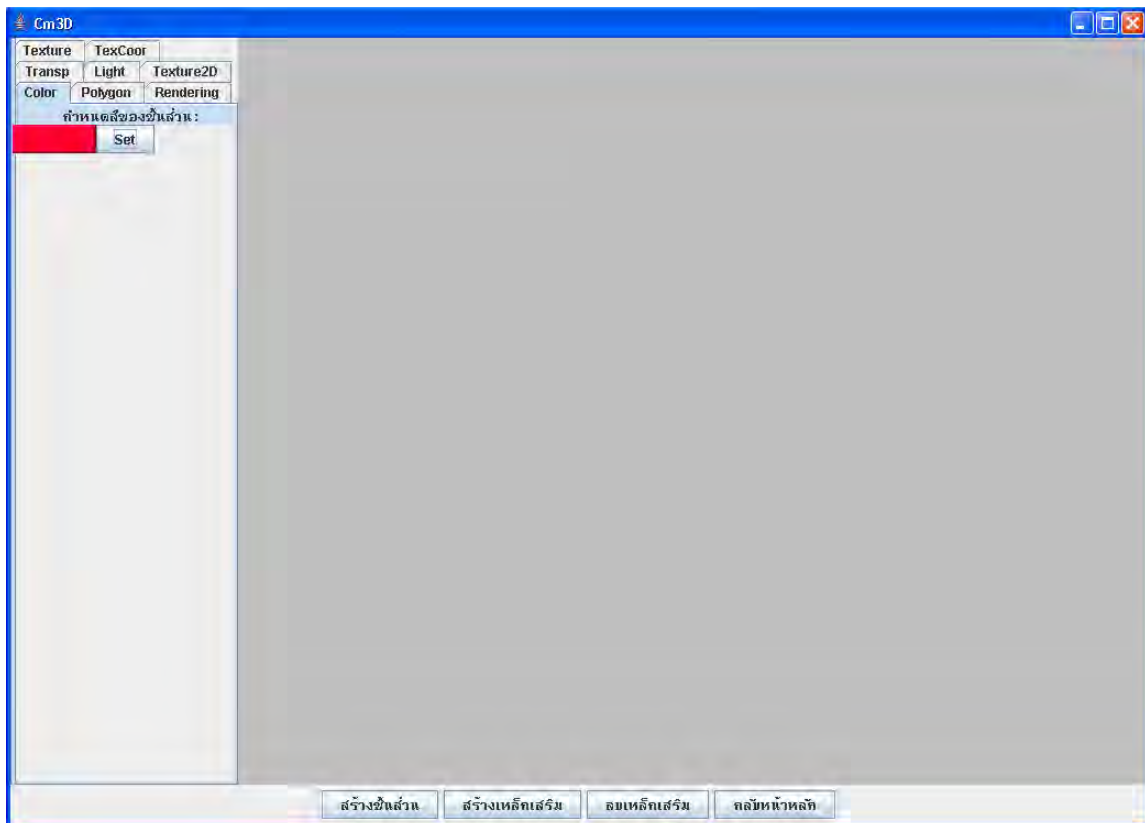
ในกรณีที่มีข้อมูลของอาคารซึ่งจัดเก็บอยู่ในฐานข้อมูล ผู้ใช้งานต้องการสร้างอาคารจากข้อมูลเดิมในฐานข้อมูล ให้ผู้ใช้งานคลิกที่ปุ่ม “สร้างอาคาร” ในหน้าต่างหลักของโปรแกรม จากนั้นโปรแกรมจะประมวลผลและสร้างอาคารแบบจำลอง 3 มิติของอาคารบนระนาบ 3 มิติ รูปที่ ข.5 แสดงการสร้างแบบจำลอง 3 มิติของอาคารที่สร้างจากข้อมูลเดิมในฐานข้อมูล หากต้องการลบอาคารออกจากระนาบ 3 มิติ ให้ผู้ใช้งานคลิกที่ปุ่ม “ลบอาคาร” ที่หน้าต่างหลักของโปรแกรม ชิ้นส่วนของอาคารจะถูกลบออกจากระนาบ 3 มิติ



รูปที่ ข.5 การสร้างแบบจำลอง 3 มิติของอาคารจากข้อมูลเดิมในฐานข้อมูล

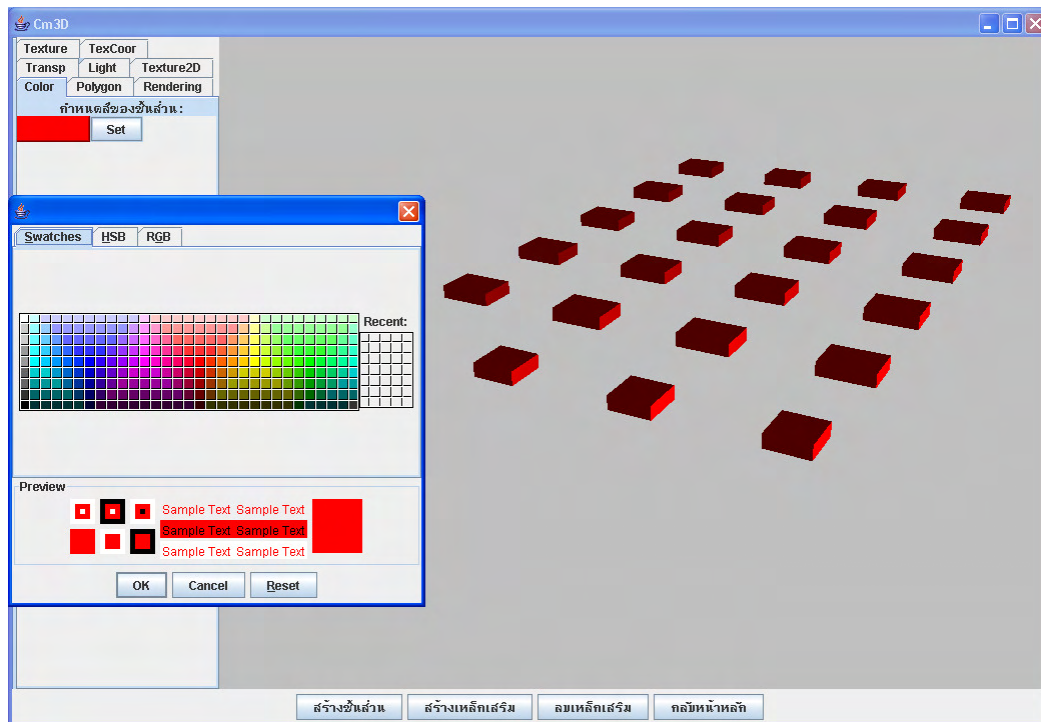
การกำหนดคุณสมบัติของชิ้นส่วน

การกำหนดสีของชิ้นส่วน ประกอบด้วย การกำหนดมุมมอง การกำหนดไม่ให้แสดงผลชิ้นส่วน การกำหนดมุมมองโปรังใส การกำหนดแสง การกำหนดวัสดุพื้นผิว การกำหนด Mode การแสดงผลวัสดุพื้นผิวและการกำหนดความละเอียดของวัสดุพื้นผิว โดยเมื่อเข้ายังหน้าต่างย่อยของชิ้นส่วนจะปรากฏ Tab panel ให้ผู้ใช้งานเลือกว่าต้องการกำหนดคุณสมบัติใด รูปที่ ข.6 แสดงหน้าต่างสำหรับกำหนดคุณสมบัติของชิ้นส่วน ซึ่งมีรายละเอียด ดังต่อไปนี้



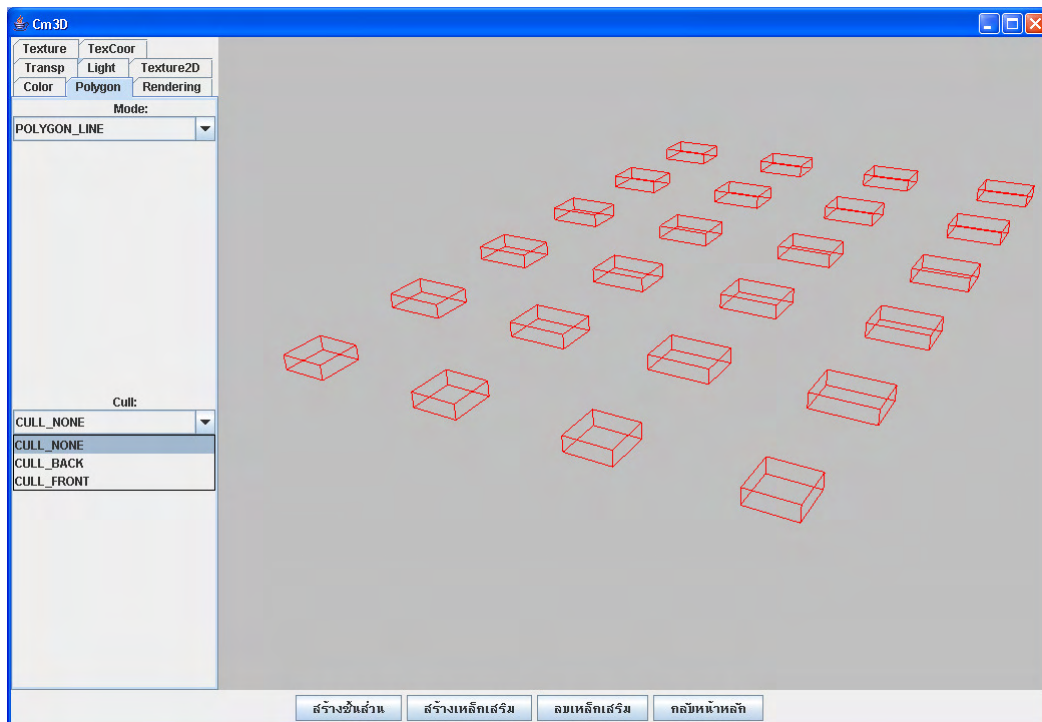
รูปที่ ข.6 หน้าต่างสำหรับกำหนดคุณสมบัติของชิ้นส่วน

การกำหนดสีของชิ้นส่วน เมื่อคลิกที่ Tab “Color” จะเข้ามายัง Tab panel สำหรับกำหนดสีของชิ้นส่วน รูปที่ 2.7 แสดงหน้าต่างสำหรับกำหนดสีของชิ้นส่วน โดยเมื่อคลิกที่ปุ่ม “Set” จะปรากฏหน้าต่างสำหรับกำหนดสีของชิ้นส่วน เมื่อผู้ใช้งานเลือกสีแล้ว ให้คลิกปุ่ม “OK” สีของชิ้นส่วนจะเปลี่ยนไปตามสีที่กำหนด

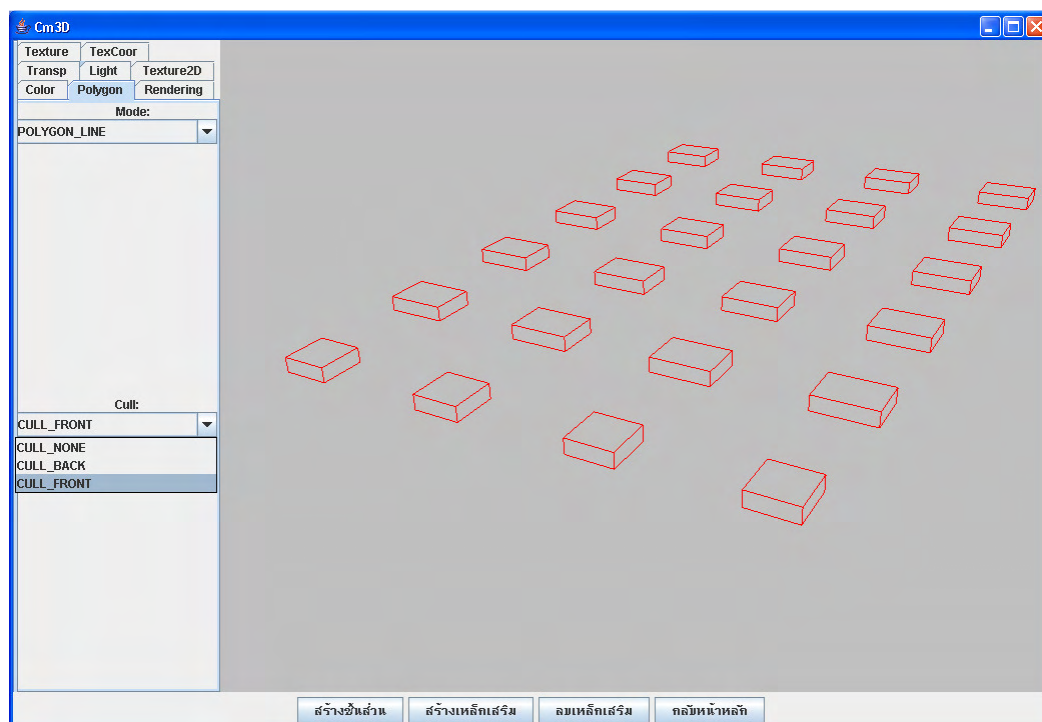


รูปที่ ข.7 หน้าต่างสำหรับกำหนดสีของชิ้นส่วน

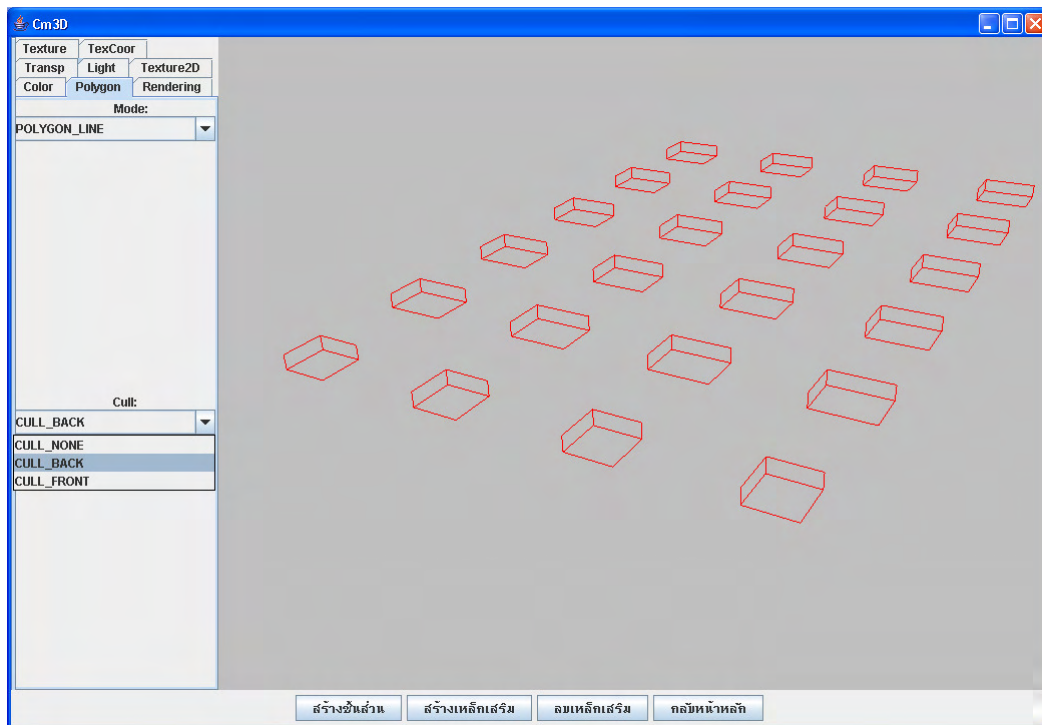
การกำหนดมุมมอง เมื่อคลิกที่ Tab “Polygon” จะเข้ามายัง Tab panel สำหรับกำหนดมุมมองของชิ้นส่วน โดยผู้ใช้งานสามารถเลือก Mode การแสดงผลชิ้นส่วนได้ 3 แบบ ได้แก่ มุมมองแบบแรเงา, มุมมองแบบเส้นโครง และมุมมองแบบจุด โดยเมื่อมุมมองเป็นแบบ Wire Frame หรือมุมมองแบบจุด ผู้ใช้สามารถเลือกการแสดงผลชิ้นส่วน โดยการเลือกที่ Cull รูปที่ ข.8 ถึง ข.10 แสดงการกำหนดมุมมองแบบเส้นโครงของชิ้นส่วนซึ่งสามารถเลือกได้แบบ NONE_CULL, CULL_BACK และCULL_FRONT



รูปที่ ข.8 การกำหนดมุมมองแบบเส้นโครง ของชิ้นส่วน

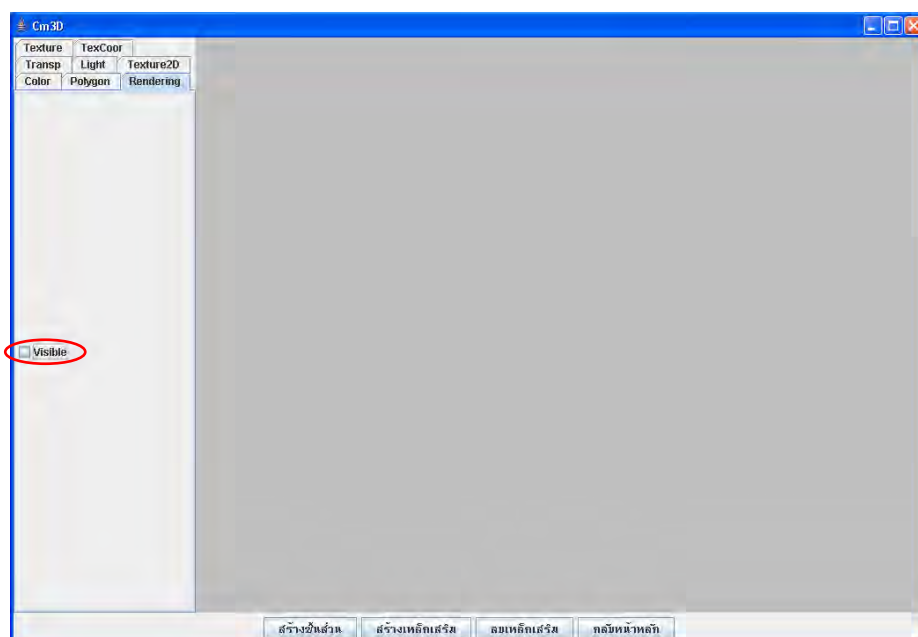


รูปที่ ข.9 การกำหนดมุมมองแบบเส้นโครงและกำหนด Cull เป็นแบบ Cull Front



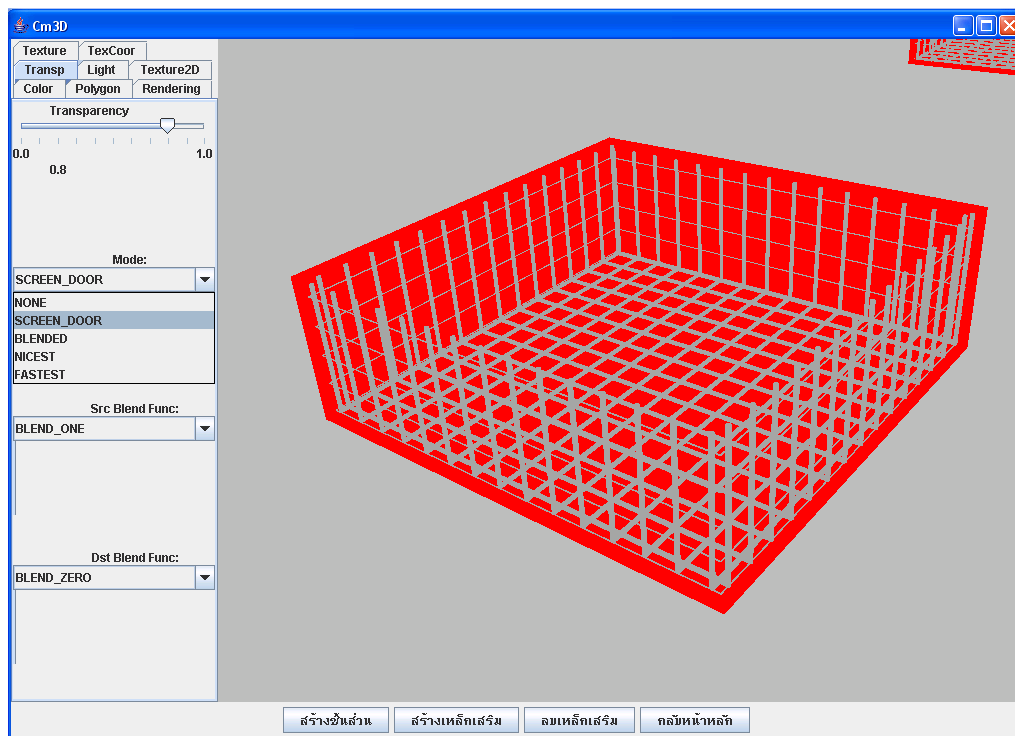
รูปที่ ข.10 การกำหนดมุมมองแบบเส้นโครง และกำหนด Cull เป็นแบบ Cull Back

การกำหนดไม่ให้แสดงผลชิ้นส่วนเมื่อคลิกที่ Tab “Rendering” จะเข้ามายัง Tab panel สำหรับกำหนดไม่ให้แสดงผลชิ้นส่วน รูปที่ ข.11 แสดงหน้าต่างสำหรับกำหนดไม่ให้แสดงผลชิ้นส่วน โดยผู้ใช้งานคลิกเอาเครื่องหมาย “/” ที่ Check box “Visible” ออก เป็นการปิดแสดงผลชิ้นส่วนบนระนาบ 3 มิติ



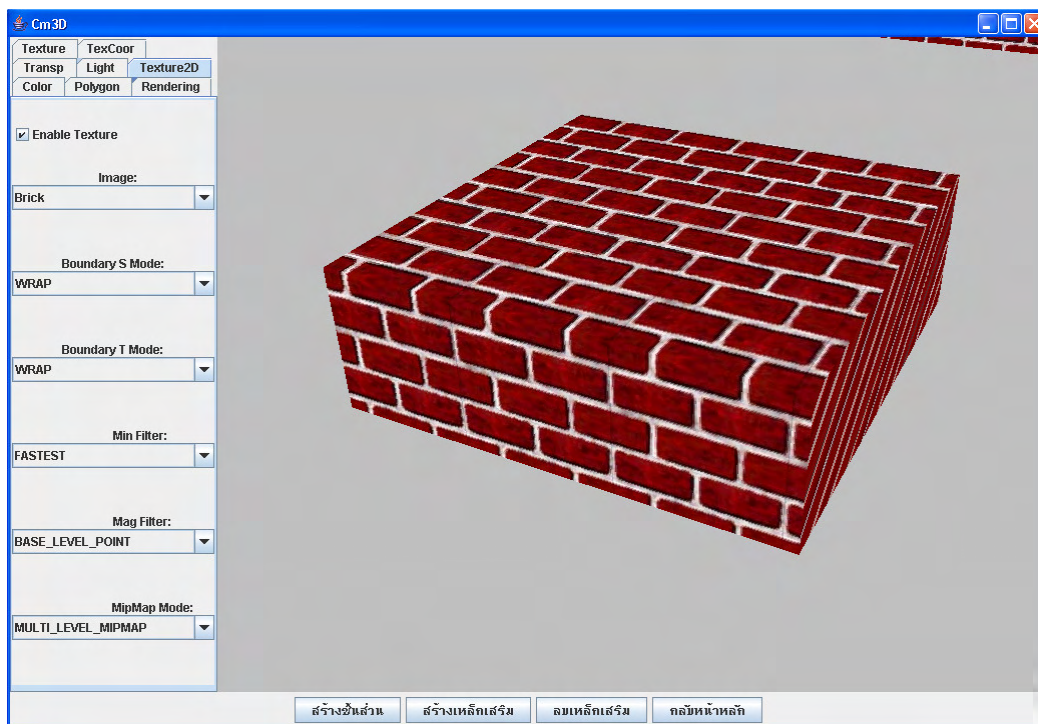
รูปที่ ข.11 การกำหนดไม่ให้แสดงผลชิ้นส่วน

การกำหนดมุมมองโปร่งใสเมื่อคลิกที่ Tab “Transp” จะเข้ามายัง Tab panel สำหรับกำหนดความโปร่งแสงของชิ้นส่วน รูปที่ ข.12 แสดงหน้าต่างสำหรับกำหนดความโปร่งแสงของชิ้นส่วน โดยผู้ใช้งานคลิกเลือก Mode สำหรับกำหนดลักษณะความโปร่งแสง และ Slider bar สำหรับปรับเปอร์เซ็นต์ความโปร่งแสงของชิ้นส่วนได้



รูปที่ ข.12 การกำหนดมุมมองโปร่งใส

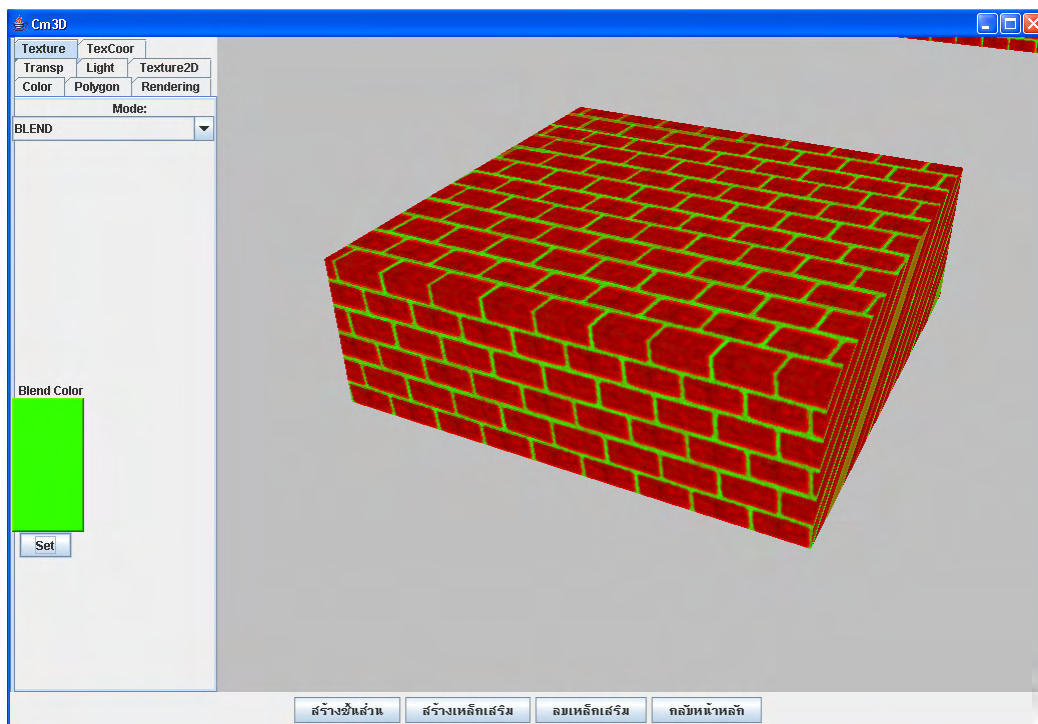
การกำหนดวัสดุพื้นผิว เมื่อคลิกที่ Tab “Texture2D” จะเข้ามายัง Tab panel สำหรับกำหนดวัสดุพื้นผิวของชิ้นส่วน รูปที่ ข.13 แสดงหน้าต่างสำหรับกำหนดวัสดุพื้นผิวของชิ้นส่วน โดยผู้ใช้งานคลิกเลือก ที่ Check box “Enable Texture” จะสามารถกำหนดวัสดุพื้นผิวให้กับชิ้นส่วนได้โดยเลือกชนิดของวัสดุได้ตามรายละเอียดของ Combo box “Image” และสามารถกำหนด Mode การแสดงผลได้ใน 2 ทิศทาง คือทิศทางในแนวนอนกำหนดที่ Combo box “Boundary S Mode” และทิศทางในแนวตั้งกำหนดที่ Combo box “Boundary T Mode”



รูปที่ ข.13 การกำหนดวัสดุพื้นผิวของชิ้นส่วน

การกำหนด Mode การแสดงผลวัสดุพื้นผิวเมื่อคลิกที่ Tab “Texture” จะเข้ามายัง Tab panel สำหรับกำหนด Mode การแสดงผลวัสดุพื้นผิวของชิ้นส่วน รูปที่ ข.14 แสดงหน้าต่างสำหรับกำหนด Mode การแสดงผลวัสดุพื้นผิวของชิ้นส่วน โดยผู้ใช้งานสามารถเลือก Mode สำหรับกำหนดการแสดงผลได้ 4 แบบ ได้แก่

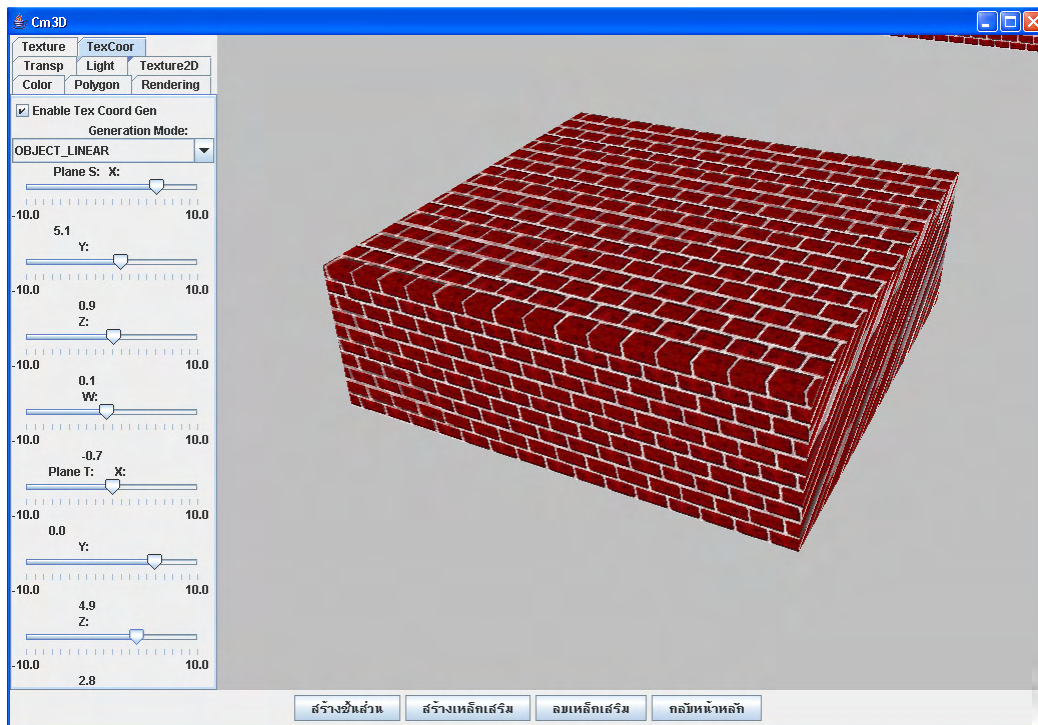
- Mode “Replace” เป็นการนำภาพวัสดุที่เลือกมาแทนที่พื้นผิวโดยตรง
- Mode “Modulate” เป็นการนำสีของแสงที่ผู้ใช้ได้กำหนดค่าไว้มากระทำต่อชิ้นส่วนทำให้เกิดแสงเงา สามารถมองเห็นเส้นขอบของชิ้นส่วนชัดเจน
- Mode “Blend” เป็นการนำเอาสีที่ผู้ใช้เลือกจาก Blend Color มากระทำต่อชิ้นส่วนมาผสมกับสีของแสงเงาที่ผู้ใช้งานกำหนดไว้ ทำให้มีความหลากหลายในการกำหนดสีให้กับชิ้นส่วน



รูปที่ ข.14 หน้าต่างสำหรับกำหนด Mode การแสดงผลวัสดุพื้นผิวของชิ้นส่วน

การกำหนดความละเอียดของวัสดุพื้นผิวเมื่อคลิกที่ Tab “TexCoor” จะเข้ามายัง Tab panel สำหรับกำหนดความละเอียดของวัสดุพื้นผิวของชิ้นส่วน รูปที่ ข.15 แสดงหน้าต่างสำหรับกำหนดความละเอียดของวัสดุพื้นผิวของชิ้นส่วน เมื่อผู้ใช้งานคลิกที่ Check box “Enable text Coord Gen” จะสามารถกำหนดความละเอียดของพื้นผิวได้ โดยผู้ใช้งานสามารถเลือก Generate Mode สำหรับกำหนดการแสดงผลได้ 3 แบบ ได้แก่ Mode “OBJECT_LINEAR”, Mode EYE_LINEAR และ Mode “SPHERE_MAP” การกำหนดความละเอียดของพื้นผิว สามารถกำหนดได้ใน 2 ทิศทาง คือ

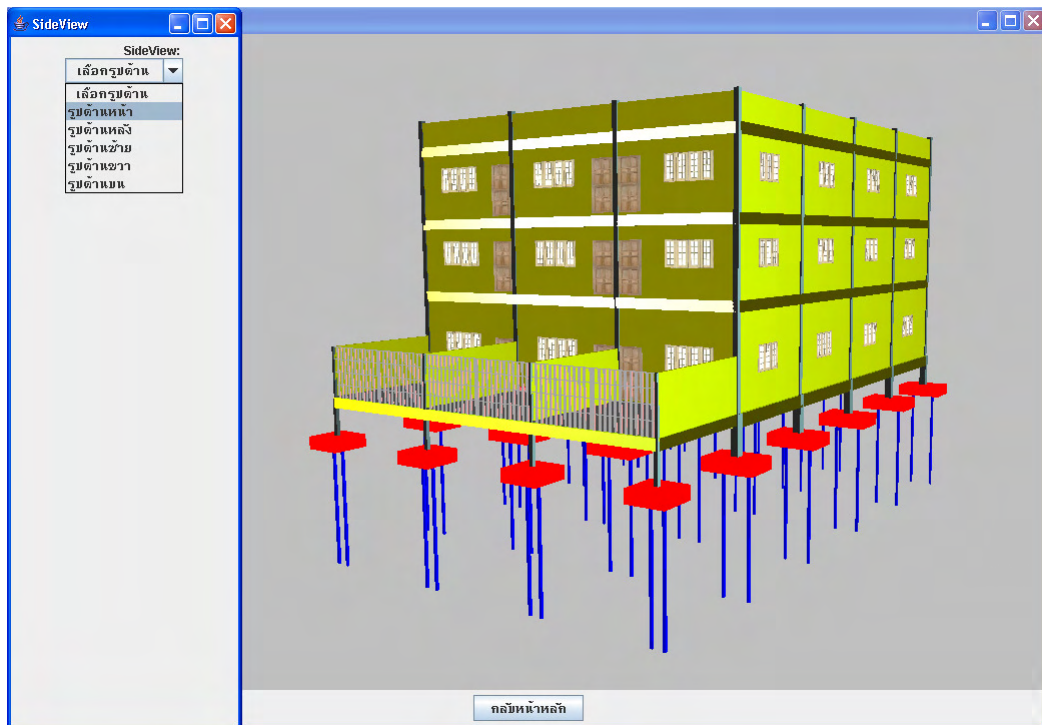
- ทิศทางตามแนวนอน ให้กำหนดในแนว Plane S ซึ่งสามารถกำหนดขนาดความละเอียดตามแนวแกน X, แกน Y, แกน Z และสามารถเลื่อนตำแหน่งของพื้นผิวได้ที่ Slid bar “W”
- ทิศทางตามแนวตั้ง ให้กำหนดในแนว Plane T ซึ่งสามารถกำหนดขนาดความละเอียดตามแนวแกน X, แกน Y, แกน Z และสามารถเลื่อนตำแหน่งของพื้นผิวได้ที่ Slid bar “W” เช่นเดียวกัน



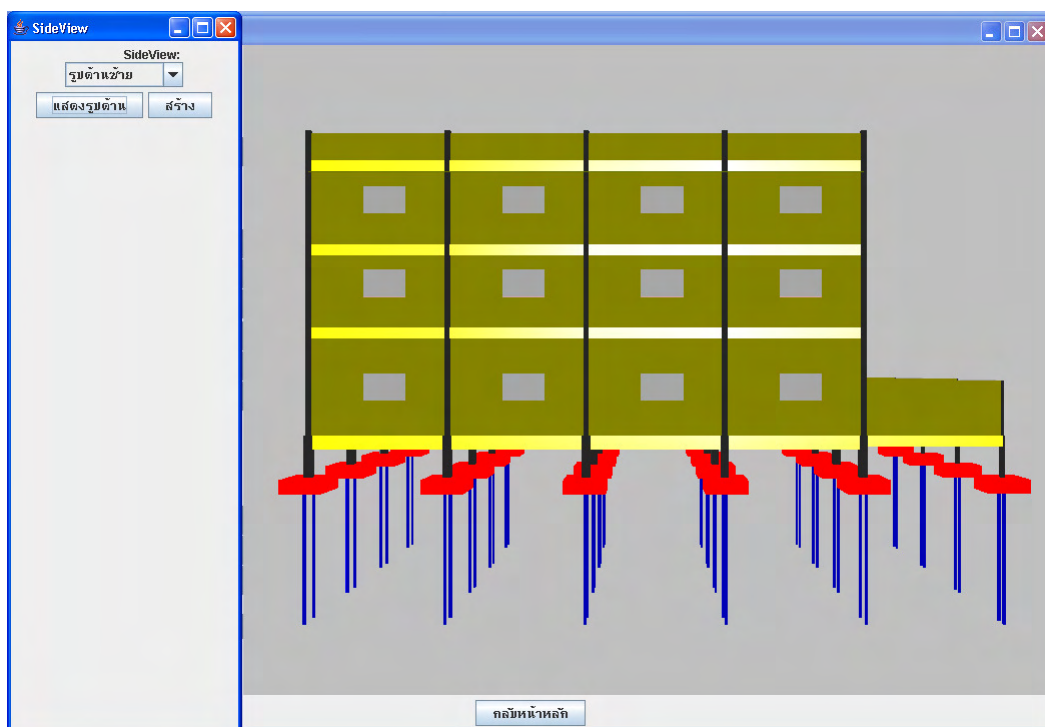
รูปที่ ข.15 แสดงการกำหนดความละเอียดของวัสดุพื้นผิว

การแสดงผลรูปด้านของอาคาร

การแสดงผลรูปด้านของอาคารสามารถทำได้โดยที่หน้าจอหลักของโปรแกรมให้ผู้ใช้คลิกที่ปุ่ม “หมายเลข 4” จากนั้นจะปรากฏหน้าต่างสำหรับเลือกชนิดของรูปด้าน รูปที่ ข.16 แสดงหน้าต่างสำหรับการแสดงผลรูปด้านของอาคาร ให้ผู้ใช้เลือกชนิดของรูปด้านที่ต้องการพิจารณา ซึ่งประกอบด้วย รูปด้านหน้า รูปด้านหลัง รูปด้านซ้าย รูปด้านขวาและรูปด้านบน จากนั้นจะปรากฏปุ่มสำหรับกำหนดให้แสดงรูปด้านและปุ่มสร้างสำหรับยกเลิกการแสดงผลรูปด้าน เมื่อผู้ใช้คลิกที่ปุ่ม “แสดงรูปด้าน” จะปรากฏรูปด้านของอาคารตาม que ผู้ใช้เลือกพิจารณาในระนาบ 3 มิติ พร้อมทั้งปรากฏ Panel สำหรับควบคุมตำแหน่งของอาคาร รูปที่ ข.17 แสดงรูปด้านซ้ายของอาคาร เมื่อผู้ใช้คลิกที่ปุ่ม “สร้าง” จะเป็นการยกเลิกการแสดงผลรูปด้านของอาคาร



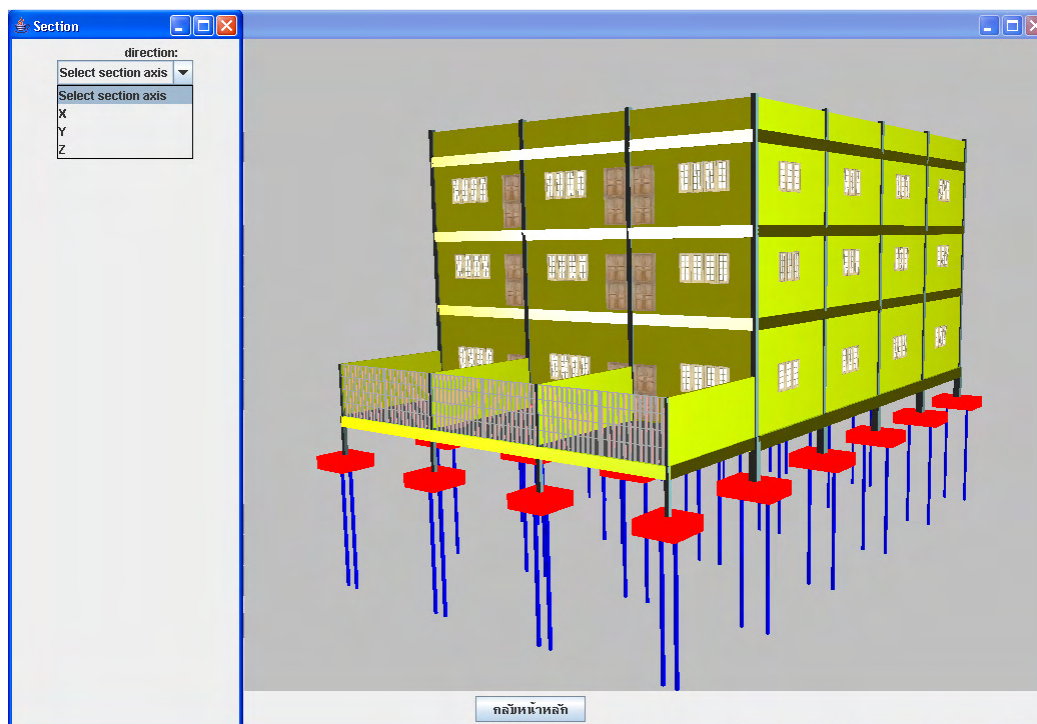
รูปที่ ข.16 หน้าต่างสำหรับการแสดงผลรูปด้านของอาคาร



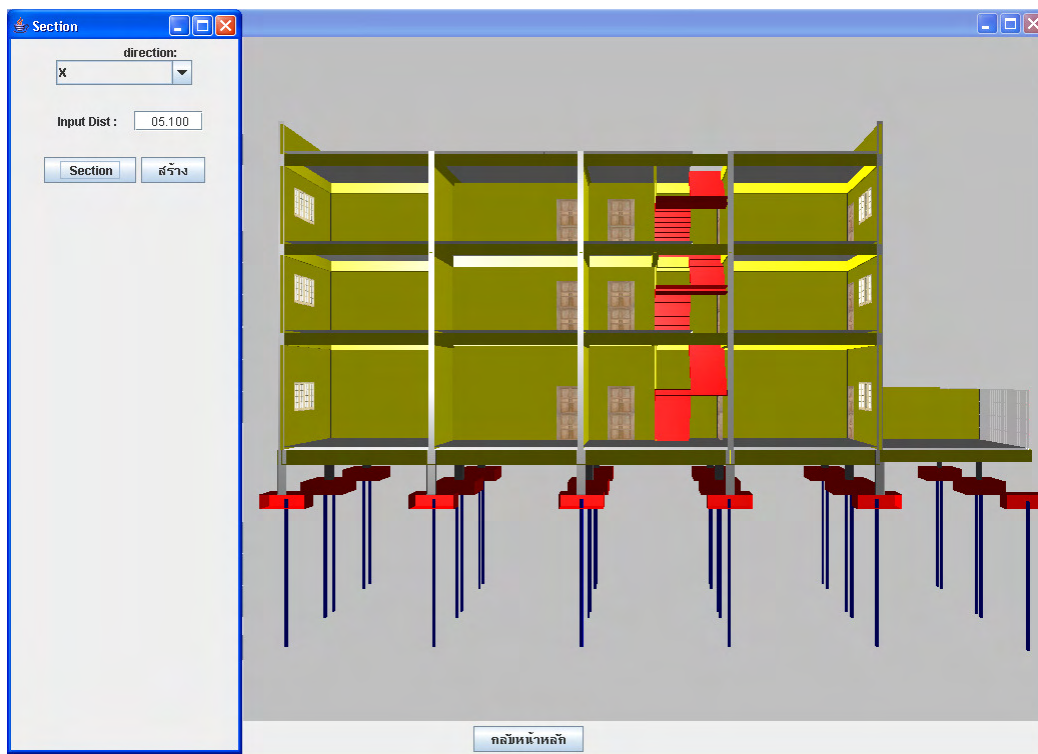
รูปที่ ข.17 รูปด้านซ้ายของอาคาร

การแสดงผลรูปตัดทางขวางของอาคาร

การแสดงผลรูปตัดทางขวางของอาคารสามารถทำได้โดยที่หน้าจอหลักของโปรแกรมให้ผู้ใช้คลิกที่ปุ่ม “หมายเลข 5” จากนั้นจะปรากฏหน้าต่างสำหรับเลือกชนิดของรูปตัดทางขวาง รูปที่ ข.18 แสดงหน้าต่างสำหรับกำหนดค่าเพื่อสร้างรูปตัดทางขวางของอาคาร ให้ผู้ใช้เลือกชนิดของรูปตัดทางขวางด้านที่ต้องการพิจารณา ซึ่งประกอบด้วย รูปตัดตามแนวแกน X รูปตัดตามแนวแกน Y และรูปตัดตามแนวแกน Z จากนั้นจะปรากฏ Input box สำหรับรับข้อมูลระยะห่างจากตำแหน่ง +0.00 ไปยังระนาบที่ต้องการตัดอาคาร ให้ผู้ใช้ป้อนข้อมูลและเมื่อคลิกที่ปุ่ม “Section” จะปรากฏรูปตัดทางขวางของอาคารตาม que ผู้ใช้กำหนดค่า เมื่อคลิกที่ปุ่ม “สร้าง” จะเป็นการยกเลิกการแสดงผลรูปตัดทางขวางของอาคาร รูปที่ ข.19 แสดงรูปตัดทางขวางตามแกน X ของอาคาร



รูปที่ ข.18 หน้าต่างสำหรับกำหนดค่าเพื่อสร้างรูปตัดทางขวางของอาคาร



รูปที่ ข.19 รูปตัดทางขวางตามแกน X ของอาคาร

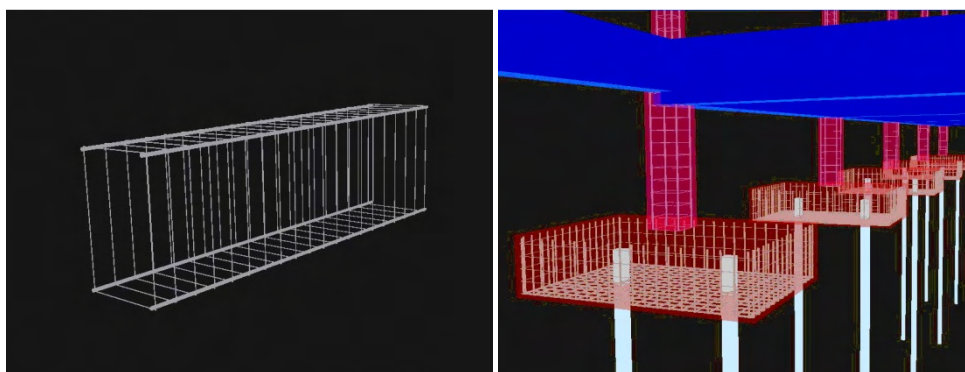
การแสดงผลอาคารตามลำดับการวางแผนการก่อสร้าง

การแสดงผลอาคารตามลำดับการวางแผนการก่อสร้างเป็นการกำหนดให้ขึ้นส่วนแต่ละชั้นส่วนแสดงผลขึ้นในระนาบ 3 มิติตามลำดับการวางแผนการทำงาน โดยผู้ใช้งานสามารถกำหนดวันที่จะทำการก่อสร้างขึ้นส่วนแต่ละชั้นได้โดยการกำหนดวันก่อสร้างไปพร้อมกับการสร้างขึ้นส่วนอาคารชั้นนั้นๆ หรือจะกำหนดวันที่ทำการก่อสร้างขึ้นส่วนในตารางของฐานข้อมูลของโปรแกรม MS Access โดยตรง โดยที่หน้าจอหลักของโปรแกรมให้ผู้ใช้งานคลิกที่ปุ่ม “หมายเลข 6” จากนั้นโปรแกรมจะลบแบบจำลอง 3 มิติของอาคารทั้งหมดออกจากระนาบ 3 มิติ แล้วจะเริ่มแสดงผลขึ้นส่วนของอาคารขึ้นมาใหม่ตามลำดับของแผนการทำงานที่ได้กำหนดไว้

การกำหนดมุมมองของอาคารโดยใช้เมาส์

- การเลื่อนตำแหน่งวัตถุในระนาบ 3 มิติ ให้คลิกขวาที่เมาส์ค้างไว้ จะสามารถเลื่อนวัตถุไปยังตำแหน่งที่ต้องการพิจารณาได้
- การ “Zoom” ไปยังตำแหน่งที่ต้องการพิจารณา ให้กดปุ่ม “Alt” ที่คีย์บอร์ดและคลิกซ้ายที่เมาส์ค้างไว้ หากเลื่อนตำแหน่งของเมาส์ไปทางขวามือจะสามารถทำการ “Zoom in ” เข้าไปยังตำแหน่งที่ต้องการพิจารณา แต่หากเลื่อนตำแหน่งของเมาส์ไปทางซ้ายมือจะเป็นการ “Zoom out ” ออกจากวัตถุ
- การหมุนวัตถุในระนาบ 3 มิติ ให้คลิกซ้ายที่เมาส์ค้างไว้ แล้วเลื่อนตำแหน่งของเมาส์ วัตถุจะหมุนไปในทิศทางที่ต้องการ

รูปที่ ข.20 แสดงตัวอย่างการใช้เมาส์ “Zoom” เข้าไปยังตำแหน่งที่ต้องการพิจารณาโดยสามารถ “Zoom” เข้าไปยังตำแหน่งของเหล็กเสริมและตำแหน่งของฐานรากเพื่อพิจารณารายละเอียดของชิ้นส่วน



รูปที่ ข.20 การใช้เมาส์ Zoom เข้าไปยังตำแหน่งที่ต้องการพิจารณา

การจัดเก็บภาพอาคาร

ที่หน้าต่างหลักของโปรแกรม ให้ผู้ใช้งาน คลิกที่ปุ่ม “หมายเลข 7” โปรแกรมจะทำการบันทึกภาพของอาคารที่ปรากฏในระนาบ 3 มิติ เป็นไฟล์ภาพสกุล jpg ไปจัดเก็บในแฟ้มข้อมูลภาพหนึ่งของโปรแกรม

การสร้างฐานข้อมูลใหม่

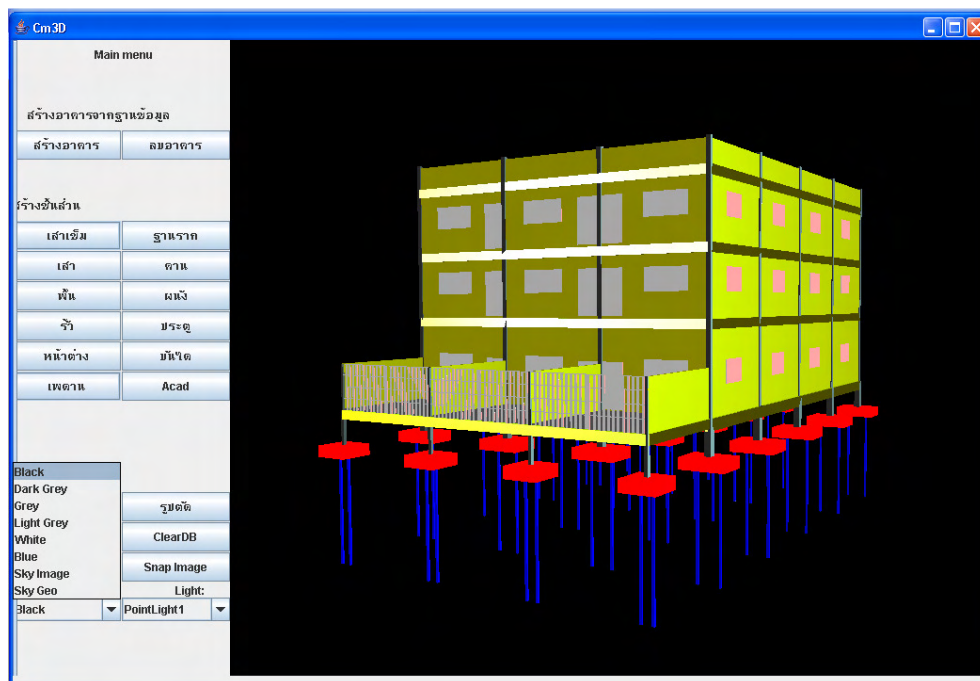
การสร้างฐานข้อมูลใหม่ ให้ผู้ใช้งานคัดลอกไฟล์ฐานข้อมูลของโปรแกรมไปจัดเก็บไว้ก่อน จากนั้นที่หน้าต่างหลักของโปรแกรม ให้ผู้ใช้งานคลิกที่ปุ่ม “หมายเลข 8” โปรแกรมจะทำการลบข้อมูลทั้งหมดในฐานข้อมูล ไฟล์ฐานข้อมูลจะเป็นไฟล์ที่ว่างเปล่า ผู้ใช้งานสามารถสร้างอาคารหลังใหม่ได้ตามปกติ

การออกจากโปรแกรม

การออกจากโปรแกรม ให้ผู้ใช้งานคลิกที่ปุ่ม “หมายเลข 9” ที่หน้าต่างหลักของโปรแกรม จากนั้นโปรแกรมจะสิ้นสุดการทำงานและออกจากโปรแกรมตามลำดับ

การกำหนดสีและภาพพื้นหลังของระนาบ 3 มิติ

การกำหนดสีและภาพพื้นหลังของระนาบ 3 มิติให้ผู้ใช้งานคลิกที่ Combo Box “หมายเลข 10” ที่หน้าต่างหลักของโปรแกรม จากนั้นซึ่งจะมีรายละเอียดสีและภาพของพื้นหลังให้ผู้ใช้งานสามารถเลือกกำหนดได้ตามที่ต้องการ เมื่อเลือกแล้วพื้นหลังของระนาบ 3 มิติจะเปลี่ยนไปตามที่ผู้ใช้งานกำหนด รูปที่ ข.21 แสดงการกำหนดสีพื้นหลังของระนาบ 3 มิติให้เป็นสีดำ



รูปที่ ข.21 การกำหนดสีพื้นหลังของระนาบ 3 มิติให้เป็นสีดำ

การกำหนดแสงเงาของแบบจำลอง 3 มิติ

การกำหนดแสงเงาของแบบจำลอง 3 มิติ ให้ผู้ใช้งานคลิกที่ Combo Box “หมายเลข 11” ที่หน้าต่างหลักของโปรแกรม จากนั้นซึ่งจะมีรายละเอียดลักษณะของแสงเงา ให้ผู้ใช้งานสามารถเลือกกำหนดได้ตามที่ต้องการ เมื่อเลือกแล้วลักษณะของแสงเงาของแบบจำลอง 3 มิติ จะเปลี่ยนไปตามที่ผู้ใช้งานกำหนด

ภาคผนวก ค

ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติ

Integration of java-based BIM with spatial database

S. Malaikrisanachalee^{1,*}, H. Vathananukij²

Received: October 2009 , Accepted: May 2010

Abstract

Java is an object-oriented program that has abundant open-source libraries for application development and 3D model rendering. Spatial database is the database that can efficiently store and manage geographic information data through various spatial data management techniques. This paper explores the rationale of coupling java with spatial database to develop an effective platform for future Building Information Modeling (BIM) application. The paper methodically presents the prototype system integration design to demonstrate how the system can be developed. The paper also meticulously presents the logical and physical data models in designing optimum BIM database for a reinforced concrete building. An 8-storey reinforced concrete building was used as an implementation case study to validate the proposed prototype system design and investigates the implementation issues. The outcome shows that not only the proposed prototype system offers technological advantages over the traditional BIM applications, its open-source solution can also overcome the financial constraint that currently inhibits the implementation of BIM especially for medium and small enterprises.

Keywords: BIM, spatial database, project management, construction database, 3D model.

1. Potentials of Java-based BIM

Building Information Modeling (BIM) has garnered momentous attention from the architecture, engineering and construction (AEC) industry in recent years since BIM potentially can minimize project time, cost and conflict as well as improve project communication, coordination and simulation [1,2]. BIM lies on the basis of using a three-dimensional (3D) digital model for managing spatial, aspatial and temporal information of building elements to support decision makings in project analysis, design, planning, construction and monitoring throughout the project life cycle. Java provides a competent platform for BIM application development as it comes with abundant open-source libraries and operational graphical user interface classes that offer array of solutions for application development. Java is also equipped with the high-level 3D graphic application programming interface that provides building blocks for composing and animating 3D objects, which can ease the development process for 3D visualization and animation in

BIM application [3,4]. In addition, java generates binary executable that can run on any operating system and hardware platform, which enables the application to run on any computer without restriction.

Since java is an object-oriented program, it supports modeling of real-world building elements where spatial, aspatial and temporal information of building elements can be efficiently encapsulated by the notion of class, object and method. Furthermore, the object-oriented approach potentially provides a means for establishing relationship constraints between building elements. For example, relationship constraints can be established between slabs, beams and columns where slabs must always be connected to their supporting beams and beams must always be connected to their supporting columns. This makes the object-oriented 3D model logically more realistic as opposed to the traditional non-object 3D model, whose building elements are simply represented by simple geometric objects (e.g. point, line, polygon and box) without associated information of real-world building elements.

Through java applet or servlet technology, java potentially can support the implementation of web-based BIM and allow effective integration between BIM and an enterprise project management system in order to allow project participants to easily access building information and closely monitor project status through the Internet. In addition, with the availability of

* Corresponding Author: fengshm@ku.ac.th

¹ Assistant Professor, Faculty of Engineering, Kasetsart University, 50 Ngam Wong Wan Rd., Chatuchak, Bangkok 10900

² Associate Professor, Faculty of Engineering, Kasetsart University, 50 Ngam Wong Wan Rd., Chatuchak, Bangkok 10900

java-based 3D geospatial viewer technology such as Nasa's World Wind Java, it is possible to integrate java-based BIM with Geographic Information System (GIS) to enhance the application potential to the like of city information modeling [5].

2. Prototype system integration design

Spatial database management system can serve as a good data management platform for BIM as it offers an efficient means for managing and analyzing building geometry data where size, shape and position of building elements can be efficiently encoded, processed and retrieved through spatial data management techniques. Figure 1 illustrates an overview of system integration between java-based BIM with spatial database where geometry data of building elements as well as their related attributes are manipulated by PostgreSQL, the open-source object-relational database management system, and PostGIS, the geospatial extension to PostgreSQL that allows spatial data to be stored inside PostgreSQL database.

Integrating spatial database with BIM can greatly improve flexibility for BIM data modeling and data query performance. Since PostGIS supports various types of 2D, 3D and 4D spatial objects as summarized in Table 1 [6], it provides superior modeling platform for spatio-temporal data of building elements. Furthermore, PostGIS can store spatial data in binary format, known as Well Known Binary (WKB), which needs less data storage and yields less data processing time than the traditional ASCII-based coordinate strings. In addition, PostGIS includes support for spatial index called Generalized Search Trees (GiST), which can be used to drastically reduce query time for a large BIM database.

PostGIS includes support for a number of spatial queries such

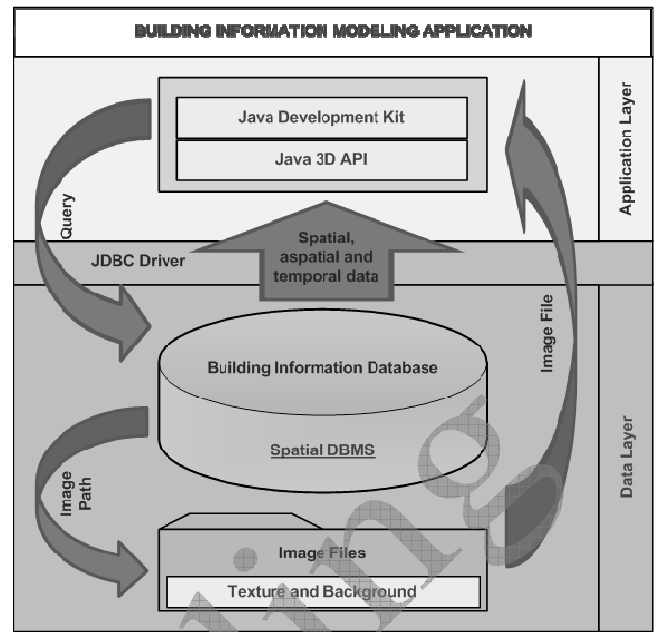


Fig. 1. Prototype System Integration Design

as distance, length, boundary, centroid and area calculations that can serve as useful tools for quantity takeoff in estimating building cost. PostGIS also provides spatial queries for examining spatial relation between building elements including union, intersection, equal, touch, contain, within and buffer operations. These queries can be grateful for verification of building design conflicts as well as evaluation of topological relationship between building elements.

The 3D model application interface includes the standard Java Development Kit (JDK), java 3D API (javax.media.j3d), java 3D utility (com.sun.j3d.utils), java 3D vector math (javax.vecmath) and java abstract window toolkit (AWT) (java.awt) [7]. The java 3D API, widely known as java 3D core, is the minimum requirement for creating, rendering and manipulating 3D objects while the java 3D utility compliments the java 3D core to ease the complication of java 3D programming. The java 3D vector math is needed for supporting multidimensional mathematical objects that are used in java 3D programming while the java abstract window toolkit provides an interface for 3D rendering.

Supplemental image files are used for background and texture setting [7]. Java sets texture of an object by loading an image file from an external folder and draping it over the object to make the object look more realistic. Texture may be predefined based on object type and can be changed on-the-fly per user request. Java supports most types of image files and there is not minimum requirement for image size or resolution. However, an image with higher resolution provides finer texture quality while using more processing time.

Connection between the 3D model application interface and PostgreSQL database can be established through the JDBC driver that provides protocol for java to connect to an external database. It is noted that PostGIS also provides the JDBC extension objects that provide useful data access functions for java to conveniently manage geometry objects in PostgreSQL database [6].

Table 1. Spatial Data Models Supported by PostGIS

Geometry Type	Geometry Dimension		
	2D (x,y)	3D (x,y,z) or (x,y,m ¹)	4D (x,y,z,m ¹)
Point	SF ²	ESF ³	ESF ³
Line	SF ²	ESF ³	ESF ³
Polygon	SF ²	ESF ³	ESF ³
Multi Point	SF ²	ESF ³	ESF ³
Multi Line	SF ²	ESF ³	ESF ³
Multi Polygon	SF ²	ESF ³	ESF ³
Geometry Collection	SF ²	ESF ³	ESF ³
Curve	SQL-MM ⁴	SQL-MM ⁴	SQL-MM ⁴
Compound Curve	SQL-MM ⁴	SQL-MM ⁴	SQL-MM ⁴
Multi Curve	SQL-MM ⁴	SQL-MM ⁴	SQL-MM ⁴
Curve Polygon	SQL-MM ⁴	SQL-MM ⁴	SQL-MM ⁴
Multi Surface	SQL-MM ⁴	SQL-MM ⁴	SQL-MM ⁴

¹Measure value such as linear referencing measurement or Unix time

²Simple Features for SQL specification defined by OGC

³Extended Simple Features for 3D and 4D spatial data representation

⁴SQL Multimedia specification for circularly interpolated curves

3. BIM data modeling for reinforced concrete building

Figure 2 presents the logical data model of a reinforced concrete building in a spatial database. As can be seen, a reinforced concrete building is an aggregation of Entity, which represents an abstract class of all building elements. Entity can be divided into two subclasses: Structure and System. Structure is an abstract class of building's structure elements including footing, beam, floor, column, wall and stair. Structure consists of Geometry, Reinforcing, Material and Time classes. System is an abstract class of building's system elements including electrical and mechanical. System consists of Geometry, Material, Cost and Time classes.

Geometry, which illustrates location, shape and size of each structure element, is implicitly represented by aggregation of element's cross sections. Rather than explicitly encoding coordinate strings of all geometry vertices, this modeling approach can greatly reduce data structure complexity as well as physical storage space by simply maintaining the geometry data of element cross sections at its head and end. For example, if geometry of beam's head and tail sections are known, other geometrical attributes of the beam including length, volume, surface area, centroid and angular alignment can be implicitly derived using spatial queries provided in spatial database. Nevertheless, if cross section of a structure element is not consistent throughout, an intermediate cross section is needed at the location where the cross section is changed.

Reinforcing is a representation of steel reinforcing details in each structure element. Reinforcing is an aggregation of Steel,

which illustrates detail of each reinforcing steel. Steel can be divided into two subclasses: Main and Stirrup. Main represents major reinforcing steel in a structure element while Stirrup represents shear reinforcing steel. Both Main and Stirrup inherits common properties from Steel, which includes Geometry and Property. Geometry represents locations and alignments of Steel while Property represents physical properties including unit cost of Steel. In order to minimize data storage, rather than storing geometry data of all individual steel bars, Geometry shall store only geometry data of one referenced steel bar through Reference class while geometry data of other steel bars can be implicitly encoded through Offset class.

Material represents the physical attributes of material (e.g. concrete) used for each structure element. Material shall also include unit cost for supporting automatic cost estimation of the element. Time represents the temporal aspect of each structural element. Typically, Time includes construction schedule and actual construction time of the element.

System is a representation of each system component in a building. System can be divided into two subclasses: Mechanical and Electrical. Both Mechanical and Electrical have common properties, which includes Geometry, Material, Cost and Time that inherits from System. Geometry illustrates shape and location of each system component while Material represents physical attributes of the component. Time and Cost represent temporal and financial aspects, respectively, regarding construction, operation and maintenance of each system component.

4. Prototype system implementation

An 8-storey reinforced concrete parking building was used as an implementation case study in integrating java-based BIM with spatial database. Figure 3 illustrates the physical database of the building that was derived from the aforementioned logical data model. Several interesting notions in creating and populating the database include:

" Explicit relationship between the Structure and System tables is not available. Rather, the two tables have implicit spatial relationship that can be derived through their geometries.

" Since geometry of each system component can come in various shape and dimension, the geometry field in the System table shall use the Geometry Collection data type provided by PostGIS, which can support modeling of point, line and polygon altogether in one tuple to attain data modeling flexibility.

" While the reference_geometry field in the Reinforce table is meant to store geometry data of the referenced reinforcing steel bar, the x_offset, y_offset and z_offset fields in the Offset table is used to store the offset information between the referenced and its successive reinforcing steel bars. The number field in the Offset table indicates the number of reinforcing steel bars.

" Since PostGIS provides flexibility to accurately encode complete reinforcing details such as bar hooks, splices and spiral stirrup, the estimated volume and weight of reinforcing steel can be calculated more accurately.

" The geometry field in the Section table shall use

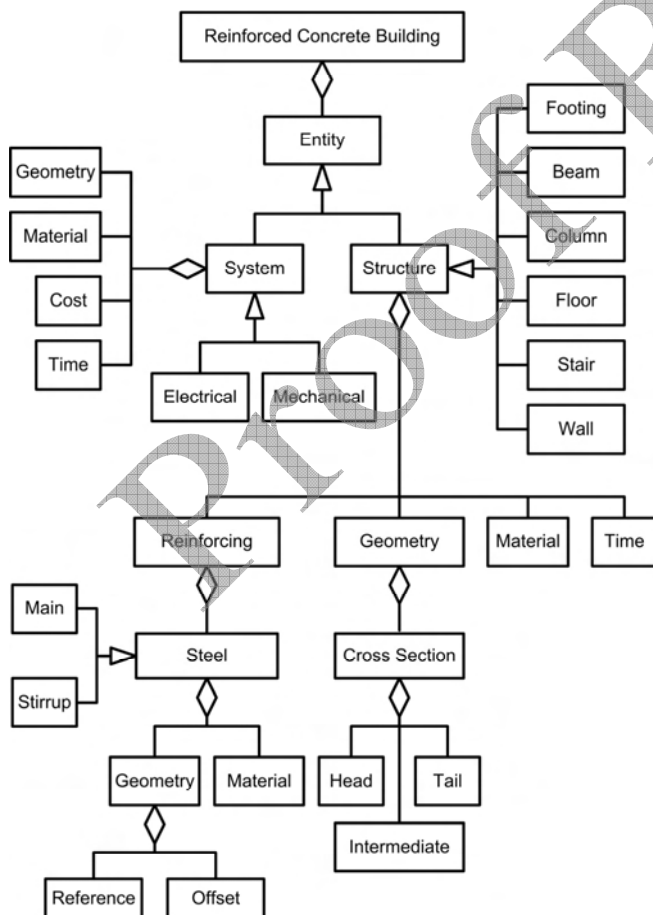


Fig. 2. Logical Data Model of Reinforced Concrete Building

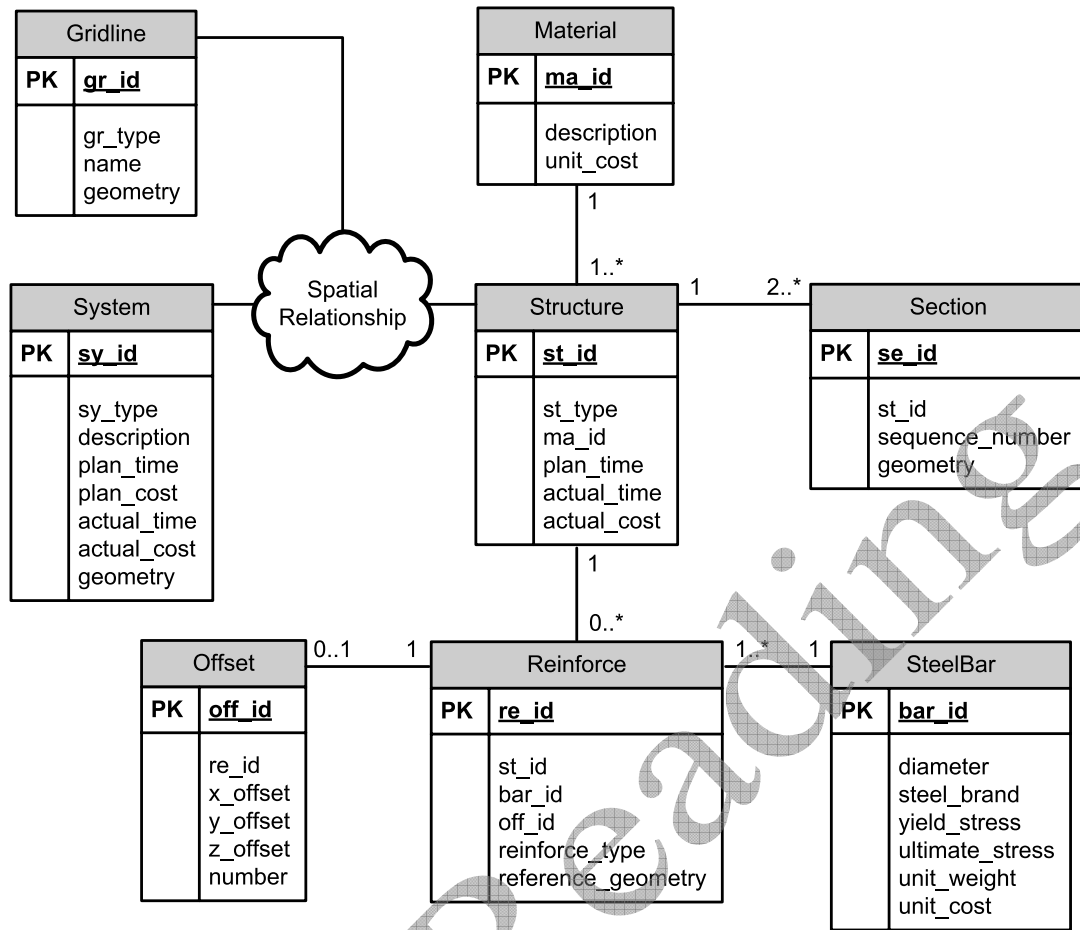


Fig. 3. BIM Database for Reinforced Concrete Building

MultiPolygon data type provided by PostGIS to achieve flexibility to model the structure element with hollow or irregular shape.

" The sequence_number field in the Section table indicates the position (e.g. head, intermediate, tail) of that section on the structure element.

" The budget cost of each structure element can be automatically calculated from the estimated material and steel reinforcing amounts time their unit costs. However, the actual cost for the entire structure element shall be directly entered into the actual_cost field in the Structure table to simplify the cost tracking process.

" The Gridline table allows transformation between the Cartesian coordinate system and human-understandable grid system and vice versa to improve project communication. While PostGIS can support both local and real-world Cartesian coordinate systems, such coordinates cannot be used to communicate effectively among project participants. For example, construction workers would not understand if they are told to dig a hole at coordinate (5, 5, 0). Instead, they could understand if they are told to dig a hole at certain referenced grid location.

" GiST index, the spatial index provided by PostGIS, shall be generated on all geometry fields in the database to minimize processing times for spatial operations and queries of building geometry data.

" The PostgreSQL database maintains transactional integrity

using multiversion model (Multiversion Concurrency Control) where multiple users can access and edit BIM data at the same time without causing a problem to the database.

Figure 4 illustrates the programming approach in generating 3D objects in java. Initially, the Canvas3D needs to be created. The Canvas3D is an instance of the Canvas class in the AWT package that provides an interface for displaying java 3D objects. The SimpleUniverse from the java utility package is constructed to provide 3D locale for viewing java 3D objects through scene graphs.

The BranchGroup (BG) is the root of a scene graph that is inserted into the locale. One scene graph is constructed for each object type to divide a 3D building model into multiple layers. For example, a 3D building model may include ten different layers: pile, footing, beam, column, slab, wall, roof, door, window, and steel reinforcing. This approach improves the ease of use where a user can choose the layers to be displayed.

The TransformGroup (TG) controls spatial transformation of an individual 3D object. For example, if a building has twenty beams, all twenty beams are referenced to the same BranchGroup but the location, width, length, height, rotating angles, and texture of each individual beam are configured through the TransformGroup. In generating each 3D building element in java, the geometry data are manipulated and retrieved from the spatial database and used to construct the TransformGroup of that element.

Figure 5 presents different views of the 8-storey reinforced concrete parking building created by java. The java 3D model can automatically generate different views of the building: front, back, side, top, plan and section views. However, due to coding imperfection, the 2D views generated from the java 3D model in Figure 5 are not truly two dimensional as the piles in the back are still can be seen in the front, side and section views. Nevertheless, this is by no mean the limitation of java as more programming improvement will be able to overcome the imperfection.

The java 3D model has the ability display objects in shade, transparent and wireframe modes. Furthermore, volume and surface area of an object can be automatically calculated. Because the java 3D model is object-oriented, distinguishing the difference between the formwork and painting areas of an object is possible. Since cost and activity duration are by-products of quantity, cost estimate and project scheduling can be done automatically by java if all supporting data are available.

It is noted that transformation between the AutoCAD 3D solid model and the java 3D model is possible. One major obstacle, however, is that the two systems have discrepancy in their coordinate orientation. Furthermore, since the AutoCAD 3D solid model is not object-oriented, the real-world object type (e.g. beam, column) of each geometric object (e.g. polygon, box) exported from AutoCAD is unknown. To improve the transformation between the two systems, the real-world object type should be indicated in the AutoCAD 3D solid model by using specific color for each real-world object type to allow java to determine the real-world object type based on its color [8].

5. Conclusion

Java-based BIM employs an object-oriented approach, which offers a more logically realistic model than the traditional non-object 3D model represented by simple geometric objects. Java-based BIM provides great tools for 3D rendering and is flexible to work on any operating platform. In addition, java-based BIM offers great potential for future extension by incorporating with GIS or an enterprise project management system. Integrating java-based BIM with spatial database is a promising alternative for future BIM application development since spatial database could provide significant improvement for BIM data modeling and management. Additionally, spatial database is equipped with a number of spatial queries that can be useful tools for quantity takeoff, verification of building design conflicts and evaluation of topological relationship between building elements. Not only the proposed prototype system offers technological advantages over the traditional BIM applications, it is also an open-source solution that is free to acquire, customize and distribute, which can overcome the financial constraint that inhibits the implementation of BIM among small and medium enterprises. Nevertheless, in order to fully exploit the proposed prototype system, multiple implementation issues are still required further research including the programming technique to overcome insurmountable data processing time when the model contains a large number of reinforcing steels.

6. Acknowledgement

The authors gratefully acknowledge supports from the Thailand Research Fund.

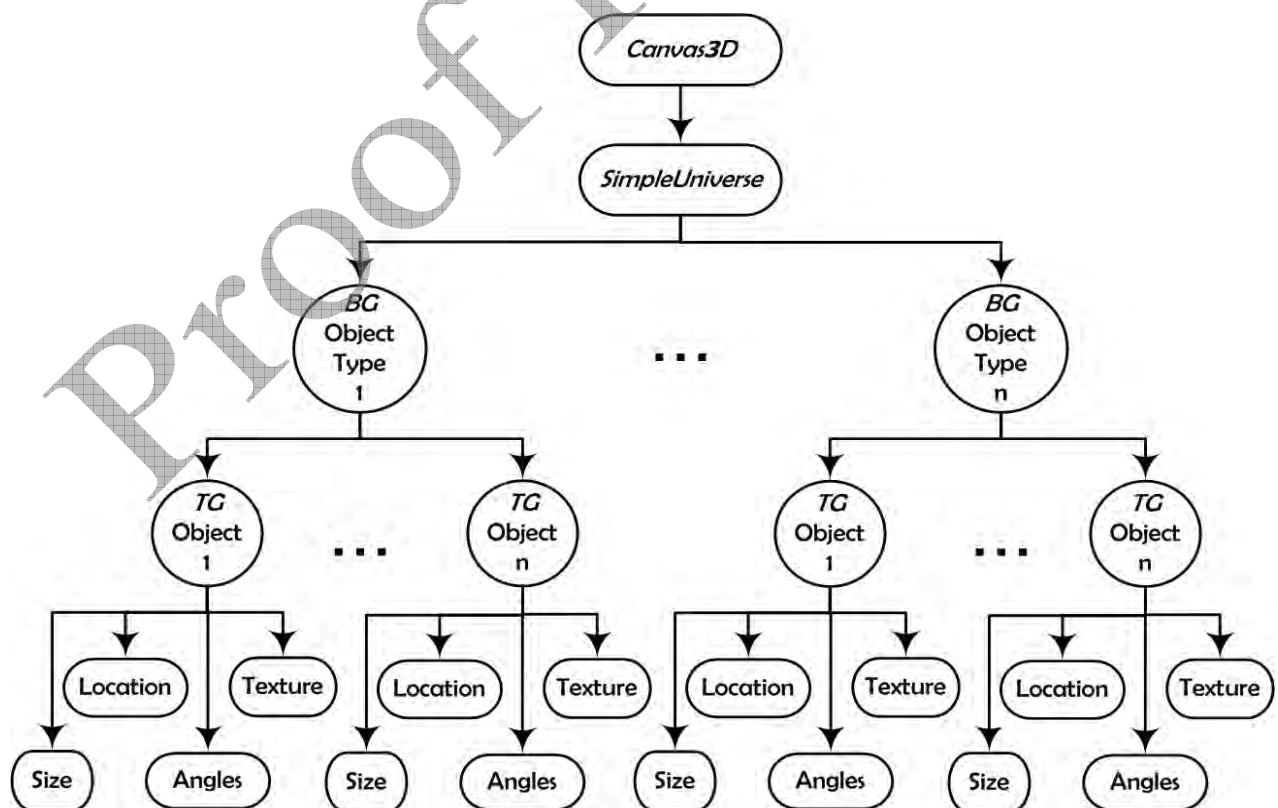


Fig. 4. 3D Model Generation in Java Environment

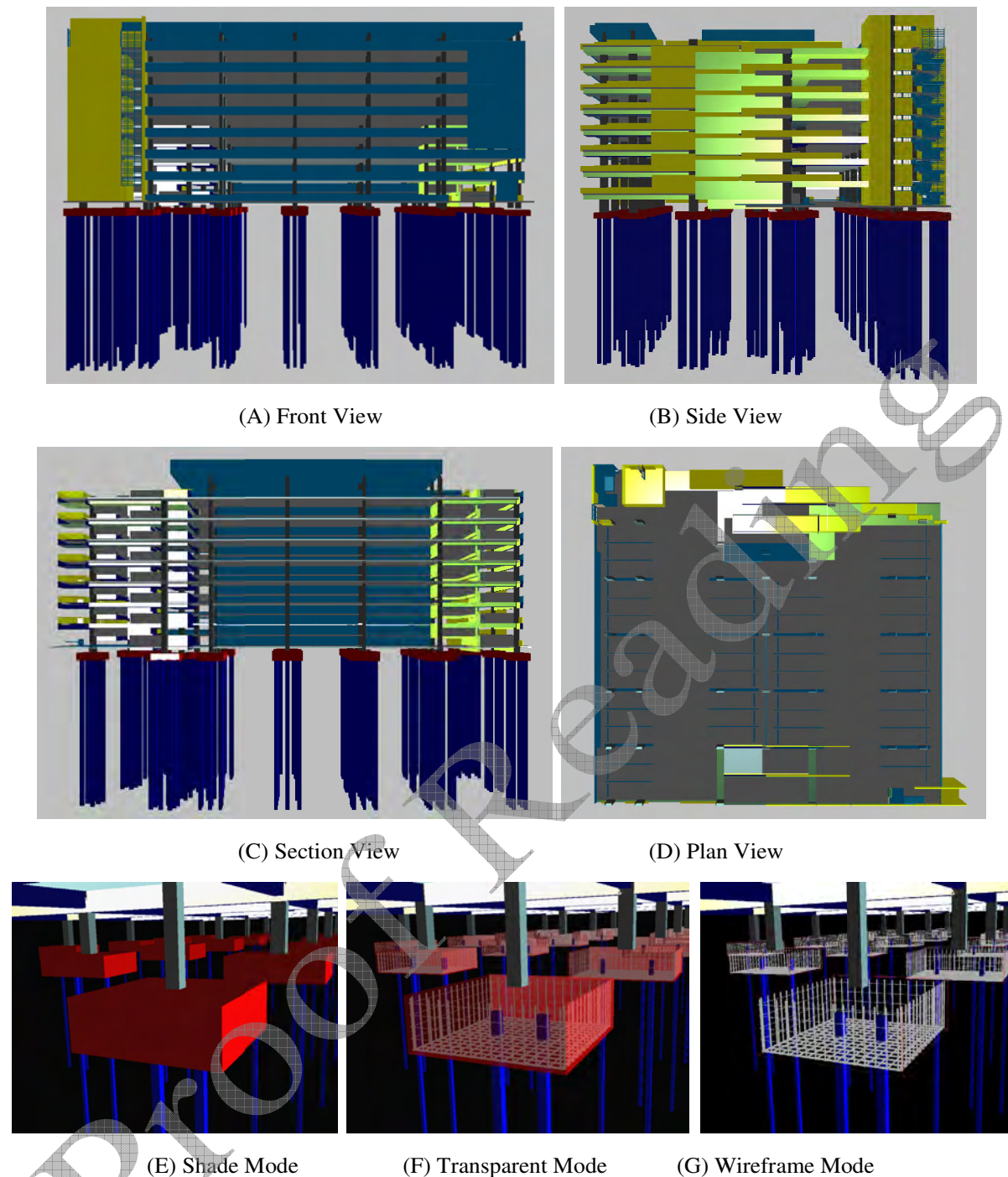


Fig. 4. 3D Model Generation in Java Environment

References

- [1] Khemlani, L. "Should We BIM? Pushing the State of the Art in AEC", Cadalyst, USA, Volume 6, 2003
- [2] Khemlani, L., "The Eureka Tower: A Case Study of Advanced BIM Implementation", AECbytes, USA, Volume 6, 2004
- [3] Sun Microsystems. "Getting Started with the Java 3D API", Tutorial version 1.6.2, Sun Microsystems Inc., California, USA, 2001
- [4] Twilleager, D.; "Introduction to Graphics Programming with Java 3D", Available source: <http://www.euclideanspace.com>, 2006
- [5] Weber, A., Heinemann, A. and Messerli, P., "Use of NASA World Wind Java SDK for Three-Dimensional Accessibility
- [6] Visualization of Remote Areas in Lao P.D.R.", 6th ICA Mountain Cartography Workshop, Lenk, Switzerland, 2008
- [7] Refrations Research, "PostGIS 1.4.0 Manual", Available source: <http://postgis.refrations.net/>, 2009
- [8] Sangkaew, W., Malaikrisanachalee, S. and Vathananukit, H., "Java-based Three-Dimensional Object-Oriented Model for Building Information Modeling.", In Proceedings, 13th National Convention on Civil Engineering, Choburi, Thailand, 2008
- [9] Srisomboon, W. and Malaikrisanachalee, S. "Four-Dimensional Object-Oriented Construction Project Planning.", In Proceedings, 13th National Convention on Civil Engineering, Choburi, Thailand, 2008