บทที่ 5

ส่วนจำเพาะช่วยในการเขียนแบบโครงสร้างของอาคาร

(Building Structure Drafting Module for AutoCAD - BSDMA)

5.1 แนวความคิดในการพัฒนา

ถึงแม้ว่าการใช้คอมพิวเตอร์ในการเขียนแบบ จะช่วยให้การเขียนแบบเป็นไปด้วยความสะดวกสบาย ถูกต้อง แม่นยำ รวดเร็วขึ้นกว่าเดิมมากก็ตาม แต่ในการเขียนแบบโครงสร้างด้วยคอมพิวเตอร์โดยใช้โปรแกรมออโตแคด ผู้เขียนแบบจะต้องทำการลากเส้นทีละเส้น หาตำแหน่งอ้างอิง เขียนเส้นบอกระยะ เขียนตัวอักษรกำหนดรายละเอียดของเหล็กเสริม ดำเนินการทีละขั้นตอน จนสำเร็จได้รายละเอียดแบบโครงสร้างขึ้นมา 1 แบบ นับว่าก็ยังใช้เวลามาก และทำงานในหลายขั้นตอนจึงจะได้ผลงานออกมา จากประสบการณ์ในการทำงานจริง โดยพิจารณาถึงวิธีการและรูปแบบของการเขียนแบบโครงสร้างพบว่า โครงสร้างต่างๆที่ต้องเขียนแบบ มีอยู่หลายครั้งที่ใช้รูปแบบเดียวกัน แต่จะแตกต่างกันที่ขนาด ระยะ เท่านั้น น่าจะมีคำสั่งพิเศษที่สามารถเขียนงานรูปแบบนั้นได้ ในขนาด ระยะที่ต่างกัน โดยการป้อนค่าขนาด ระยะ ที่ต้องการแล้ว โปรแกรมจะเขียนรูปให้ทันทีโดยอัตโนมัติ การพัฒนาส่วนจำเพาะเพื่อช่วยในการเขียนแบบโครงสร้างโดยเลือกพัฒนาสำหรับรูปแบบโครงสร้างที่พบเห็นและนิยมใช้กันบ่อย ๆ จะเป็นวิธีการที่แก้ปัญหานั้นได้ ทำให้การเขียนแบบด้วยโปรแกรมออโตแคด มีประสิทธิภาพมากขึ้น ลดเวลา และขั้นตอนการทำงานลงไปได้มาก

5.2 ขั้นตอนการพัฒนา

การพัฒนาส่วนจำเพาะที่ช่วยในการเขียนแบบโครงสร้างได้ดำเนินการตามขั้นตอนต่างๆ ดังนี้

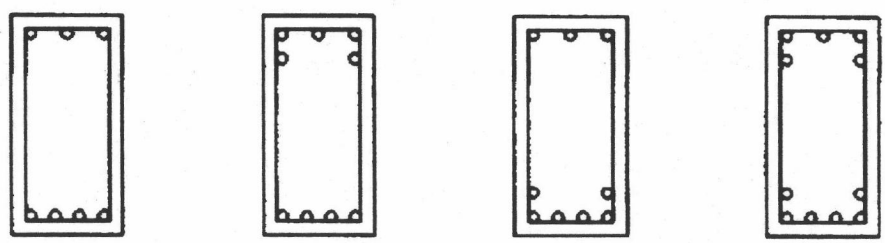
5.2.1 วิเคราะห์ ศึกษา และกำหนดถึงรูปแบบโครงสร้างที่พบเห็น และนิยมใช้กันบ่อย ๆ จนเป็นมาตรฐานของโครงสร้างทั่ว ๆ ไป เพื่อจะนำมาพัฒนาสร้างส่วนจำเพาะในการเขียนแบบ ส่วนโครงสร้างนั้น ๆ ได้แก่

5.2.1.1 **รูปแบบโครงสร้างฐานราก** โครงสร้างของฐานรากจะเริ่มตั้งแต่ฐานรากที่ใช้เสาเข็มเพียงต้นเดียว ซึ่งนิยมใช้กันมากในอาคารขนาดเล็กทั่ว ๆ ไป ไปจนถึงฐานรากที่ใช้เข็ม 5 ต้น ซึ่งสามารถรับน้ำหนักได้มากขึ้น และมีรูปแบบที่ค่อนข้างจะเป็นมาตรฐาน ยกเว้นฐานรากที่ใช้เข็มตั้งแต่ 5 ต้นขึ้นไป การออกแบบฐานราก รูปทรงของฐานราก และการจัดกลุ่มของเสาเข็ม จะแปรเปลี่ยนไปตามการออกแบบของวิศวกรผู้ออกแบบนั้น ๆ ไม่มีมาตราฐานกำหนดไว้ตายตัว รูปทรง

ของฐานรากอาจจะเป็นรูปหกเหลี่ยม รูปสี่เหลี่ยม หรือรูปทรงอื่นๆ ขึ้นอยู่กับการออกแบบโดยพิจารณาถึงลักษณะสภาพพื้นที่นั้นๆ ตำแหน่งของฐานรากข้างเคียง การรับน้ำหนักต่างๆ เป็นต้น และในปัจจุบันเทคโนโลยีทางการก่อสร้างได้พัฒนาขึ้นมาก เสาเข็มสามารถออกแบบให้รับน้ำหนักได้มากขึ้น แนวโน้มในปัจจุบันฐานรากจะใช้เสาเข็มน้อยลง เพื่อสะดวกและลดเวลาในการทำงานก่อสร้าง (รูปที่ 2.1 - 2.4)

5.2.1.2 **รูปแบบโครงสร้างหน้าตัดเสา** โครงสร้างของเสาที่พบกันโดยทั่วไป ส่วนมากจะมีหน้าตัดของเสาเป็นรูปสี่เหลี่ยมผืนผ้า หรือรูปสี่เหลี่ยมจัตุรัส และมีขนาดต่างๆ กันออกไปตามน้ำหนักที่เสานั้นต้องรับบรรทุก รูปแบบของการเสริมเหล็กภายในเสา โดยปกติทั่วไปจะเริ่มด้วยการเสริมเหล็กยืน 4 เส้น และมีเหล็กปลอกเพียง 1 ชุด ถ้ามีการรับน้ำหนักมากขึ้น วิศวกรจะเพิ่มขนาดของหน้าตัดเสา, เพิ่มขนาดของ และจำนวนของเหล็กยืน และในบางครั้งอาจจะต้องเพิ่มชุดของเหล็กปลอก เพื่อให้เหมาะสมกับน้ำหนักที่บรรทุกในจุดตำแหน่งของเสานั้น การพัฒนานี้ได้เลือกพัฒนารูปแบบหน้าตัดเสาแบบต่าง ๆ ที่ใช้กันบ่อย ๆ (ดังรูปที่ 2.5)

5.2.1.3 **รูปแบบโครงสร้างหน้าตัดคาน** โครงสร้างหน้าตัดของคานจะคล้ายคลึงกับหน้าตัดของเสา แต่พฤติกรรมการรับน้ำหนักของคานจะแตกต่างไปจากของเสา คานจะรับน้ำหนักบรรทุกตามนอนตลอดแนวยาวของคานนั้น โดยปกติลักษณะคานที่ใช้ทั่วไปจะเป็นรูปสี่เหลี่ยมผืนผ้า การเสริมเหล็กในคานประกอบด้วย เหล็กเสริมแนวบน และเหล็กเสริมแนวล่างของคาน โดยยึดเหล็กเสริมทั้งสองแนวไว้ด้วยเหล็กปลอก ถ้ามีน้ำหนักบรรทุกมากขึ้น วิศวกรจะเพิ่มขนาดของหน้าตัดคาน เพิ่มขนาดและจำนวนของเหล็กเสริมบนและล่าง หรืออาจจะเพิ่มจำนวนชุดเหล็กปลอก และกำหนดระยะห่างระหว่างเหล็กปลอกให้ถี่ขึ้นก็ได้ ในการวิจัยนี้ได้เลือกรูปแบบของหน้าตัดคานที่นิยมใช้กันมาก โดยสามารถกำหนดจำนวนเหล็กเสริมบนและล่างได้ถึง 2 ระดับ (ดังรูปที่ 5.1)

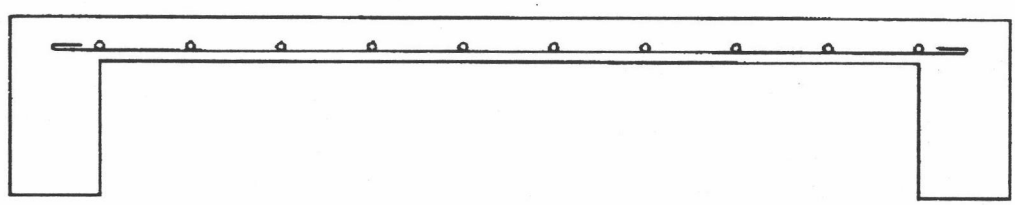


รูปที่ 5.1 รูปแบบการเสริมเหล็กของหน้าตัดคานที่นิยมใช้กันทั่วไป

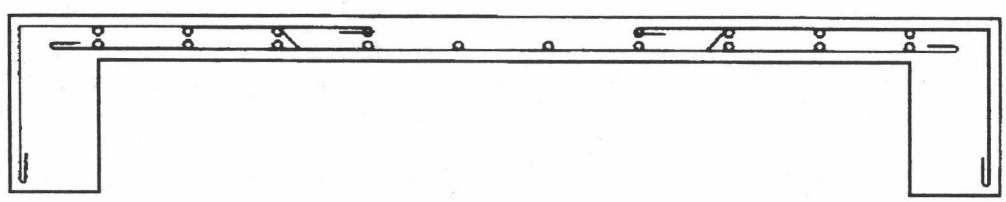
5.2.1.4 **รูปแบบโครงสร้างพื้นคอนกรีต** รูปแบบโครงสร้างพื้นคอนกรีตที่ได้วิเคราะห์ เพื่อนำมาพัฒนาเป็นส่วนจำเพาะช่วยในการเขียนแบบโครงสร้างนั้นได้แบ่งแยกออกเป็น 2 ประเภท ตาม ลักษณะการทำงานก่อสร้าง คือ

- **โครงสร้างพื้นคอนกรีตหล่อกับที่** ซึ่งประกอบไปด้วยรูปแบบโครงสร้างของ

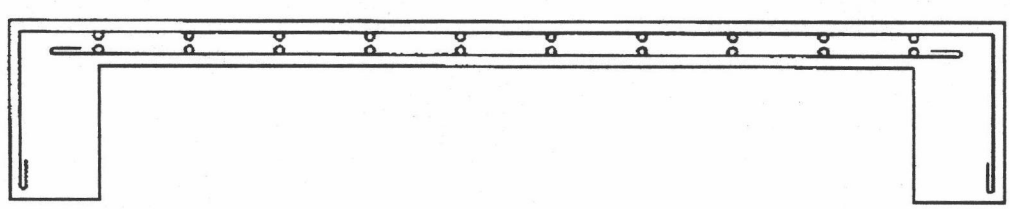
- พื้นคอนกรีตช่วงสั้น ถ่ายน้ำหนักทางเดียว (Short Span One Way Slab_**SS1WS**) (รูปที่ 5.2)
- พื้นคอนกรีตช่วงสั้น ถ่ายน้ำหนักสองทาง (Short Span 2 Way Slab_**SS2WS**) (รูปที่ 5.3)
- พื้นคอนกรีตช่วงสั้น ถ่ายน้ำหนักสองทาง เสริมเหล็ก 2 ชั้น (Short Span 2 Way Slab_Double Bar_**SS2WDB**) (รูปที่ 5.4)
- พื้นคอนกรีตช่วงยาว ถ่ายน้ำหนักสองทาง (Long Span 2 Way Slab_**LS2WS**) (รูปที่ 5.5)
- พื้นคอนกรีตช่วงยาว ถ่ายน้ำหนักสองทาง เสริมเหล็ก 2 ชั้น (Long Span 2 Way Slab Double Bar_**LS2WDB**) (รูปที่ 5.6)
- พื้นคอนกรีตยื่น (Cantiliver Slab_**SLC**) (รูปที่ 5.7)



รูปที่ 5.2 พื้นคอนกรีตช่วงสั้น ถ่ายน้ำหนักทางเดียว (Short Span One Way Slab_**SS1WS**)

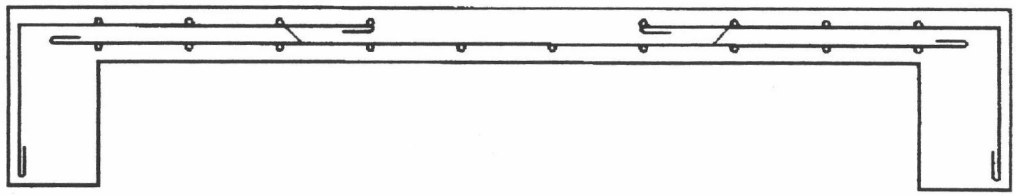


รูปที่ 5.3 พื้นคอนกรีตช่วงสั้น ถ่ายน้ำหนักสองทาง (Short Span 2 Way Slab_**SS2WS**)

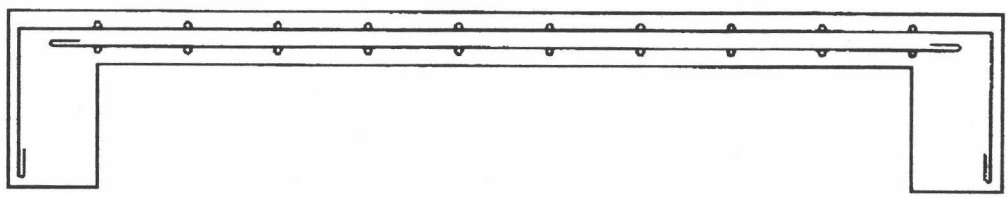


รูปที่ 5.4 พื้นคอนกรีตช่วงสั้น ถ่ายน้ำหนักสองทาง เสริมเหล็ก 2 ชั้น

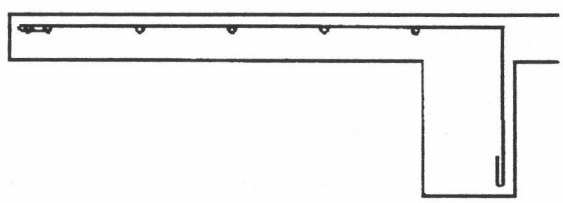
(Short Span 2 Way Slab_Double Bar_**SS2WDB**)



รูปที่ 5.5 พื้นคอนกรีตช่วงยาว ถ่ายน้ำหนักสองทาง (Long Span 2 Way Slab_**LS2WS**)



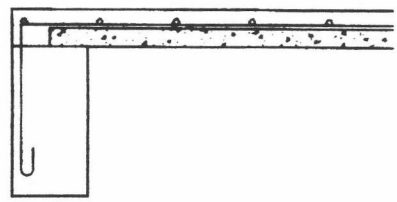
รูปที่ 5.6 พื้นคอนกรีตช่วงยาว ถ่ายน้ำหนักสองทาง เสริมเหล็ก 2 ชั้น
(Long Span 2 Way Slab Double Bar_**LS2WDB**)



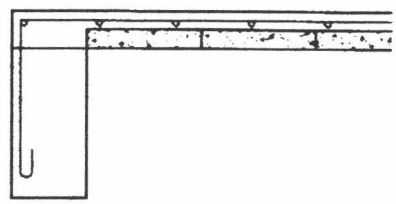
รูปที่ 5.7 พื้นคอนกรีตยื่น (Cantiliver Slab_**SLC**)

- **โครงสร้างพื้นคอนกรีตสำเร็จรูปชนิดแผ่นเรียบ (Pre-stressed Concrete Slab:PLANK)** ซึ่งเป็นแผ่นพื้นคอนกรีตสำเร็จรูปที่มีการเสริมเหล็กอัดแรงภายในมาจากโรงงานผู้ผลิตเป็นที่นิยมใช้กันมากในปัจจุบัน เพราะสะดวก สบาย ประหยัดเวลาและทำงานได้ง่าย โดยปกติจะมีขนาดความหนาประมาณ 5 ซม. และความกว้างต่อแผ่นประมาณ 30-35 ซม. ความยาวจะแปรตามช่วงของคานที่พาดรับน้ำหนัก แต่ประมาณไม่เกิน 4.00 ม. การเขียนโครงสร้างพื้นประเภทนี้จะเขียนได้ใน 2 ลักษณะคือ

- พื้นคอนกรีตสำเร็จรูปทางยาว (Pre-stressed concrete Slab Long section_**PSL**) (รูปที่ 5.8)
- พื้นคอนกรีตสำเร็จรูปทางสั้น (Pre-stressed concrete Slab Cross section_**PSC**) (รูปที่ 5.9)

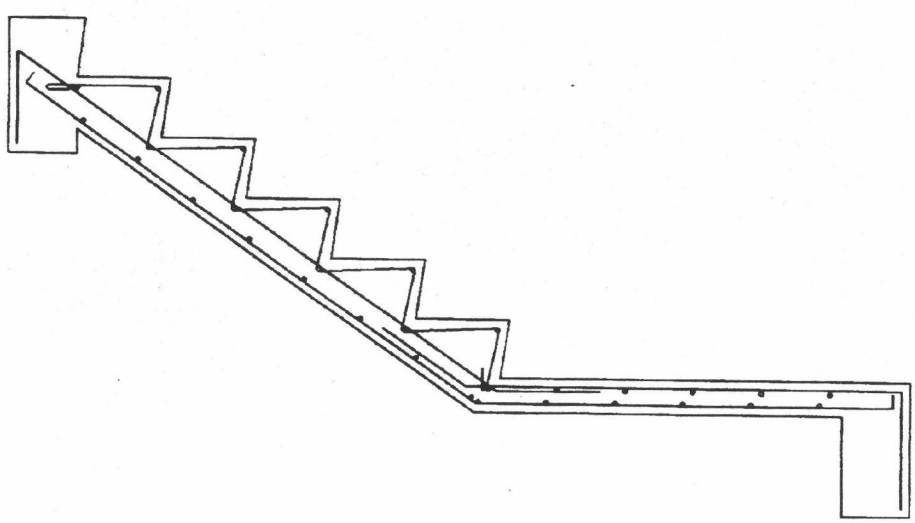


รูปที่ 5.8 พื้นคอนกรีตสำเร็จรูปทางยาว (Pre-stressed concrete Slab Long section_**PSL**)



รูปที่ 5.9 พื้นคอนกรีตสำเร็จรูปทางสั้น (Pre-stressed concrete Slab Cross section_**PSC**)

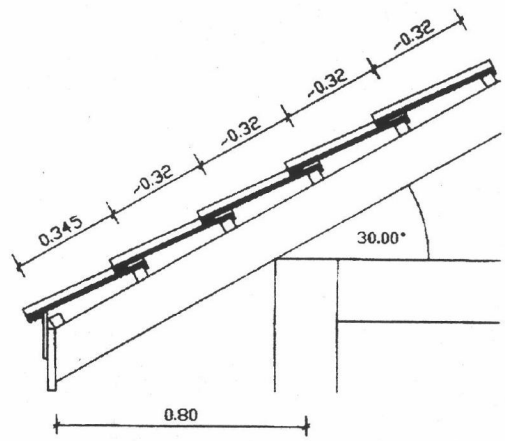
5.2.1.5 **รูปแบบโครงสร้างบันไดคอนกรีต** บันไดคอนกรีตจะมีรูปแบบแตกต่างกันออกไปได้มากมาย เช่น บันไดพับผ้า, บันไดรูปตัวแอล, บันไดโค้ง บันไดเวียน ฯลฯ และมีการถ่ายเทน้ำหนักบรรทุกจากบันไดลงสู่คานหรือเสา ในหลายลักษณะต่าง ๆ กัน การสร้างส่วนจำเพาะเพื่อใช้ในการเขียนแบบบันไดคอนกรีตได้เลือกใช้รูปแบบบันไดต่อเนื่อง 1 ช่วง (1FLIGHT) เพราะนิยมใช้กันโดยทั่วไป และสามารถนำไปประกอบใช้กับบันไดอื่น ๆ ที่ซับซ้อนขึ้นได้ เช่น บันไดแบบพับผ้า หรือบันไดรูปตัวแอล (รูปที่ 5.10)



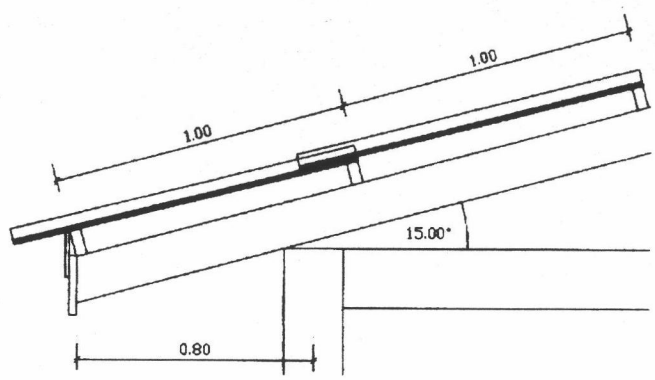
รูปที่ 5.10 การเสริมเหล็กบันไดคอนกรีต 1 ช่วง

5.2.1.6 **รูปแบบโครงสร้างหลังคา** หลังคาเป็นโครงสร้างของอาคารที่มีความหลากหลาย มีรูปแบบต่างๆ กันออกไปได้มากมายเช่นเดียวกับบันได หลังคาอาจจะแบ่งได้ตามลักษณะของโครงสร้างได้ดังนี้

- หลังคาคอนกรีตเสริมเหล็ก มักจะพบเห็นกันทั่วไปในอาคารคอนกรีตเสริมเหล็กที่มีดาดฟ้า ในการออกแบบจะถือว่าหลังคาคอนกรีตเสริมเหล็กเปรียบเสมือนพื้นคอนกรีตเสริมเหล็กชนิดหนึ่ง ใช้วิธีการเสริมเหล็กเช่นเดียวกับพื้นคอนกรีต ดังนั้นการเขียนแบบรายละเอียดของหลังคาคอนกรีตเสริมเหล็กจะเหมือนกับการเขียนแบบรายละเอียดของพื้นคอนกรีต
- หลังคาโครงสร้างประเภทอื่น หลังคาประเภทนี้จะประกอบไปด้วยโครงสร้างหลังคาและวัสดุมุงหลังคา วัสดุโครงสร้างหลังคาอาจจะเป็นได้ทั้งไม้เนื้อแข็ง และเหล็กรูปพรรณต่างๆ หรืออาจจะเป็นวัสดุอื่นๆก็ได้ ส่วนวัสดุมุงหลังคาที่นิยมใช้กันได้แก่ กระเบื้องมุงหลังคาต่างๆ เช่น กระเบื้องลอนต่างๆ กระเบื้องคอนกรีต ฯลฯ



รูปที่ 5.11 แบบขยายการมุงหลังคากระเบื้องคอนกรีต



รูปที่ 5.12 แบบขยายการมุงหลังคากระเบื้องลอนคู่

5.2.2 ศึกษาข้อมูลนำเข้าของการเขียนแบบส่วนโครงสร้างต่างๆ เพื่อใช้เป็นพารามิเตอร์สำหรับโปรแกรมทำงาน จากการศึกษาถึงการเขียนแบบส่วนโครงสร้างต่าง ๆ พบว่าข้อมูลที่ผู้เขียนแบบจะต้องเรียนรู้เพื่อนำไปใช้ในการเขียนแบบโครงสร้าง แต่ละส่วนนั้นมีดังนี้

5.2.2.1 **ข้อมูลนำเข้าสำหรับการเขียนแบบฐานราก** จะประกอบไปด้วย ขนาดความกว้าง ความยาว และความหนาของฐานราก ระยะหุ้มของคอนกรีต รายละเอียดการเสริมเหล็กต่างๆ ทั้งขนาด และจำนวนของเหล็กเสริม ความหนาของชั้นคอนกรีตหยาบรองก้นหลุม ความหนาของชั้นทรายอัดแน่น และจำนวน ประเภท ขนาดของเสาเข็มที่ใช้ รวมไปถึงน้ำหนักปลอดภัยที่รับได้ของเสาเข็มแต่ละต้น

5.2.2.2 **ข้อมูลนำเข้าสำหรับการเขียนแบบเสา** จะประกอบไปด้วย ขนาดความกว้าง ความยาวของหน้าตัดเสา ระยะหุ้มของคอนกรีต รายละเอียดการเสริมเหล็กต่างๆ ทั้งขนาด และจำนวนของเหล็กเสริมเสา ทั้งเหล็กยืน และเหล็กปลอก รวมถึงระยะห่างของเหล็กปลอก

5.2.2.3 **ข้อมูลนำเข้าสำหรับการเขียนแบบคาน** จะประกอบไปด้วย ขนาดความกว้าง ความยาวของหน้าตัดคาน ระยะหุ้มของคอนกรีต รายละเอียดการเสริมเหล็กต่างๆ ทั้งขนาดและจำนวนของเหล็กเสริมคานทั้งเหล็กเสริมบน เหล็กเสริมล่าง และเหล็กปลอก รวมถึงระยะห่างของเหล็กปลอก

5.2.2.4 **ข้อมูลนำเข้าสำหรับการเขียนแบบพื้นคอนกรีต** จะประกอบไปด้วยขนาดความหนา ความยาวของพื้นคอนกรีต ระยะหุ้มของคอนกรีต รายละเอียดการเสริมเหล็กต่างๆ ทั้งขนาดและจำนวนของเหล็กเสริมบน เหล็กเสริมล่าง รวมถึงระยะห่างของเหล็กเสริม

5.2.2.5 **ข้อมูลนำเข้าสำหรับการเขียนแบบบันไดคอนกรีต** จะประกอบไปด้วยจำนวนขั้นบันได ความสูง ความกว้างของขั้นบันได ขนาดความหนาของบันได ระยะหุ้มของคอนกรีต รายละเอียดการเสริมเหล็กต่างๆ ทั้งขนาด และจำนวนของเหล็กเสริมบน เหล็กเสริมล่าง เหล็กเสริมขั้นบันได รวมถึงระยะห่างของเหล็กเสริมต่างๆ

5.2.2.6 **ข้อมูลนำเข้าสำหรับการเขียนแบบหลังคากระเบื้อง** จะประกอบไปด้วยชนิดของกระเบื้องที่ใช้มุงหลังคา มุมยกของหลังคา ขนาดของโครงหลังคาต่างๆ รวมถึงระยะยื่นของชายคา

5.2.3 สร้างฟังก์ชันในภาษาออโตลิสป์ (AutoLISP function) ที่ช่วยในการทำงานต่างๆ เช่น ฟังก์ชันช่วยในการเขียนรูปสี่เหลี่ยมผืนผ้า ฟังก์ชันช่วยในการเขียนตัวอักษรให้รายละเอียดของเหล็กเสริม หรือ ฟังก์ชันช่วยในการเขียนเส้นบอกระยะ เป็นต้น

5.2.4 พัฒนาโปรแกรมภาษาออโตลิสป์ ที่ใช้เขียนแบบส่วนโครงสร้างต่างๆ (AutoLISP drawing routine file) โดยให้รับข้อมูลนำเข้าจากการพิมพ์ที่บรรทัดคำสั่ง

5.2.5 ออกแบบตัวประสานกับผู้ใช้แบบกราฟิก (Graphic User`s Interface) เพื่อให้ผู้ใช้ป้อนค่าข้อมูลต่างๆ โดยที่ในโปรแกรมออโตแคดจะใช้กรอบสนทนาเป็นเครื่องมือในการติดต่อ และโต้ตอบกับผู้ใช้

5.2.6 สร้างกรอบสนทนาโดยใช้ภาษาดีซีแอล (Dialog Control Language)

5.2.7 พัฒนาโปรแกรมขับกรอบสนทนาในภาษาออโตลิสป์ (AutoLISP dialog box driver) สำหรับควบคุมการทำงานต่างๆ ของกรอบสนทนา

5.2.8 สร้างแฟ้มสไลด์ (Slide file) ที่ใช้แสดงผลในกรอบสนทนา

5.2.9 ปรับปรุงแก้ไขโปรแกรมภาษาออโตลิสป์ที่ใช้เขียนส่วนของโครงสร้างให้สามารถรับข้อมูลที่ผู้ใช้ป้อนเข้าทางกรอบสนทนาได้

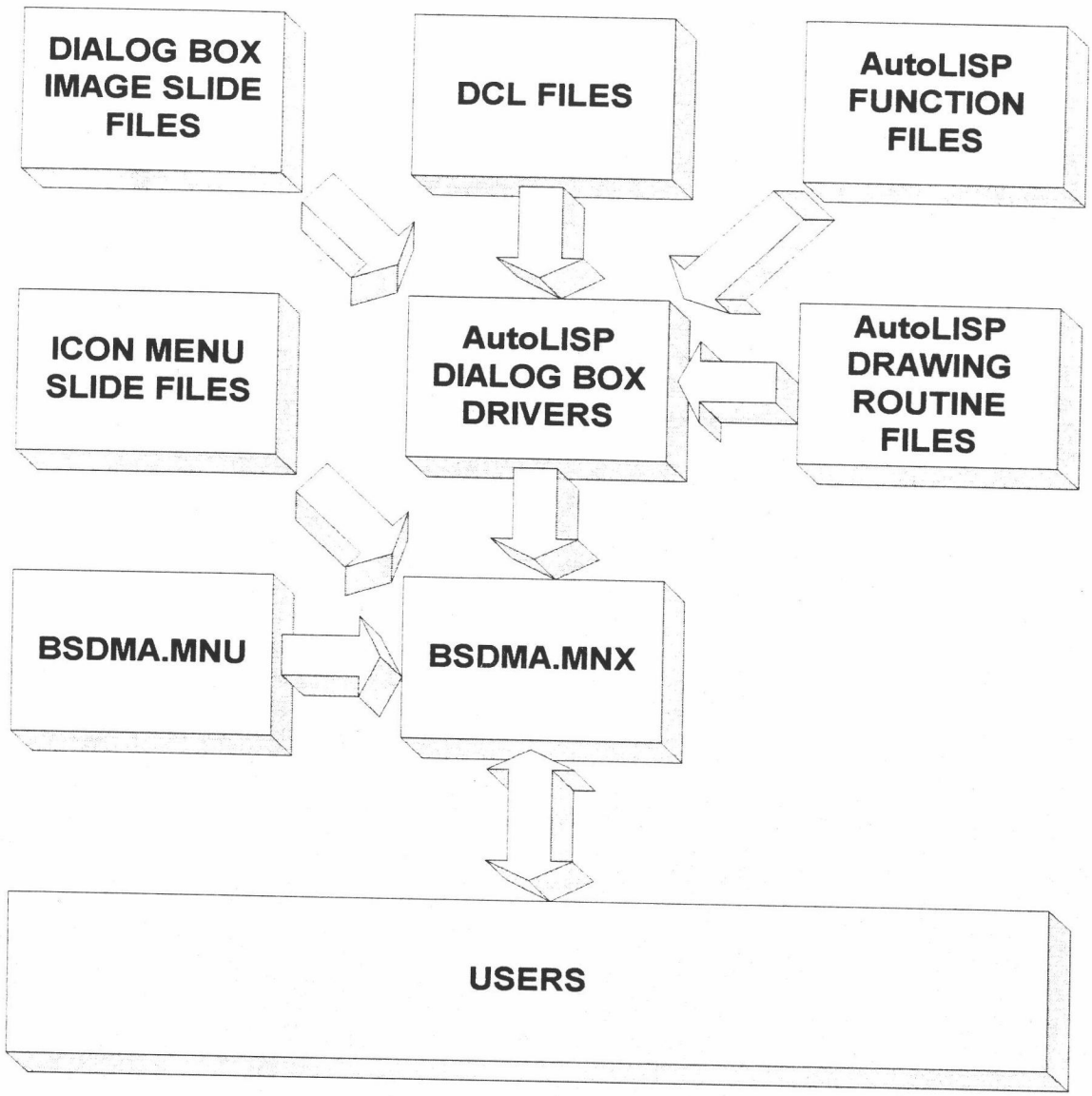
5.2.10 ปรับปรุงรายการเลือกของโปรแกรมออโตแคดให้สามารถเรียกใช้งานโปรแกรมที่พัฒนานี้ได้ ทั้งในส่วนของ Pop up menu และ Icon menu

5.2.11 สร้างแฟ้มสไลด์ที่ใช้แสดงผลใน Icon menu

5.2.12 ทำการทดสอบโปรแกรม และแก้ไขปรับปรุงโปรแกรมที่พัฒนาแล้วให้ใช้งานได้อย่างสมบูรณ์ ถูกต้อง

5.3 การทำงานของโปรแกรม

5.3.1 องค์ประกอบของ BSDMA



รูปที่ 5.13 แผนภูมิการทำงานของส่วนจำเพาะ BSDMA

BSDMA ประกอบไปด้วยแฟ้มต่าง ๆ 4 ประเภท ที่ทำงานเกี่ยวเนื่องกันสามารถแยกประเภทต่าง ๆ ได้ดังนี้ (รูปที่ 5.13)

5.3.1.1 แฟ้มรายการเลือก (Menu file) คือ BSDMA.MNU เป็นประเภทแฟ้มรายการเลือกของโปรแกรมออโตแคดใช้เป็นรายการเลือกสำหรับเรียกการทำงานของส่วนต่างของ BSDMA สร้างได้โดยการนำเอาแฟ้ม ACAD.MNU ต้นฉบับ มาเขียนเพิ่มเติมในส่วนของการเรียกใช้ส่วนจำเพาะช่วยในการเขียนแบบโครงสร้างเพิ่มเข้าไป โดยเขียนเพิ่มเข้าไปที่ Pop up menu และ Icon menu เมื่อออโตแคดบรรจุ BSDMA.MNU เข้าไปทำงานจะทำการแปล BSDMA.MNU ให้เป็นแฟ้มรายการเลือกที่สามารถประมวลผลได้ (Menu executable file) จะได้เป็น BSDMA.MNX และนำไปใช้งานต่อไป

ใน BSDMA.MNU จะมีทางเลือกให้เลือกว่าจะเขียนส่วนของโครงสร้างใด และในแต่ละทางเลือกจะมีคำสั่งอ้างอิงไปยังแฟ้มสไลด์ และแฟ้มออโตลิสป์ที่เกี่ยวข้องเพื่อเรียกมาทำงาน เพื่อให้ได้ผลตามที่ต้องการ

5.3.1.2 แฟ้มสไลด์ (Slide file) เป็นแฟ้มรูปภาพของโปรแกรมออโตแคด ใช้สำหรับแสดงผลภาพ ประกอบรายการเลือก และกรอบสนทนาเพื่อให้ผู้ใช้สามารถเข้าใจและป้อนข้อมูลในกรอบสนทนาได้อย่างง่ายๆ แบ่งได้ 2 ประเภท ดังนี้

- **แฟ้มสไลด์สำหรับการแสดงผลในรายการเลือก** เป็นการแสดงผลในรูปแบบที่เรียกว่า สัญรูป (Icon) ซึ่งอยู่ในส่วนของ Icon menu ของ BSDMA.MNU การแสดงผลภาพสัญรูปนี้ ภาพที่ออกมาจะมีขนาดเล็ก ไม่สามารถแสดงรายละเอียดได้มาก ดังนั้นภาพสไลด์ที่จะใช้สำหรับทำเป็นสัญรูปต้องมีขนาดเหมาะสมและไม่มีรายละเอียดมากนัก แต่สามารถสื่อสารให้ผู้ใช้เข้าใจได้ว่าเป็นรูปภาพของโครงสร้างส่วนใด มีตัวอักษรขนาดใหญ่กำกับในภาพเพื่อบอกว่าเป็นภาพของโครงสร้างส่วนใด แฟ้มประเภทนี้ได้แก่ I_B1.SLD, I_C1.SLD, I_F1.SLD ฯลฯ เป็นต้น

- **แฟ้มสไลด์สำหรับการแสดงผลในกรอบสนทนา** ใช้ในการแสดงผลภาพประกอบในกรอบสนทนา เพื่อสื่อสารให้ผู้ใช้เข้าใจในส่วนต่างๆ ของโครงสร้างที่จะเขียนว่าในแต่ละส่วนเรียกว่าอะไร ทำให้ผู้ใช้สามารถป้อนค่าในกรอบสนทนาได้ง่ายและถูกต้อง รูปภาพสไลด์ที่ใช้สำหรับแสดงผลในกรอบสนทนา จะต้องมีขนาดใหญ่พอสมควรและมีรายละเอียดตัวอักษรบ่งชี้ว่าส่วนใดเรียกว่าอะไร ชื่อส่วนประกอบต่างๆ ที่บอกไว้ในภาพควรจะตรงกับชื่อส่วนประกอบที่รอการป้อนค่าในกรอบสนทนา ตัวอักษรที่แสดงต้องมีขนาดใหญ่พอสมควร พอที่จะสามารถอ่านออกได้ เมื่อเวลาภาพนั้นถูกแสดงในกรอบสนทนา แฟ้มประเภทนี้ได้แก่ B1.SLD, C1.SLD, F1.SLD ฯลฯ เป็นต้น

5.3.1.3 แฟ้มออโตลิสป์ (AutoLISP file) เป็นแฟ้มภาษาออโตลิสป์ ซึ่งสามารถเขียนโปรแกรมใช้สั่งการทำงานออโตแคดได้ ใน BSDMA นี้สามารถแบ่งแฟ้มภาษาออโตลิสป์ ตามการใช้งานได้ 3 ประเภท ดังนี้

- **แฟ้มออโตลิสป์ฟังก์ชัน (AutoLISP function file)** เป็นฟังก์ชันที่เขียนขึ้นในภาษาออโตลิสป์ เพื่อช่วยในการทำงานต่าง ๆ เช่น ช่วยในการคำนวณมุม ช่วยในการเขียนเส้น ช่วยในการบอกระยะ ฯลฯ แบ่งออกได้ 3 หมวดใหญ่ ๆ ได้ดังนี้

1. **ฟังก์ชันช่วยในการทำงานทั่วไป (General function)** ได้แก่

ฟังก์ชัน DTR	ใช้เปลี่ยนค่ามุมจากองศา (DEGREE) ไปเป็นเรเดียน (RADIAN)
ฟังก์ชัน RTD	ใช้เปลี่ยนค่ามุมจากเรเดียนไปเป็นองศา
ฟังก์ชัน CHKANG	ใช้ตรวจสอบค่ามุม
ฟังก์ชัน DM	ใช้เขียนเส้นบอกระยะของจุด 2 จุด
ฟังก์ชัน DRWCSTL	ใช้เขียนรูปเหล็กเสริมทางขวาง โดยกำหนด ระยะห่างระหว่างเหล็กเสริม
ฟังก์ชัน DRWSTL	ใช้เขียนรูปเหล็กเสริมทางขวางโดยกำหนดจำนวนเหล็ก และระยะรวมการเสริมเหล็ก
ฟังก์ชัน FMIDPT	ใช้คำนวณหาพิกัดจุดกึ่งกลางของจุด 2 จุด
ฟังก์ชัน HOOK	ใช้เขียนการงอเหล็กที่ปลายของเหล็กเสริม
ฟังก์ชัน INDXL	ใช้เขียนเส้นชี้บอกรายละเอียด
ฟังก์ชัน INDXLL	ใช้เขียนเส้นชี้บอกรายละเอียดทางด้านซ้าย
ฟังก์ชัน REC	ใช้เขียนสี่เหลี่ยมผืนผ้าตามความกว้าง ความยาวที่กำหนด
ฟังก์ชัน SETDMTCK	ใช้ตั้งระบบการเขียนเส้นบอกขนาดเป็น TICK
ฟังก์ชัน STIRTXT	ใช้เขียนตัวอักษรบอกรายละเอียดเหล็กปลอก
ฟังก์ชัน STLTXT	ใช้เขียนตัวอักษรบอกรายละเอียดเหล็กเริม
ฟังก์ชัน STR2TD	ใช้แยกอักขระที่สกัดได้จาก Pop up list ออกเป็น 2 ส่วน คือ ส่วนที่เป็นข้อมูลประเภทของเหล็กเสริม และส่วนที่เป็นข้อมูลขนาดของเหล็กเสริม
ฟังก์ชัน STTXT	ใช้เป็นตัวอักษรบอกรายละเอียดเหล็กเสริมทางด้านซ้าย
ฟังก์ชัน TITLETXT	ใช้เป็นตัวอักษรที่เป็นชื่อของส่วนโครงสร้างนั้นและบอกขนาดของ SCALE ที่ใช้เขียน

2. **ฟังก์ชันที่ใช้สำหรับการเขียนฐานราก (Footing function)** ได้แก่

ฟังก์ชัน DRWPILE ใช้เขียนรูปเสาเข็มตามขนาดที่กำหนด
ฟังก์ชัน DRWFSTL ใช้เขียนเหล็กเสริมในฐานรากในรูปตัด
ฟังก์ชัน DRWSLINE ใช้เขียนเส้นเหล็กเสริมของฐานรากในรูปแปลน
ฟังก์ชัน DRWHTCH ใช้เขียนแรเงาหน้าตัดในรูปแบบที่กำหนด
ฟังก์ชัน DRW3STL ใช้เขียนเหล็กเสริมฐานราก 3 เหลี่ยมที่ใช้เข็ม 3 ต้น
ฟังก์ชัน PILETXT ใช้เขียนอักษรบอกรายละเอียดเสาเข็ม
ฟังก์ชัน MATTXT ใช้เขียนอักษรบอกรายละเอียดและขนาดความหนาของวัสดุ
ฟังก์ชัน FSTLTXT ใช้เขียนอักษรบอกรายละเอียดเหล็กเสริมในฐานราก
ฟังก์ชัน EBSTLTXT ใช้เขียนอักษรบอกรายละเอียดเหล็กเสริมปลายในฐานราก
ฟังก์ชัน XTEND ใช้เขียนเส้นสัญลักษณ์ที่แสดงว่าเป็นส่วนที่ถูกตัดออกมา
ฟังก์ชัน COLHISTL ใช้เส้นเหล็กเสริมของเสาในฐานราก
ฟังก์ชัน MRKPILE ใช้เขียนกากบาทตำแหน่งจุดกึ่งกลางเสาเข็มในแปลนของฐานราก
ฟังก์ชัน DRW3FP ใช้เขียนเส้นรอบรูปฐานราก 3 เหลี่ยม ที่ใช้เข็ม 3 ต้น
ฟังก์ชัน FINDFILE ใช้หาตำแหน่งจุดกึ่งกลางเสาเข็มของฐานราก 3 เหลี่ยมที่ใช้เข็ม 3 ต้น

3. **ฟังก์ชันที่ใช้ในการเขียนพื้น (Slab function)** ได้แก่

ฟังก์ชัน SLBSTX ใช้เขียนอักษรบอกรายละเอียด เหล็กเสริมในพื้น
ฟังก์ชัน DWSLBSTL ใช้เขียนเหล็กเสริมทางขวางในแผ่นพื้นโดยกำหนดจำนวนระยะห่างการเสริมเหล็ก ไม่มีการคำนวณเศษ
ฟังก์ชัน DRWSLBCB ใช้เขียนเหล็ก เสริมทางขวางในแผ่นพื้นโดยกำหนดระยะห่างระหว่างเหล็กเสริม และคำนวณเศษของการเสริมเหล็ก
ฟังก์ชัน ARRYSTL ใช้เขียนเหล็กเสริมทางขวางในแผ่นพื้น โดยกำหนดจุดตั้งต้น จุดปลายและระยะห่างระหว่างเหล็กเสริม

4. **ฟังก์ชันที่ใช้ในการเขียนบันได (Stair function)** ได้แก่

ฟังก์ชัน FTHRDB ใช้คำนวณหาพิกัดตำแหน่งเหล็กเสริมบริเวณจมูกบันได
ฟังก์ชัน FTHRDB1 ใช้คำนวณหาพิกัดตำแหน่งเหล็กเสริมทางยาวของบันได
ฟังก์ชัน TNGNTPT ใช้คำนวณหาพิกัดจุดสัมผัสของวงกลม 2 วงที่กำหนด

- **แฟ้มออโตลิสป์ที่ใช้เขียนส่วนโครงสร้าง (AutoLISP drawing routine file)** เป็นแฟ้มในภาษาออโตลิสป์ ที่ใช้ในการเขียนรูปส่วนของโครงสร้างต่าง ๆ ตามต้องการ ซึ่งจะทำการรับค่าตัวแปรที่

กำหนดใช้งานจากผู้ใช้ โดยผ่านทางกรอบสนทนาและจะเรียกฟังก์ชันที่เกี่ยวข้องเข้ามาใช้เป็นเครื่องมือในการทำงานในแต่ละขั้นตอน แฟ้มเหล่านี้ได้แก่

G_B1.LSP	ใช้เขียนรูปหน้าตัดคานแบบที่ 1
G_B2.LSP	ใช้เขียนรูปหน้าตัดคานแบบที่ 2
G_B3.LSP	ใช้เขียนรูปหน้าตัดคานแบบที่ 3
G_B4.LSP	ใช้เขียนรูปหน้าตัดคานแบบที่ 4
G_C1.LSP	ใช้เขียนรูปหน้าตัดเสาแบบที่ 1
G_C2.LSP	ใช้เขียนรูปหน้าตัดเสาแบบที่ 2
G_C3.LSP	ใช้เขียนรูปหน้าตัดเสาแบบที่ 3
G_C4.LSP	ใช้เขียนรูปหน้าตัดเสาแบบที่ 4
G_C5.LSP	ใช้เขียนรูปหน้าตัดเสาแบบที่ 5
G_C6.LSP	ใช้เขียนรูปหน้าตัดเสาแบบที่ 6
G_F1.LSP	ใช้เขียนฐานรากแบบที่ 1 มีเสาเข็ม 1 ต้น
G_F2.LSP	ใช้เขียนฐานรากแบบที่ 2 มีเสาเข็ม 2 ต้น
G_F3.LSP	ใช้เขียนฐานรากแบบที่ 3 มีเสาเข็ม 3 ต้น
G_F4.LSP	ใช้เขียนฐานรากแบบที่ 4 มีเสาเข็ม 4 ต้น
G_F5.LSP	ใช้เขียนฐานรากแบบที่ 5 มีเสาเข็ม 5 ต้น
G_SS1WS.LSP	ใช้เขียนพื้นคอนกรีตช่วงสั้น ถ่ายน้ำหนักทางเดียว
G_SS2WS.LSP	ใช้เขียนพื้นคอนกรีตช่วงสั้น ถ่ายน้ำหนักสองทาง
G_SS2WDB.LSP	ใช้เขียนพื้นคอนกรีตช่วงสั้น เสริมเหล็ก สองชั้น
G_LS2WS.LSP	ใช้เขียนพื้นคอนกรีตช่วงยาว ถ่ายน้ำหนักสองทาง
G_LS2WDB.LSP	ใช้เขียนพื้นคอนกรีตช่วงยาว เสริมเหล็ก สองชั้น
G_PSC.LSP	ใช้เขียนพื้นคอนกรีตสำเร็จรูป ทางขวาง
G_PSL.LSP	ใช้เขียนพื้นคอนกรีตสำเร็จรูป ทางยาว
G_SLC.LSP	ใช้เขียนพื้นคอนกรีตยื่น
G_STUP.LSP	ใช้เขียนบันไดคอนกรีต
G_CCTILE.LSP	ใช้เขียนแบบขยายหลังคากระเบื้องคอนกรีต
G_RMTILE.LSP	ใช้เขียนแบบขยายหลังคากระเบื้องลอน

5.3.1.4 แฟ้มดีซีแอล (DCL file) เป็นแฟ้มที่เขียนขึ้นในภาษาดีซีแอล ใช้ในการสร้างตัวกรอบสนทนา เพื่อให้ผู้ใช้ป้อนค่า และนำไปเขียนแบบส่วนโครงสร้างที่ต้องการ แฟ้มดีซีแอลใน BSDMA มีดังนี้

DDB1.DCL	ใช้สร้างกรอบสนทนาของรูปหน้าตัดคานแบบที่ 1
DDB2.DCL	ใช้สร้างกรอบสนทนาของรูปหน้าตัดคานแบบที่ 2
DDB3.DCL	ใช้สร้างกรอบสนทนาของรูปหน้าตัดคานแบบที่ 3
DDB4.DCL	ใช้สร้างกรอบสนทนาของรูปหน้าตัดคานแบบที่ 4
DDC1.DCL	ใช้สร้างกรอบสนทนาของรูปหน้าตัดเสาแบบที่ 1
DDC2.DCL	ใช้สร้างกรอบสนทนาของรูปหน้าตัดเสาแบบที่ 2
DDC3.DCL	ใช้สร้างกรอบสนทนาของรูปหน้าตัดเสาแบบที่ 3
DDC4.DCL	ใช้สร้างกรอบสนทนาของรูปหน้าตัดเสาแบบที่ 4
DDC5.DCL	ใช้สร้างกรอบสนทนาของรูปหน้าตัดเสาแบบที่ 5
DDC6.DCL	ใช้สร้างกรอบสนทนาของรูปหน้าตัดเสาแบบที่ 6
DDF1.DCL	ใช้สร้างกรอบสนทนาของฐานรากแบบที่ 1 มีเสาเข็ม 1 ต้น
DDF2.DCL	ใช้สร้างกรอบสนทนาของฐานรากแบบที่ 2 มีเสาเข็ม 2 ต้น
DDF3.DCL	ใช้สร้างกรอบสนทนาของฐานรากแบบที่ 3 มีเสาเข็ม 3 ต้น
DDF4.DCL	ใช้สร้างกรอบสนทนาของฐานรากแบบที่ 4 มีเสาเข็ม 4 ต้น
DDF5.DCL	ใช้สร้างกรอบสนทนาของฐานรากแบบที่ 5 มีเสาเข็ม 5 ต้น
DDSS1WS.DCL	ใช้สร้างกรอบสนทนาของพื้นคอนกรีตช่วงสั้น ถ่ายน้ำหนักทางเดียว
DDSS2WS.DCL	ใช้สร้างกรอบสนทนาของพื้นคอนกรีตช่วงสั้น ถ่ายน้ำหนักสองทาง
DDSS2WDB.DCL	ใช้สร้างกรอบสนทนาของพื้นคอนกรีตช่วงสั้น เสริมเหล็ก สองชั้น
DDLS2WS.DCL	ใช้สร้างกรอบสนทนาของพื้นคอนกรีตช่วงยาว ถ่ายน้ำหนักสองทาง
DDLS2WDB.DCL	ใช้สร้างกรอบสนทนาของพื้นคอนกรีตช่วงยาว เสริมเหล็ก สองชั้น
DDPSC.DCL	ใช้สร้างกรอบสนทนาของพื้นคอนกรีตสำเร็จรูป ทางขวาง
DDPSL.DCL	ใช้สร้างกรอบสนทนาของพื้นคอนกรีตสำเร็จรูป ทางยาว
DDSLC.DCL	ใช้สร้างกรอบสนทนาของพื้นคอนกรีตยื่น
DDSTUP.DCL	ใช้สร้างกรอบสนทนาของบันไดคอนกรีต
DDCCTILE.DCL	ใช้สร้างกรอบสนทนาของหลังคากระเบื้องคอนกรีต
DDRMTILE.DCL	ใช้สร้างกรอบสนทนาของหลังคากระเบื้องลอน

5.3.2 ขั้นตอนการทำงานของ BSDMA

ขั้นตอนการทำงานของ BSDMA สามารถอธิบายได้ตามลำดับดังนี้

5.3.2.1 เมื่อเข้าโปรแกรมออโตแคด แล้วจะเรียกใช้ BSDMA โดยพิมพ์คำสั่ง MENU ที่บรรทัดคำสั่ง (Command line) แล้วเลือก BSDMA.MNU โปรแกรมออโตแคดจะค้นหาแฟ้ม BSDMA.MNU และแปล (compile) แฟ้ม BSDMA.MNU ให้เป็นแฟ้ม BSDMA.MNX และบรรจุแฟ้ม BSDMA.MNX ในหน่วยความจำเพื่อใช้งาน และแสดงรายการเลือกใช้งานบนจอภาพ

5.3.2.2 เมื่อเลือกส่วนโครงสร้างอาคารที่ต้องการจะเขียนจากรายการเลือก ในกรณีที่เป็น Icon menu โปรแกรมออโตแคดจะเรียกแฟ้มสไลด์สำหรับรายการเลือกมาแสดงผลบนจอภาพให้เลือกใช้

5.3.2.3 เมื่อกดเลือกรูปแบบที่ต้องการใน Icon menu แล้วโปรแกรมออโตแคดจะทำงานตามคำสั่งที่เขียนไว้ในแฟ้มรายการเลือก โดยเรียกแฟ้มดีซีแอลที่เกี่ยวข้องมาทำงาน

5.3.2.4 แฟ้มดีซีแอลจะสร้างกรอบสนทนาตามรูปแบบที่กำหนดไว้ในแฟ้มดีซีแอลนั้น และเรียกแฟ้มสไลด์สำหรับกรอบสนทนาที่เกี่ยวข้องมาแสดงผลรอให้ผู้ใช้ทำงานกับกรอบสนทนา

5.3.2.5 ในระหว่างที่ผู้ใช้ป้อนค่าข้อมูลต่างๆ ลงในกรอบสนทนา โปรแกรมออโตแคดจะทำการตรวจสอบการป้อนข้อมูลของผู้ใช้ ว่าเป็นประเภทข้อมูลที่ถูกต้องตามความต้องการของโปรแกรมหรือไม่ ถ้าไม่ถูกต้องก็จะแสดงข้อความเตือนความผิดพลาดให้ผู้ใช้ทราบ

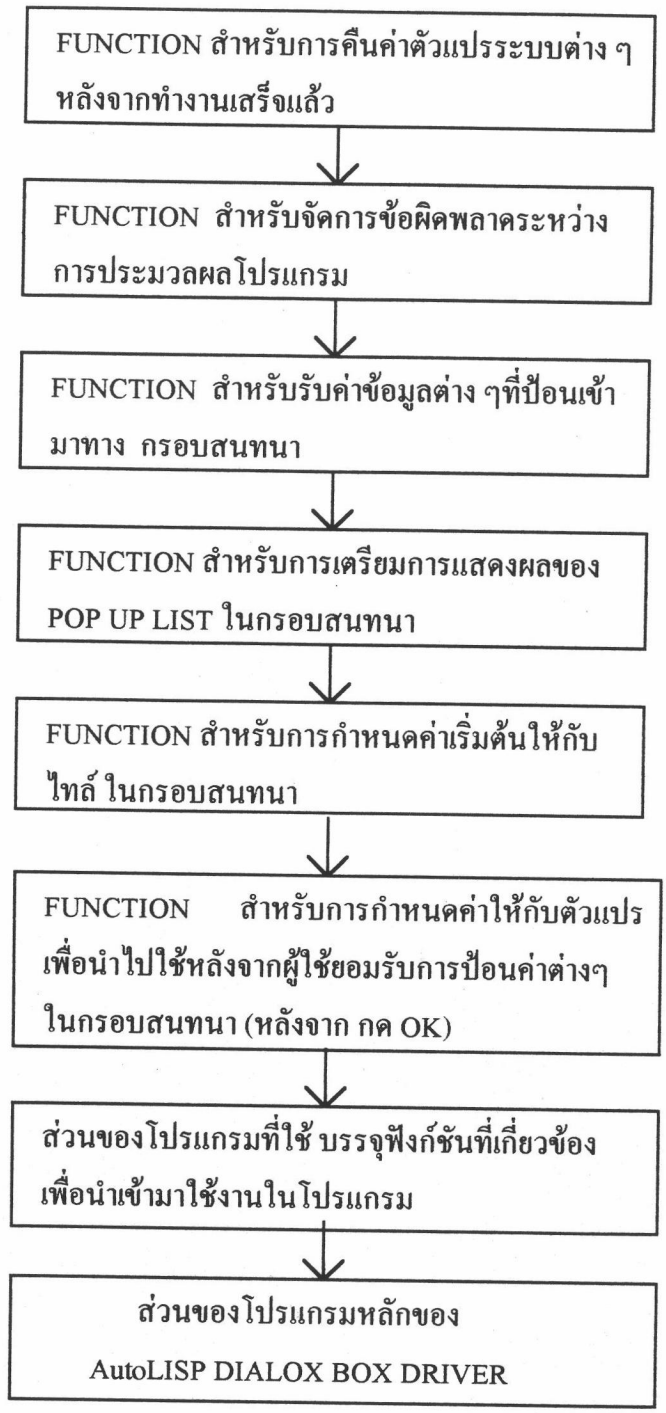
5.3.2.6 เมื่อผู้ใช้ป้อนค่าข้อมูลต่างๆ ลงในกรอบสนทนาเรียบร้อยแล้ว และกดปุ่ม OK เป็นการยอมรับค่าข้อมูลทั้งหมดที่ป้อนเข้าไป โปรแกรมออโตแคดจะทำการกำหนดค่าตัวแปรใช้งานให้พร้อมที่จะทำงานได้

5.3.2.7 โปรแกรมออโตแคดจะแจ้งให้ผู้ใช้เลือกจุดที่จะเขียนแบบโครงสร้างนั้น

5.3.2.8 เมื่อผู้ใช้กำหนดจุดที่จะเขียนแบบโครงสร้างแล้ว โปรแกรมออโตแคดจะส่งผ่านค่าตัวแปรที่เตรียมไว้แล้ว พร้อมกับค่าพิกัดตำแหน่งการเขียนภาพให้กับแฟ้มออโตลิสป์ที่ใช้ในการเขียนรูปส่วนของโครงสร้าง เพื่อทำการเขียนภาพโครงสร้างนั้นๆ จนสำเร็จ

5.3.3 ส่วนประกอบทั่วไปของโปรแกรมขับกรอบสนทนา ใน BSDMA

การเขียนโปรแกรมขับกรอบสนทนานับว่าเป็นงานที่ซับซ้อนมากที่สุดงานหนึ่งในการพัฒนาวิจัยชิ้นนี้ เพราะจะต้องเขียนโปรแกรมควบคุมการทำงานทั้งหมดของกรอบสนทนา ไม่ว่าจะเป็นการเฝ้าตรวจสอบการป้อนค่าของผู้ใช้ การกำหนดการทำงานให้กับไทล์ต่างๆ การเตรียมการแสดงผลของไทล์เช่น Pop up list เป็นต้น รวมถึงการเรียกใช้ฟังก์ชันต่างๆ ที่เกี่ยวข้องมาทำงาน ส่วนประกอบต่างๆ ของโปรแกรมขับกรอบสนทนา สามารถแยกออกได้ดังรูปที่ 5.14 ลองพิจารณาตัวอย่างโปรแกรมขับกรอบสนทนาของการเขียนแบบเสาประเภทที่ 1 (ddc1.lsp)



รูปที่ 5.14 แผนภูมิส่วนประกอบของโปรแกรมขับกรอบสนทนาใน BSDMA

```
(defun RESTORE()
  (setvar "BLIPMODE" oldblp) (setvar "CMDECHO" oldech)
```

```
  (setvar "PICKBOX" oldpbx)  (setq *ERROR* olderr)
)
(defun *ERROR* (msg)
  (princ " \nERROR: ") (princ msg)
  (unload_dialog dcl_id)
  (restore)
  (princ)
)
;=============================================================================
(defun GET_CNAME()
  (setq cnamet (get_tile "cname")) (str_ok cnamet "column name")
)
(defun GET_CW(/ ok)
  (setq cwt (distof (get_tile "cw") 2))
  (real_ok cwt "column width")
  (if (= ok 0) (progn (setq cwt cw) (set_tile "cw" (rtos cwt 2 3))))
)
(defun GET_CD(/ ok)
  (setq cdt (distof (get_tile "cd") 2))
  (real_ok cdt "column depth")
  (if (= ok 0) (progn (setq cdt cd) (set_tile "cd" (rtos cdt 2 3))))
)
(defun GET_CVR(/ ok)
  (setq cvrt (distof (get_tile "cvr") 2))
  (real_ok cvrt "covering")
  (if (= ok 0) (progn (setq cvrt cvr) (set_tile "cvr" (rtos cvrt 2 3))))
)
(defun GET_ASTRP(/ ok)
  (setq astrpt (distof (get_tile "astrp") 2))
  (real_ok astrpt "spacing")
  (if (= ok 0) (progn (setq astrpt astrp) (set_tile "astrp" (rtos astrpt 2 3))))
```

```
)
;-------------------------------------------------------------------
(defun GET_MB(index_string)
 (if (/= index_string "")
   (progn
     (setq index (atoi index_string))
     (setq mb (nth index list_item))
     (set_tile "mb" mb)
   );endprogn
 );endif
)
(defun GET_STRP(index_string)
 (if (/= index_string "")
   (progn
     (setq index (atoi index_string))
     (setq strp (nth index list_item))
     (set_tile "strp" strp)
   );endprogn
 );endif
)
;=====================================================================
(defun MAKE_LIST(popkey)          ;MAKE LIST
  (setq list_item (list "RB 6" "RB 9" "RB 12" "RB 15" "RB 19" "RB 22" "RB 25"
                        "DB 10" "DB 16" "DB 19" "DB 20" "DB 25" "DB 28"))
  (start_list popkey)
  (mapcar ' add_list list_item)
  (end_list)
)
(defun SET_IVALUE()
  (setq cname "C1") (set_tile "cname" cname)
  (if (not cw) (setq cw 0.2)) (set_tile "cw" (rtos cw 2 3))
```

```
  (if (not cd) (setq cd 0.2)) (set_tile "cd" (rtos cd 2 3))
  (if (not cvr) (setq cvr 0.025)) (set_tile "cvr" (rtos cvr 2 3))
  (if (not astrp) (setq astrp 0.2)) (set_tile "astrp" (rtos astrp 2 3))
  (if (not mb) (setq mb "RB 6")) (set_tile "mb" mb)
  (if (not strp) (setq strp "RB 6")) (set_tile "strp" strp)
)
(defun DO_SETVAR()
  (setq cname (get_tile "cname"))
  (setq cw (atof (get_tile "cw")))
  (setq cd (atof (get_tile "cd")))
  (setq cvr (atof (get_tile "cvr")))
  (setq astrp (atof (get_tile "astrp")))
     (str2td mb) (setq tmb typ) (setq dmb di)
          (setq dmbx (/ dmb 1000))
     (str2td strp) (setq tstrp typ) (setq dstrp di)
          (setq dstrpx (/ dstrp 1000))
)
;=========================LOAD ASSOCIATE LISP=========================
(if (not chktype) (load "c:/acadr12/bsdma/chktype"))
(if (not g_c1) (load "c:/acadr12/bsdma/g_c1"))
;---------------------------------------------------------------
(defun c:ddc1(/ cnamet cwt cdt cvrt astrpt ok list_item what_next
          oldblp oldech oldpbx olderr)

  (setq oldblp (getvar "BLIPMODE")
     oldech (getvar "CMDECHO")
     oldpbx (getvar "PICKBOX")
     olderr *ERROR*
  )
  (setvar "BLIPMODE" 0) (setvar "CMDECHO" 0) (setvar "PICKBOX" 0)
  ;----------------------MAIN PROCEDURE-------------------------
```

```
(setq dcl_id (load_dialog "ddc.dcl"))      ;LOAD DIALOG
(if (not (new_dialog "ddc_dlg" dcl_id)) (exit))

;; initialize the tiles
(set_ivalue)
(mode_tile "cname" 2)
(start_image "c_image")
  (setq max_x (dimx_tile "c_image"))
  (setq max_y (dimy_tile "c_image"))
  (slide_image 0 0 max_x max_y "c1")
  (end_image)
(make_list "mb")
(make_list "strp")

(action_tile "cname" "(get_cname)")
(action_tile "cw" "(get_cw)")
(action_tile "cd" "(get_cd)")
(action_tile "cvr" "(get_cvr)")
(action_tile "astrp" "(get_astrp)")
(action_tile "mb" "(get_mb $value)")
(action_tile "strp" "(get_strp $value)")
(action_tile "accept" "(do_setvar)(done_dialog 1)")
(action_tile "cancel" "(done_dialog 0)")

(setq what_next (start_dialog))
(if (= what_next 1)
   (progn
     (g_c1)
   );endprogn
);endif
(unload_dialog dcl_id)
```

```
(restore)
(princ)
)
;=============================================END=============================================
```

ในโปรแกรมควบคุมกรอบสนทนาของการเขียนแบบเสาประเภทที่ 1 (ddc1.lsp) จะพบว่ามีส่วนประกอบต่างๆ คือ

5.3.3.1 ฟังก์ชันสำหรับการคืนค่าตัวแปรระบบต่างๆ

ฟังก์ชัน RESTORE เป็นฟังก์ชันที่ใช้สำหรับคืนค่าตัวแปรระบบต่างๆ ได้แก่ "BLIPMODE", "CMDECHO" และ "PICKBOX" โดยการเก็บค่าเดิมของตัวแปรเหล่านี้ไว้ก่อนที่โปรแกรมทำงาน ในระหว่างการทำงาน ค่าตัวแปรเหล่านี้จะมีการเปลี่ยนแปลงไป และเมื่อจบการทำงานแล้วก็จะคืนค่าเดิมให้กับตัวแปรเหล่านี้

5.3.3.2 ฟังก์ชันสำหรับจัดการข้อผิดพลาดในระหว่างการประมวลผล

ฟังก์ชัน *ERROR* เป็นฟังก์ชันที่ออโตลิสป์กำหนดไว้สำหรับจัดการกับความผิดพลาดที่เกิดขึ้นในระหว่างการทำงาน เมื่อมีความผิดพลาดเกิดขึ้น ระบบจะแจ้งข้อความแสดงความผิดพลาดผ่านทางฟังก์ชัน *ERROR* และจะทำการปิดกรอบสนทนาที่เปิดอยู่ คืนค่าตัวแปรระบบต่างๆให้เป็นปกติตามเดิม

5.3.3.3 ฟังก์ชันสำหรับการรับค่าข้อมูลที่ป้อนเข้ามาทางกรอบสนทนา และตรวจสอบประเภทของค่าที่ป้อนเข้ามาว่าตรงกับที่กำหนดไว้หรือไม่

ฟังก์ชัน GET_CNAME, GET_CW, GET_CD, GET_CVR, GET_ASTRP, GET_MB และ GET_STRP เป็นฟังก์ชันที่ใช้ในการรับค่าข้อมูลที่ป้อนเข้าทางกรอบสนทนา โดยผ่านไทล์ต่างๆ และจะทำการตรวจสอบประเภทของข้อมูลว่าถูกต้องตรงกับที่ต้องการหรือไม่โดยการใช้ฟังก์ชัน REAL_OK, INT_OK ฯลฯ ฟังก์ชัน GET_MB และ GET_STRP จะใช้รับค่าขนาดและประเภทของเหล็กเสริมหลัก และ เหล็กปลอก ที่กำหนดไว้ใน Pop up list

5.3.3.4 ฟังก์ชันสำหรับสร้างรายการ (List) และเตรียมการแสดงผลของ Pop up list

ฟังก์ชัน MAKE_LIST ใช้สำหรับสร้างรายการตัวเลือกใน Pop up list ต่างๆ เช่นในการเลือกประเภท ขนาดของเหล็กเสริมหลัก และเหล็กปลอก และใช้ในการแสดงผลของ Pop up list นั้น

5.3.3.5 ฟังก์ชันสำหรับกำหนดค่าเริ่มต้นให้กับไทล์แต่ละไทล์ในกรอบสนทนา
ฟังก์ชัน SET_IVALUE ใช้กำหนดค่าเริ่มต้นให้กับไทล์แต่ละไทล์ในกรอบสนทนา ในตัวแปรบางตัวถ้าเกิดมีค่าเดิมของตัวแปรตัวนั้น จะรับเอาค่าเดิมมาใช้

5.3.3.6 ฟังก์ชันสำหรับกำหนดค่าให้กับตัวแปรส่วนกลางเพื่อใช้ในการเขียนแบบโครงสร้าง เมื่อผู้ใช้ป้อนค่าต่างๆในกรอบสนทนาเรียบร้อยแล้ว กดเลือกปุ่ม OK จะเป็นการยอมรับการตั้งค่าเหล่านั้น เป็นจังหวะที่ดีที่จะกำหนดค่าต่างๆให้กับตัวแปรส่วนกลาง เพื่อใช้ในการเขียนแบบโครงสร้าง เพราะไม่มีการเปลี่ยนแปลงค่าใดๆอีก เนื่องจากกรอบสนทนาได้ปิดแล้ว และใช้ฟังก์ชัน DO_SETVAR ในการกำหนดค่าให้กับตัวแปรส่วนกลาง

5.3.3.7 ส่วนของโปรแกรมที่เรียกใช้ฟังก์ชันที่เกี่ยวข้องในการทำงานของกรอบสนทนา จะมีการตรวจสอบว่าในหน่วยความจำได้มีการบรรจุฟังก์ชันที่ต้องการใช้แล้วหรือไม่ ถ้ายังไม่มีในหน่วยความจำ โปรแกรมจะทำการบรรจุฟังก์ชันที่ต้องการใช้ลงในหน่วยความจำ ในกรณีนี้จะบรรจุฟังก์ชัน CHKTYPE (ซึ่งใช้ในการตรวจสอบชนิดของค่าที่ป้อนเข้าทางกรอบสนทนา) และฟังก์ชัน G_C1 (ซึ่งใช้ในการเขียนแบบโครงสร้างเสาแบบที่ 1)

5.3.3.8 โปรแกรมหลักของ AutoLISP dialog box driver

- เริ่มแรกจะเป็นการกำหนดชื่อให้กับฟังก์ชันหลัก (defun c:**ddc1**()) และให้สามารถเรียกใช้ได้เสมือนคำสั่งหนึ่งของออโตแคด
- ต่อมาเป็นการนำค่าตัวแปรระบบที่กล่าวมาแล้วไปเก็บยังตัวแปรที่กำหนด และตั้งค่าตัวแปรระบบใหม่ให้สอดคล้องกับการทำงานของโปรแกรม
- ขั้นต่อมาจะเรียกแฟ้มดีซีแอลเพื่อสร้างรูปกรอบสนทนา ตามที่กำหนดไว้ในแฟ้มดีซีแอลนั้นๆ และแสดงผลกรอบสนทนาออกมาทางจอภาพ
- ถ้าการแสดงผลกรอบสนทนาเรียบร้อย จะทำการกำหนดค่าเริ่มต้นให้กับไทล์ต่างๆ เตรียมไทล์ต่างๆให้พร้อมใช้งาน เรียกภาพจากสไลด์แฟ้มที่กำหนดไว้มาแสดงผลใน Image ทำการสร้างรายการตัวเลือก ในขั้นตอนนี้เรียกว่า "Initializing tile"
- ต่อมาจะกำหนดการทำงานให้กับไทล์ต่างๆ โดยใช้ฟังก์ชัน action_tile เมื่อมีการใช้งานไทล์ใดก็จะส่งผ่านการทำงานไปยังฟังก์ชันที่กำหนดสำหรับไทล์นั้น อาทิเช่น ฟังก์ชัน GET_CNAME GET_MB GET_CVR ฯลฯ ประเด็นที่สำคัญ คือ จะต้องมีการกำหนดฟังก์ชันการทำงานให้กับ Accept key (ปุ่ม OK) และ Cancel key (ปุ่ม Cancel) เมื่อมีการกดเลือกปุ่มใดปุ่มหนึ่งนี้ จะเป็นการจบการทำงานกับกรอบสนทนา และปิดกรอบสนทนา ด้วยฟังก์ชัน done_dialog

การจะตรวจสอบว่าผู้ใช้จบการทำงานกับกรอบสนทนาด้วยวิธีใด ต้องกำหนดตัวเลข Index ให้กับฟังก์ชัน done_dialog เช่น กำหนดว่าผู้ใช้ยอมรับค่าต่างๆในกรอบสนทนา กดเลือกปุ่ม OK จะเป็น(done_dialog 1) ฟังก์ชัน done_dialog จะให้ค่าตัวเลข 1 และเมื่อผู้ใช้ยกเลิกการทำงานกับกรอบสนทนา กดเลือกปุ่ม Cancel จะเป็น(done_dialog 0) ฟังก์ชัน done_dialog จะให้ค่าตัวเลข 0 ตัวเลขเหล่านี้ใช้ในการตรวจสอบความประสงค์ของผู้ใช้

- ขั้นตอนสุดท้ายโปรแกรมจะตรวจสอบการจบกรอบสนทนาของผู้ใช้ ถ้าผู้ใช้ยกเลิกกรอบสนทนา โดยการกดเลือก Cancel โปรแกรมจะปิดกรอบสนทนา และคืนค่าตัวแปรระบบตามเดิม แต่ถ้าผู้ใช้ยอมรับค่าต่างๆในกรอบสนทนา กดเลือกปุ่ม OK แสดงว่าผู้ใช้ยอมรับค่าต่างๆในกรอบสนทนา และต้องการทำงานเขียนแบบโครงสร้างนั้น โปรแกรมจะทำการปิดกรอบสนทนา กำหนดค่าให้กับตัวแปรส่วนกลาง และเรียกใช้ฟังก์ชัน G_C1 เพื่อทำการเขียนแบบโครงสร้างนั้นต่อไป