

บทที่ 8
ผังไฮโป (HIPO Chart)

วัตถุประสงค์ของบทนี้

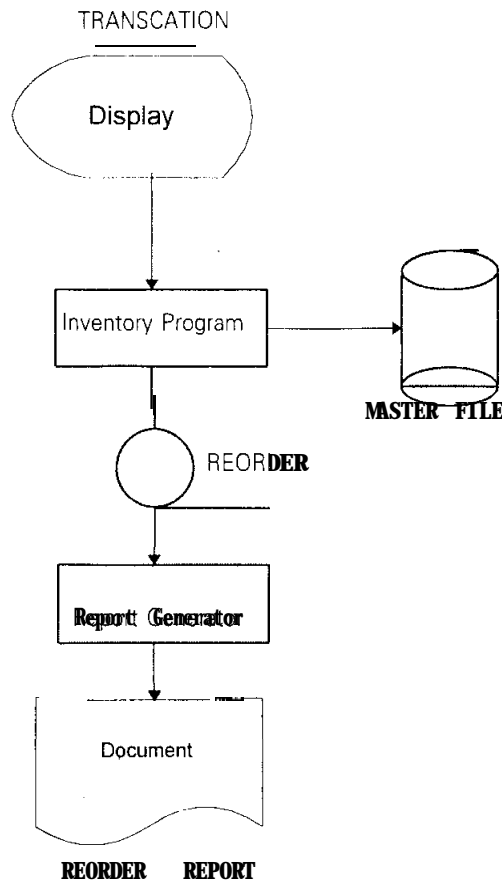
1. การใช้เครื่องมือในการออกแบบระบบ
2. ประโยชน์ของการนำผังไฮโปไปใช้งาน
3. วิธีการออกแบบระบบโปรแกรมโดยใช้ผังไฮโป

ผังไฮโป (Hierarchical Plus Input Processing and Output :HIPO)

ผังไฮโปเป็นผังที่พัฒนาโดยบริษัทไอบีเอ็มเพื่อที่จะช่วยให้ผู้เขียนโปรแกรมใช้เป็น Structure Tool ในการนำไปสู่การเขียนโปรแกรมของระบบต่อไป

ผังไฮโปนั้นจะแสดงลำดับชั้นของการปฏิบัติงานของระบบโปรแกรม โดยที่ผังดังกล่าวจะรูปแบบที่ลำดับจากบนลงล่าง (Top Down) โดยที่โปรแกรมที่แตกมาเป็นโปรแกรมส่วนนั้นเราจะเรียกว่า โมดูล (Module) โดยที่โมดูลที่เราแบ่งย่อยออกมานั้นก็คือ โปรแกรมย่อย (Procedure) หนึ่งๆที่เราจะเรียกใช้ไปปฏิบัติงานนั่นเอง โดยปกติเกณฑ์ในการแตกออกเป็นโมดูลย่อยนั้นมีหลักการก็คือ โมดูลย่อยนั้นจะมีลักษณะดังนี้คือไม่ควรจะเป็นโปรแกรมที่ยาวเกินไปจนเกินความสามารถที่เราจะไล่ตรวจสอบได้ และโมดูลนั้นควรจะปฏิบัติงานเพียงกิจกรรมเดียวเสร็จ (Single , Complete Logical Function)

การเขียนไฮโป นั้นสามารถดำเนินการได้โดยอาศัยผังของระบบ (System Flow Chart) หรืออาศัยจากผังกระแสข้อมูล(DFD) ให้พิจารณาจากตัวอย่างในระบบตรวจสอบคลังสินค้าต่อไปนี้

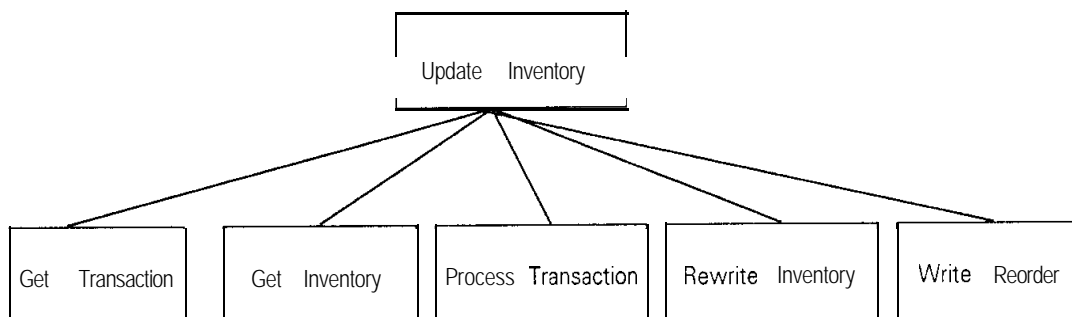


ภาพที่ 8.1 ระบบการสั่งซื้อ

จากผังระบบที่ปรากฏอยู่นั้น เราสามารถที่จะนำไปออกแบบผังไฮโป เพื่อที่จะนำออกแบบระบบโปรแกรมใน ระบบงานจัดการเรื่องสินค้าคงคลังออกมาได้ดังกิจกรรมที่จะดำเนินการต่อไปนี้

1. Get a Transaction
2. Get master record
3. Process the transaction
4. Rewrite the matter record
5. Write a reorder record (if necessary)

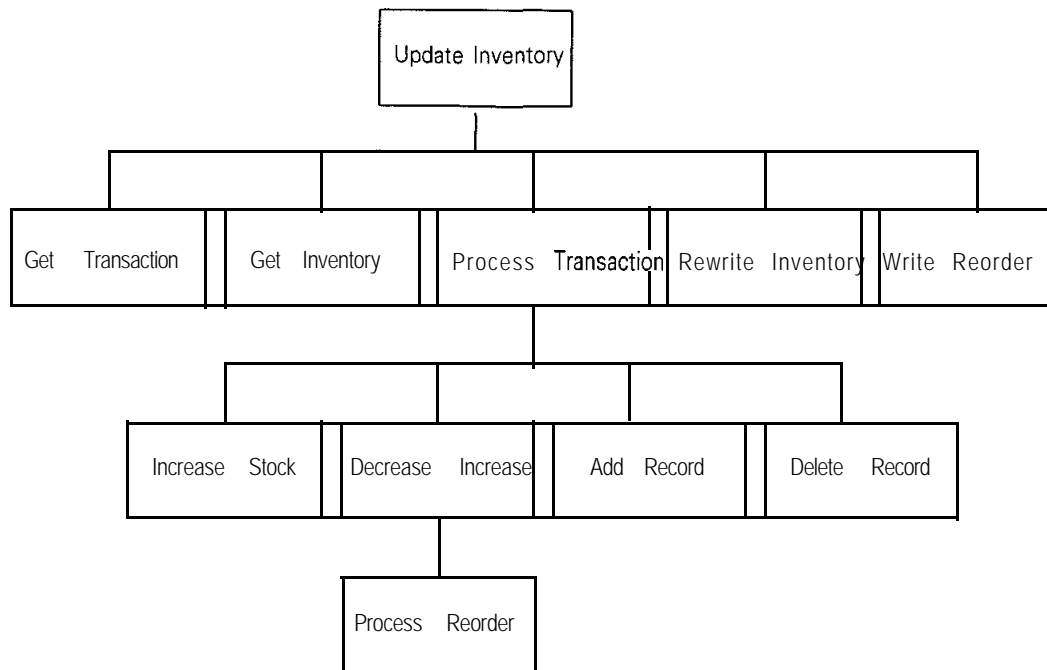
กิจกรรมที่กล่าวถึงเหล่านี้ เราจะรวมเรียกว่าการปรับแก้ยอดสินค้าคงคลัง (Inventory Control) โดยที่เราจะนำกิจกรรมเหล่านี้ไปเขียนเป็นผังไฮโปในระดับที่ 1 ได้ดังนี้



ภาพที่ 8.2 ผังไฮโปแสดงระบบโปรแกรมการปรับยอดสินค้า

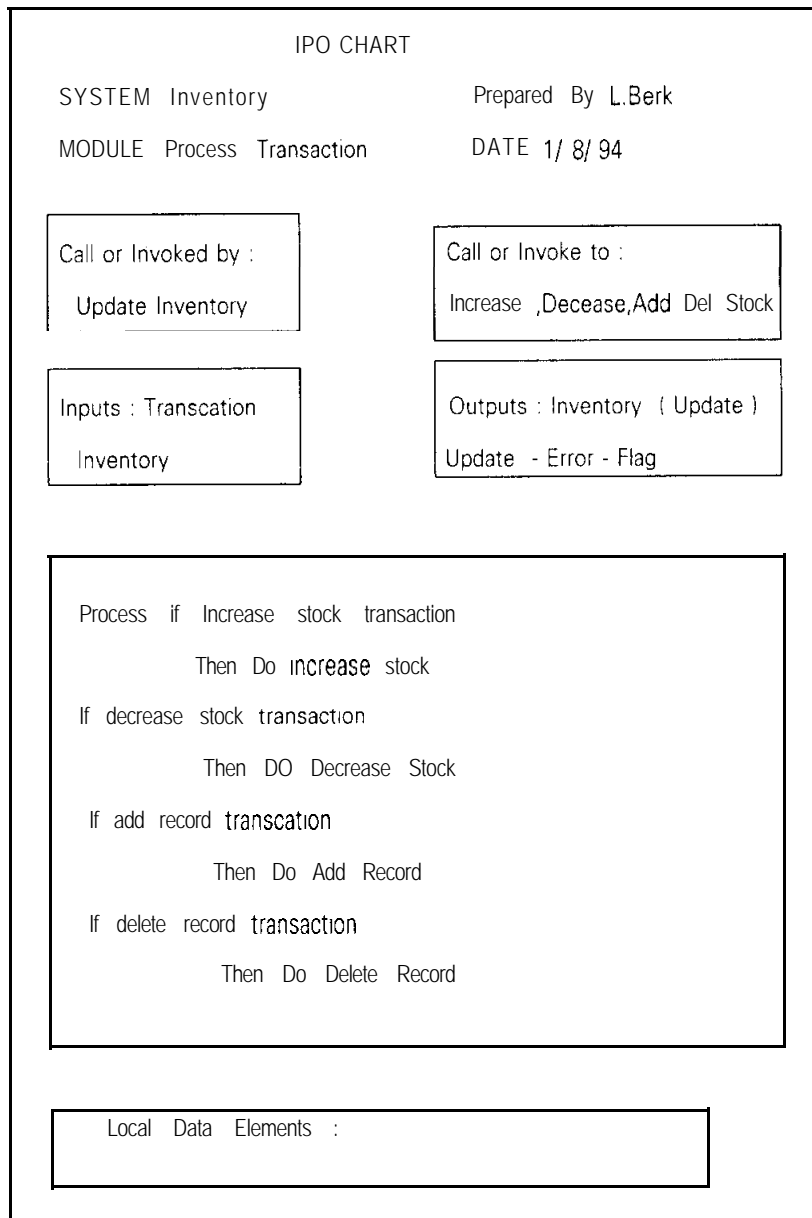
เมื่อเรานำผังไฮโปในระดับที่ 1. ไปกระจายกิจกรรมที่ปรากฏภายในกิจกรรมของ Process Transaction แล้วเราจะสามารถจำแนกกิจกรรมออกได้เป็นกิจกรรมย่อยๆเป็นผังไฮโปรูปที่ 2 ได้ดังนี้

ผังไฮโปในระดับที่ 2 ปรากฏดังนี้

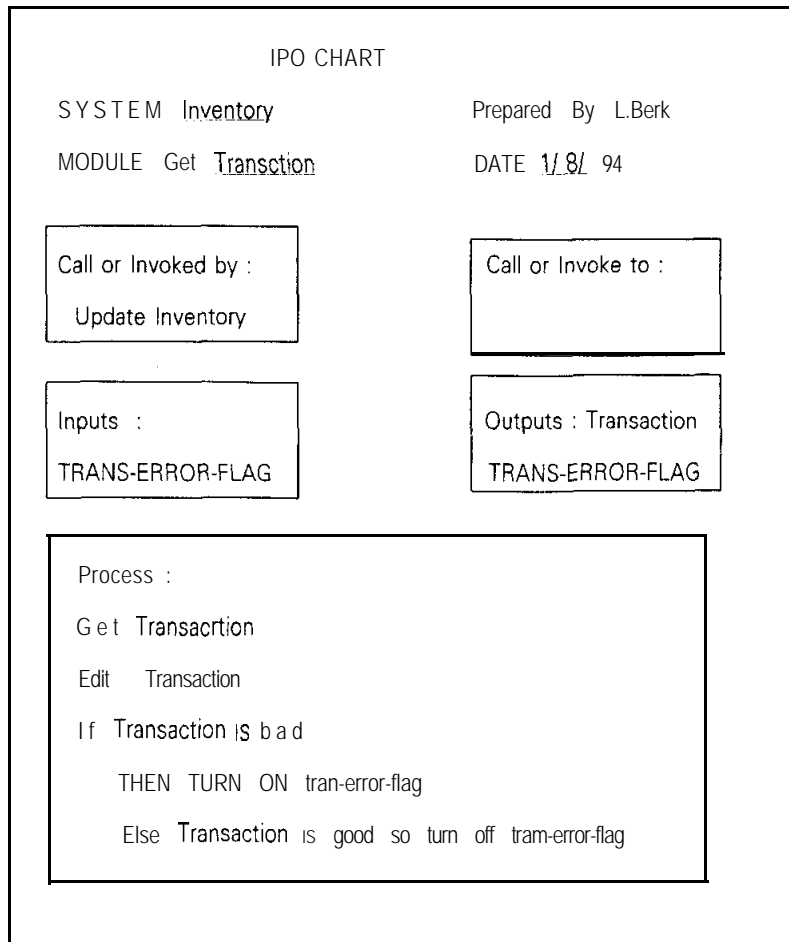


ภาพที่ 8.3 แสดงโปรแกรมย่อยๆต่าง

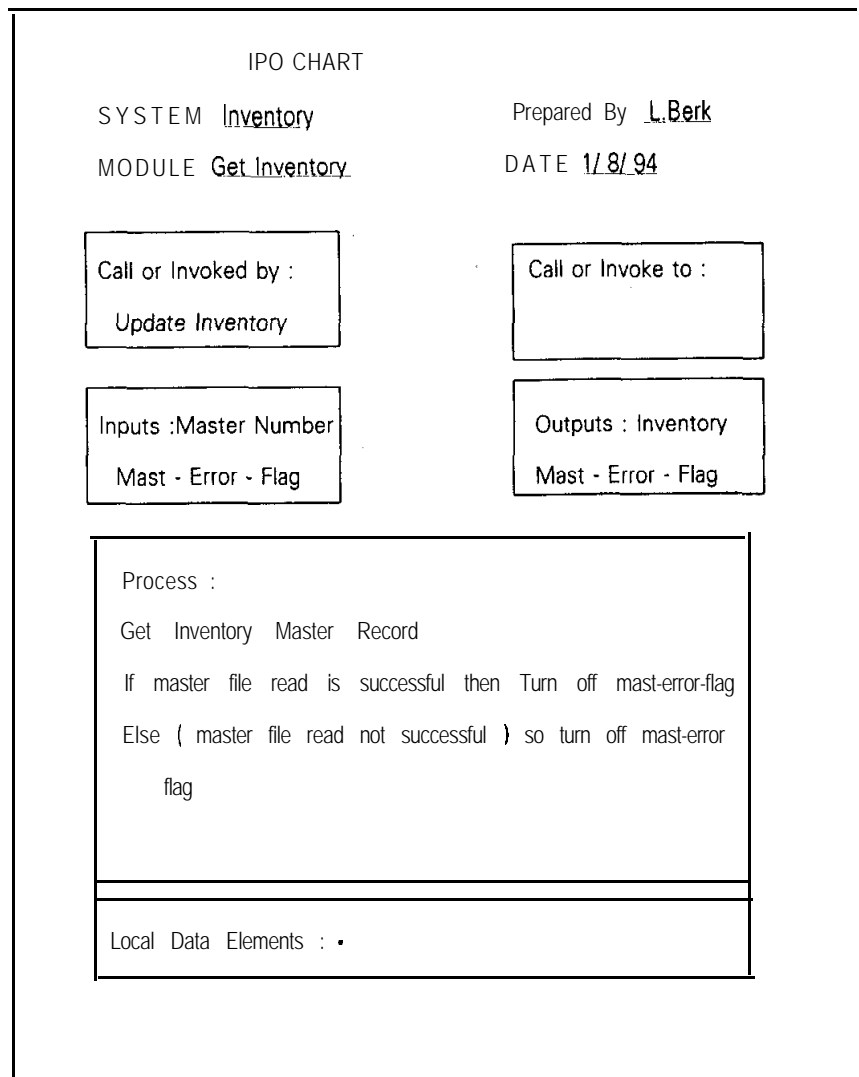
ภายหลังออกแบบไฮโปแล้วเราจะเขียนผังรายละเอียดของไฮโปได้ดังภาพที่ 8.4 โดยที่วิธีการเขียนผังไฮโปนั้นเราสามารถกระทำได้หลายรูปแบบ ดังตัวอย่างวิธีการแบบหนึ่งดังนี้คือ



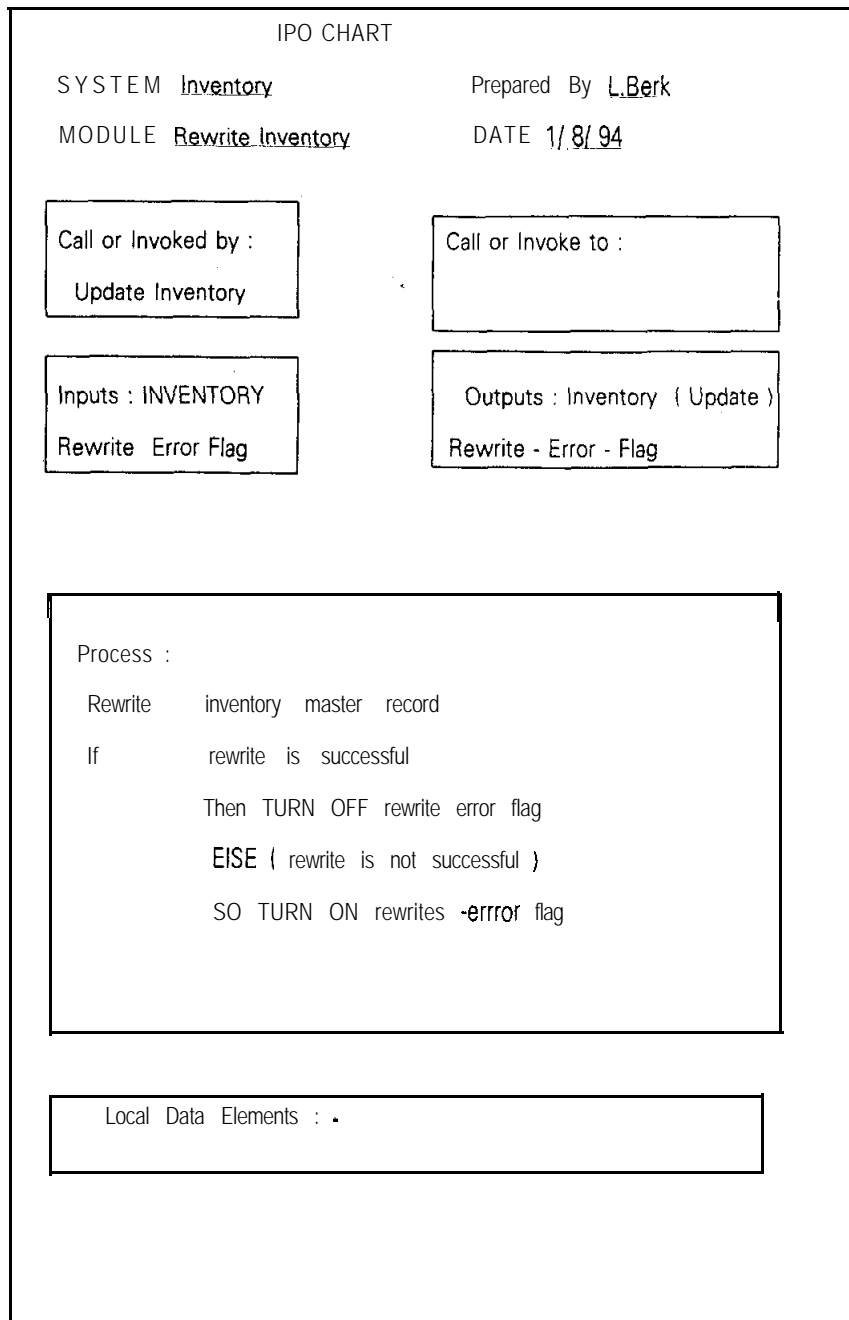
ภาพที่ 8.4 (ต่อ)



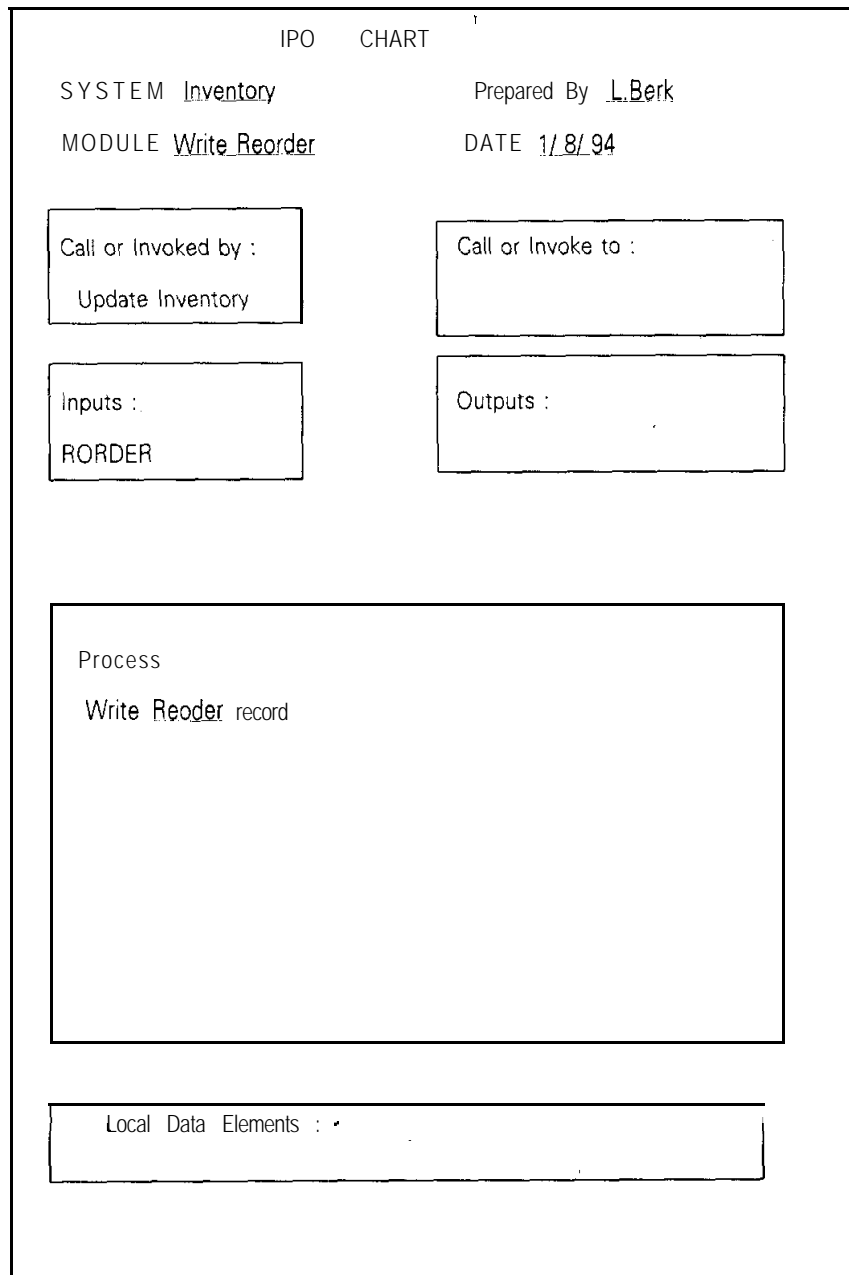
ภาพที่ 8.4 การเขียนรายละเอียดในผังไอโป



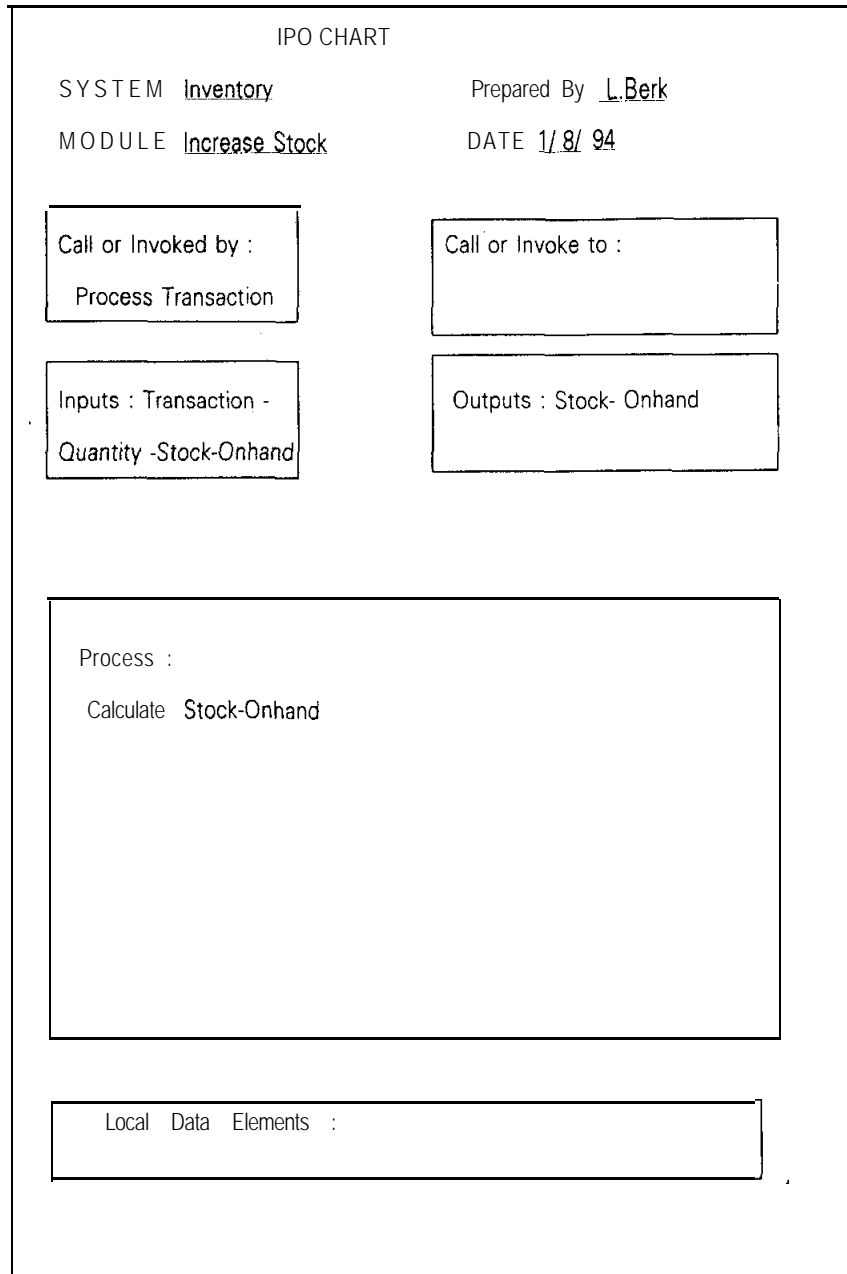
ภาพที่ 8.4 (ต่อ)



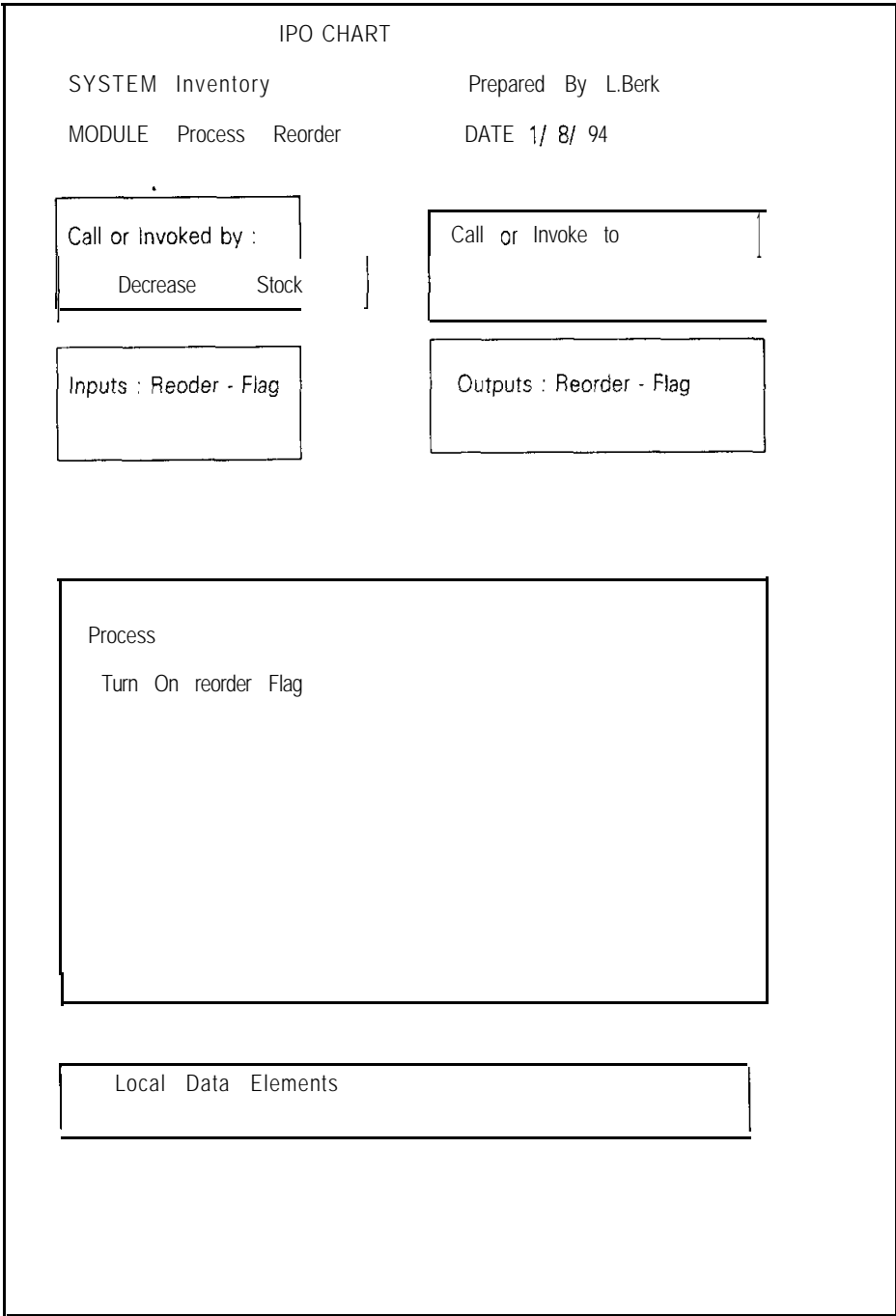
ภาพที่ 8.4 (ต่อ)



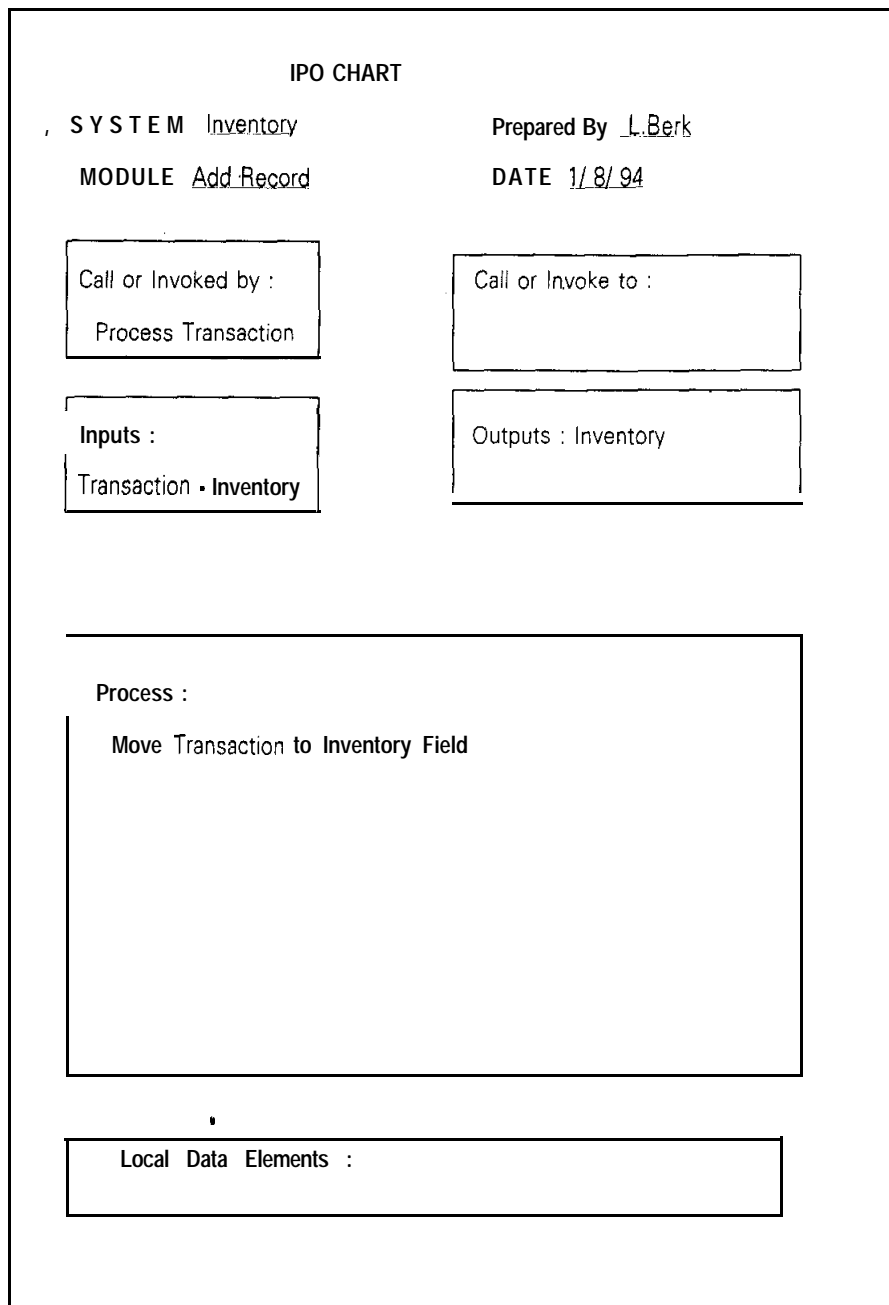
ภาพที่ 8.4 (ต่อ)



ภาพที่ a.4 (ต่อ)



ภาพที่ 8.4 (ต่อ)



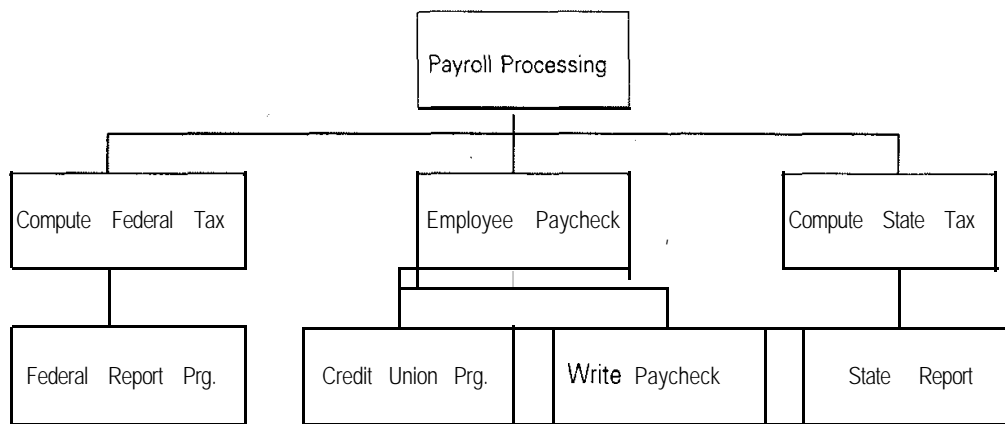
ภาพที่ 8.4 (ต่อ)

หลักการเขียนผังไฮโป

ผังดังกล่าวจะประกอบด้วยองค์ประกอบอยู่ 3 ส่วนคือ

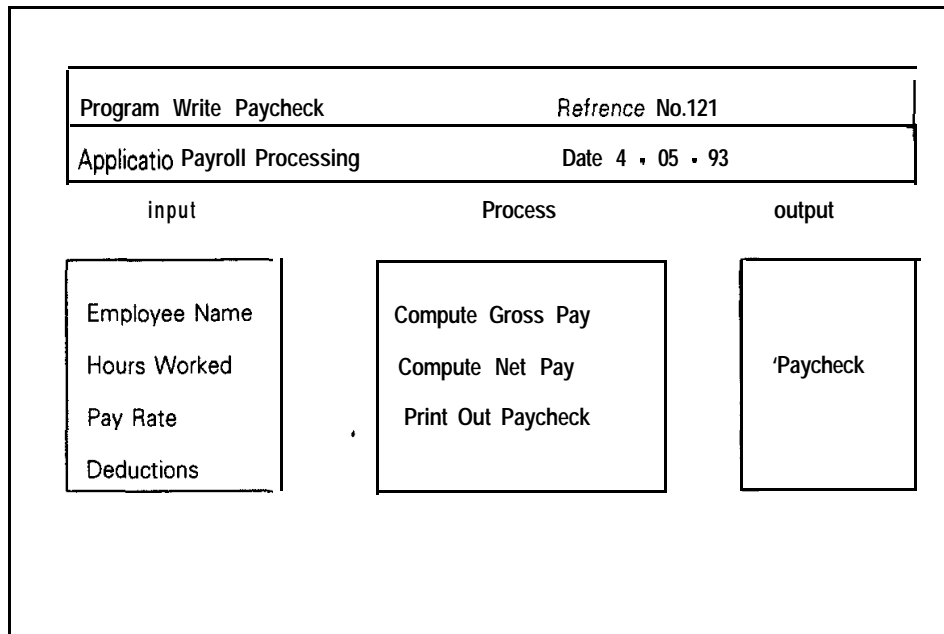
1. ผังแสดงลำดับการปฏิบัติการของระบบ เราอาจจะเรียกได้ว่าเป็น Top Down Structure of the Program หรือที่เรียกได้อีกอย่างว่าเป็น Visual table of content
2. แสดงถึงองค์ประกอบคือ IPO
3. แสดงถึงรายละเอียดของ IPO

ภาพแสดงถึง HIPO ของระบบการคิดบัญชีเงินเดือน



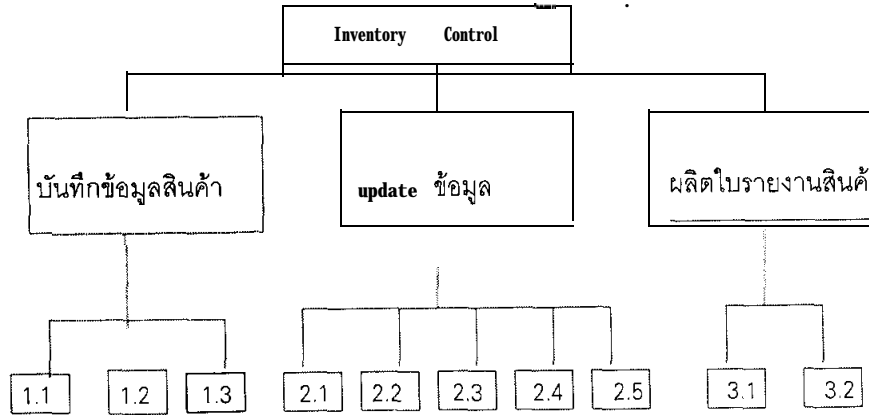
ภาพที่ 8.5 ผังไฮโปของการคิดบัญชีเงินเดือน

ส่วนการเขียน IPO ของแต่ละโปรแกรมนั้นเราอาจเขียนได้ในลักษณะรูปต่อไปนี้คือ
 (ในที่นี้จะใช้โปรแกรม Write Paycheck เป็นตัวอย่างในการเขียนผังไอโปกอีกแบบที่นิยมใช้กัน



ภาพที่ 8.6 การเขียนผังไอโปกอีกรวิธีที่นิยมใช้กัน

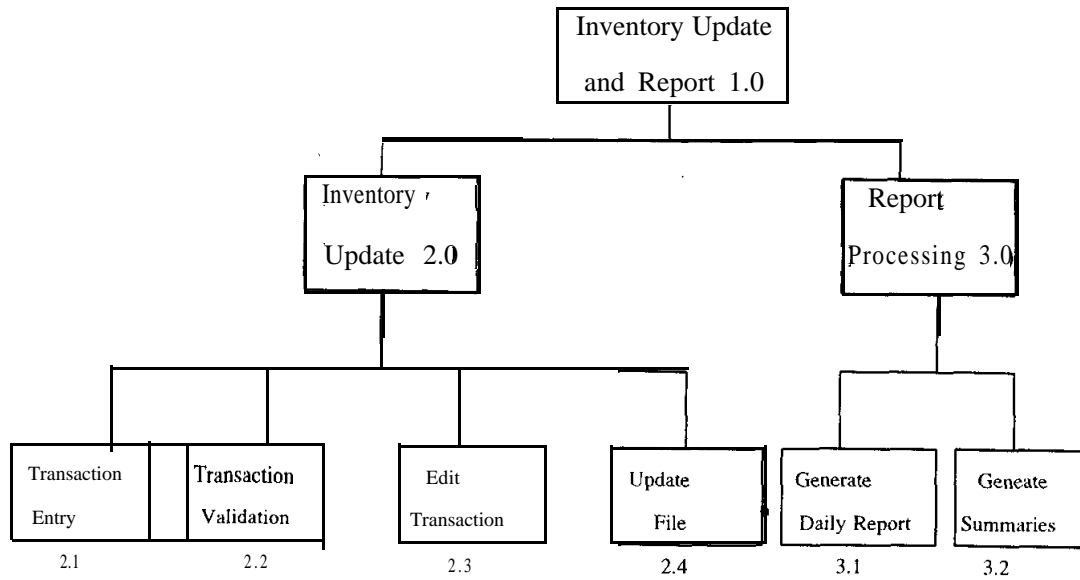
ตัวอย่างแสดงระบบการควบคุมสินค้าคงคลัง แสดงด้วยผัง HIPO จากฝ่าย ผลิตสินค้า
(Production)



ภาพที่ 8.7 แสดงการแบ่งเป็นโปรแกรมย่อยๆ

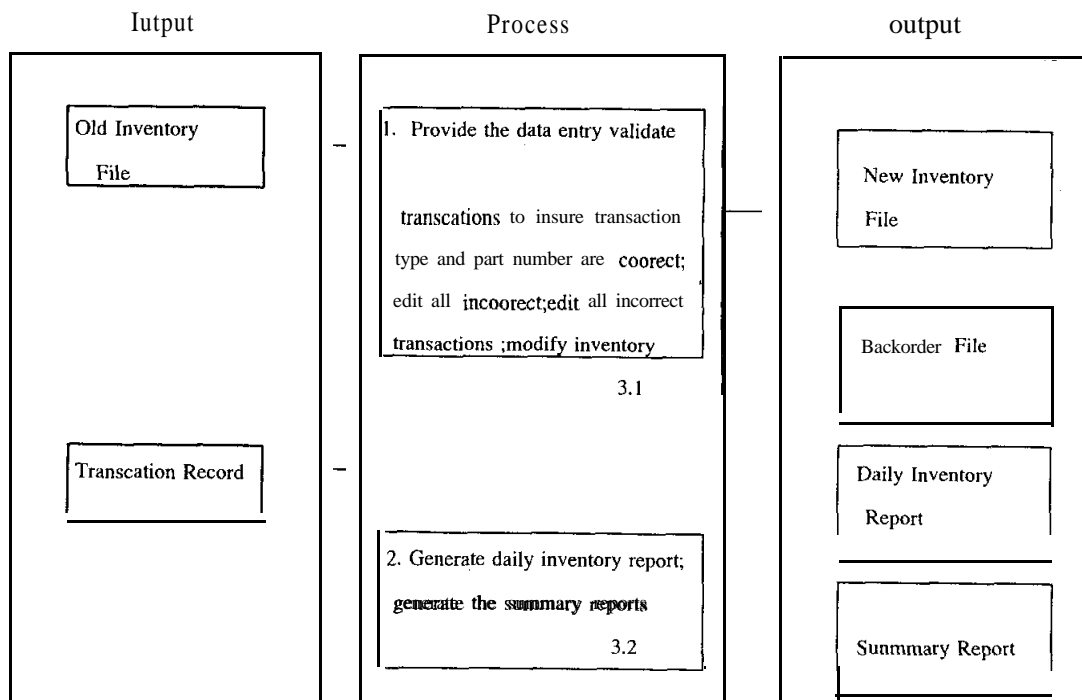
- 1.1 = การรับใบสั่งซื้อจากลูกค้า
- 1.2 = ระบุถึงเพิ่มสินค้าที่เกี่ยวข้องด้วย
- 1.3 = แสดงผลใบสั่งซื้อที่ผิดพลาด
- 2.1 = หาจำนวนใบสั่งซื้อที่จะยกเลิก
- 2.2 = การลดจำนวนสินค้าคงคลัง
- 2.3 = ปรับปรุงยอดขายรวม
- 2.4 = เปลี่ยนแปลงวันที่ ที่ทำงาน
- 2.5 = คำนวณสั่งซื้อสินค้า ที่ต้องสั่งซื้อ
- 3.1 = แสดงรายการที่ต้องแสดงออกมา
- 3.2 = พิมพ์รายงานบอกสถานภาพของใบสั่งซื้อ

ตัวอย่างต่อไปนี้จะเป็นการเขียนไฮโปในระบบการทำเรื่องควบคุมคลังสินค้า เช่นเดียวกับฟังก์ชันที่ปรากฏก่อนหน้านี้ แต่เราจะแตกรายละเอียดแตกต่างไป พร้อมกันนั้นจะแสดงภาพของไฮโปด้วย



ภาพที่ 8.8 แสดงไฮโปของระบบควบคุมสินค้าคงคลัง

ภาพแสดงฟังก์ชันของไฮโปจากโปรแกรมย่อยที่ 3.0



ภาพที่ 8.9 ไฮโปของโปรแกรมย่อยที่ 3.0

การเขียน HIPO จุดประสงค์เพื่อแสดงถึง Module หรือ Function ที่ใช้ทำงาน และจะต้องดูว่าแต่ละ Module นั้น สามารถกระจายออกทำงานอื่นอีกได้หรือไม่ ถ้าทำได้จะต้องกระจายจนถึง Module สุดท้ายที่มีลักษณะเป็น Single logical Function คือ ไม่สามารถกระจายต่อไปได้อีก

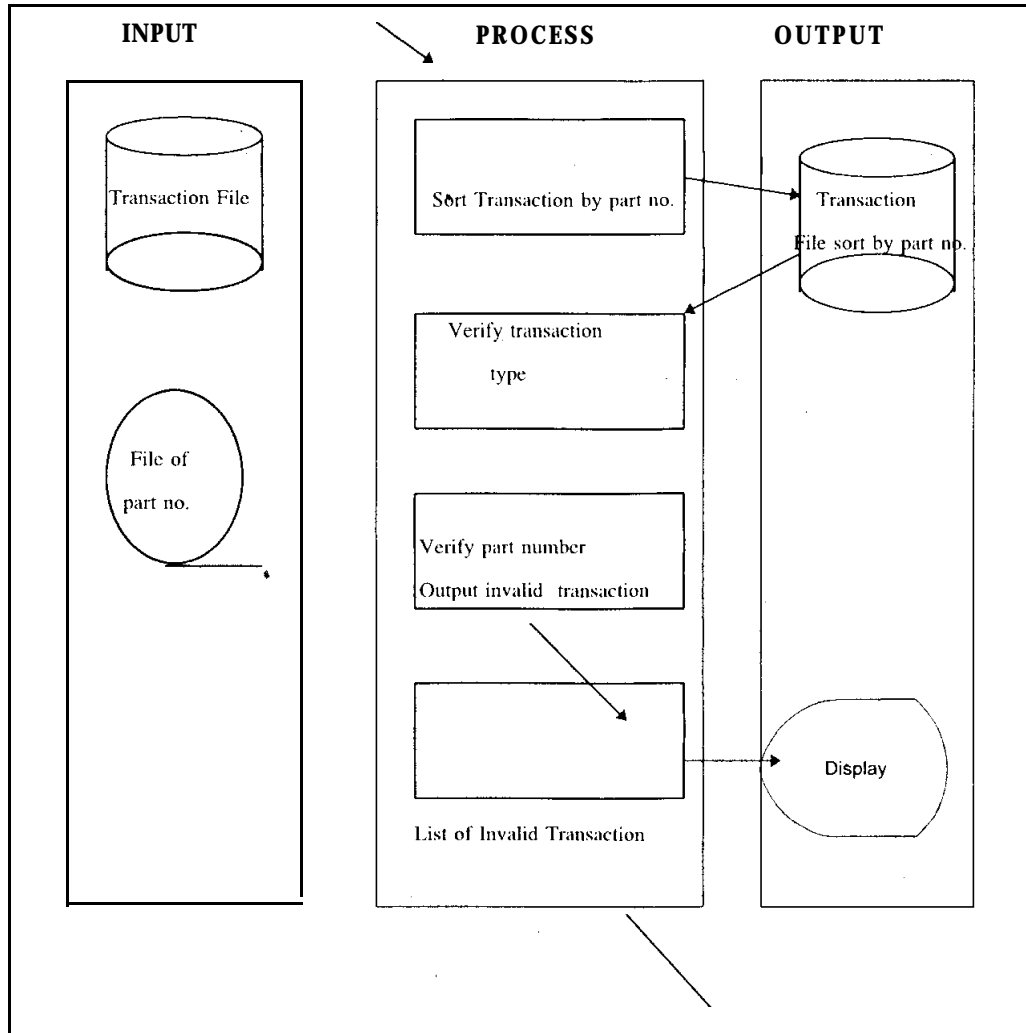
การใช้ประโยชน์จาก HIPO

1. ใช้ประเมินและตรวจสอบระบบโปรแกรมที่ออกแบบไว้
2. ใช้ตรวจสอบหาข้อผิดพลาดก่อนที่จะนำไปเขียนโปรแกรม
3. ผู้เขียนโปรแกรมใช้เป็นคู่มือในการดูแล รักษา ปรับปรุงโปรแกรมในระบบ

หลักการในการออกแบบโปรแกรมย่อยที่เราเรียกว่าโมดูลนั้นเราควรคำนึงปัจจัยต่อไปนี้

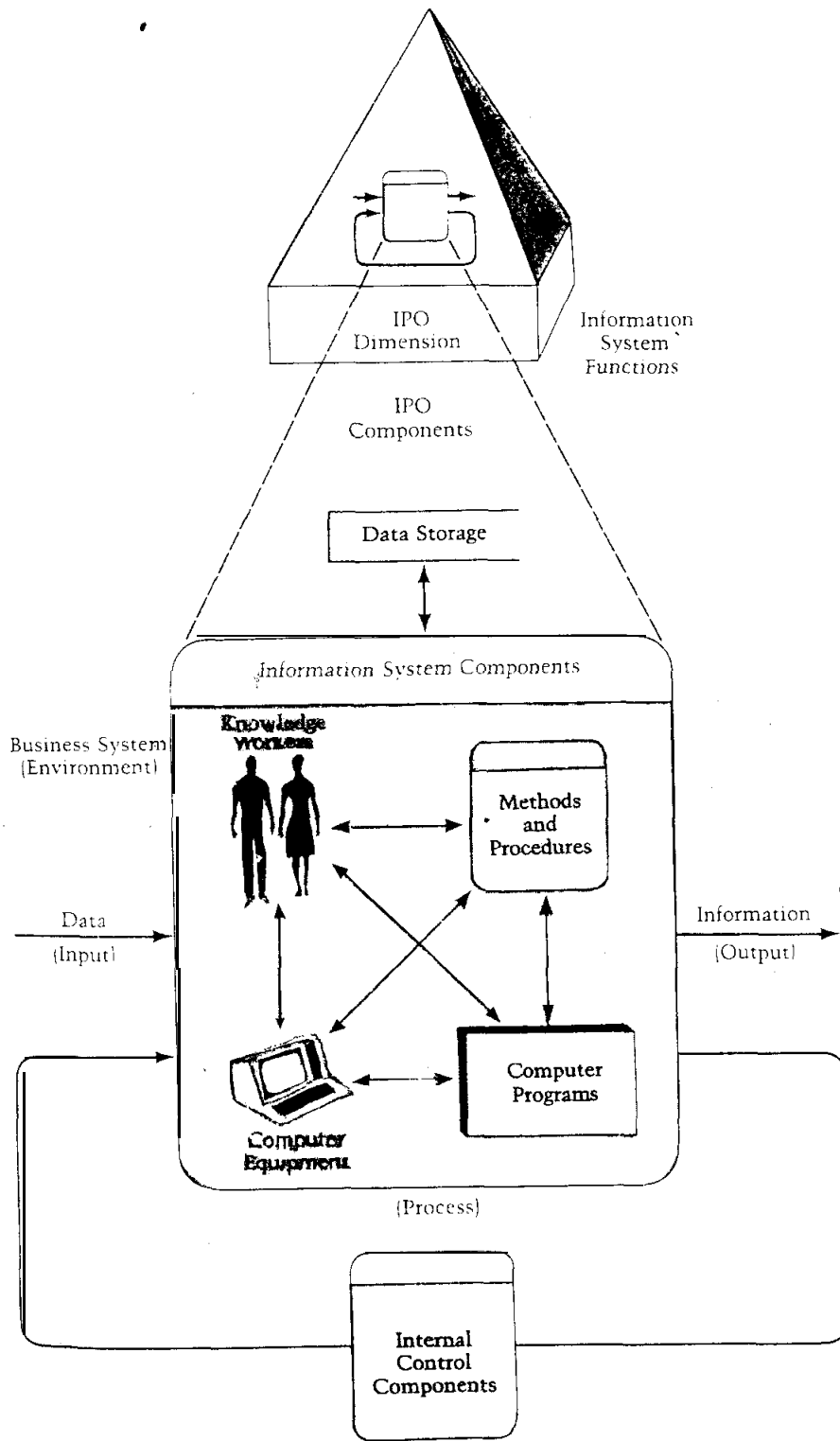
1. ง่ายแก่การใช้งาน (Simplicity) การออกแบบที่ใช้นั้นนับว่าสิ่งที่ดีสำหรับผู้ที่นำไปใช้งาน
2. พยายามให้โมดูลนั้นเป็น โมดูลที่สามารถทำงานเพียง 1 กิจกรรม (Modular Structure)
3. พยายามให้ระบบนั้นประกอบด้วยส่วนของ Audit Trail เพื่อที่เราตรวจสอบระบบข้อมูล เพื่อไว้ใช้ตรวจสอบได้ เวลามีปัญหา
4. การออกแบบระบบนั้น เราจะคำนึงถึงผู้ใช้เป็นหลัก (End User Orientation) แต่เราจะต้องคำนึงถึงทรัพยากรคือระบบเครื่องคอมพิวเตอร์ด้วย
5. การออกแบบระบบนั้นจะต้องคำนึงถึง ลักษณะการใช้งานของมนุษย์ร่วมด้วย (User Friendly)
6. ระบบที่ดีจะต้องมีความเที่ยงตรง (Consistent)
7. ระบบที่ดีจะต้องมีประสิทธิภาพทั้งในแง่ของการใช้งาน และการใช้ทรัพยากรของระบบ (Efficiency)
8. ระบบที่ดีจะต้องมีมาตรการควบคุมรักษาความปลอดภัย (Security and Control) ในแง่ความปลอดภัยนั้น โปรแกรมจะต้องดูแลทุกส่วนของระบบ

ในกรณีของตัวอย่างไฮโปภาพนี้เราสามารถที่จะเขียนผังไอโป ในลักษณะของการใช้อุปกรณ์ประกอบรายละเอียดของโมดูลนั้นจะสามารถเขียนได้ดังนี้ (จะใช้โมดูลที่ 2 เป็นตัวอย่าง)



To diagram 2.3

ภาพที่ 8.10 การเขียนผังไอโปโดยเน้นที่อุปกรณ์



ภาพที่ 8.11 แสดง Input-Process-Output ซึ่งเป็นส่วนหนึ่งในระบบสารสนเทศ

แบบฝึกหัดท้ายบท

1. ผังไฮโปคืออะไร เจตนาจะใช้เพื่อประโยชน์อะไร
2. จงออกแบบระบบบัญชีเงินเดือนของพนักงานที่คิดค่าจ้างรายวัน โดยที่จะจ่ายเงิน
ทุกๆ 2 สัปดาห์ กำหนดเงื่อนไขที่ปฏิบัติตามระบบงานโดยทั่วไป
3. จงเปรียบเทียบการใช้ผังไฮโปแต่ละแบบที่ใช้กันว่า มีข้อดีข้อเสียแตกต่างกันอย่างไร
4. ผังไฮโปนั้น เรานำไปใช้ประโยชน์ในงานใด