เอกสารอ้างอิง

1. สุรศักดิ์ นานานุกูล, การบริหารงานผลิต (กรุงเทพมหานคร: สำนักพิมพ์ไทยวัฒนาพานิช, 2517)

2. Marvin V. Zelkowitz in Principles of Software Engineering and Design, Prentice-Hall, Inc., Englerwood Cliffs, N.J. 1979.

3. R.W. Bemer in Manageable Software Engineering, Software Engineering VOLUME-I, Academic Press, Inc., (1970).

4. Jean E. Sammet in Perspective on Methods of Improving Software Development, Software Engineering VOLUME-I, Academic Press, Inc., (1970).

5. C.E. Walston and C.P. Felix "A Method of Programming Measurement and Estimation" IBM Syst.J. 16, (1977).

6. Victor R. Basili, Karl Freburger "Programming Measurement and Estimation in the Software Engineering Laboratory", The Journal of System and Software 2, (1981):47-57.

7. Earl Chrysler, "Some Basic Determinants of Computer Programming Productivity" Communications of the ACM, Vol. 21 No. 6., June 1978.

8. Asian Institute of Technology, "Software Engineering." Asian Institute of Technology, 1979

9. Edward T. Chen, "Program Complexity and Programmer Productivity" IEEE Transactions on Software Engineering, Vol. SE - 4, No. 3, May 1978.

10. เอกชัย ชัยประเสริฐสิทธิ์, การวิเคราะห์ซอฟต์แวร์และการลดถอย, ฉบับพิมพ์ครั้งที่ 3, กรุงเทพมหานคร: สำนักพิมพ์มหาวิทยาลัยธรรมศาสตร์, 2525

11. ผศ.สมจิต วัฒนาชยากุล, "การวิเคราะห์การลดถอยและสหสัมพันธ์" สถิติ-วิเคราะห์เบื้องต้น, (กรุงเทพมหานคร: สาขาวิชาคณิตศาสตร์-สถิติ คณะศิลปศาสตร์ มหาวิทยาลัยธรรมศาสตร์, เมษายน 2524)

12. กำจัด เขียวขจี, "วิธีการควบคุมและติดตามผลโครงการ", ประสบการณ์การบริหารงานคอมพิวเตอร์โดยความร่วมมือของกลุ่มผู้ใช้คอมพิวเตอร์ (BESSTT GROUP), สมาคมคอมพิวเตอร์แห่งประเทศไทย, 15 พฤษภาคม 2524.

13. Tadashi Yoshioka in Programming Standards and its Technology, Standardized Technology and Engineering for Programming Support, Center of the International Cooperation for Computerization, (1982)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

การวัดผลผลิตโดยวัดจากจำนวนบรรทัดของโปรแกรมอย่างเดียวยังไม่สมบูรณ์นัก จำนวนบรรทัดของโปรแกรมเปรียบเสมือนปริมาณ จะมีวิธีวัดคุณภาพของผลผลิตนั้นได้หรือไม่ มีผู้เสนอแนะวิธีการวัดโดยนำเอาเวลาทำงานของคอมพิวเตอร์สำหรับงานนั้นและจำนวนเนื้อที่หน่วยความจำ ซึ่งโปรแกรมนั้นใช้ไปทั้งหมดมารวมในการพิจารณาด้วย ตัวอย่างเช่น สมมติให้มีผู้ทำโปรแกรม 2 คน คือ นาย ก. และ นาย ข. นาย ก. ใช้เวลา 2 สัปดาห์ในการทำโปรแกรมและแก้ไขโปรแกรมจนใช้งานได้ถูกต้อง โปรแกรมของนาย ก ใช้เวลาคอมพิวเตอร์ 1 นาที มีจำนวนทั้งหมด 400 คำสั่ง ส่วนนาย ข. ใช้เวลา 4 สัปดาห์ ทำงานชิ้นเดียวกัน ใช้เวลาของคอมพิวเตอร์ 0.5 นาที จำนวนคำสั่ง 600 บรรทัด เราจะถือว่านาย ข. มีผลผลิตมากกว่านาย ก. หรือไม่ ? ไม่มีคำตอบที่แน่นอนตายตัว ต้องขึ้นอยู่กับวัตถุประสงค์ของหัวหน้างานที่กำหนดมา ถ้าเราวัดด้วยจำนวนบรรทัดของโปรแกรมที่ผลิตได้ต่อวัน จะพบว่า นาย ก. มีผลผลิตมากกว่า นาย ข. เกือบ 2 เท่า แต่ทว่าเวลาที่นาย ข. ใช้มากกว่านาย ก. นั้น นาย ข. ใช้ในการลดเวลาทำงานของคอมพิวเตอร์ลงเหลือเพียงครึ่งหนึ่งของ นาย ก.

จากองค์ประกอบต่อไปนี้ คือ

1. เวลาที่ใช้ไปในการทำโปรแกรม
2. เวลาที่คอมพิวเตอร์ใช้ไปในการทำโปรแกรมนั้น
3. ขนาดของหน่วยความจำที่โปรแกรมนั้นใช้

ผู้จัดการฝ่ายคอมพิวเตอร์สามารถพิจารณาผลผลิตของผู้ทำโปรแกรมแต่ละคนด้วยองค์ประกอบ 3 ตัวนี้ โดยการกำหนดน้ำหนักความสำคัญให้กับองค์ประกอบแต่ละตัวต่าง ๆ กัน เช่น ถ้าเน้นว่างานต้องเสร็จภายใน 2 สัปดาห์ ส่วนเวลาคอมพิวเตอร์ที่ใช้และจำนวนหน่วยความจำที่ใช้มีความสำคัญน้อยกว่า หรือจะเน้นที่การใช้เวลาคอมพิวเตอร์ให้น้อยลงก็เพิ่มน้ำหนักของคะแนนให้แก่องค์ประกอบเรื่องเวลาของคอมพิวเตอร์

ในการหาค่าเปรียบเทียบเราใช้สูตร

$E = W_1$* เวลาที่ใช้ในการทำโปรแกรม + W_2 * เนื้อที่หน่วยความจำที่ใช้ +
W_3* เวลาทำงานของคอมพิวเตอร์

E = ผลลัพธ์ของการวัด

W_n= น้ำหนักซึ่งผู้จัดการกำหนดให้แก่องค์ประกอบแต่ละตัว

เพื่อความสะดวกในการกำหนดค่าของ W จึงให้ผลรวมของ

$$W_1 + W_2 + W_3 = 1$$

และเพื่อให้ค่าตัวเลขในทางปฏิบัติสามารถนำมาเปรียบเทียบได้ คือให้ค่าขององค์ประกอบแต่ละตัวของผู้ทำโปรแกรมคนที่ดีที่สุดมีค่าเป็น 1 และมีค่าลดหลั่นกันลงมาสำหรับผู้ทำโปรแกรมคนอื่น ๆ

ค่าขององค์ประกอบ	นาย ก.	นาย ข.
เวลาที่ใช้ไปในการทำโปรแกรม	1	0.5
เนื้อที่หน่วยความจำที่ใช้	1	0.667
เวลาที่คอมพิวเตอร์ทำงาน	0.5	1.0

จากตัวอย่างของนาย ก. และ นาย ข. สมมุติให้ทั้ง 2 คน ได้คะแนนขององค์ประกอบตามตาราง

– ถ้าผู้จัดการมีความเห็นว่าเวลาที่ใช้ไปในการทำโปรแกรมและเนื้อที่หน่วยความจำมีความสำคัญเท่ากัน แต่เวลาที่คอมพิวเตอร์ทำงานมีความสำคัญมากกว่าองค์ประกอบ 2 ตัวแรก 8 เท่า จะได้ว่า

$W_1 = W_2 = 0.1$, $W_3 = 0.8$, $W_1+W_2+W_3=1.0$ แทนค่าในสมการ

E (นาย ก) $=0.1\ (1) +0.1\ (1) + 0.8\ (0.5) = 0.6$

E (นาย ข) $= 0.1\ (0.5) + 0.1\ (0.667) + 0.8\ (1) = 0.917$

จะได้ นาย ข. มีประสิทธิภาพเหนือกว่า นาย ก.

– ถ้าผู้จัดการกำหนดคะแนนความสำคัญใหม่เป็น

$W_1 = 0.6$, $W_2 = 0.1$, $W_3 = 0.3$, $(W_1+W_2+W_3= 1.0)$

แทนค่าในสมการ

E (นาย ก.) $= 0.6\ (1) + 0.1\ (1) + 0.3\ (0.5) = 0.85$

E (นาย ข.) $= 0.6\ (0.5) + 0.1\ (0.667) + 0.3\ (1) = 0.667$

จะพบว่า นาย ก. มีประสิทธิภาพสูงกว่า นาย ข.

– ถ้าผู้จัดการกำหนดใหม่ให้

$W_1 = W_2 = W_3$ จะได้ว่า E (นาย ก.) $= 0.883$

E (นาย ข.) $= 0.772$

ซึ่งมีประสิทธิภาพใกล้เคียงกัน

นี่เป็นวิธีการหนึ่งที่มีผู้เสนอเพื่อการวัดผลผลิตโดยพิจารณาหลาย ๆ องค์ประกอบที่เกี่ยวข้องกับการทำโปรแกรม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

การทำโปรแกรมในรูปแบบของ STEPS

โดยทั่ว ๆ ไป การประมวลผลโปรแกรมแบบกลุ่มของผู้ใช้งานคอมพิวเตอร์ โปรแกรมเหล่านี้เป็นภาษาโคบอลและมีขั้นตอนการทำงานเฉลี่ย 500 ขั้นตอน หรืออย่างมากที่สุดก็ 3,000 ขั้นตอน โปรแกรมเหล่านี้มักจะคล้ายคลึงกัน สิ่งที่แตกต่างกัน คือ ข้อมูลและตำแหน่ง Key

เมื่อโปรแกรมเหล่านี้ถูกนำมาจัดเรียงตามรูปแบบของการประมวลผล เช่น การปรับปรุงให้ทันสมัย (updating) และ การประมวลผลสอบถาม (inquiries) ประมาณร้อยละ 90 จะมีรูปแบบเหมือนกับรูปใดรูปหนึ่งใน รูป ผ. 1

เราจะยกตัวอย่างโปรแกรมของการปรับปรุงข้อมูลให้ทันสมัย ดังรูป ผ.2 มาใช้ในการพิจารณา

ขั้นตอนการทำงานของการ update แฟ้มข้อมูลหลักจะมี ดังนี้

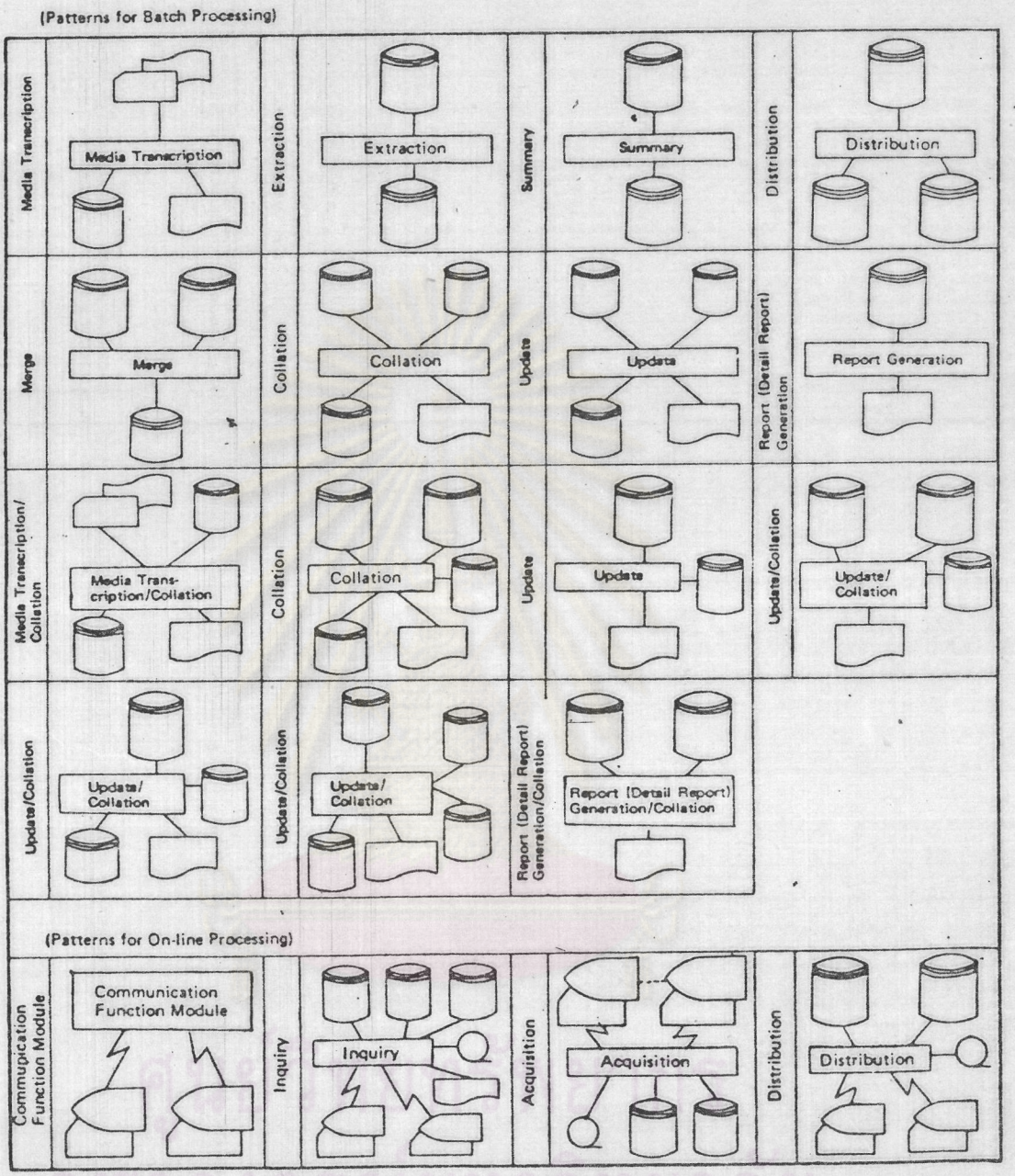
1. หนึ่งระเบียนของข้อมูลจะถูกอ่านออกมาจากแฟ้มรายการเปลี่ยนแปลง ซึ่งจัดเรียงลำดับตามตัวกำหนด (keys) แล้ว

2. หนึ่งระเบียนจะถูกอ่านมาจากแฟ้มข้อมูลหลักซึ่งจัดเรียงลำดับด้วยตัวกำหนดเดียวกัน

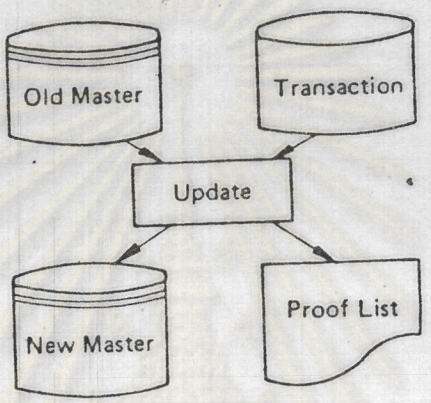
3. ทำการเปรียบเทียบตัวกำหนดของทั้งสองแฟ้มข้อมูล

4. ถ้าแฟ้มรายการเปลี่ยนแปลงมีค่าใหญ่กว่าข้อมูลหลัก ก็ทำการคัดลอกระเบียนนั้นลงในแฟ้มข้อมูลหลักใหม่ จนกระทั่งพบตัวกำหนดที่เหมือนกันหรือใหญ่กว่า

5. ถ้าตัวกำหนดของแฟ้มรายการเปลี่ยนแปลง และแฟ้มข้อมูลหลักมีค่าเท่ากัน ข้อมูลที่บรรจุอยู่ในแฟ้มข้อมูลหลักจะถูกปรับปรุงให้ทันสมัยด้วยข้อมูลในรายการเปลี่ยนแปลง



รูป ผ.1 รูปแบบมาตรฐานของการประมวลผลแบบต่าง ๆ



รูป ผ. 2 รูปแบบมาตรฐานของการปรับปรุงข้อมูลให้ทันสมัย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

6. ถ้าแฟ้มข้อมูลหลักมีค่าใหญ่กว่า แสดงว่าแฟ้มข้อมูลเก่าไม่ได้มีการบันทึกข้อมูลที่มีตัวกำหนดตัวนี้ ข้อมูลในระเบียบของแฟ้มรายการเปลี่ยนแปลงจะถูกศึกษา และถ้าพบว่าเป็นข้อมูลใหม่ ก็จะทำการแทรกข้อมูลนั้นเข้าไป หรือไม่ก็ทำการแจ้งการประมวลผลผิดพลาด

จากขั้นตอนต่าง ๆ ข้างต้น ในขั้นตอนที่ 5 ซึ่งเป็นขบวนการปรับปรุงแฟ้มข้อมูลให้ทันสมัย หรือขั้นตอนที่ 6 ซึ่งเป็นการแทรกข้อมูลหรือการแจ้งข้อผิดพลาดของการประมวลผล ทั้งสองขั้นตอนนี้เป็นส่วนสำคัญในทางธุรกิจ ทั้งสองขั้นตอนนี้เราสามารถนำมาทำอยู่ในรูปแบบมาตรฐานและทำเป็นโปรแกรมมาตรฐานนั่นคือ การทำให้ส่วนของโปรแกรมเหล่านี้เป็นมาตรฐาน ซึ่งสามารถทำได้หลายวิธี แต่จะขอเสนอวิธีการหนึ่งของ STEPS คือ การทำรูปแบบของโปรแกรมให้เป็นมาตรฐาน ซึ่งมีรูปแบบดังนี้

ข้อดีของวิธีการนี้ คือ

1. โปรแกรมต้น (source program) เป็นภาษาโคบอล ซึ่งเป็นภาษาที่มาตรฐานสูง

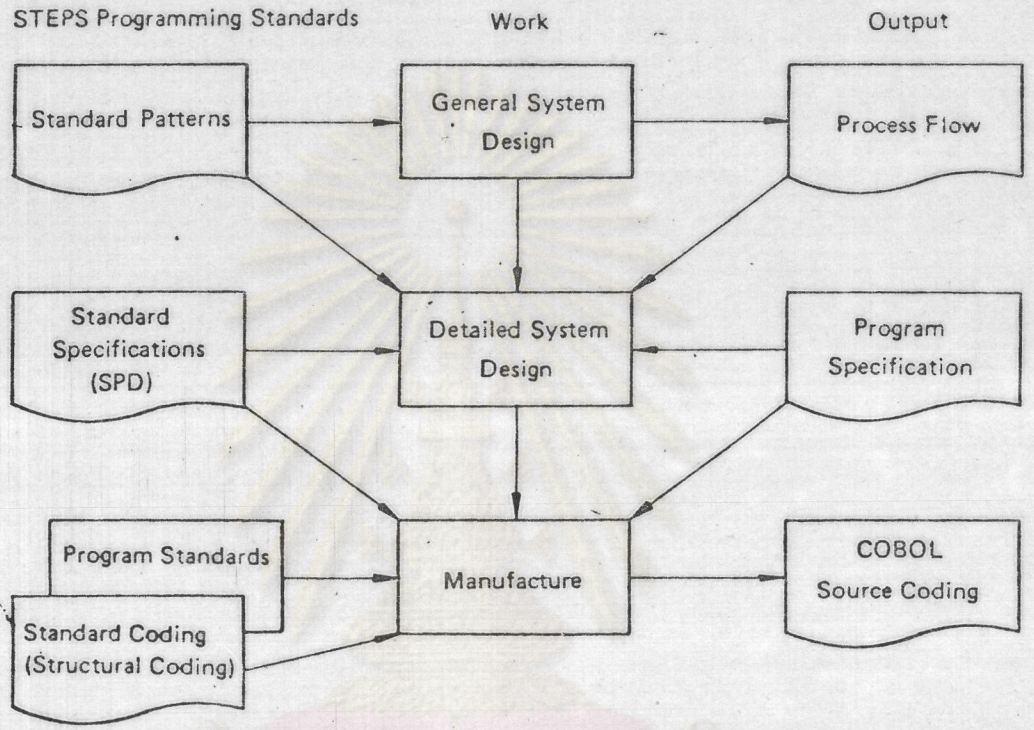
2. โปรแกรมมาตรฐานนี้สามารถทำให้อยู่ในรูปแบบเฉพาะสำหรับผู้ใช้แต่ละคนได้มากที่สุด

3. ให้ผลผลิตสูง

4. โปรแกรมมีลักษณะโครงสร้างที่ดีและง่ายในการบำรุงรักษา

แนวความคิดพื้นฐานของการทำโปรแกรมมาตรฐานของ STEPS ดังรูป ผ. 3

ในขั้นแรก ผู้พัฒนาโปรแกรมจะเขียนผังการทำงานให้เข้ากับรูปแบบมาตรฐานที่กำหนดในช่วงของการออกแบบระบบโดยทั่ว ๆ ไป ซึ่งจะมีรายละเอียดจำเพาะของโปรแกรมมาตรฐานสำหรับแต่ละรูปแบบมาตรฐาน แล้วผู้ออกแบบโปรแกรมก็จะทำการเปรียบเทียบรายละเอียดของโปรแกรมที่ได้กับข้อมูลจำเพาะของความต้องการในช่วงของการออกแบบระบบในรายละเอียด ซึ่งจะตัดงานส่วนที่ไม่จำเป็นออกไป และกำหนด



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

รูป ฆ.3 ลักษณะการทำโปรแกรมตามรูปแบบมาตรฐานของ STEPS

ขบวนการต่าง ๆ ที่เป็นเนื้อหาที่แท้จริงทางธุรกิจเพื่อเพิ่มหรือแทรกเข้าไป

ต่อไปในช่วงของการทำโปรแกรม โปรแกรมมาตรฐานจะถูกปรับปรุงให้เข้ากับงานหรือลบทิ้งบางส่วน และเพิ่มเติมสอดแทรกงานด้านธุรกิจเหล่านั้นเข้าไป

ในการทำงานเหล่านี้ จะมีเอกสารที่ใช้ 2 อย่าง คือ SPD (Structured Programming Diagram) และตารางตรรก (Logic Table)

SPD จะแทนการจัดลำดับขั้นของหน่วยย่อยของโปรแกรม (Program modules hierachically) และแสดงความสัมพันธ์ของแต่ละหน่วยย่อยของโปรแกรม

ตารางตรรกใช้ในการอธิบายรายละเอียดจำเพาะของขบวนการทางธุรกิจ

การลงรหัสของโปรแกรมมาตรฐานในรูปของ SPD จะแบ่งออกเป็น 5 ระดับ ดังนี้

- IDENTIFICATION DIVISION
- ENVIRONMENT DIVISION
- DATA DIVISION
  - FILE SECTION
  - WORKING - STORAGE SECTION
- PROCEDURE DIVISION
  - LEVEL 1 ระดับที่ 1
  - LEVEL 2 ระดับที่ 2
  - LEVEL 3 ระดับที่ 3
  - SUBROUTINES ระดับที่ 4
  - USERS LEVEL ระดับที่ 5

ระดับที่ 1. กำหนดสิ่งแวดล้อมของขบวนการทำงานโดยการแบ่งส่วนขั้นตอนต่าง ๆ ซึ่งปกติมี 3 ส่วน คือ

1. ส่วนเริ่มต้น
2. ส่วนประมวลผล
3. ส่วนปิดท้าย

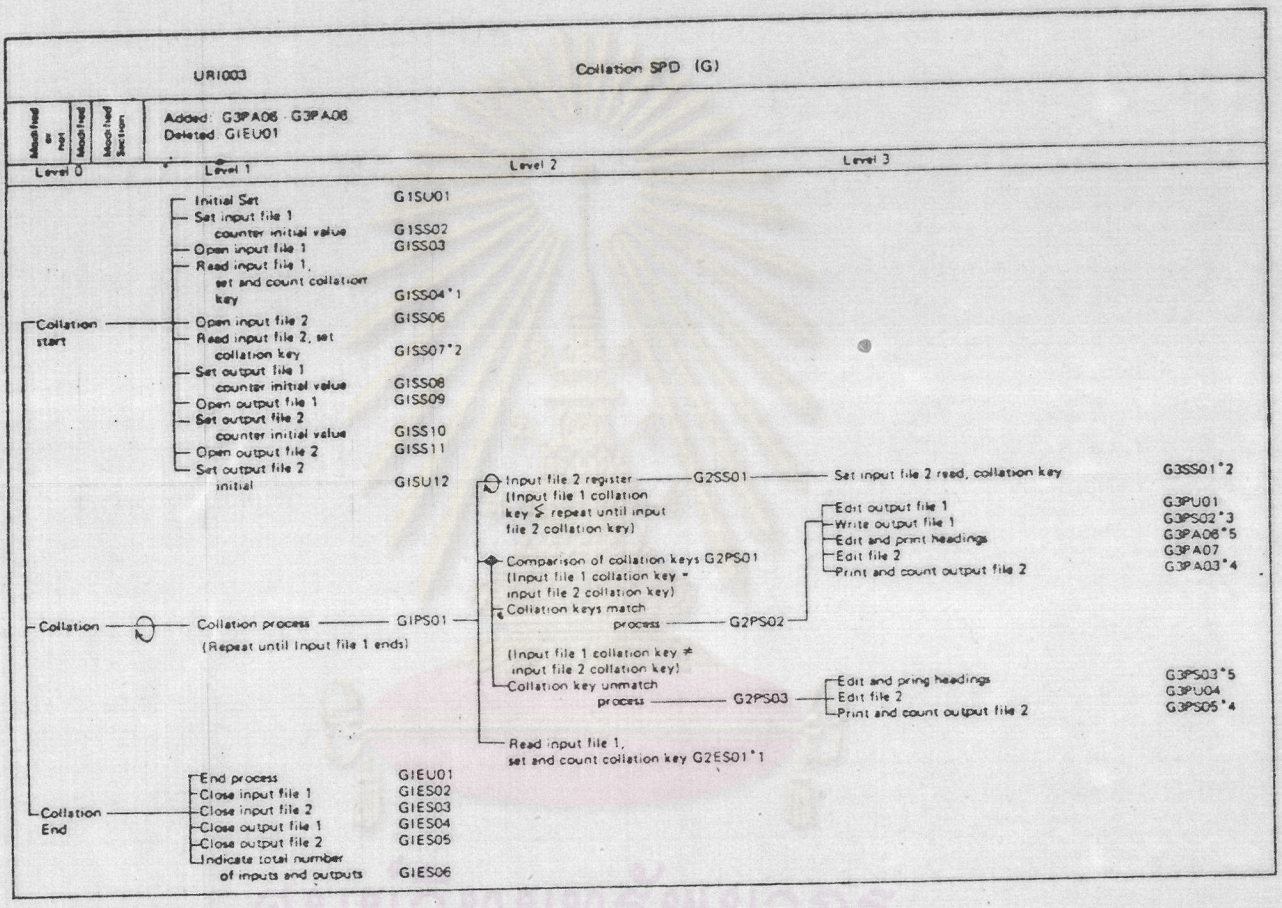
ระดับที่ 2. เป็นระดับที่แสดงถึงเงื่อนไขและทางเลือกสำหรับการปฏิบัติงานโดยการเรียกใช้หน่วยย่อยของโปรแกรมต่าง ๆ ซึ่งมีอยู่ในระดับที่ 4 และ 5

ระดับที่ 3. เป็นส่วนที่แสดงถึงรายละเอียดของขบวนการทำงานซึ่งต่อมาจากระดับที่ 2

ระดับที่ 4. เป็นหน่วยย่อยของโปรแกรมต่าง ๆ(modules, subroutines) ซึ่งอาจถูกเรียกใช้ซ้ำกันในหลาย ๆ ที่

ระดับที่ 5. เป็นส่วนของการทำงานที่แท้จริงทางธุรกิจถูกกำหนดโดยผู้ใช้ ซึ่งเป็นผู้ที่ทราบความต้องการของงาน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูป ผ.4 เอกสารการจัดลำดับชั้นของหน่วยย่อยของโปรแกรม แสดงความสัมพันธ์ของแต่ละหน่วยย่อยของโปรแกรม

ประวัติการศึกษา

นายยงยศ พรหมประกรณ์ ได้รับปริญญาวิทยาศาสตรบัณฑิต จากคณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2518 และเข้าศึกษาระดับปริญญามหาบัณฑิตในภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2523

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย