บทที่ 3

การวิเคราะห์ความต้องการ

การพัฒนาระบบนั้นประกอบด้วยกันหลายขั้นตอน ก่อนที่เราจะทำการออกแบบระบบเราต้องทราบก่อนว่าจะต้องทำอะไรอย่างไรบ้าง ซึ่งข้อมูลต่างๆเราได้จากลูกค้า โดยผู้พัฒนาระบบต้องทำการรวบรวมข้อมูลต่างๆของผู้ใช้ระบบทั้งหมด อาจเป็นการสังเกตุ การออกแบบสอบถาม การสัมภาษณ์ หรือรวบรวมจากแหล่งภายในองค์กรหรือภายนอกองค์กร ความต้องการต่างๆของระบบนั้นเราสามารถแบ่งได้เป็น 2 ชนิดคือ functional และ non-functional โดยนำความต้องการที่ได้เขียนเป็นเอกสารกำหนดความต้องการเพื่อใช้สำหรับอ้างอิงในการพัฒนาระบบในขั้นต่อๆไป

3.1

ความต้องการคืออะไร

ความต้องการ(Requirement) คือคุณลักษณะหรือรายละเอียดต่างๆที่กำหนดให้ระบบสามารถกระทำได้ โดยทั่วไปการที่เราจะพัฒนาระบบใหม่ขึ้นมาแทนที่ระบบเก่านั้น ต้องเกิดปัญหาหรือความไม่พอใจในการทำงานของระบบเดิม เช่นมีความผิดพลาดในการทำงาน ต้องการเพิ่มประสิทธิภาพในการทำงาน ต้องการนำเทคโนโลยีใหม่ๆเข้าไปช่วยเพื่อเพิ่มช่องทางด้านการตลาด แก้ไขปัญหาของพนักงาน แก้ไขปัญหาทางการผลิต เป็นต้น ดังนั้นการวิเคราะห์ความต้องการจึงเป็นการกำหนดถึงปัญหาของระบบ ค้นหาถึงปัญหาและความต้องการของลูกค้า เพราะในการพัฒนาระบบนั้นถ้าทราบถึงปัญหาที่เด่นชัด ทราบถึงจุดมุ่งหมายที่จะต้องกระทำจะทำให้คิดหนทางในการแก้ปัญหาได้ง่ายขึ้น

CT 484 77

เพราะในการพัฒนาระบบนั้นถ้าทราบถึงปัญหาที่เด่นชัด ทราบถึงจุดมุ่งหมายที่จะต้องกระทำจะทำให้คิดหนทางในการแก้ปัญหาได้ง่ายขึ้น

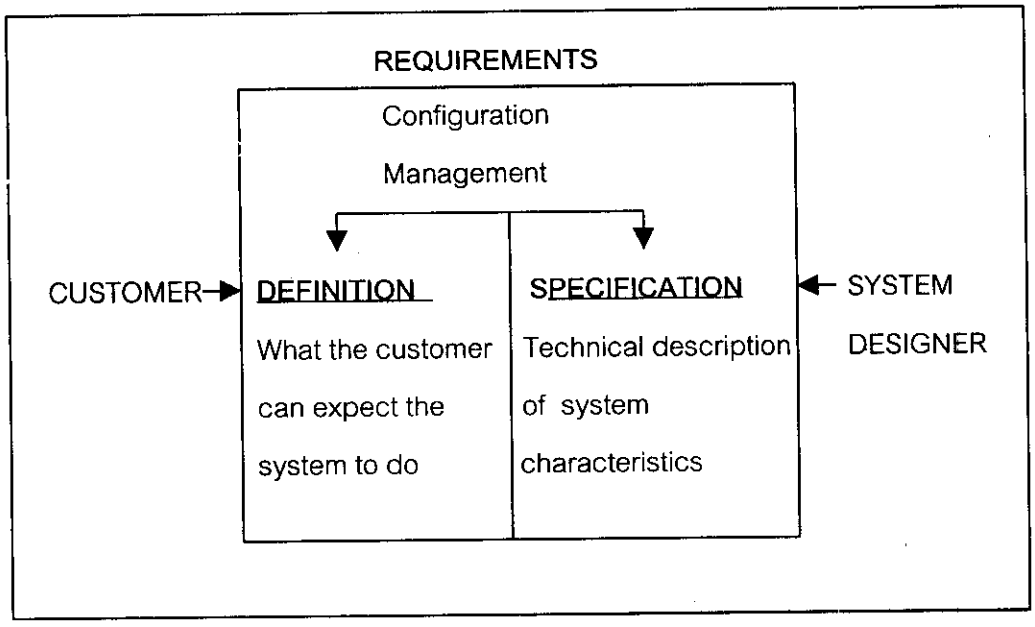
Two Sets of Requirements

ในการพัฒนาระบบนั้นเราจะทำการวิเคราะห์ความต้องการโดยจัดทำเป็นเอกสารสำหรับผู้ที่เกี่ยวข้อง 2 กลุ่ม คือ ลูกค้า และนักออกแบบระบบ ดังรายละเอียดดังนี้

1. Requirements Definition Document เป็นเอกสารที่เขียนสำหรับลูกค้า โดยเขียนด้วยภาษาที่ลูกค้าเข้าใจได้ง่าย มีการอธิบายถึงปัญหาและสิ่งที่ลูกค้าต้องการเป็นภาพรวมของระบบใหม่ที่สามารถกระทำได้ กำหนดไว้ในเอกสารอย่างชัดเจนเพื่อใช้แทนความเข้าใจระหว่างลูกค้าและผู้พัฒนา
2. Requirement Specification Document เป็นเอกสารที่บรรยายคุณลักษณะของระบบในเทอมของเทคนิค เป็นเอกสารที่ใช้สำหรับนักออกแบบระบบ ซึ่งนักออกแบบระบบสามารถทำความเข้าใจได้ง่าย

ความแตกต่างระหว่างความต้องการทั้งสองกลุ่มนั้น กรณีของลูกค้าเพียงทราบว่าระบบทำอะไรได้บ้างเท่านั้น แต่สำหรับนักออกแบบต้องทราบถึงข้อกำหนดกฎเกณฑ์ที่มากกว่านั้นทางด้านเทคนิค เช่นการปฏิบัติงานของระบบหนึ่ง ระบบปัจจุบันมีการประมวลผลแบบแบทช์ ซึ่งมีปัญหาในเรื่องของการตรวจสอบข้อมูล ถ้าลูกค้าต้องการให้ระบบสามารถประมวลผลได้ทันที เช่นสามารถปรับปรุงข้อมูลในฐานข้อมูลได้ทันที จะบรรยายความต้องการนี้เป็นเอกสารกำหนดความต้องการว่าการประมวลผลข้อมูลสามารถปรับปรุงข้อมูลในฐานข้อมูลได้ทันที แต่ถ้าสำหรับผู้ออกแบบระบบต้องทราบรายละเอียดมากกว่านั้นว่า การสืบค้นหรือปรับปรุงข้อมูลในฐานข้อมูลต้องกระทำด้วยความเร็วเป็นอย่างไร ปริมาณต่อหน่วยเวลาเป็นอย่างไร ซึ่งเป็นข้อบังคับของระบบ อาทิเช่นกำหนดให้การประมวลผลต้องเสร็จสิ้นภายในเวลา 5 วินาที ผู้ออกแบบระบบต้องหาวิธีการแก้ปัญหาที่เหมาะสมในการออกแบบระบบให้สามารถกระทำตามข้อกำหนดได้ รูปภาพที่ 3.2 แสดงถึงความต้องการทั้ง 2 กลุ่ม จะเห็นได้ว่าข้อมูลต่างๆของเอกสารความต้องการของทั้งสองชนิดอยู่ภายใต้การดำเนินการของ Configuration management ดังภาพ

รูปภาพที่ 3.2 Two Sets of Requirements



Functional and Nonfunctional

ความต้องการเป็นสิ่งต่างๆที่ระบบสามารถกระทำได้ อาจเป็นหน้าที่หรือกิจกรรมหรือการเปลี่ยนแปลงกิจกรรมจากกิจกรรมหนึ่งไปเป็นอีกกิจกรรมหนึ่ง Yeh ได้มีการจัดแบ่งความต้องการออกเป็น 2 กลุ่ม ดังนี้

1. Functional Requirements เป็นความต้องการของระบบที่มีการปฏิสัมพันธ์กับสภาพแวดล้อม เช่น แหล่งข้อมูลเข้า แหล่งข้อมูลออก กิจกรรมที่ต้องกระทำในระบบ เป็นต้น
2. Nonfunctional Requirements เป็นความต้องการซึ่งเป็นข้อจำกัดหรือข้อบังคับที่ใช้ในการแก้ปัญหาของระบบ เช่นความเร็วในการประมวลผล ,ปริมาณต่อหน่วยเวลา ,ฮาร์ดแวร์หรือซอฟต์แวร์ที่กำหนด ,งบประมาณที่มีให้ ,เวลาในการพัฒนา เป็นต้น

3.2

ชนิดของความต้องการ

การกำหนดความต้องการ เป็นการบรรยายทุกสิ่งทุกอย่างที่ระบบมีการปฏิสัมพันธ์กับสภาพแวดล้อม รวมทั้งข้อกำหนดหรือข้อจำกัดต่างๆ แบ่งเป็นหลายชนิด ได้แก่

1. Physical Environment

   เป็นการรวบรวมข้อมูลของสภาพแวดล้อมทางกายภาพของระบบ ในเรื่องของการทำงาน ข้อจำกัดต่างๆที่ส่งผลต่อการทำงาน โดยบรรยายในเรื่องของ

   1.1 ที่ตั้งของระบบ

   1.2 กิจกรรมต่างๆที่กระทำในระบบอยู่ที่ใดบ้าง มีอยู่ที่เดียว หรือหลายที่

   1.3 ที่ตั้งของระบบมีข้อจำกัดทางสภาพแวดล้อมหรือไม่ เช่นมีความร้อนสูง มีความชัน หรือมีการแทรกของคลื่นแม่เหล็ก ฯลฯ

2. Interfaces

   เป็นการบรรยายการปฏิสัมพันธ์ของระบบกับสภาพแวดล้อมภายนอก เช่น

   2.1 แหล่งข้อมูลเข้า มาจากที่ใดบ้าง มีที่มาเพียงแหล่งเดียวหรือมาจากหลายแหล่ง

   2.2 แหล่งข้อมูลออกส่งไปที่ใดบ้าง มีที่เดียวหรือหลายที่

   2.3 มีข้อกำหนดหรือรูปแบบของข้อมูลที่ใช้ในการปฏิสัมพันธ์เป็นอย่างไร กรณีใดที่ถือว่าข้อมูลถูกต้องหรือไม่ถูกต้อง

   2.4 ปริมาณข้อมูลมากน้อยเท่าใดที่ใช้สำหรับปฏิสัมพันธ์ อาทิเช่นกระทำโดยทันทีทันใดหรือกระทำเป็นกลุ่ม

3. Users and Human factors

   การรวบรวมข้อมูลที่เกี่ยงข้องกับผู้ใช้ระบบทั้งหมด เช่น

   3.1 ผู้ใช้ระบบมีใครบ้าง ?

   3.2 ผู้ใช้ระบบแบ่งออกได้เป็นกี่กลุ่ม และใช้ระบบในกิจกรรมใดบ้าง ?

   3.3 ผู้ใช้ระบบในแต่ละกลุ่มต้องมีความรู้หรือความเชี่ยวชาญในเรื่องใดบ้าง ?

3.4 ชนิดของการฝึกอบรมที่เหมาะสมกับผู้ใช้แต่ละกลุ่ม

3.5 ประสบการณ์ของผู้ใช้แต่ละกลุ่มสามารถเข้าใจและใช้งานระบบได้ดีเพียงใด มีเปอร์เซนต์ใช้ระบบผิดมากน้อยเพียงใด

4. Functionality

เป็นการบรรยายกิจกรรมและข้อกำหนดต่างๆที่ต้องการในระบบ เช่น

4.1 ระบบทำงานเมื่อใด ?

4.2 หน้าที่ต่างๆของระบบที่ต้องการกระทำมีอะไรบ้าง ?

4.3 ช่วงชีวิตของระบบ เมื่อใดที่ระบบจะมีการปรับเปลี่ยน วิธีการปรับเปลี่ยนระบบกระทำอย่างไร ?

4.4 ข้อกำหนดของเวลาในการประมวลผล, เวลาตอบสนอง หรือปริมาณงานต่อหน่วยเวลา

5. Documentation

เป็นการบรรยายถึงการใช้งานเอกสารทั้งหมดในระบบ

5.1 มีชนิดของเอกสารอะไรบ้าง ? จำนวนของเอกสารที่ต้องการ

5.2 ตำแหน่งที่เก็บเอกสารต่างๆอยู่ที่ใดบ้าง ?

6. Data

เป็นการบรรยายข้อมูลทั้งหมดที่ใช้ในระบบ

6.1 รูปแบบของข้อมูลเข้า และรูปแบบของข้อมูลออกที่ระบบต้องการ

6.2 ปริมาณของข้อมูลที่ได้รับ และที่ส่งออกไป

6.3 บูรณภาพของข้อมูลที่ต้องการ ความถูกต้องของข้อมูล

6.4 ปริมาณของข้อมูลทั้งหมดที่ผ่านระบบ

6.5 ระดับของความถูกต้องในการคำนวณ โดยเฉพาะค่าที่เป็นทศนิยม

6.6 ระยะเวลาในการรักษาข้อมูล

7. Resources

บรรยายถึงทรัพยากรต่างๆที่ระบบต้องการ เช่น

7.1 บุคลากร, วัสดุ, อุปกรณ์, ฮาร์ดแวร์, ซอฟต์แวร์ที่ต้องการใช้สำหรับสร้างหรือนำไปใช้งาน หรือบำรุงรักษาระบบ

7.2 ความเชี่ยวชาญที่ผู้พัฒนาต้องมี

7.3 ปริมาณของสื่ออุปกรณ์ที่ใช้ในระบบ เช่น ขนาดของหน่วยความจำสำรองที่ต้องการ

7.4 กำลังไฟฟ้า, เครื่องทำความร้อน หรือเครื่องทำความเย็น ที่ต้องการ

7.5 ข้อกำหนดของเวลาในการพัฒนา

7.6 ข้อจำกัดของงบประมาณในการพัฒนาหรือข้อกำหนดให้ใช้เฉพาะฮาร์ดแวร์หรือซอฟต์แวร์ที่ต้องการเท่านั้น

8. Security

บรรยายถึงความปลอดภัยที่ต้องการ เช่น

8.1 บรรยายถึงความต้องการในการควบคุมการประมวลผลของสารสนเทศต่างๆในระบบของผู้ใช้ในกลุ่มต่างๆ ในกรณีที่ผู้ใช้แต่ละคนใช้ข้อมูลและโปรแกรมไม่เหมือนกัน

8.2 บรรยายถึงวิธีการแบ่งแยกข้อมูล, โปรแกรม , ระบบปฏิบัติการของผู้ใช้แต่ละคน

8.3 บรรยายถึงวิธีการการสำรองข้อมูลและโปรแกรม กำหนดว่าเมื่อใด, อย่างไร และจัดเก็บไว้ที่ใด ?

8.4 บรรยายถึงการป้องกันปัญหาในด้านอื่นๆเช่นระบบการป้องกันขโมย ,ระบบป้องกันไฟฟ้าตก ,ไฟฟ้าดับ, ไฟไหม้, น้ำท่วม, แผ่นดินไหว หรือ ลมพายุ เป็นต้น

9. Quality Assurance

บรรยายถึงความต้องการในการประกันคุณภาพของระบบ

9.1 บรรยายถึงความต้องการที่แสดงถึงความน่าเชื่อถือได้ของระบบ

9.2 ลักษณะเด่นของระบบที่สามารถแสดงออกมาให้เห็นโดยชัดเจน

9.3 ความสามารถในการสืบหาและวินิจฉัยข้อผิดพลาดให้กับผู้ใช้ได้

9.4 บรรยายวิธีการปฏิบัติเมื่อระบบเกิดล้มเหลว เช่น ระยะเวลาที่เริ่มเปิดเครื่องใหม่ ระเบียบที่ต้องปฏิบัติในระหว่างที่ระบบล้มเหลว เป็นต้น

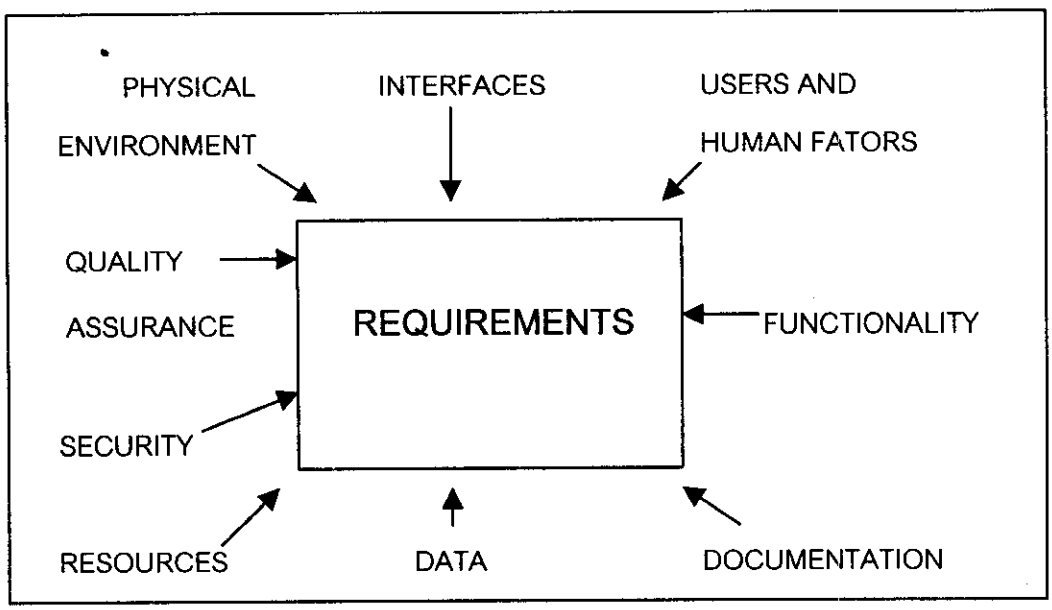
9.5 บรรยายถึงวิธีการในการบำรุงรักษาระบบ เช่นการปรับเปลี่ยนการออกแบบระบบ , การแก้ไขความผิดพลาดที่อาจเกิดขึ้นในระบบ,การปรับเปลี่ยนฮาร์ดแวร์หรือซอฟต์แวร์ระบบให้มีประสิทธิภาพมากขึ้น เป็นต้น

9.6 บรรยายถึงวิธีการในการวัดประสิทธิภาพของระบบ เช่น อัตถประโยชน์ของทรัพยากรต่างๆ, เวลาในการตอบสนอง เป็นต้น

9.7 บรรยายถึงการติดตั้งระบบในสภาพแวดล้อมต่างๆ กรณีของฮาร์ดแวร์หรือระบบปฏิบัติการที่แตกต่างกัน เป็นต้น

เราสามารถสรุปชนิดของความต้องการได้ตามรูปที่ 3.3

รูปที่ 3.3 Types of Requirements



3.3

คุณลักษณะของความต้องการ

หลังจากที่ได้รวบรวมข้อมูลต่างๆในหัวข้อที่ 3.2 แล้วต่อจากนั้นนำมาเขียนเป็นเอกสารที่บรรยายความต้องการต่างๆที่ระบบสามารถกระทำได้ซึ่งเขียนเป็นสองแนวทางโดยเขียนด้วยภาษาเขียน

CT 484 83

ปกติ และเขียนในเทอมของเทคนิคทางคอมพิวเตอร์ เอกสารต่างๆเหล่านี้เป็นประโยชน์กับผู้ใช้ 3 กลุ่มด้วยกัน คือ

- เอกสารกำหนดความต้องการ เป็นประโยชน์สำหรับผู้พัฒนาระบบสามารถทำความเข้าใจกับลูกค้าถึงสิ่งที่ลูกค้าต้องการได้ตรงกัน
- เอกสารระบุความต้องการ สามารถทำให้ผู้ออกแบบระบบเข้าใจถึงสิ่งที่ลูกค้าต้องการและหาวิธีการแก้ปัญหาให้ตรงกับวัตถุประสงค์ที่ต้องการได้
- ทีมงานทดสอบ สามารถทำการทดสอบระบบได้ตามความต้องการของลูกค้าโดยอ้างอิงถึงเอกสารความต้องการนี้ เมื่อมีการส่งมอบระบบให้แก่ลูกค้า

ดังนั้นความต้องการที่ดีย่อมเป็นประโยชน์ต่อการพัฒนาเป็นอย่างมาก คุณลักษณะของความต้องการที่ดีนั้นเราสามารถแบ่งออกเป็น 7 ลักษณะดังนี้

1. Correct ความถูกต้องของความต้องการ โดยผู้พัฒนาและลูกค้าต้องทำความเข้าใจถึงสิ่งที่ต้องการเพื่อให้แน่ใจว่าไม่มีความผิดพลาดเกิดขึ้น
2. Consistent ความคงที่ของความต้องการ ความต้องการที่ดีต้องมีความชัดเจนไม่คลุมเคลือ เช่นความต้องการของลูกค้าในตอนแรกกำหนดให้ระบบมีความสามารถในการให้บริการกับผู้ใช้ในช่วงเวลาหนึ่งได้สูงสุดไม่เกิน 10 คน แต่ต่อมาลูกค้าเกิดเปลี่ยนใจต้องการให้ระบบใหม่สามารถบริการแก่ลูกค้าได้สูงสุดถึง 20 คน ลักษณะนี้ถือว่าเป็นความไม่แน่นอนของความต้องการเพราะส่งผลต่อการแก้ไขปัญหาในการออกแบบระบบ ทำให้เกิดความล่าช้าในการพัฒนา
3. Compleate ความสมบูรณ์ของความต้องการ เป็นสิ่งที่สำคัญการที่เราจะพัฒนาระบบต้องทราบข้อมูลให้ครบถ้วน อทิเช่นการขายสินค้า ต้องทราบด้วยว่าลูกค้ามีกี่ประเภท มีการให้ส่วนลดอย่างไร มีการให้วงเงินเครดิตของลูกค้าอย่างไร กรณีที่สินค้าไม่มีอยู่ในสต๊อคจะดำเนินการอย่างไร การชำระเงินด้วยวิธีใดบ้าง ซึ่งข้อมูลต่างๆเหล่านี้ต้องบรรยายโดยละเอียดและครบถ้วน
4. Realistic ความต้องการที่ดีต้องสามารถกระทำให้เป็นจริงได้ ผู้พัฒนาต้องทราบว่าสิ่งใดที่กระทำได้ สิ่งใดที่ไม่สามารถกระทำได้

5. **Needed** ความต้องการที่ดีต้องกำหนดเฉพาะที่จำเป็นเท่านั้นเพราะบางครั้งความต้องการที่ลูกค้าต้องการอาจเป็นสิ่งที่ไม่จำเป็นไม่เป็นประโยชน์ต่อระบบ เช่นลูกค้าต้องการเครื่องไมโครคอมพิวเตอร์ยี่ห้อหนึ่งเนื่องจากมีชื่อเสียงดี แต่ถ้าเครื่องยี่ห้อนี้มีตัวโปรเซสเซอร์ที่ไม่ดีนักไม่สามารถนำมาแก้ปัญหาของระบบได้ เราต้องอภิปรายกับลูกค้าให้เข้าใจถึงความไม่จำเป็นของความต้องการในข้อนี้
6. **Verifiable** ความต้องการที่ดีต้องสามารถตรวจสอบและพิสูจน์ได้ เช่นความต้องการให้ระบบมีเวลาในการตอบสนองในเวลา 5 วินาที ต้องมีวิธีการทดสอบได้ว่าระบบสามารถกระทำได้ดังความต้องการจริง
7. **Traceable** ความต้องการที่ดีต้องสามารถติดตามได้ ว่ามีที่มาที่ไปเป็นอย่างไร เพราะกิจกรรมต่างๆในระบบเป็นกลุ่มของความต้องการที่เกี่ยวข้องกัน ซึ่งมีลำดับของการดำเนินงานที่ต่อเนื่องกัน

ตัวอย่าง การกำหนดความต้องการที่ดีและไม่ดี

Accuracy shall be sufficient to support mission planning.

การกำหนดความต้องการข้างต้นมีคุณลักษณะที่ไม่ดี เนื่องจากเราไม่สามารถทดสอบให้เห็นจริงได้ เนื่องจากไม่ทราบว่าความถูกต้องของสิ่งใดที่จะส่งผลต่อการสนับสนุนการวางแผนในเรื่องอะไรอีก ความต้องการนี้คลุมเคลือและไม่ชัดเจน

Position error shall be less than 50 feet along orbit, less than 30 feet off orbit.

สำหรับการกำหนดความต้องการนี้สามารถตรวจสอบได้เพราะมีความชัดเจนของความต้องการจัดว่าเป็นการกำหนดความต้องการที่ดี

System shall provide real-time response to queries.

การกำหนดความต้องการนี้ไม่ดี เพราะไม่กำหนดว่าให้ชัดเจนว่าการตอบสนอง real-time มีค่าเป็นเท่าใด การกำหนดที่ดีควรมีการบรรยายดังนี้

System shall respond to queries in not more than two seconds.

ซึ่งการกำหนดวิธีหลังนี้ สามารถทดสอบถึงการทำงานของระบบได้ว่าใช้เวลาเท่าใด โดยกำหนดอย่างชัดเจนมีประโยชน์สำหรับนักออกแบบระบบ

3.4

การกำหนดความต้องการ

การกำหนดความต้องการนั้นโดยทั่วไปจะบรรยายเป็นข้อความ บางครั้งข้อความที่เขียนกำกวมไม่ชัดเจน การติดตามกิจกรรมหรือทำความเข้าใจกับความสัมพันธ์ของระบบย่อยต่างๆกระทำได้ยาก ดังนั้นการกำหนดความต้องการต้องมีวิธีการที่ดีในการบรรยายระบบที่จะสร้างขึ้น เทคนิคในการกำหนดความต้องการมีดังนี้

1. Static Descriptions of Requirements

เป็นการบรรยายระบบที่เกี่ยวข้องกับเอ็นติตี้ เอ็ดทิบิ้ว และความสัมพันธ์ ซึ่งเป็นการปฏิบัติงานหรือกิจกรรมหลักๆของระบบ มีด้วยกันหลายวิธี เช่น

1.1 Indirect reference เป็นการบรรยายถึงปัญหาและการแก้ปัญหาของระบบ ซึ่งมีค่าคงที่ เช่นการหาค่าจากสมการทางคณิตศาสตร์

$$Y(X) = 3X^4 + 4X^2 - 1$$

1.2 Recurrence Relations เป็นการกำหนดความต้องการที่มีความสัมพันธ์แบบเรียกตัวเอง โดยเรียกการทำงานในลำดับก่อนมาทำงาน เช่นสมการ Fibonacci

$$F(0) = 1$$

$$F(1) = 1$$

$$F(n+1) = F(n) + F(n-1)$$

1.3 Axiomatic Definition เป็นการกำหนดคุณลักษณะพื้นฐานของระบบ รวมทั้งการสร้างกฎเกณฑ์ใหม่ของระบบที่เรียกว่าทฤษฎี(theorems) โดยทั่วไปใช้สำหรับการพัฒนาระบบผู้เชี่ยวชาญ โดยระบบจะเก็บรวบรวมข้อมูลต่างๆนำมาสร้างเป็นเงื่อนไขและกฎเกณฑ์ต่างๆตามความรู้หรือความเชี่ยวชาญของมนุษย์ และนำมาประมวลผลเพื่อสร้างสารสนเทศตามที่ต้องการ

1.4 Expression as a Language การกำหนดนิพจน์ของภาษาโปรแกรม ระบบจะทำการประมวลผลกลุ่มของข้อความ เพื่อตรวจสอบว่าเป็นคำสั่งภาษาโปรแกรมที่ถูก

ต้องหรือไม่ เช่นตัวแปลภาษาจะตรวจสอบคำสั่งที่ผู้ใช้ป้อนว่าผิดไวยากรณ์ภาษาหรือไม่ การกำหนดความต้องการที่นิยมเรียกว่า regular language ในรูปของ finite state machine ซึ่งสามารถแสดงให้เห็นถึงรูปแบบของคำและนิพจน์ที่ถูกต้องของภาษาโปรแกรมได้ นอกจากนี้ยังสามารถเขียนเงื่อนไขของไวยากรณ์ภาษาในรูปของ Backus-Naur form ได้ดังตัวอย่าง

- ASCII characters
- Expressions (<expr>)
- Terms
- Factors
- Scale factors (<scale>)
- Digits
- Addition operators (<addop>)

โดยกำหนดความสัมพันธ์ระหว่างตัวอักษรต่างๆดังตัวอย่างต่อไปนี้

<condition> ::= <bool-term> | <bool-term> or <condition>

<bool-term> ::= <bool-factor> | <bool-factor> and <bool-term>

<bool-factor> ::= <expr> <relop> <expr> | (<condition>)

<relop> ::= < | <= | = | > | >= | <>

<expr> ::= <addop> <expr>

<addop> ::= + | -

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

1.5 **Data Abstraction** เป็นเทคนิคที่ใช้ในการนิยามข้อมูลในรูปแบบ data-type dictionary กล่าวคือจัดแบ่งข้อมูลออกเป็นกลุ่มๆโดยรายการข้อมูลที่เกี่ยวข้องกันรวมไว้ด้วยกันพร้อมทั้งกำหนดชื่อให้แต่ละกลุ่มข้อมูลดังตัวอย่างของข้อมูลนักศึกษาในมหาวิทยาลัยแห่งหนึ่ง มีการนิยามข้อมูลเป็นดังนี้

Semester Record

Semester Type

Semester date

Grade point average

Completed hours

Semester Type

(Fall,Spring,Summer)

Address Information

Telephone number

Street address

City

State

Zip code

Student Record

Name

Student number

Address Information

Number of semesters

{Semester record }

นอกจากนี้ต้องกำหนดถึงการวิธีการจัดการข้อมูลต่างๆเหล่านี้ เช่นการสร้าง, การสอบถาม, การแก้ไขปรับปรุง, การลบข้อมูลให้ชัดเจน

2. **Dynamic Descriptions of Requirements** เป็นการบรรยายความต้องการ ที่มีการเคลื่อนไหวไม่คงที่เช่นการเปลี่ยนสถานะหรือเปลี่ยนกิจกรรมจากกิจกรรมหนึ่งไปเป็นอีกกิจกรรมหนึ่งเมื่อเกิดเหตุการณ์บางสิ่งบางอย่างเกิดขึ้น ซึ่งเทคนิคในการบรรยายในเอกสารนั้นมีอยู่ด้วยกันหลายวิธี ดังนี้

2.1 **Decision Tables** ตารางตัดสินใจ เป็นการบรรยายถึงกิจกรรมที่ต้องปฏิบัติ เมื่อมีข้อมูลหรือเงื่อนไขในการทำงานที่ต่างๆกัน

ตัวอย่างต่อไปนี้เป็นตัวอย่างของระบบสั่งซื้อสินค้าที่มีเหตุการณ์ที่เกิดขึ้น 3 เหตุการณ์ที่แตกต่างกันเมื่อมีการตรวจสอบสต๊อคสินค้าคือกรณีที่หนึ่งสินค้าในสต๊อคเพียงพอ กรณีที่สองมีสินค้าในสต๊อคบางส่วน และกรณีที่สามสินค้าขาดสต๊อค ดังนั้นกิจกรรมที่เกิดขึ้นในระบบนั้นแตกต่าง

กันไปแล้วแต่กรณี ในที่นี้ถ้าสินค้าในสต๊อคมีเพียงพอ จะทำการเตรียมใบส่งของ บันทึกยอดสั่งซื้อในใบส่งของ และส่งใบส่งของที่ได้จัดทำไปให้ลูกค้า ในกรณีอื่นๆก็จะมีกิจกรรมที่กระทำแตกต่างกัน ดังตารางตัดสินใจตามตารางที่ 3.1 ดังนี้

ตารางที่ 3.1 แสดงตารางตัดสินใจ

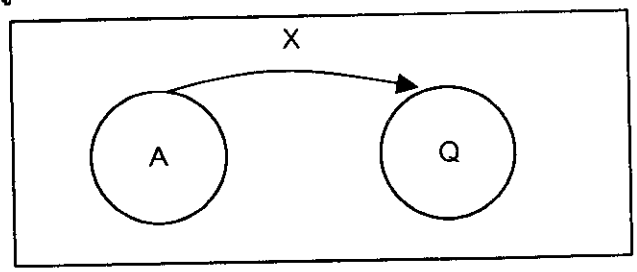
Order System	Rules		
	1	2	3
Sufficient Stock Quantity	Y	N	N
Partial Quantity in Stock		Y	N
Out of Stock			Y
Prepare invoice	X	X	
Enter Order Qty on invoice	X		
Enter Stock Qty on invoice		X	
Ship invoice Qty to Cus	X	X	
Order-Stock on Backorder		X	
Order Qty on Backorder		X	

จะเห็นได้ว่าการใช้ตารางตัดสินใจแทนการบรรยายจะทำให้เราเข้าใจได้ง่ายขึ้น

2.2 Functional Descriptions and Transition Diagrams เป็นการกำหนดความต้องการของระบบในรูปของไดอะแกรมการเปลี่ยนแปลงของกิจกรรมต่างๆ โดยรูปวงกลมแทนสถานะ ลูกศรแทนการเปลี่ยนสถานะ จากรูปภาพที่ 3.4 ประกอบด้วยสถานะ A ซึ่งเมื่อเกิดเหตุการณ์ X ระบบจะมีการเปลี่ยนสถานะจากสถานะ A ไปเป็นสถานะ Q เขียนเป็นการกระทำของระบบได้ดังนี้

F(state A, event X) = Action Q

รูปภาพที่ 3.4 Transition from State A to Q on Input X



ในกรณีระบบมีหลายสถานะและมีหลายเหตุการณ์สามารถเขียนเป็นสมการได้ดังนี้

$F(S_i , C_j) = S_k$ สมมุติว่าสถานะ S_i เมื่อมีเหตุการณ์ C_j

ส่งผลให้เปลี่ยนสถานะเป็น S_k ดังตารางที่ 3.2 เราสามารถเขียนเป็นไดอะแกรมจากข้อมูลในตารางที่ 3.2 ได้ดังรูปภาพที่ 3.5

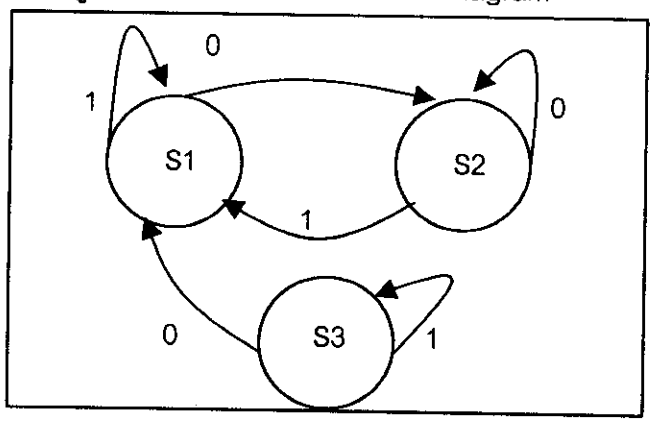
ตารางที่ 3.2 Transition Table

Current State	Input	Next state
S1	0	S2
S1	1	S1
S2	0	S2
S2	1	S1
S3	0	S1
S3	1	S3

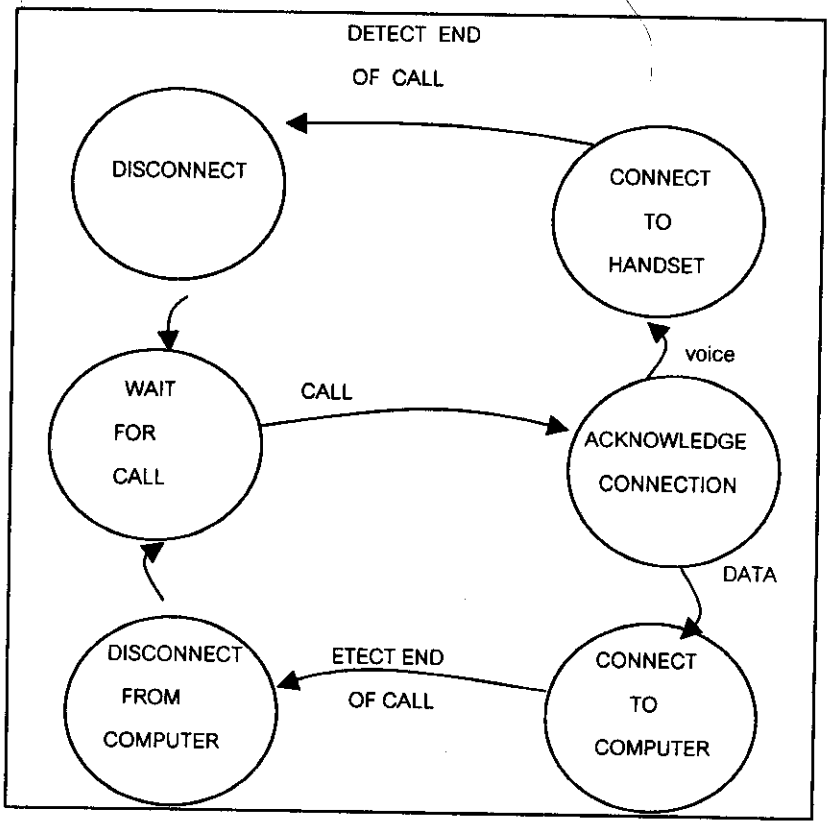
โดยไดอะแกรมในรูปภาพที่ 3.5 นั้นเหมาะสำหรับการกำหนดความต้องการสำหรับระบบขนาดเล็ก ที่มีหลายสถานะ ทำให้เห็นการเปลี่ยนสถานะได้ชัดเจน

รูปภาพที่ 3.6 แสดงถึงการกำหนดความต้องการของการเชื่อมต่อโทรศัพท์ระหว่างเครื่องคอมพิวเตอร์ในข่ายงาน

รูปภาพที่ 3.5 State Transition Diagram



รูปภาพที่ 3.6 Transition Diagram for Automatic Call Router



2.3 Event Tables ในการกำหนดความต้องการนั้นกรณีที่มีหลายเหตุการณ์ที่สามารถ

ทำให้เกิดการเปลี่ยนสถานะที่แตกต่างกัน เราสามารถนำมาเขียนเป็นตารางกำหนดความต้องการที่เรียกว่า event table ได้ดังตารางที่ 3.3 ดังนี้

ตารางที่ 3.3 Event Table

Mode	Event			
	Event1	Event2	Event3	Event4
Graphics	Action1	Action8	0	X
Architecture	X	Action2 Followed by Action3	Action5 and 6 in parallel	0
Native	0	Action4	Action1,2,3	Action7

จากตารางที่ 3.3 เป็นการกำหนดความต้องการของบริษัทในการผลิตคอมพิวเตอร์แห่งหนึ่งที่มีการสร้างระบบอัตโนมัติแบ่งเป็น 3 โหมด โดยกรณีของโหมด Graphic สามารถแสดงผลเป็นรูปภาพต่างๆได้ โหมด Architecture สามารถแสดงรูปภาพพิมพ์เขียว ได้โหมด Native สามารถให้โปรแกรมเมอร์สามารถสร้างงานประยุกต์ที่ต้องการได้ โดยเหตุการณ์ที่สามารถปฏิสัมพันธ์กับโหมดต่างๆมีอยู่ด้วยกัน 4 เหตุการณ์ ส่งผลให้กระทำกิจกรรมต่างๆแตกต่างกันไป ในที่นี้เลข 0 หมายถึง ไม่มีผลใดๆเกิดขึ้น อักษร X หมายถึง เป็นเหตุการณ์ที่ไม่สามารถเกิดขึ้นได้หรือเป็นไปไม่ได้

2.4 Petri Nets เทคนิคหนึ่งที่เรานำมาใช้ในการกำหนดความต้องการคือการผสมผสานหลายๆเทคนิคเข้าด้วยกัน เมื่อระบบมีความซับซ้อนมากขึ้น มีการเปลี่ยนแปลงสถานะซึ่งขึ้นอยู่กับความสัมพันธ์ของเหตุการณ์ต่างๆ โดยเฉพาะในลักษณะของการประมวลผลแบบขนาน (parallel processing) เทคนิค Petri Nets มีความเหมาะสมที่สุด เทคนิคนี้แทนความต้องการในรูปความสัมพันธ์ของสถานะและเหตุการณ์ดังนี้

F(State A , Event) -> State S

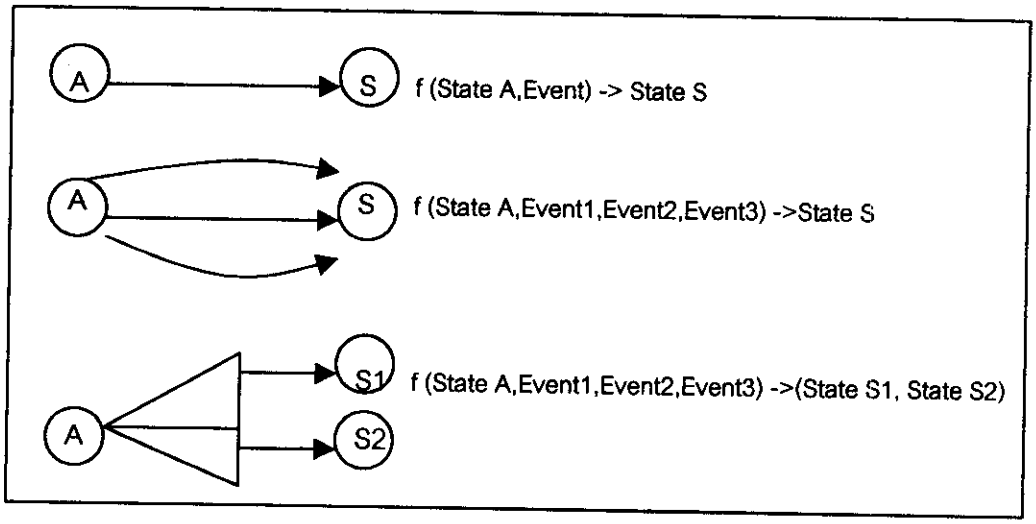
กรณีที่มีหลายเหตุการณ์ที่ทำให้เกิดผลลัพธ์ได้เหมือนกันหรือแตกต่างกันเราสามารถกำหนดความต้องการของกิจกรรมต่างๆในระบบโดยเขียนความสัมพันธ์ได้ดังนี้

F(State A, Event1, Event2,...,Event N) -> State S

F(State A, Event1, Event2,...,Event N) -> State1, State2,...,State N

เทคนิคนี้สามารถเขียนในรูปแบบของกราฟโดยกำหนดให้โหนดแทนสถานะใดๆของระบบ ลูกศรแทนการเปลี่ยนสถานะ ดังรูปภาพที่ 3.7

รูปภาพที่ 3.7 Three Types of Transitions



3.5

วิธีระบุรายละเอียดความต้องการ

ในหัวข้อนี้จะกล่าวถึงการเอกสารระบุความต้องการในเทอมของเทคนิคที่ผู้ออกแบบระบบสามารถนำไปหาหนทางแก้ปัญหาของระบบได้ วิธีการในการระบุรายละเอียดสามารถกระทำได้หลายๆวิธีแบ่งเป็น 2 วิธีการคือ

1. Manual Method เป็นวิธีการที่เหมาะกับโครงงานขนาดเล็ก ซึ่งช่วยให้เข้าใจถึงโครงสร้างและความสัมพันธ์ของกิจกรรมต่างๆได้ดี วิธีการที่นิยมได้แก่

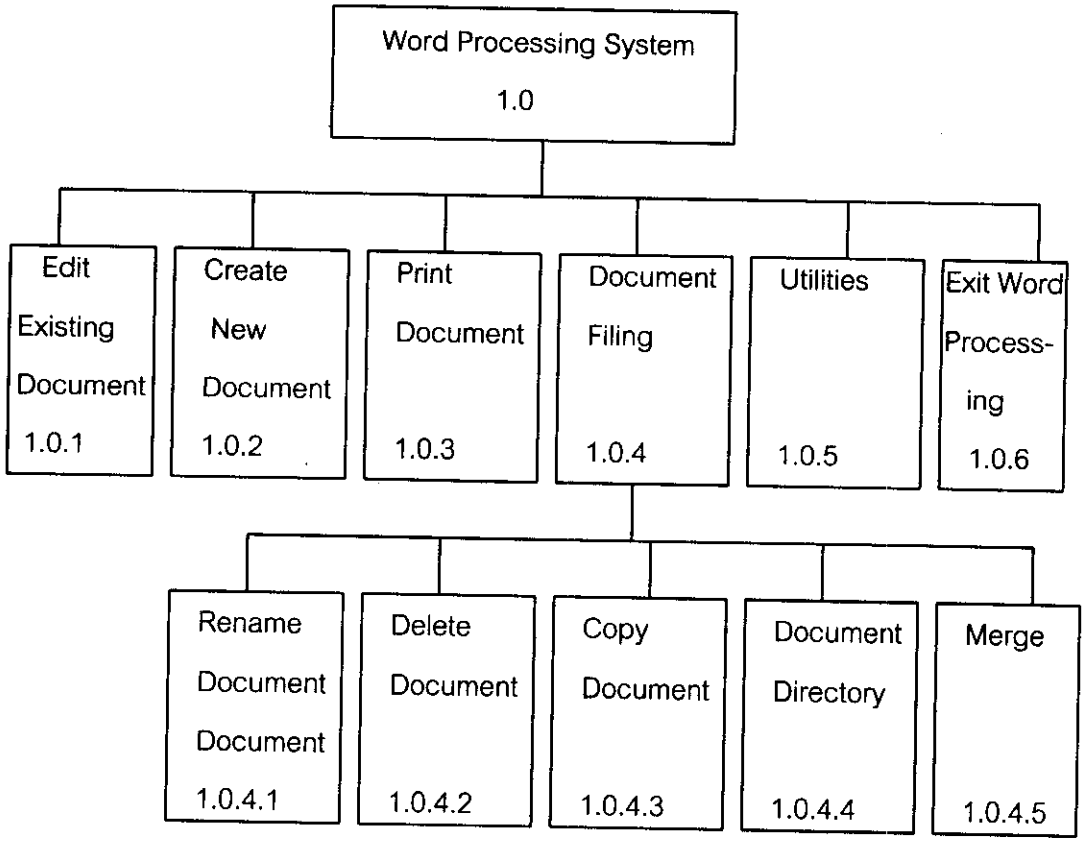
1.1 HIPO Charts ย่อมาจาก Hierarchy and Input-Process-Output เป็นเครื่องมือที่แสดงความสัมพันธ์ของหน้าที่ที่กระทำในระบบ โดยแบ่งระบบออกเป็นหน้าที่ย่อยๆเป็นลำดับชั้นเริ่มจากระดับบนสุดและแบ่งเป็นระดับย่อยลงมา ใช้สัญลักษณ์สี่เหลี่ยมแทนหน้าที่ต่างๆของระบบ ความสัมพันธ์ของหน้าที่ต่างๆจะเชื่อมโยงเป็นลำดับชั้น ดังตัวอย่างที่แสดงดังรูปภาพที่ 3.8

จากรูปภาพที่ 3.8 เป็นการระบุความต้องการของระบบประมวลผลคำโดยกำหนดถึงหน้าที่ต่างๆที่ระบบสามารถกระทำได้เป็นลำดับชั้น สำหรับข้อมูลเข้า, การประมวลผลและผลลัพธ์ของหน้าที่ต่างๆเหล่านั้น ต้องมีการระบุในไดอะแกรมที่เรียกว่า IPO (input-process-output)

รูปภาพที่ 3.9 แสดงถึง IPO ของการพิมพ์เอกสาร จะเห็นได้ว่าการระบุหน้าที่ต่างๆของระบบใช้หมายเลขในการอ้างอิง เพื่อให้ง่ายต่อการทำความเข้าใจ และสามารถติดตามรายละเอียดได้ง่าย

1.2 Hierarchical Data Structure เราสามารถใช้โครงสร้างลำดับชั้นเพื่อระบุความต้องการในกรณีเป็นการทำงานของระบบ อทิเช่นใช้เป็นเครื่องมือในตรวจสอบความสัมพันธ์ของข้อมูลที่กำหนดขึ้น โดยมองข้อมูลเปรียบเสมือนโครงสร้างต้นไม้ ข้อมูลในระดับล่างจะเป็นเซตย่อยของข้อมูลที่อยู่ในระดับสูงกว่า ดังตัวอย่างตามรูปภาพที่ 3.10

รูปภาพที่ 3.8 Table of Contents for Word Processing System

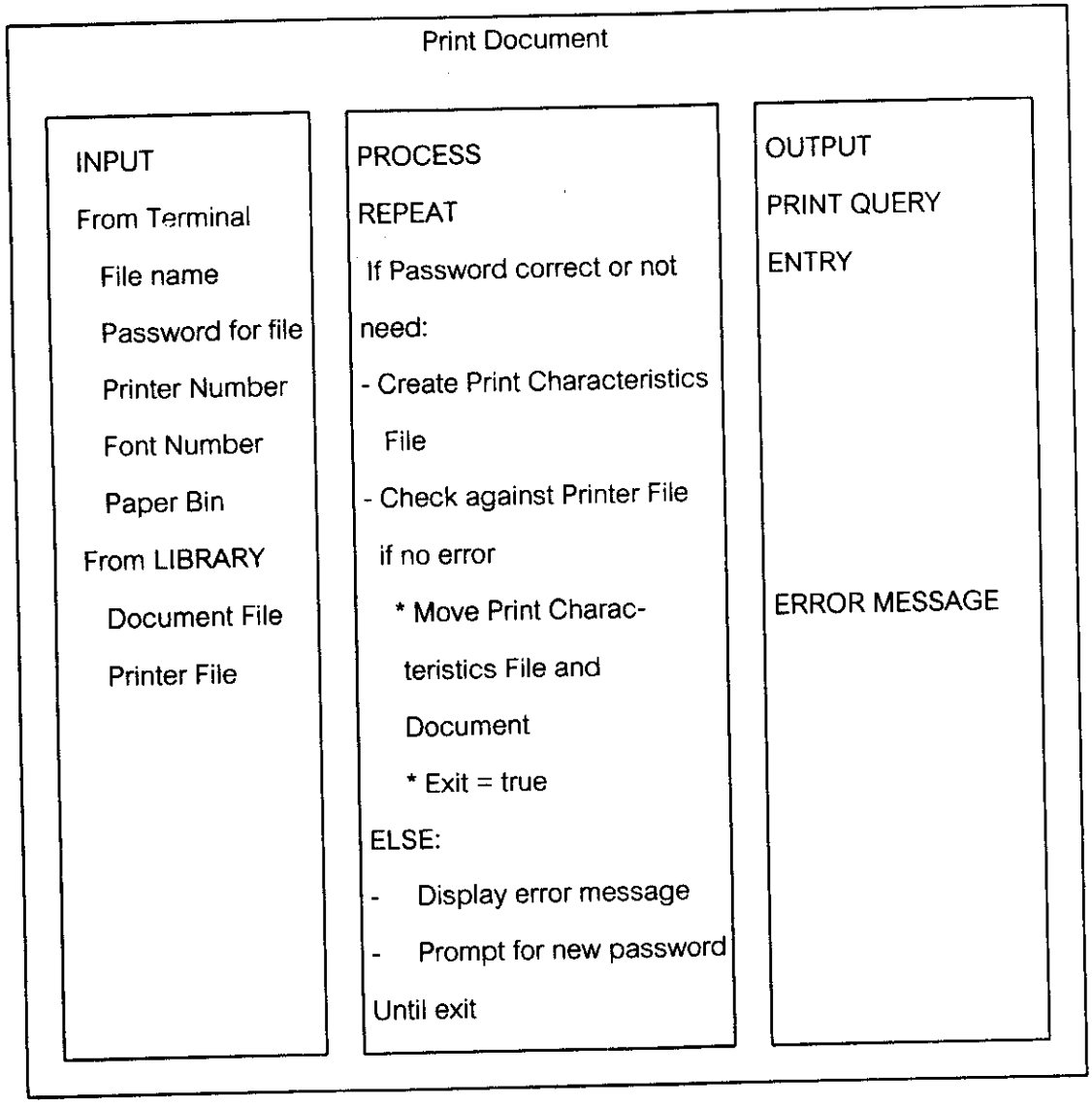


เทคนิคที่สามารถระบุความต้องการได้คล้ายกันอีกวิธีหนึ่งเรียกว่า Warnier Diagram เป็นการระบุความสัมพันธ์โดยใช้เครื่องหมาย braces ({) แทนความสัมพันธ์ของข้อมูลในระดับขั้นต่างๆ โดยแสดงจากซ้ายไปขวา ข้อมูลซ้ายสุดจะเป็นข้อมูลในระดับสูงสุด ข้อมูลระดับขวาสุดเป็นข้อมูลในระดับขั้นต่ำสุด ดังรูปภาพที่ 3.11

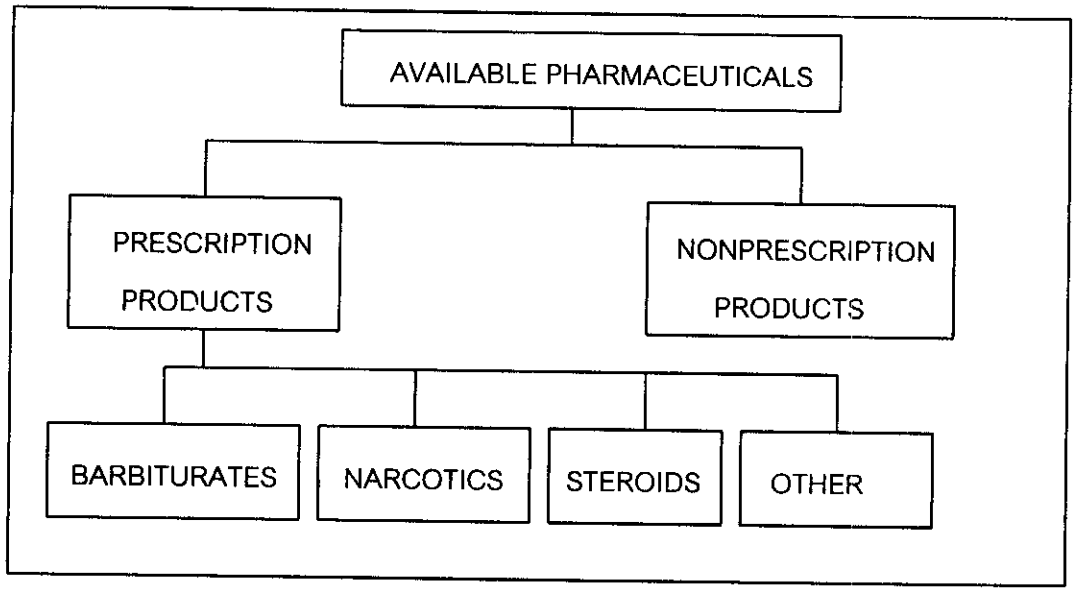
โดยเครื่องหมายบวกที่มีวงกลมนั้นหมายถึงความเกี่ยวพันธ์ซึ่งกันและกัน และกรณีที่ข้อมูลมีหลายชุดจะมีจำนวนของการกระทำซ้ำระบุอยู่หลังข้อมูลในที่นี้ระบุในวงเล็บหลังข้อมูลเป็นจำนวนที่กระทำซ้ำกัน

1.3 **Data Flow Diagrams** เป็นเทคนิคในการระบุความต้องการของข้อมูลในระบบ เป็นไดอะแกรมที่ทำให้เข้าใจถึงการไหลของข้อมูล การประมวลผลข้อมูล การจัดเก็บข้อมูลในระบบ คล้ายกับ input-process-output (IPO) ของ HIPO chart โดยข้อมูลที่เข้าจะแทนด้วยลูกศรเข้า การประมวลผลแทนด้วยวงกลม และข้อมูลออกจะแทนด้วยลูกศรที่ออกจากวงกลม ดังรูปภาพที่ 3.12

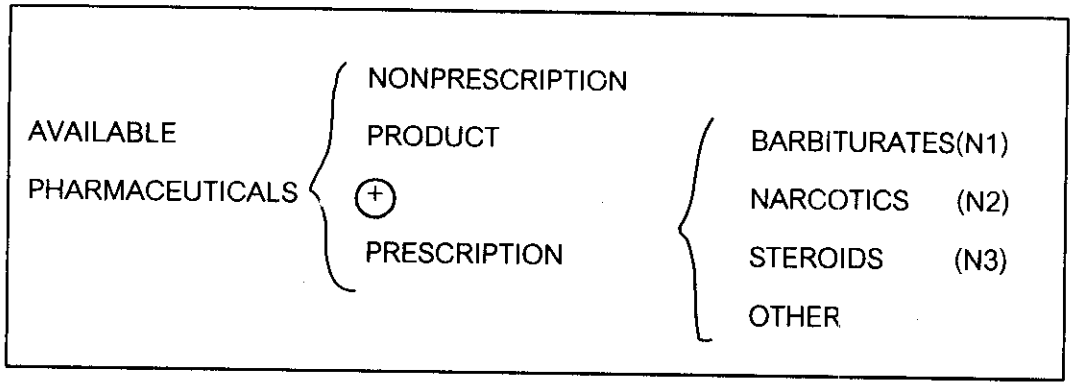
รูปภาพที่ 3.9 Input-Process-Output for Box 1.0.3 ของรูปภาพที่ 3.3



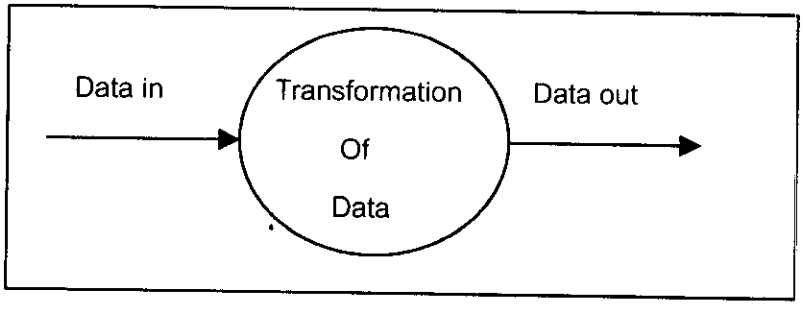
รูปภาพที่ 3.10 Hierarchical Data Structure



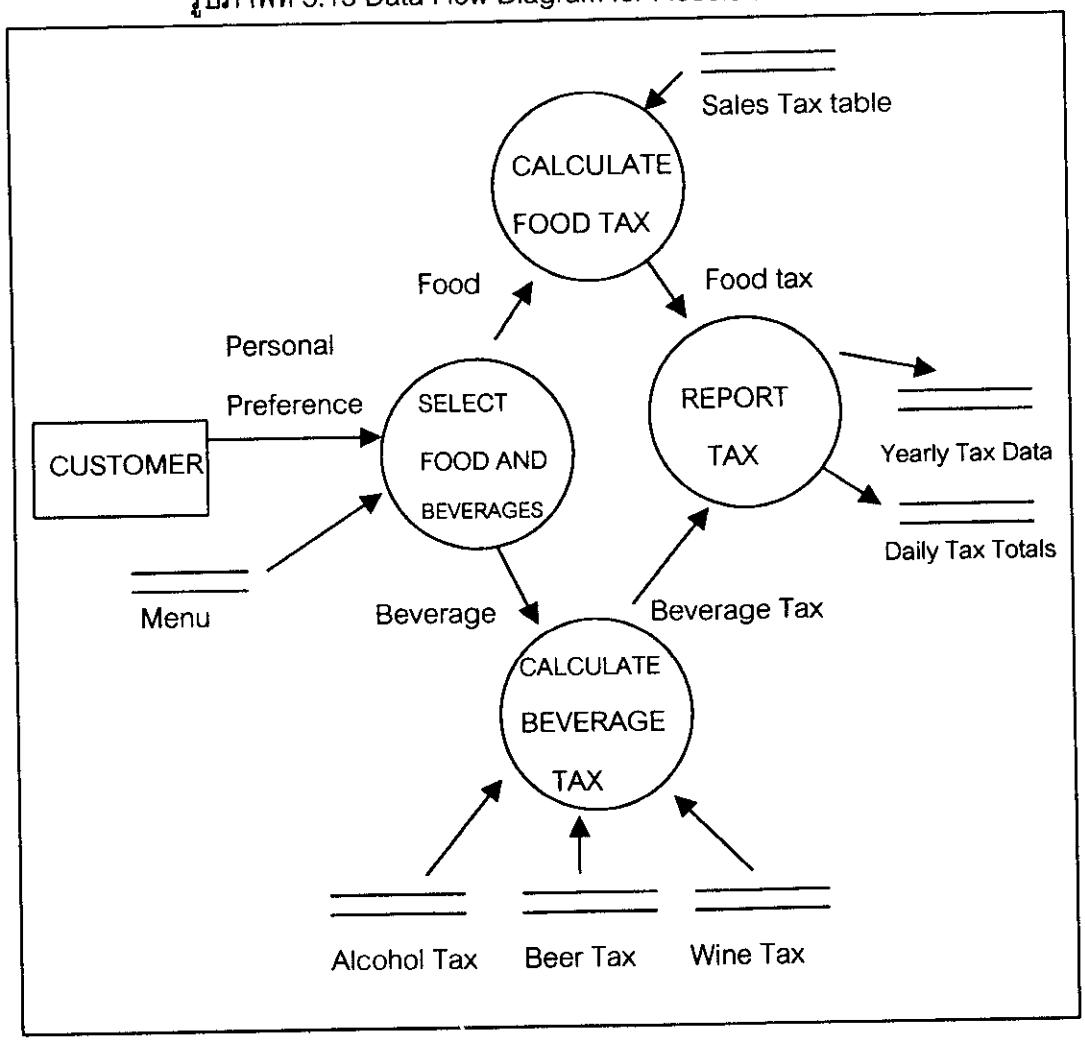
รูปภาพที่ 3.11 Warnier Diagram



รูปภาพที่ 3.12 Data Flow



รูปภาพที่ 3.13 Data Flow Diagram for Piccolo's Restaurant



รูปภาพที่ 3.13 เป็นการระบุความต้องการโดยใช้ Data Flow Diagram ของระบบคิดภาษีในการสั่งอาหารของลูกค้า โดยการคิดจะแยกการคำนวณเป็นสองกลุ่มคือกลุ่มอาหารและกลุ่มของเครื่องดื่มแอลกอฮอร์ และนำภาษีที่คิดได้ทั้งสองกลุ่มออกเป็นรายงานภาษีนำไปบันทึกในแฟ้มประจำวัน และแฟ้มประจำปี

2. **Automated Methods** เป็นวิธีการที่ใช้โปรแกรมสำเร็จรูปที่พัฒนาขึ้นมาเพื่อช่วยกำหนดความต้องการ ซึ่งสามารถวาดผัง กราฟกิจกรรม รวมทั้งสร้างพจนานุกรมข้อมูลของระบบได้ ที่นิยมใช้กันมีดังนี้

2.1 PSL/PSA วิธีการนี้ ใช้ Problem Statement Language (PSL) สำหรับกำหนดความต้องการของระบบ โดยบรรยายการไหลของข้อมูลระหว่างระบบกับสภาพแวดล้อม กำหนดขอบเขตของระบบ โครงสร้างของระบบเป็นลำดับชั้น ความสัมพันธ์ระหว่างข้อมูลต่างๆ กิจกรรมและการประมวลผลทั้งหมดในระบบ

ตัวอย่าง ระบบ Weaver Farm ซึ่งใช้ PSL ผลิตแผนงานการบำรุงรักษาระบบ โดยสร้างแผนงานในการตัดหญ้าในแต่ละวันในฟาร์มได้ดังนี้

PROCESS: daily-mowing-schedule

DESCRIPTION: For week day, schedules areas of farm to be moved.

GENERATES: mowing-schedule;

RECEIVES: turf-file , history-file , cell-file ;

SUBPARTS ARE: check-type , check-frequency , check-history , schedule;

PART OF: daily-maintenance-schedule;

DERIVES: mow-need

USING: turf-file , cell-file;

PROCEDURE:

1. Check cell file for type of turf.
2. Check type of turf for frequency of mowing : mow-need.
3. Determine new mow-day.
4. If today = mow-day, put on schedule.

HAPPENS: daily

TRIGGERED BY work-day ;

TERMINATION CAUSES: holiday ;

SECURITY IS: none;

สำหรับ Problem statement Analyzer (PSA) เป็นการสร้างรายงานจากรายละเอียดของ PSL โดยสร้างพจนานุกรมข้อมูล รายงานความสัมพันธ์ของข้อมูล การไหลของข้อมูล คุณลักษณะของข้อมูล เก็บประวัติของข้อมูลและสืบค้นความผิดพลาดที่เกิดขึ้น

เช่น ถ้ามีการกำหนดโครงสร้างของข้อมูลไว้ แต่ระบบไม่ได้นำมาใช้งาน PSA สามารถวิเคราะห์ให้ทราบได้ รวมทั้งสร้างรายงานการอ้างอิงข้อมูลที่มีการการเปลี่ยนแปลง

2.2 Software Requirements Engineering Methodology (SREM) เป็นเครื่องมือซอฟต์แวร์ที่พัฒนาโดย TRW Corporation สำหรับระบบ real-time ซึ่งมีการแบ่งออกเป็นสองส่วนเหมือนกับ PSA โดย SREM ประกอบด้วย

-RSL(Requirements Statement Language)

-REVS (Requirements Engineering Validation System)

โดย RSL บรรยายในเทอมของการไหลของการประมวลผลในข่ายงาน ใช้รูปภาพแทนการอธิบาย แต่ละข่ายงานเราจะเรียกว่า R-net รูปภาพที่ 3.14 แสดงข่ายงานของการประมวลผลของระบบฝากถอนเงินอัตโนมัติของธนาคารแห่งหนึ่ง ซึ่งจากภาพสัญลักษณ์บวก(+) ภายในวงกลม แทนเงื่อนไขของการประมวลผล สำหรับampersand (&)ภายในวงกลม แทนการประมวลผลแบบขนาน (parallel) เครื่องหมายสามเหลี่ยมแทนการส่งผ่านข้อมูลแบบสัมพันธ์ (synchronization)

จากไดอะแกรม R-net รูปภาพที่ 3.14 สามารถแปลเป็นคำสั่ง RSL ได้ดังนี้

```
R-NET : PROCESS_TRANSACTION
   STRUCTURE :
         INPUT_INTERFACE_ACCOUNT_REQUEST_RECORD
         EXTRACT_DATES
         DO (REQUEST = TRANSACTION)
               RECORD_TRANSACTION
               TERMINATE
         OTHERWISE
               FIND_ACCOUNT-RECORD
               COMPUTE_SAVINGS_BALANCE
               AND COMPUTE_CHECKING_BALANCE
```

AND COMPUTE_MONEY-MARKET_BALANCE

PRINT_BALANCE

TERMINATE

END

R-net เป็นเครื่องมือที่ใช้สำหรับระบุความต้องการที่เป็น functional สำหรับการระบุความต้องการที่เป็น nonfunctional ต้องมีการสร้าง validation point ขึ้นในที่นี้คือกิจกรรมที่มีเครื่องหมาย * ซึ่งใช้เป็นจุดสำหรับวัดหรือทดสอบความต้องการที่ลูกค้าต้องการ จากรูปภาพที่ 3.14 มีการกำหนดจุดในการทดสอบ 2 จุดซึ่งถ้าลูกค้าต้องการให้กระทำภายในเวลา 5 วินาที การทดสอบจะเริ่มจากการค้นหาระเบียนบัญชีของลูกค้าพบแล้วทำการประมวลผลจนกระทั่งพิมพ์ยอดคงเหลือ โดยเทคนิคนี้ใช้ REVS สำหรับแสดงความต้องการที่เป็น nonfunctional

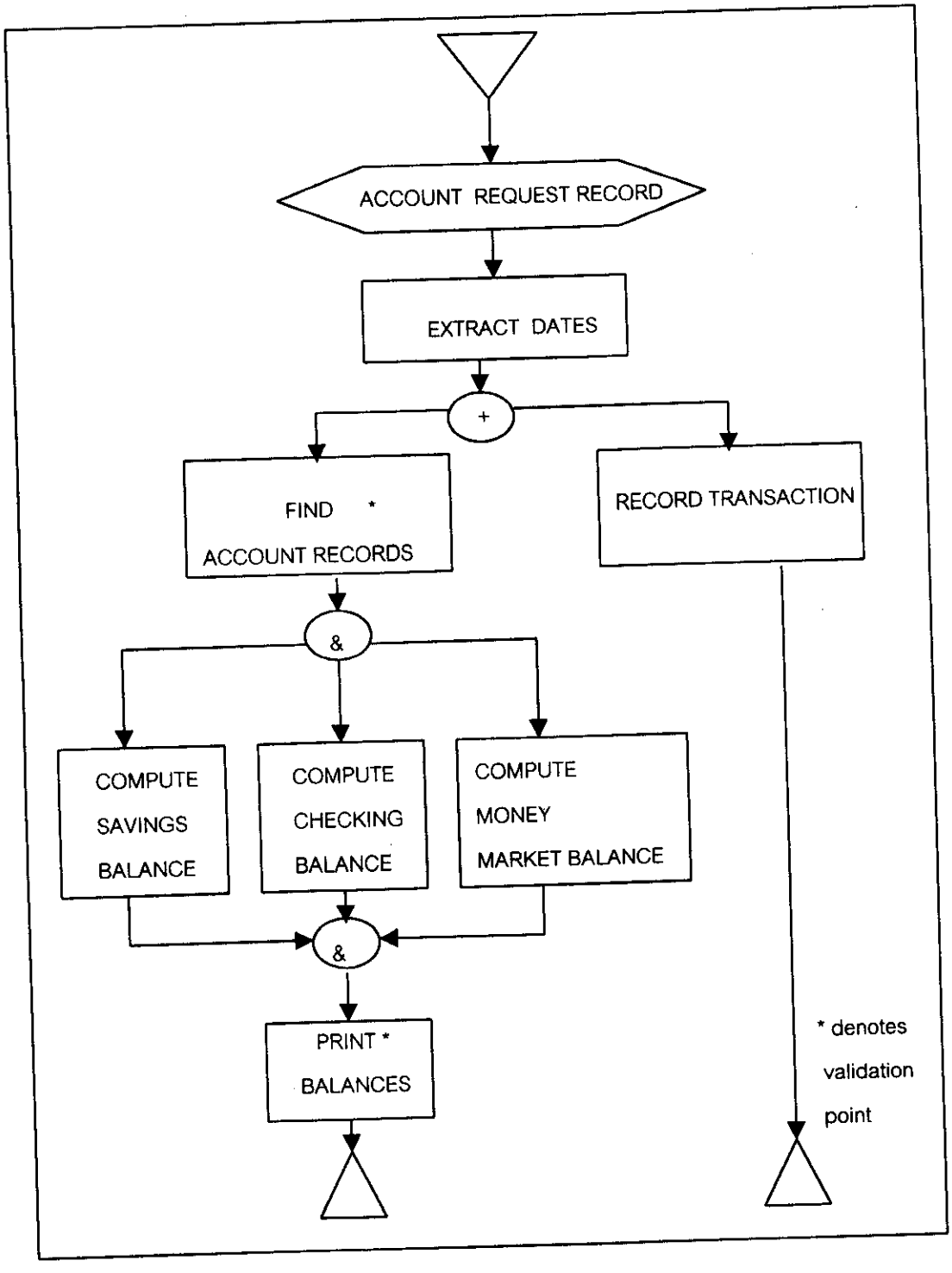
ระบบ REVS จะอ่านคำสั่ง RSL เพื่อทำการแปลคำสั่ง RSL ให้อยู่ในรูปของฐานข้อมูลเรียกว่า Abstract System Semantic Model (ASSM) เพื่อผลิตรายงานชนิดต่างๆออกมา อาทิเช่นรายงานสรุปทางเลือกของการปฏิบัติงานต่างๆ วิเคราะห์ความเป็นไปได้ของระบบ การไหลของข้อมูลผ่านระบบ สร้างแบบจำลองการทำงานของระบบ เป็นต้น สามารถสรุปขั้นตอนการทำงานของ SREM ดังข้อมูลจากตาราง 3.4

2.3 Structure Analysis and Design Technique (SADT) เป็นเทคนิคที่พัฒนาโดย SofTech, Inc การกำหนดความต้องการจะแบ่งออกเป็นสองส่วนคือ

- Structured analysis (SA) เป็นการระบุความต้องการโดยใช้ไดอะแกรม
- Design (DT) เป็นการบรรยายผลลัพธ์

โดย SA นั้นจะเป็นไดอะแกรมของกิจกรรม, ไดอะแกรมข้อมูล,การเปลี่ยนสถานะ โดยไดอะแกรมจะประกอบด้วยองค์ประกอบ 4 อย่างคือ ข้อมูลเข้า, พารามิเตอร์ควบคุม, มนุษย์หรือเครื่องจักร และผลลัพธ์ โดยทุกๆกิจกรรมจะวาดเป็นสี่เหลี่ยมมีลูกศรชี้เข้า 3 ทางและลูกศรชี้ออก 1 ทาง ดังรูปภาพที่ 3.15และ 3.16 แสดงการแทนไดอะแกรมกิจกรรม และไดอะแกรมข้อมูลตามลำดับ

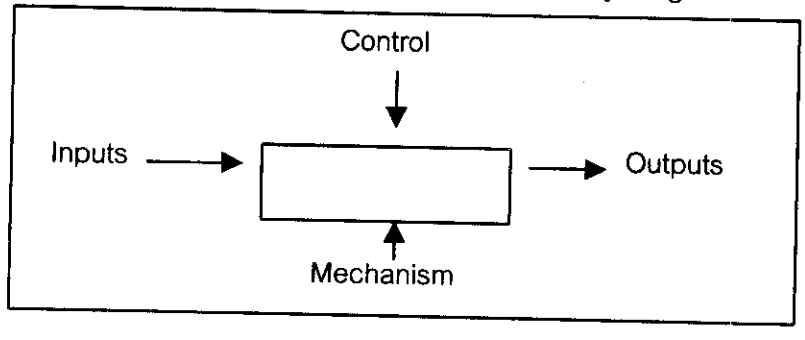
รูปภาพที่ 3.14 Requirement Network



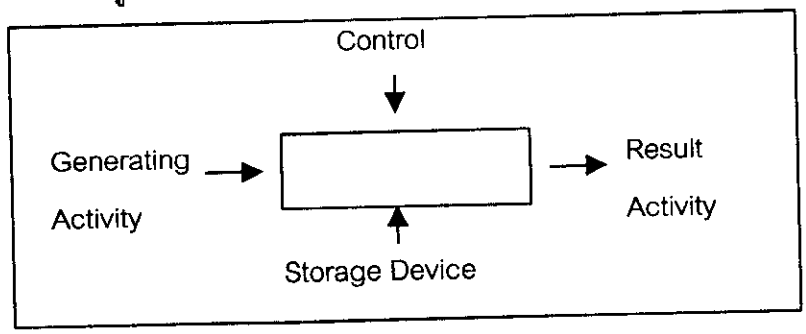
ตารางที่ 3.4 SREM Steps

Phase	Focus	Criteria
Define kernel	Identify I/O , R-nets, Transformations.	All input messages processed. All output message generated.
Establish baseline.	Clean up data base. Plot R-nets.	All naming consistent.
Define data.	Define input, output For each transformation.	No data used before Given a value
Establish traceability	Generate consistent traceability requirements	All top requirements satisfied
Simulate functionality	Simulate subsystem Functions performed	Validation that all are processed correctly.
Identify performance	Define traceable. Testable performance Subsystem requirements	Each path constrained by response time and accuracy
Demonstrate feasibility	Rapid prototype of all critical algorithms.	Accuracy requirements Satisfied by prototype algorithms.

รูปภาพที่ 3.15 Structured Analysis Activity Diagram



รูปภาพที่ 3.16 Structured Analysis Data Diagram



จากรูปภาพทั้งสอง mechanism เป็นอุปกรณ์ภายนอกที่ใช้เป็นเครื่องมือในการประมวลผล control ป็นข้อมูลซึ่งเป็นข้อจำกัดหรือระเบียบกฎเกณฑ์ซึ่งมีผลต่อการเปลี่ยนสถานะของกิจกรรม ในข่ายงานระบบบล๊อกสี่เหลี่ยมผืนผ้าเราเรียกว่าโหนด(node) เครื่องมือนี้ใช้สำหรับสร้างกราฟิก ที่แสดงถึงโครงสร้างของระบบและความสัมพันธ์ซึ่งเรียกว่า system blueprint

รูปภาพที่ 3.17 แสดงภาพรวมของระบบประมวลผลคำ มีลำดับของกิจกรรมต่างๆที่ เชื่อมโยงกัน รวมทั้งการสร้างเอกสาร ,การตรวจสอบคำสะกด, การพิมพ์เอกสาร จะเห็นได้ว่า ไดอะแกรมข่ายงานไม่แสดงกิจกรรมโดยละเอียด

รูปภาพที่ 3.18 แสดงตัวอย่างของไดอะแกรมข้อมูล และกิจกรรมอย่างละเอียด ที่เกี่ยว ข้องในข่ายงาน

2.4 **Gist** เป็นภาษาระบุความต้องการที่พัฒนาโดย The Information Sciences Institute โดยนำข้อมูลมาจัดเป็นกลุ่ม รวมทั้งกำหนดสถานะทั้งหมดของกิจกรรมใน ระบบในเทอมของ finite state model ซึ่งเทคนิคนี้คล้ายกับ data abstraction methods ซึ่งเป็นการนิยามชนิดของข้อมูล ค่าของข้อมูลที่ถูกต้องรวมทั้งการปฏิบัติงาน กับข้อมูลต่างๆเหล่านี้

ตัวอย่าง ระบบ Weaver Farm ซึ่งจัดกลุ่มของต้นไม้ ที่สร้างด้วยคำสั่ง Gist ได้ดังนี้

type TREE includes (BIRCH, ELM ,OAK , PINE , DOGWOOD);

สำหรับความสัมพันธ์ระหว่างข้อมูลสามารถสร้างได้ดังตัวอย่างต่อไปนี้

relation SHADES(TREE,CANOPY,HEIGHT)

where HEIGHT< CANOPY (TOP)

การเปลี่ยนกิจกรรมจากสถานะหนึ่งไปเป็นอีกสถานะ สามารถระบุได้ดังนี้

action LOGGING(TREE,CANOPY)

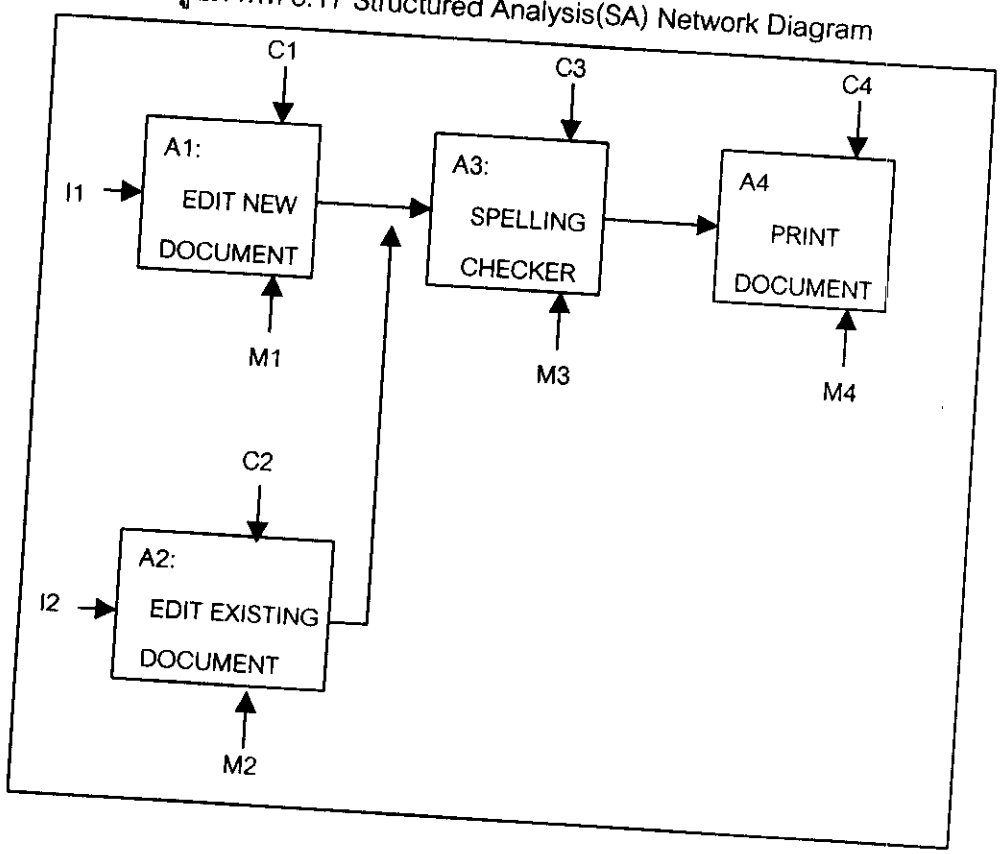
precondition TREE:HEIGHT <= CANOPY

definition if SHADES(TREE,CANOPY,HEIGHT) then HEIGHT=0

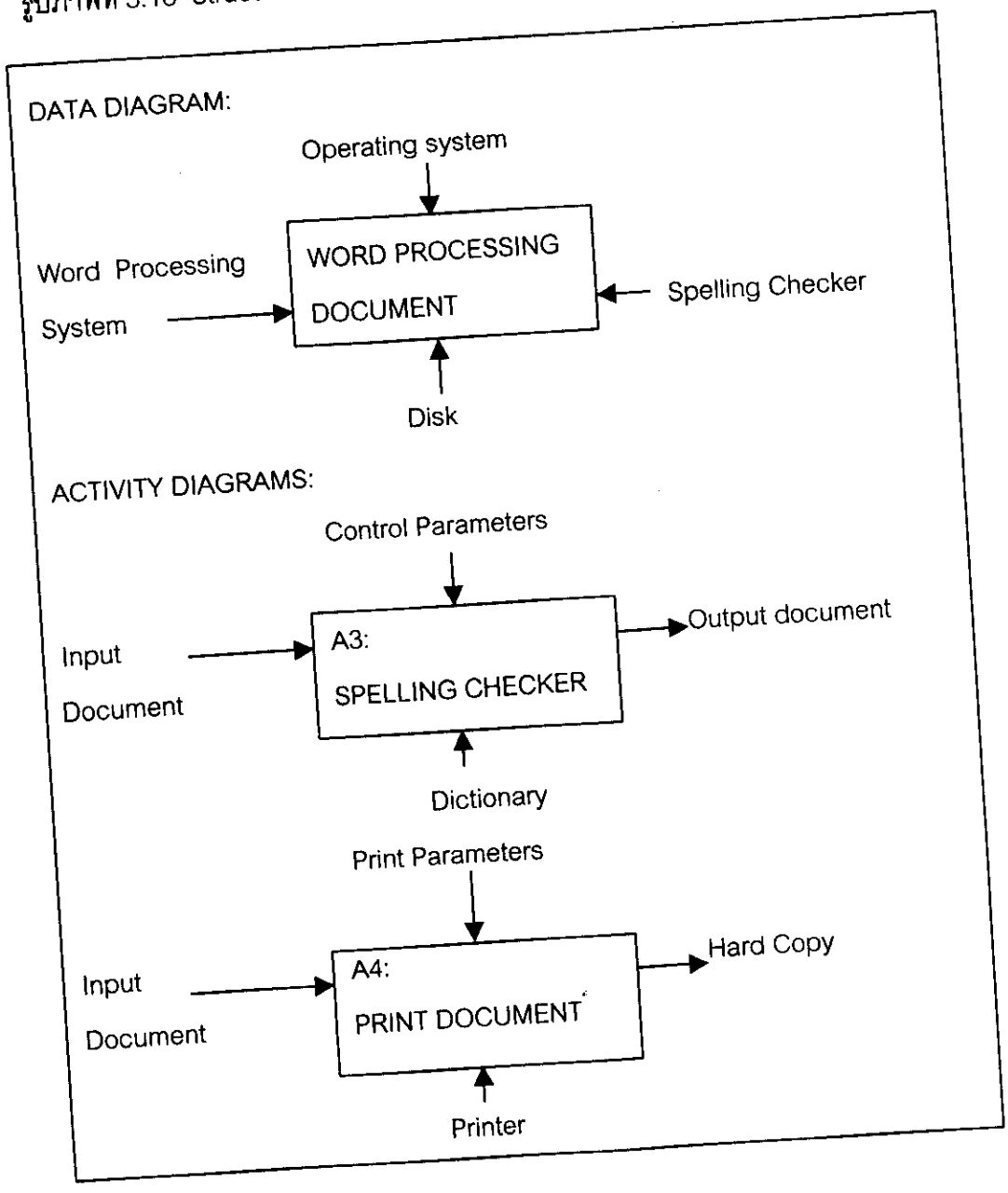
end

ซึ่ง ภาษา Gist นั้นเราสามารถระบุถึงความต้องการของระบบได้ สำหรับการตรวจสอบความต้องการนั้นเรามีเครื่องมืออัตโนมัติที่เรียกว่า Initial Operating Capability (IOC) โดยใช้ข้อมูลทดสอบ ตรวจสอบผลลัพธ์ของความต้องการ ทำให้ความต้องการที่ระบุมีความถูกต้อง

รูปภาพที่ 3.17 Structured Analysis(SA) Network Diagram



รูปภาพที่ 3.18 Structures Analysis(SA) detailed Data and Activity Diagrams



3.6

เอกสารความต้องการ

ความต้องการต่างๆที่กำหนดหรือระบุรายละเอียดด้วยเทคนิคหรือเครื่องมือใดๆก็ตาม ต้องมีการจัดเก็บเป็นเอกสารเพื่อใช้สำหรับใช้ในการพัฒนาและบำรุงรักษาระบบ โดยมีความชัดเจนและเข้าใจได้ง่าย กรณีที่เป็นไดอะแกรมต้องมีข้อความแสดงให้ครบถ้วนถูกต้อง ควรใช้ตัวเลขเป็นค่าที่ใช้อ้างอิงถึงกิจกรรมต่างๆ เพื่อให้ง่ายต่อการติดตาม และ จัดการโครงร่างเมื่อมีการเปลี่ยนแปลงความต้องการเกิดขึ้น

Requirements Definition Document

เป็นเอกสารสำหรับกำหนดความต้องการสำหรับลูกค้า เพื่อบรรยายให้ลูกค้าทราบว่าระบบสามารถกระทำกิจกรรมใดได้บ้าง การบรรยายควรประกอบไปด้วยข้อมูลต่างๆดังนี้

1. general purpose อันดับแรกต้องบรรยายถึงจุดประสงค์ทั่วไปของการพัฒนาระบบ อ้างถึงระบบอื่นๆที่เกี่ยวข้อง โดยอธิบายถึงประโยชน์ที่ได้รับจากการพัฒนาระบบนี้
2. background and objective ต่อมาเป็นการบรรยายถึงความเป็นมาและวัตถุประสงค์ของระบบที่ต้องการพัฒนา โดยอธิบายถึงปัญหาที่เกิดขึ้นในระบบปัจจุบันว่ามีสิ่งใดบ้างโดยเขียนให้ชัดเจน
3. description of the approach เป็นโครงร่างความต้องการของลูกค้า เมื่อลูกค้าประสงค์ที่จะแก้ปัญหาของระบบปัจจุบัน รายละเอียดนี้เป็นการบรรยายถึงปัญหามิใช่การแก้ปัญหา โดยบรรยายในลักษณะที่ว่าระบบควรเป็นเช่นไรถึงแก้ปัญหาให้กับลูกค้าได้ กรณีมีข้อกำหนดที่ลูกค้าต้องการต้องบรรยายในเอกสารให้ชัดเจนด้วย
4. detailed characteristics of the proposed system หลังจากบรรยายภาพรวมของปัญหาและความต้องการทั้งหมดของระบบแล้ว ต้องบรรยายคุณลักษณะโดยละเอียดของระบบที่จะพัฒนาด้วย โดยกำหนดขอบเขต ฟังก์ชันต่างๆของระบบ การปฏิสัมพันธ์ระหว่างกัน รวมทั้งบรรยายถึงคุณลักษณะของข้อมูล ความสัมพันธ์

ระหว่างข้อมูลและฟังก์ชัน ข้อมูลเข้าและข้อมูลออกของแต่ละโปรเซส รวมถึงกำหนดถึงประสิทธิภาพของระบบที่ต้องการด้วยเช่น ความถูกต้อง เวลาที่ต้องการเป็นต้น

5. environment กำหนดถึงสภาพแวดล้อมที่ต้องการ เช่นการสนับสนุน(support), ความปลอดภัย(security) , ความเป็นส่วนตัว(privaacy) , ฮาร์ดแวร์หรือซอฟต์แวร์ที่ต้องการ เป็นต้น

Requirements Specification Document

เอกสารระบุความต้องการเป็นเอกสารที่เขียนแนวเดียวกับเอกสารกำหนดความต้องการ แตกต่างกันที่วิธีการบรรยาย โดยเอกสารกำหนดความต้องการจะเขียนสำหรับลูกค้าที่ทำให้ลูกค้าเข้าใจได้ง่ายแต่เอกสารระบุความต้องการเขียนสำหรับผู้พัฒนา ซึ่งเขียนในเทอมของเทคนิค เช่นถ้าลูกค้าต้องการให้ทำการคำนวณค่าที่ซับซ้อนค่าหนึ่งการบรรยายเป็นการบรรยายเป็นตัวหนังสือที่เข้าใจได้ง่าย แต่ถ้าเป็นเอกสารระบุความต้องการเราจะบรรยายเป็นสูตรหรือสมการที่สามารถนำไปออกแบบหรือพัฒนาได้

ตัวอย่าง ของการบรรยายความต้องการ satellite tracking system โดยเอกสารกำหนดความต้องการของลูกค้ามีการบรรยายดังนี้

INITIATE TRACK ON IMAGE:

Logical processing shall be done to INITIATE TRACK ON IMAGE . This shell have as input HANDDOVER_DATA . This shell have as output HOIQ ,STATE DATA,and IMAGE ID. This logical processing shell, when appropriate ,identify a new instance of IMAGE. This logical processing ,when appropriate shell identify the type of entity instance as being IMAGE ON TRACK NOTE: A request for pulses is made by entering a formal record into the HOIQ whice feeds the pulse-send procedure.

สมมุติว่าเลือกใช้โปรแกรมอัตโนมัติ RSL เพื่อสร้างเอกสารระบุความต้องการ ปรากฏผลดังนี้

ALPHA: INITIATE_TRACK_ON_IMAGE.

INPUTS: HANDOVER_DATA

OUTPUTS: HOIQ , STATE_DATA , IMAGE_ID

CREATES: IMAGE .

SETS: IMAGE_ON_TRACK.

DESCRIPTION: A REQUEST FOR PULSE IS MADE BY ENTERING A FORMAL RECORD REQUEST INTO THE HOIQ WHICH FEEDS THE PULSE SENDING PROCEDURES.

3.7

REQUIREMENTS REVIEW

ก่อนที่จะเริ่มพัฒนาระบบนั้นต้องมีการตรวจสอบถึงความต้องการอีกครั้งหนึ่งเราเรียกว่า requirements review โดยมีการประชุมกันระหว่างตัวแทนจากผู้พัฒนาและลูกค้า อันประกอบไปด้วย ทีมงานออกแบบ ทีมงานทดสอบ ทีมงานจัดการโครงร่าง ผู้ใช้ระบบ ผู้บริหารระดับต่างๆ ฯลฯ โดยทำการตรวจสอบในเรื่องดังต่อไปนี้

1. ตรวจสอบถึงเป้าหมายและวัตถุประสงค์ของระบบ
2. ทำการเปรียบเทียบความต้องการที่กำหนดกับเป้าหมายและวัตถุประสงค์ว่าความต้องการที่กำหนดนั้นตรงกับเป้าหมายที่วางไว้และเป็นสิ่งที่จำเป็นในการพัฒนา
3. ตรวจสอบถึงสภาพแวดล้อมที่ได้กำหนดหรือระบุไว้ โดยตรวจสอบถึงการปฏิสัมพันธ์ระหว่างระบบกับระบบภายนอกอื่นๆ เช่น การไหลของสารสนเทศ โครงสร้างของระบบโดยตรวจสอบถึงความถูกต้องและความสมบูรณ์ การระบุฟังก์ชัน

ต่างๆของระบบต้องมีความคงที่และมีขอบเขตตามที่ลูกค้าต้องการ ข้อกำหนดต่างๆ ต้องตรวจสอบว่าสามารถกระทำได้หรือพัฒนาได้จริง

4. ถ้ามีความเสี่ยงในการพัฒนาหรือในระบบงานจริง ต้องทำการประเมินเพื่อหาหนทางที่ดีที่สุดในการแก้ปัญหาโดยตกลงถึงวิธีการที่ใช้
5. ทำการตกลงในเรื่องของการทดสอบระบบ , วิธีการในการแก้ไขความต้องการให้สมบูรณ์ ,วิธีการที่ทีมงานทดสอบตรวจสอบความต้องการ ,ผู้ที่มีหน้าที่เตรียมข้อมูลทดสอบ

เมื่อความต้องการได้ถูกตรวจสอบโดยสมบูรณ์ ทั้งผู้พัฒนาและลูกค้าจะรู้สึกสบายใจ เพราะมีความเข้าใจที่ตรงกัน ผู้พัฒนาสามารถพัฒนาระบบให้ตรงกับวัตถุประสงค์ที่ลูกค้าต้องการได้

3.8

สรุป

สำหรับในบทนี้เป็นขั้นตอนแรกในการพัฒนาซอฟต์แวร์มีวัตถุประสงค์เพื่อให้ทราบว่าปัญหาของระบบคืออะไร เพื่อกำหนดและระบุถึงความต้องการ ดังนั้นผู้พัฒนาต้องเริ่มจาก การวิเคราะห์ถึงปัญหา โดยรวบรวมข้อมูลต่างๆของระบบงานปัจจุบันจากการสัมภาษณ์ การออกแบบสอบถาม การสังเกตุ หรือรวบรวมเองจากแหล่งข้อมูลภายในองค์กรและภายนอกองค์กรเป็นเอกสารต่างๆที่มีอยู่ในระบบ

จากนั้นนำความต้องการต่างๆมาสร้างเป็นเอกสารกำหนดความต้องการ โดยใช้ static technique หรือ dynamic technique ในรูปของสมการ ตารางการตัดสินใจ หรือ transaction diagram เราสามารถบรรยายระบบในเทอมของฟังก์ชันหรือการเปลี่ยนสถานะของกิจกรรมได้

ต่อจากนั้นสร้างเอกสารระบุรายละเอียดความต้องการ โดยใช้วิธีการ manual หรืออัตโนมัติ ขึ้นอยู่กับขนาด ความต้องการ และความซับซ้อนของโครงงาน ที่นิยมกันคือ HIPO charts , Data Flow Diagram ,PSL/PSA , SREM , Gist

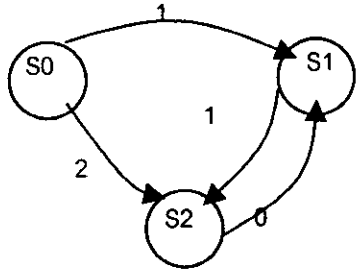
โดยเอกสารความต้องการที่ดีนั้นต้องมีการบรรยายที่ดีต้องมีความชัดเจน ถูกต้อง คงที่ สมบูรณ์ เป็นจริงได้ จำเป็น ตรวจสอบ และติดตามได้ง่าย เพราะเอกสารนี้จะถูกนำไปใช้ในการอ้างอิงในการพัฒนาและทดสอบระบบ เมื่อมีการส่งมอบระบบให้กับลูกค้า

ดังนั้นก่อนที่จะออกแบบระบบเพื่อหาหนทางในการแก้ปัญหาให้ได้ตามความต้องการของลูกค้า ต้องทำการตรวจสอบความต้องการระบบให้แน่ใจอีกครั้งหนึ่งพร้อมทั้งตกลงในเรื่องของความเสี่ยงและวิธีการดำเนินงานต่างๆให้เข้าใจตรงกัน ซึ่งเทคนิคดังกล่าวนี้เรียกว่า requirements review

3.9

แบบฝึกหัด

1. ความต้องการหมายถึงอะไร จงอธิบายและยกตัวอย่างประกอบ
2. จงอธิบายความแตกต่างระหว่าง Requirement definition document กับ Requirement specification document
3. จงอธิบายความแตกต่างระหว่าง Functional requirements กับ Nonfunctional requirements
4. ความต้องการแบ่งออกเป็นกี่ชนิด อะไรบ้าง ยกตัวอย่างให้เห็นจริง
5. คุณลักษณะของความต้องการที่ดีมีอะไรบ้าง
6. จงเขียนตารางตัดสินใจที่รวบรวมถึงกฎเกณฑ์ต่างๆของเกมส์ checker
7. จงเขียน event table เพื่อบรรยายผลลัพธ์ของการหาค่ารากของสมการยกกำลังสองใดๆ
8. จงเขียนตาราง Transition จาก State Transition Diagram ต่อไปนี้



9. จงเขียน HIPO chart ของระบบการคิดเงินเดือนของบริษัทแห่งหนึ่ง

3.10

ปฏิบัติ

1. ให้นักศึกษารวบรวมข้อมูลความต้องการของระบบที่ต้องการพัฒนา ทำแผนงานในการพบปะพูดคุยกับผู้ใช้ระบบ ลูกค้า ผู้จัดการในระดับต่างๆ โดยเตรียมหัวข้อสัมภาษณ์หรือแบบสอบถาม หรือ สังเกตุ หรือรวบรวมจากเอกสารที่มีอยู่ในระบบ โดยรวบรวมชนิดความต้องการต่างๆจากหัวข้อ 3.2 ของบทนี้
2. ให้นักศึกษาเขียน Requirement Definition Document เพื่อเป็นเอกสารที่บรรยายให้เห็นถึงปัญหาและความต้องการของระบบ
3. นักศึกษาเขียน Requirement Specification Document โดยเลือกเทคนิคหรือวิธีการที่เหมาะสมที่ผู้ออกแบบระบบและผู้ทดสอบระบบสามารถทำความเข้าใจได้ง่าย

หมายเหตุ แนวทางการเขียนเอกสารให้ใช้โครงร่างในหัวข้อที่ 3.6 โดยเริ่มจากบรรยายจุดประสงค์ทั่วไป ความเป็นมา แนวทางการแก้ปัญหา รายละเอียดของระบบที่ต้องการ สภาพแวดล้อม ส่งอาจารย์