

## การพัฒนาาระบบจัดการเวอร์ชันชุดคำสั่ง



นายสุเมธ เตชาพิสุทธิ์



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาดำเนินการตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

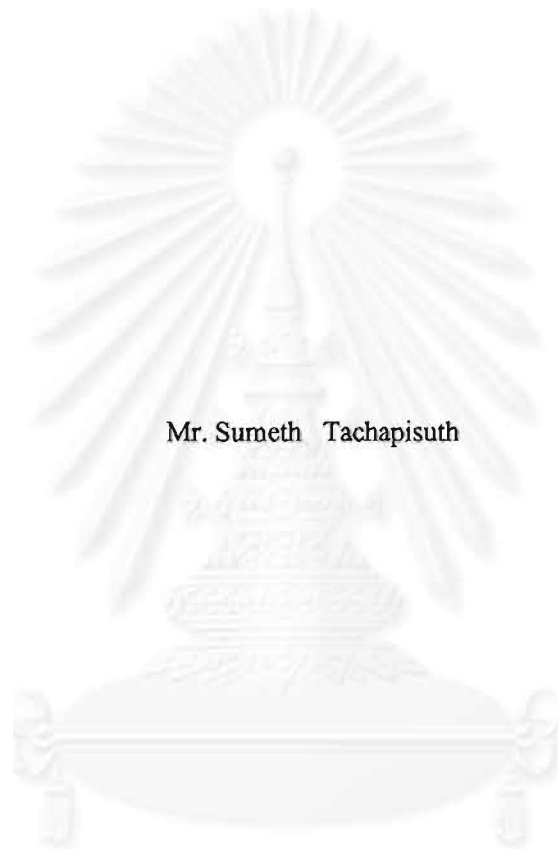
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2542

ISBN 974-333-643-5

ลิขสิทธิ์ของ จุฬาลงกรณ์มหาวิทยาลัย

# DEVELOPMENT OF A SOURCE CODE VERSIONS MANAGEMENT SYSTEM



Mr. Sumeth Tachapisuth

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

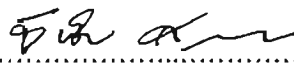
Academic Year 1999

ISBN 974-333-643-5

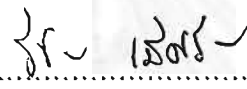
หัวข้อวิทยานิพนธ์      การพัฒนาระบบจัดการเวอร์ชันชุดคำสั่ง  
โดย                              นาย สุเมธ เตชาพิสุทธิ์  
ภาควิชา                        วิศวกรรมคอมพิวเตอร์  
อาจารย์ที่ปรึกษา            อาจารย์ วิวัฒน์ วัฒนาวุฒิ  
อาจารย์ที่ปรึกษาร่วม      รองศาสตราจารย์ ดร. วันชัย รื้อไพบูลย์

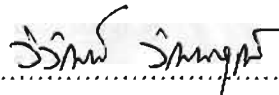
---


คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัยเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาโท

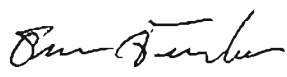
  
.....คณบดีคณะวิศวกรรมศาสตร์  
(รองศาสตราจารย์ ดร. ชัชชัย สุมิตร)

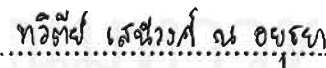
คณะกรรมการสอบวิทยานิพนธ์

  
.....ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ วิชาญ เลิศวิภาตระกูล)

  
.....อาจารย์ที่ปรึกษา  
(อาจารย์ วิวัฒน์ วัฒนาวุฒิ)

  
.....อาจารย์ที่ปรึกษาร่วม  
(รองศาสตราจารย์ ดร. วันชัย รื้อไพบูลย์)

  
.....กรรมการ  
(ผู้ช่วยศาสตราจารย์ ธนาวรรณ จันทร์คนไพบูลย์)

  
.....กรรมการ  
(อาจารย์ ดร. ทวีชัย เสนีวงศ์ ณ อยุธยา)

สุเมธ เตชาพิสุทธิ : การพัฒนาระบบจัดการเวอร์ชันซอร์สโค้ด (A DEVELOPMENT OF SOURCE CODE VERSIONS MANAGEMENT SYSTEM) อ. ที่ปรึกษา : อาจารย์ วิวัฒน์ วัฒนาวุฒิ, อ. ที่ปรึกษาร่วม : รศ. ดร. วันชัย รั้วไพบูลย์, 87 หน้า. ISBN 974-333-643-5.

ในวิทยานิพนธ์ฉบับนี้ ระบบจัดการเวอร์ชันซอร์สโค้ด ได้ถูกพัฒนาขึ้นโดยใช้เทคนิคการบีบอัดข้อมูลแบบพิเศษ ที่เรียกว่า วิธีการเก็บข้อมูลแบบเคลด้า เพื่อให้ประหยัดเนื้อที่ในการเก็บ โดยจัดเก็บเฉพาะซอร์สโค้ดเวอร์ชันล่าสุดและสิ่งที่แก้ไขในแต่ละเวอร์ชัน ซึ่งเป็นการลดการจัดเก็บในส่วนที่ซ้ำซ้อนกัน ผู้พัฒนาโปรแกรมสามารถเรียกซอร์สโค้ดของโปรแกรมเวอร์ชันย้อนหลังกลับมาใช้ได้หากต้องการ นอกจากนี้ยังมีการเพิ่มขีดความสามารถของระบบโดยขณะที่ ระบบจัดการเวอร์ชันซอร์สโค้ดทำการจัดเก็บซอร์สโค้ดเข้าสู่ระบบนั้น จะทำการตรวจสอบความสัมพันธ์ระหว่างมอดูลต่างๆ ของซอร์สโค้ด และทำการเตือนในกรณีที่เกิดความขัดแย้งขึ้น

หลักการงานหลักคือ การตรวจสอบการเรียกใช้มอดูล และการส่งผ่านข้อมูลระหว่างมอดูล โดยไม่ต้องอาศัยข้อมูลใดๆ จากเอกสารออกแบบของโปรแกรมอีก ด้วยความสามารถนี้ทำให้ระบบจัดการเวอร์ชันซอร์สโค้ดที่พัฒนาขึ้นในการวิทยานิพนธ์นี้เป็นเครื่องมือ โครงแบบซอฟต์แวร์ชิ้นหนึ่งที่ใช้งานง่าย และสะดวกขึ้นในการที่จะจัดการและประกันความสมบูรณ์ของเอกสารในระบบซอฟต์แวร์ที่จะเกิดขึ้นจากการเปลี่ยนซอร์สโค้ด และลดผลกระทบที่เกิดขึ้นวงกว้างทั่วทั้งระบบที่กำลังพัฒนา เมื่อมีความผิดพลาดเกิดขึ้นเพียงจุดใดจุดหนึ่ง

ระบบจัดการเวอร์ชันซอร์สโค้ดนี้ได้ทำการทดสอบกับโครงการจำลองขนาดกลางที่พัฒนาโปรแกรมด้วยภาษา โฟท์จีแอล ปรากฏว่าสามารถตรวจสอบและเตือนผู้ใช้ระบบได้อย่างถูกต้อง โดยระบบจัดการเวอร์ชันซอร์สโค้ดจะทำการแสดงข้อมูลของผลกระทบที่เกิดจากการแก้ไขเปลี่ยนแปลงมอดูลโปรแกรมที่มีความสัมพันธ์

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมคอมพิวเตอร์  
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์  
ปีการศึกษา 2542

ลายมือชื่อนิติ สุเมธ เตชาพิสุทธิ  
ลายมือชื่ออาจารย์ที่ปรึกษา วิวัฒน์ วัฒนาวุฒิ  
ลายมือชื่ออาจารย์ที่ปรึกษาร่วม วันชัย รั้วไพบูลย์

## 3972236321 : MAJOR COMPUTER SCIENCE

KEY WORD: DATA COUPLING / SOURCE CODE CONTROL SYSTEM / FUNCTIONAL  
DEPENDENCY / 4GL

SUMETH TACHAPISUTH : A DEVELOPMENT OF SOURCE CODE VERSIONS  
MANAGEMENT SYSTEM. THESIS ADVISOR : WIWAT VATANAWOOD, THESIS  
COADVISOR : ASSOC. PROF. WANCHAI RIVEPIBOON, 87 PP. ISBN 974-333-643-5

In this thesis, a source code versions management system is developed using the delta storage technique to minimize the usage of storage area. It means that source code common to more than one version are not duplicated. The developer is capable to retrieve previous versions of program source codes, if needed. Moreover, several extended features are proposed during the source code control task as to alert the functional dependency among modules of program source codes.

In our approach, the data coupling dependency checking feature requires only program source codes as input without any other information from design document. It is a practical software configuration management tool that handles and ensures the completeness of existing documents in the software system. The developer is able to manage the risk of the changes of source codes and to reduce the ripple effect of errors occurring at one location and propagating through out the system.

This system has been tested using synthetic test source codes of a medium-scale 4GL application. The results are satisfactory since the source code versions management system can correctly report the impact of changing in program modules.

ภาควิชา วิศวกรรมคอมพิวเตอร์      ลายมือชื่อนิติศ กฤษณ์ เม่งชัย  
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์      ลายมือชื่ออาจารย์ที่ปรึกษา วิวัฒน์ วิวัฒน์  
ปีการศึกษา 2542      ลายมือชื่ออาจารย์ที่ปรึกษาร่วม กฤษณ์



## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ได้สำเร็จลุล่วงไปด้วยดี เพราะได้รับความช่วยเหลือแนะแนวทาง และแก้ไขข้อบกพร่องจาก อาจารย์ วิวัฒน์ วัฒนาวุฒิ และรองศาสตราจารย์ ดร. วันชัย ธีระไพบุลย์ รวมทั้งได้รับการตรวจสอบและแก้ไขเพื่อความสมบูรณ์ และถูกต้องจากคณะกรรมการสอบวิทยานิพนธ์ ผู้วิจัยขอขอบพระคุณอาจารย์ทุกท่านเป็นอย่างสูง

ขอขอบพระคุณ บริษัท เคอร์เนลคอมพิวเตอร์ แอนด์ คอมมูนิเคชัน จำกัด ที่ให้ความช่วยเหลือในการจัดหาซอฟต์แวร์ และเครื่องคอมพิวเตอร์

ขอขอบพระคุณ บิดา มารดา ซึ่งได้ให้ความรัก และส่งเสริมให้ได้รับการศึกษาจนกระทั่งสำเร็จการศึกษาระดับปริญญาโท

ขอขอบพระคุณ ภรรยา ซึ่งคอยเป็นกำลังใจ ช่วยเหลือมาตลอดช่วงเวลาที่ทำการวิจัยฉบับนี้  
ขอขอบพระคุณเพื่อนๆ ทุกคนที่ให้กำลังใจ และคำแนะนำอันเป็นประโยชน์ต่อการทำวิจัยนี้

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฌ
สารบัญภาพ.....	ญ
บทที่	
1. บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์.....	4
1.3 ขอบเขตการวิจัย.....	4
1.4 ขั้นตอนและวิธีการดำเนินการวิจัย.....	5
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	5
2. ทฤษฎีที่นำมาใช้ในการวิจัย.....	6
2.1 การเก็บข้อมูลแบบเคลด้า.....	6
2.2 การวิเคราะห์ศัพท์.....	7
2.3 โครงสร้างและวากยสัมพันธ์ของภาษา อินฟอร์มมิกซ์ โฟท์จีแอล.....	11
3. การวิเคราะห์และออกแบบระบบ.....	16
3.1 การวิเคราะห์ระบบ.....	16
3.2 การออกแบบระบบ.....	18
3.3 การออกแบบส่วนจัดเก็บ.....	29
3.4 การออกแบบตัวประสานผู้ใช้.....	35
4. การทดสอบโปรแกรม.....	48
4.1 สภาพแวดล้อมในการทดสอบ โปรแกรม.....	48
4.2 การทดสอบโปรแกรม.....	49

## สารบัญ (ต่อ)

	หน้า
5. สรุปผลการวิจัยและเสนอแนะ .....	59
5.1 สรุปผลการวิจัย .....	59
5.2 ข้อจำกัดของระบบ .....	60
5.3 ข้อเสนอแนะในการพัฒนาเพิ่มเติม .....	60
5.4 ผลงานที่เกี่ยวข้องกับการวิจัย.....	61
รายการอ้างอิง.....	62
ภาคผนวก.....	63
ภาคผนวก ก. การใช้งานระบบจัดการเวอร์ชันชุดคำสั่ง .....	64
ภาคผนวก ข. ผลงานตีพิมพ์ในงาน NCSEC'99 .....	81
ประวัติผู้วิจัย.....	87



## สารบัญตาราง

ตารางที่	หน้า
2.1	ตัวดำเนินการของนิพจน์ปรกติ ..... 10
2.2	ชนิดข้อมูลในภาษาอินฟอร์มิกซ์ โฟท์จีแอล..... 15
3.1	โครงสร้างของตารางรายชื่อ โปรแกรม/ชุดคำสั่งมอดูล ..... 29
3.2	โครงสร้างของตารางรายชื่อฟังก์ชันที่จัดเก็บในระบบ..... 30
3.3	โครงสร้างของตารางรายละเอียดพารามิเตอร์ของฟังก์ชัน ..... 30
3.4	โครงสร้างของตารางรายชื่อผู้ใช้ระบบ..... 31
3.5	โครงสร้างของตารางสิทธิผู้ใช้ระบบในการเข้าถึง โปรแกรม/ชุดคำสั่งมอดูล ..... 31
3.6	โครงสร้างของตารางประวัติการลง โปรแกรม/ชุดคำสั่งมอดูล..... 31
3.7	โครงสร้างของตารางเก็บ โปรแกรม/ชุดคำสั่งมอดูล..... 32
3.8	โครงสร้างของตารางเก็บการเปลี่ยนแปลง โปรแกรม/ชุดคำสั่งมอดูล ..... 33
3.9	โครงสร้างของตารางรายชื่อฟังก์ชันที่มีการเรียกใช้ใน โปรแกรม/ชุดคำสั่งมอดูลชุดล่าสุด..... 33
3.10	โครงสร้างของตารางรายละเอียดพารามิเตอร์ของฟังก์ชันชุดล่าสุด..... 34
3.11	โครงสร้างของตารางประวัติหมายเหตุการแก้ไข โปรแกรม/ชุดคำสั่งมอดูล ..... 34

## สารบัญภาพ

รูปที่	หน้า
1.1	หน้าที่หลักของระบบควบคุม ..... 3
2.1	วิธีการจัดเก็บข้อมูลแบบเคลด้าและการนำข้อมูลกลับมา ..... 7
2.2	ภาพแสดงขั้นตอนในการใช้ lex ในการสร้างโปรแกรม lexical analyzer ..... 8
2.3	ตัวอย่างการกำหนดหมายเหตุในภาษาอินฟอร์มิคซ์ ไฟท์จีแอล ..... 13
3.1	แผนภาพบริบทของระบบจัดการเวอร์ชันชุดคำสั่ง ..... 18
3.2	แผนภาพกระแสข้อมูลระดับ 0 ของระบบจัดการเวอร์ชันชุดคำสั่ง ..... 19
3.3	แผนภาพกระแสข้อมูลระดับ 1 ของส่วนจัดเก็บ โปรแกรม ..... 20
3.4	แผนภาพกระแสข้อมูลระดับ 1 ของส่วนขอเบ็ก โปรแกรม ..... 21
3.5	แผนภาพกระแสข้อมูลระดับ 1 ของส่วนตรวจสอบและแสดงความสัมพันธ์ที่ขัดแย้ง ..... 22
3.6	แผนภาพกระแสข้อมูลระดับ 1 ของส่วนกำหนดผู้ใช้ระบบและสิทธิในการใช้งาน ..... 23
3.7	แผนภาพกระแสข้อมูลระดับ 1 ของส่วนตรวจสอบและแสดงความแตกต่าง ของชุดคำสั่ง โปรแกรม ..... 24
3.8	แผนภาพโครงสร้างของโปรแกรม Pcheckin ..... 25
3.9	แผนภาพโครงสร้างของโปรแกรม Pcheckout ..... 26
3.10	แผนภาพโครงสร้างของโปรแกรม Pcheckrel ..... 27
3.11	แผนภาพโครงสร้างของโปรแกรม Padmin ..... 27
3.12	แผนภาพโครงสร้างของโปรแกรม Pcheckdiff ..... 28
3.13	หน้าจอลงบันทึกเข้า ..... 36
3.14	หน้าจอแสดงความผิดพลาดของรหัสผู้ใช้หรือรหัสผ่านของผู้ใช้ ..... 36
3.15	หน้าจอแสดงความผิดพลาดเนื่องจากผู้ใช้ไม่มีสิทธิในการใช้ ..... 36
3.16	หน้าจอ โปรแกรมจัดเก็บ โปรแกรมหรือชุดคำสั่งมอดูล ..... 37
3.17	หน้าจอป้อนคำอธิบายอย่างย่อสำหรับโปรแกรมหรือชุดคำสั่งมอดูล ..... 39
3.18	หน้าจอป้อนหมายเหตุการแก้ไขสำหรับโปรแกรมหรือชุดคำสั่งมอดูล ..... 39
3.19	หน้าจอ โปรแกรมขอเบ็ก โปรแกรมหรือชุดคำสั่งมอดูล ..... 40
3.20	หน้าจอแสดงประวัติการแก้ไขของ โปรแกรมหรือชุดคำสั่งมอดูล ..... 41
3.21	หน้าจอป้อนรายละเอียดเกี่ยวกับลูกค้า ..... 42

## สารบัญภาพ (ต่อ)

รูปที่	หน้า
3.22	หน้าจอแสดงความสัมพันธ์ที่ขัดแย้งระหว่าง โปรแกรมหรือชุดคำสั่งมอดูล ..... 42
3.23	หน้าจอโปรแกรมกำหนดผู้ใช้ระบบและสิทธิในการใช้งาน ..... 44
3.24	หน้าจอโปรแกรมกำหนดผู้ใช้ระบบส่วนสิทธิในการเข้าถึงชุดคำสั่งที่จัดเก็บ ..... 45
3.25	หน้าจอโปรแกรมตรวจสอบและแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูล.... 47
4.1	หน้าต่างป้อนรหัสผู้ใช้และรหัสผ่าน ขณะ โปรแกรมทำงาน ..... 49
4.2	เพิ่มชุดคำสั่ง โปรแกรม Calcomm.4gl ที่ต้องการจัดเก็บ..... 50
4.3	หน้าจอแสดงการจัดเก็บชุดคำสั่ง โปรแกรม Calcomm.4gl ..... 51
4.4	แสดงการป้อนหน้าต่างป้อนคำอธิบายอย่างย่อ..... 52
4.5	หน้าจอโปรแกรมขอเบิกโปรแกรม Calcomm เพื่อทำการแก้ไข ..... 53
4.6	หน้าต่างป้อนชื่อเพิ่มสำหรับจัดเก็บชุดคำสั่งที่ขอเบิกจากระบบ ..... 53
4.7	ชุดคำสั่ง โปรแกรม Calcomm.4gl ที่ถูกแก้ไข โดยใช้ Notepad ..... 54
4.8	หน้าจอจัดเก็บชุดคำสั่ง โปรแกรม Calcomm.4gl ที่ได้รับการแก้ไขแล้ว ..... 55
4.9	แสดงการป้อนหน้าต่างหมายเหตุการแก้ไข โปรแกรมหรือชุดคำสั่งมอดูล ..... 56
4.10	เพิ่มชุดคำสั่ง โปรแกรม Calcomm.4gl เวอร์ชัน 1.0 ..... 56
4.11	เพิ่มชุดคำสั่ง โปรแกรม CustOrder.4gl ที่ต้องการจัดเก็บ ..... 57
4.12	หน้าจอแสดงการจัดเก็บชุดคำสั่ง โปรแกรม Custorder.4gl ..... 57
ก.1	หน้าต่างลงบันทึกเข้า..... 65
ก.2	หน้าต่างหลักโปรแกรมจัดเก็บเพิ่มข้อมูลชุดคำสั่งมอดูล..... 65
ก.3	โปรแกรมทำการแสดงชุดคำสั่งในเพิ่มข้อมูลชุดคำสั่งมอดูลที่ต้องการจัดเก็บ..... 66
ก.4	แสดงการป้อนเมื่อชุดคำสั่งมอดูลที่ต้องการจัดเก็บที่ไม่เคยมีการจัดเก็บมาก่อน..... 67
ก.5	แสดงการป้อนเมื่อชุดคำสั่งมอดูลที่ต้องการจัดเก็บที่เคยมีการจัดเก็บมาก่อน ..... 67
ก.6	แสดงการแจ้งเตือนผลกระทบเมื่อทำการจัดเก็บชุดคำสั่งมอดูล ..... 68
ก.7	หน้าต่างลงบันทึกเข้า..... 69
ก.8	หน้าต่างหลักโปรแกรมเบิกเพิ่มข้อมูลชุดคำสั่งมอดูล ..... 70
ก.9	แสดงการเบิกชุดคำสั่งมอดูลที่ต้องการแก้ไข ..... 70
ก.10	หน้าต่างสำหรับป้อนชื่อเพิ่มชุดคำสั่งมอดูลที่เบิกจากระบบ ..... 71

## สารบัญญภาพ (ต่อ)

รูปที่	หน้า
ก.11 หน้าต่างป้อนรายละเอียดเกี่ยวกับลูกค้า.....	71
ก.12 หน้าต่างแสดงประวัติการแก้ไขของชุดคำสั่งมอดูล.....	72
ก.13 หน้าต่างลงบันทึกเข้า.....	73
ก.14 หน้าต่างหลัก โปรแกรมแสดงความสัมพันธ์ที่ขัดแย้งระหว่างชุดคำสั่งมอดูล.....	73
ก.15 แสดงความสัมพันธ์ที่ขัดแย้งระหว่างชุดคำสั่งมอดูล.....	74
ก.16 แสดงรายชื่อชุดคำสั่งมอดูลอื่นที่เรียกใช้ฟังก์ชันที่กำหนดภายในชุดคำสั่งมอดูล.....	75
ก.17 หน้าต่างลงบันทึกเข้า.....	76
ก.18 โปรแกรมกำหนดผู้ใช้และสิทธิในการใช้งาน ขณะกำหนดผู้ใช้ใหม่ลงสู่ระบบ.....	76
ก.19 โปรแกรมกำหนดผู้ใช้และสิทธิในการใช้งาน ขณะแก้ไขข้อมูลผู้ใช้ระบบ.....	77
ก.20 แสดงการกำหนดสิทธิในการเข้าถึงชุดคำสั่งมอดูลของผู้ใช้ระบบ.....	78
ก.21 หน้าต่างลงบันทึกเข้า.....	79
ก.22 หน้าต่างหลัก โปรแกรมแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูล.....	80
ก.20 แสดงการเปรียบเทียบความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูล.....	80

บทที่ 1

บทนำ



## 1.1 ความเป็นมาและความสำคัญของปัญหา

ในโครงการการพัฒนาโปรแกรมที่แม้ว่าได้ทำการกำหนดขอบเขตเป็นอย่างดีและมีการควบคุมเป็นอย่างดีก็ตาม แต่วิศวกรซอฟต์แวร์มักจำเป็นต้องปรับปรุงโค้ดใหม่เพื่อให้ตรงกับคุณสมบัติใหม่ๆ และฟังก์ชันใหม่ๆ ตามที่ผู้ใช้งานต้องการหรือเพื่อให้สามารถแข่งขันทางการค้า และบางครั้งที่นักพัฒนาโปรแกรมจำเป็นต้องหันกลับมาเริ่มต้นงานใหม่ เมื่อต้องการทำการเปลี่ยนแปลงบางอย่าง นอกจากนั้นโครงการการพัฒนาโปรแกรมของบริษัทผู้พัฒนาโปรแกรม (Software House) ซึ่งจำเป็นต้องให้บริการต่อลูกค้าที่ใช้ซอฟต์แวร์ที่มีเวอร์ชันแตกต่างกันออกไปจำเป็นต้องหาเครื่องมือเข้ามาเพื่อช่วยจัดเก็บ โปรแกรมที่มีเวอร์ชันแตกต่างกัน

ในโครงการการพัฒนาโปรแกรมขนาดใหญ่ มักจะมีการพัฒนาโปรแกรมเป็นทีม แต่การพัฒนาโปรแกรมเป็นทีมก็ส่งผลให้ความสลับซับซ้อนเพิ่มมากขึ้นด้วยเช่นกัน เนื่องมาจากการทำงานของสมาชิกคนหนึ่งภายในทีมพัฒนาอาจเข้าไปแทรกแซงการทำงานของสมาชิกอื่นในทีมได้โดยไม่ได้ตั้งใจ อันเนื่องจากการที่มีความต้องการเปลี่ยนแปลงแก้ไขชุดคำสั่งมอดูลเดียวกันในช่วงระยะเวลาเดียวกัน ดังนั้นจำเป็นต้องมีมาตรการพิเศษในการควบคุมการเปลี่ยนแปลง แก้ไขชุดคำสั่งมอดูลดังกล่าว

เครื่องมือประเภท Software Configuration Management (SCM) [1] ได้ถูกนำมาเข้ามาช่วยแก้ไขปัญหาของงานควบคุมโครงการการพัฒนาโปรแกรมและเอาไว้ช่วยหลีกเลี่ยงปัญหาที่เกิดจากการพัฒนาโปรแกรมเป็นทีม รวมทั้งคอยดูแลสภาพของซอฟต์แวร์ เครื่องมือเหล่านี้ได้พัฒนาขึ้นครั้งแรก และมีใช้งานเฉพาะในเครื่องคอมพิวเตอร์ขนาดใหญ่ที่เป็นระบบหลายผู้ใช้เท่านั้น จนกระทั่งเมื่อไม่นานนี้ ได้มีการใช้เครื่องคอมพิวเตอร์ส่วนบุคคลมาทำการเชื่อมต่อเป็นเครือข่าย เพื่อนำมาใช้ในงานพัฒนาโปรแกรม จึงได้มีการพัฒนาเครื่องมือประเภท Software Configuration Management

ให้สามารถใช้งานบนเครื่องคอมพิวเตอร์ส่วนบุคคลที่ทำการเชื่อมต่อเป็นเครือข่าย และยัง ได้เพิ่มคุณสมบัติบางอย่างและปรับปรุงฟังก์ชันให้ใช้งานได้ง่ายขึ้น

ในการพัฒนาเครื่องมือควบคุมโครงการการพัฒนาโปรแกรมนั้น ปัญหาบางอย่างของเครื่องมือควบคุมได้ก่อให้เกิดเทคโนโลยีพิเศษมารองรับ เนื่องจากปัญหาของการจัดเก็บชุดคำสั่งมอดูล รวมทั้งการเปลี่ยนแปลงหลายๆครั้ง การเก็บไฟล์ทั้งหมดเอาไว้ หรือที่เรียกว่า การจัดเก็บข้อมูลแบบ โอเมกา (Omega storage) [1] ทำให้สูญเสียพื้นที่ในการจัดเก็บเป็นอย่างมาก โดยเฉพาะอย่างยิ่งถ้าหากมีการเปลี่ยนแปลงเกิดขึ้นทีละน้อย ดังนั้นเครื่องมือควบคุมจึงใช้เทคนิคการบีบอัดข้อมูลแบบพิเศษ ที่เรียกว่าวิธีการเก็บข้อมูลแบบเดลต้า (Delta storage) [1] ซึ่งชุดคำสั่งมอดูลที่ถูกจัดเก็บ จะทำการจัดเก็บไว้เพียงชุดเดียว(ซึ่งส่วนใหญ่ที่จัดเก็บไว้จะเป็นชุดคำสั่งมอดูลชุดล่าสุด) และจัดเก็บส่วนเปลี่ยนแปลงชุดคำสั่งมอดูลสำหรับเปลี่ยนกลับไปเป็นชุดคำสั่งมอดูลเวอร์ชันย้อนหลังควบคุมไว้ เมื่อต้องการใช้งานชุดคำสั่งมอดูลเวอร์ชันเก่าทำได้โดยการนำชุดคำสั่งมอดูลชุดที่เก็บไว้ มาประมวลผลกับส่วนเปลี่ยนแปลงชุดคำสั่งมอดูล เพื่อให้ได้ชุดคำสั่งมอดูลเวอร์ชันย้อนหลัง

ในการพัฒนาเครื่องมือควบคุมนี้ ต้องการให้สามารถจัดเก็บและตรวจสอบความเปลี่ยนแปลงระหว่างชุดคำสั่งมอดูลชุดเดียวกัน แต่ต่างเวอร์ชันกันได้ โดยนำโครงสร้างของตัวภาษามาช่วยในการพิจารณาความแตกต่างภายในมอดูล ดังนั้นจึงต้องนำวิธีการ เลขศัพท์ออนไลน์เซอร์ [2] (Lexical analyzer) ซึ่งเป็นวิธีการที่ใช้ในการทำงานของตัวแปลภาษา เพื่อทำหน้าที่แยกกลุ่มคำ (token) ของชุดคำสั่งมอดูล มาประยุกต์ใช้งานเพื่อการค้นหาส่วนที่เป็นความแตกต่างได้ถูกต้อง

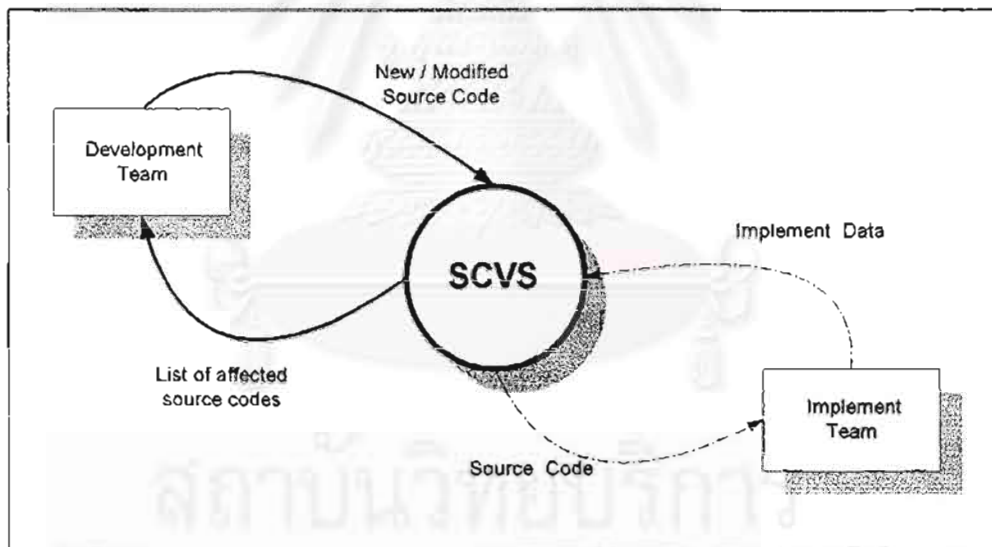
เครื่องมือที่มีใช้งานในปัจจุบัน ที่ขอกกล่าวถึงเป็นอันดับแรก คือ “Microsoft Visual SourceSafe” ผู้พัฒนาเครื่องมือนี้คือ บริษัท ไมโครซอฟต์ เวอร์ชันล่าสุด 5.0 เป็นเครื่องมือที่ใช้ในการควบคุมเวอร์ชันของโปรแกรมที่ได้ถูกพัฒนา โดยใช้เครื่องมือพัฒนาโปรแกรมที่เป็นผลิตภัณฑ์ของบริษัท ไมโครซอฟต์ เครื่องมือนี้ดังกล่าวทำงานบน ไมโครซอฟต์วินโดวส์ 95/NT วิธีที่ใช้ในการจัดเก็บชุดคำสั่งมอดูลที่ทางไมโครซอฟต์เลือกเป็นการจัดเก็บแบบเดลต้า (Delta storage) โดยใช้ระบบจัดเก็บข้อมูลเฉพาะซึ่งคล้ายกับระบบฐานข้อมูล ข้อดี ของเครื่องมือนี้ก็คือความเข้ากันได้อย่างกลมกลืนกันกับเครื่องมือชนิดอื่นๆของไมโครซอฟต์เอง ข้อเสีย ของเครื่องมือนี้ก็คือการที่เครื่องมือถูกออกแบบมาให้ใช้งานกับโปรแกรมต่างๆของไมโครซอฟต์เท่านั้น และระบบรักษาความปลอดภัยก็มีน้อยมาก โดยมีระบบรักษาความปลอดภัยเพียง 4 ระดับเท่านั้นเพื่อใช้ในการกำหนดสิทธิเพื่อเข้าไปใช้ระบบ แต่ไม่สามารถปรับเปลี่ยนให้มีระดับการควบคุมสูงกว่านี้ได้ โดย



สรุปแล้วเครื่องมือดังกล่าวจึงเหมาะสมกับผู้ใช้งานโปรแกรมต่างๆของไมโครซอฟต์เท่านั้น และเหมาะสมกับการควบคุมโครงการขนาดเล็กที่ไม่ต้องการการควบคุมมากนัก

เครื่องมืออีกชุด คือ “PVCS Version Manager” ของบริษัท Intersolv เวอร์ชันล่าสุด 5.2.20 ถือเป็นเครื่องมือควบคุมที่มีชื่อเสียงและเป็นที่รู้จักใช้งานอย่างแพร่หลายมากที่สุด เครื่องมือชุดนี้ใช้วิธีการจัดเก็บชุดคำสั่งมอดูลแบบเดลต้า (Delta storage) โดยใช้ระบบไฟล์ ข้อดี ของเครื่องมือนี้ก็คือการที่สามารถรองรับการทำงานบนหลายแพลตฟอร์ม ข้อเสีย ของเครื่องมือนี้ก็คือการใช้งานที่ยุ่งยากซับซ้อนมาก โดยสรุปแล้วเครื่องมือนี้เหมาะสมกับผู้ใช้งานที่ต้องการความสามารถในการทำงานข้ามแพลตฟอร์ม

เครื่องมือควบคุมโครงการการพัฒนาโปรแกรม มีหน้าที่หลักคือ ใช้ในการจัดเก็บชุดคำสั่งมอดูลที่ผู้พัฒนาโปรแกรมเขียนขึ้น และเมื่อมีการแก้ไขชุดคำสั่งมอดูลที่ส่งผลกระทบต่อชุดคำสั่งมอดูลอื่น เครื่องมือจะแสดงรายชื่อชุดคำสั่งมอดูลที่ได้รับผลกระทบให้แก่ผู้พัฒนาโปรแกรมทราบ (ดูรูปที่ 1.1)



รูปที่ 1.1 หน้าที่หลักของระบบควบคุม

ส่วนประกอบของเครื่องมือควบคุมโครงการการพัฒนาโปรแกรม แบ่งออกเป็น 4 ส่วนใหญ่ๆดังนี้คือ

- 1.1.1 ส่วนที่จัดเก็บชุดคำสั่งมอดูลใหม่/เปลี่ยนแปลง เป็นส่วนที่จัดเก็บชุดคำสั่งมอดูลด้วยวิธีการจัดเก็บแบบเดลต้า ซึ่งจัดเก็บอยู่ในฐานข้อมูลแบบสัมพันธ์

- 1.1.2 ส่วนที่นำชุดคำสั่งมอดูลออกมาเพื่อแก้ไขเปลี่ยนแปลง ประกอบด้วย
  - ส่วนที่ตรวจสอบสิทธิในการแก้ไขชุดคำสั่งมอดูล
  - ส่วนที่ทำหน้าที่นำชุดคำสั่งมอดูลจากวิธีการจัดเก็บแบบเคลด้า
  - ส่วนที่ทำหน้าที่บันทึกรายการผู้ที่นำชุดคำสั่งมอดูลไปแก้ไข
- 1.1.3 ส่วนที่ทำหน้าที่กำหนดสิทธิของผู้ที่ใช้งานเครื่องมือควบคุม
- 1.1.4 ส่วนที่นำชุดคำสั่งมอดูลออกมาเพื่อการติดตั้งใช้งาน

## 1.2 วัตถุประสงค์

สร้างเครื่องมือช่วยในการควบคุมการเปลี่ยนแปลงโปรแกรมสำหรับ โครงการการพัฒนาโปรแกรม

## 1.3 ขอบเขตการวิจัย

- 1.3.1 พัฒนาระบบเครื่องมือคอมพิวเตอร์ภายใต้โปรแกรมปฏิบัติการไมโครซอฟต์ วินโดวส์ 95 ขึ้นไป
- 1.3.2 เครื่องมือที่พัฒนาขึ้นในงานวิจัยครั้งนี้ ได้ออกแบบให้รองรับดูแลระบบงานที่พัฒนาขึ้นภายใต้ภาษาที่มีโครงสร้างและวากยสัมพันธ์ที่กำหนดโดยการ ใช้คำสำคัญ (key word) แต่จะยึดโครงสร้างและวากยสัมพันธ์ของภาษาเพียงภาษาหนึ่ง มาพัฒนาในการพัฒนาครั้งนี้ ซึ่งภาษาที่ใช้ในการวิจัยครั้งนี้คือ ภาษา อินฟอร์มิกซ์ โฟท์จีแอล [4,5] (Informix 4GL) โดยพัฒนารอบคลุมเฉพาะชนิดข้อมูลภายใน (Internal data type) ของภาษาเท่านั้น
- 1.3.3 สามารถเก็บรักษาชุดคำสั่งมอดูลในลักษณะเป็นเวอร์ชันได้
- 1.3.4 เปิดให้สามารถกำหนดสิทธิให้กับผู้แก้ไขเปลี่ยนแปลงชุดคำสั่งมอดูลที่ถูกจัดเก็บไว้ในระบบ
- 1.3.5 ช่วยป้องกันการแก้ไขชุดคำสั่งมอดูลเดียวกัน ในช่วงเวลาเดียวกัน
- 1.3.6 สามารถแสดงความแตกต่างระหว่างชุดคำสั่งมอดูลชุดเดียวกันที่ต่างเวอร์ชันกัน



## 1.4 ขั้นตอนและวิธีการดำเนินการวิจัย

- 1.4.1 ศึกษาทฤษฎี เลกซ์คัลลอนาไลเซอร์ และการใช้งาน lex & yacc [3]
- 1.4.2 ศึกษาความรู้ทางวินโดวส์และแนวทางการเขียนโปรแกรมสำหรับไมโครซอฟต์วินโดวส์
- 1.4.3 ศึกษาความรู้ด้านระบบฐานข้อมูลและแนวทางการเขียนโปรแกรมในการเข้าถึงระบบฐานข้อมูล
- 1.4.4 ออกแบบระบบ
  - ออกแบบค้ำประสานผู้ใช้
  - ออกแบบฐานข้อมูล
- 1.4.5 เขียนโปรแกรม
- 1.4.6 ทดสอบและประเมินผลการทำงานของโปรแกรม
- 1.4.7 สรุปผล เสนอแนะ และจัดทำรูปเล่มวิทยานิพนธ์

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 เป็นเครื่องมือช่วยในการควบคุมโครงการพัฒนาโปรแกรมขนาดใหญ่
- 1.5.2 ช่วยลดความยุ่งยากและเวลาในการดูแลควบคุมการเปลี่ยนแปลงแก้ไขโปรแกรม
- 1.5.3 ช่วยพัฒนาอุตสาหกรรมการผลิตซอฟต์แวร์

## บทที่ 2

### ทฤษฎีที่นำมาใช้ในการวิจัย

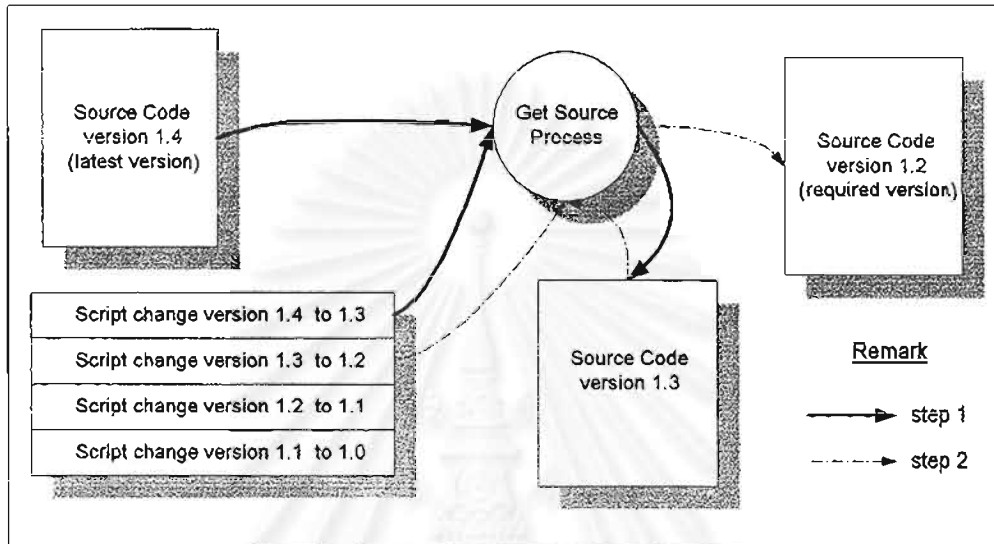
ในการพัฒนาเครื่องมือควบคุมโครงการการพัฒนาโปรแกรมนั้น เพื่อให้การจัดเก็บชุดคำสั่งมอดูลไม่สิ้นเปลืองเนื้อที่ในการจัดเก็บมาก จึงจำเป็นต้องอาศัยเทคนิคพิเศษที่เรียกว่าวิธีการเก็บข้อมูลแบบเดลต้า (Delta storage) ในการจัดเก็บ นอกจากการจัดเก็บชุดคำสั่งมอดูลแล้ว เมื่อมีการแก้ไขชุดคำสั่งมอดูลที่ส่งผลกระทบต่อชุดคำสั่งมอดูลอื่น เครื่องมือจะแสดงรายชื่อชุดคำสั่งมอดูลที่ได้รับผลกระทบให้แก่ผู้พัฒนาโปรแกรมทราบ และในงานวิจัยครั้งนี้ได้เลือกพัฒนาเครื่องมือที่ใช้ควบคุมโครงการที่พัฒนาด้วยภาษา อินฟอร์มิกซ์ โฟทจีแอล ดังนั้นผู้วิจัยจึงต้องทำการศึกษาการวิเคราะห์ศัพท์และ โครงสร้างของภาษา อินฟอร์มิกซ์ โฟทจีแอล เพื่อตรวจสอบใช้ในการตรวจสอบโครงสร้างและวากยสัมพันธ์

#### 2.1 การเก็บข้อมูลแบบเดลต้า (Delta storage) [1]

การเก็บข้อมูลแบบเดลต้า เป็นแนวคิดในการจัดเก็บข้อมูลที่พัฒนาขึ้นมาเพื่อใช้ในการเก็บข้อมูลที่เป็นชุดคำสั่งมอดูลเป็นเวอร์ชัน ซึ่งข้อมูลแบบดังกล่าวมักมีลักษณะคล้ายคลึงกัน มีการเปลี่ยนแปลงเพียงเล็กน้อยในแต่ละเวอร์ชัน การจัดเก็บแบบธรรมดาทำให้มีส่วนที่ต้องจัดเก็บซ้ำๆ กันมาก เสียพื้นที่ในการจัดเก็บ ลักษณะการจัดเก็บข้อมูลแบบเดลต้า จะทำการจัดเก็บข้อมูลชุดคำสั่งมอดูลไว้เพียงเวอร์ชันเดียว พร้อมกับจัดเก็บส่วนต่างที่ใช้ในการเปลี่ยนข้อมูลชุดคำสั่งมอดูลชุดดังกล่าว กลับไปเป็นชุดคำสั่งมอดูลเวอร์ชันอื่นๆ ทำให้ประหยัดพื้นที่ในการจัดเก็บ ซึ่งโดยทั่วไปข้อมูลชุดคำสั่งมอดูลที่ถูกเก็บไว้ นิยมเก็บเป็นชุดคำสั่งมอดูลเวอร์ชันล่าสุด ด้วยเหตุผลที่ว่าปกติแล้วชุดคำสั่งมอดูลเวอร์ชันล่าสุดมักมีการใช้งานมากกว่าเวอร์ชันอื่น

จากรูปที่ 2.1 ด้านซ้ายเป็นชุดคำสั่งโปรแกรมที่จัดเก็บไว้ด้วยวิธีการเก็บข้อมูลแบบเดลต้า โดยข้อมูลที่จัดเก็บไว้ประกอบด้วย ชุดคำสั่งโปรแกรมเวอร์ชันล่าสุด (เวอร์ชัน 1.4) และการเปลี่ยนแปลงของชุดคำสั่งแต่ละเวอร์ชัน ตั้งแต่เวอร์ชัน 1.0 ถึง 1.4 เมื่อต้องการชุดคำสั่งโปรแกรมย้อนหลังกลับเป็นเวอร์ชัน 1.2 สามารถทำได้โดยการนำชุดคำสั่งโปรแกรมเวอร์ชันล่าสุด มาทำการ

ประมวลผลร่วมกับการเปลี่ยนแปลงของชุดคำสั่งจากเวอร์ชัน 1.4 ไปเป็น เวอร์ชัน 1.3 หลังจากเสร็จสิ้นการประมวลผลในขั้นตอนแรก จะได้ชุดคำสั่งโปรแกรมเวอร์ชัน 1.3 หลังจากนั้นนำชุดคำสั่งโปรแกรมเวอร์ชัน 1.3 มาประมวลผลร่วมกับการเปลี่ยนแปลงของชุดคำสั่งจากเวอร์ชัน 1.3 ไปเป็น เวอร์ชัน 1.2 หลังจากเสร็จสิ้นการประมวลผลในขั้นตอนที่สองนี้ ก็จะได้ชุดคำสั่งโปรแกรมเวอร์ชัน 1.2 ตามที่ต้องการ



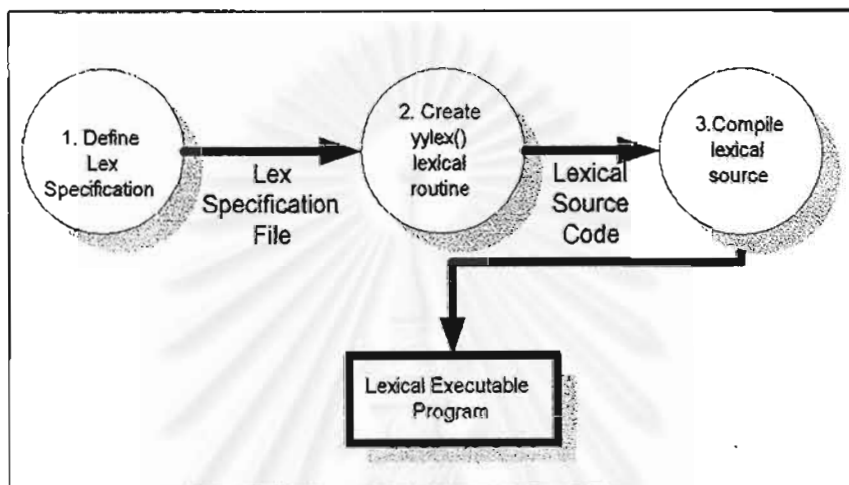
รูปที่ 2.1 วิธีการจัดเก็บข้อมูลแบบเคลดค่าและการนำข้อมูลกลับมา

## 2.2 การวิเคราะห์ศัพท์

เนื่องจากเครื่องมือควบคุม โครงการการพัฒนาโปรแกรมที่พัฒนาขึ้นในงานวิจัยนี้ ได้ทำการเพิ่มเติมความสามารถในการตรวจสอบรายงาน การแก้ไขชุดคำสั่งมอดูลที่มีผลกระทบต่อชุดคำสั่งมอดูลอื่น อันเนื่องมาจากความสัมพันธ์ในการเรียกใช้ฟังก์ชันระหว่างชุดคำสั่งมอดูล ดังนั้นเพื่อให้สามารถตรวจสอบความสัมพันธ์ดังกล่าวได้จึงต้อง ตรวจสอบในระดับของการกำหนดฟังก์ชันและการเรียกใช้งานฟังก์ชันภายในแต่ละชุดคำสั่งมอดูล โดยการทำเลกซิกัลชุดคำสั่งมอดูลในลักษณะที่คล้ายกับการทำงานของคอมไพเลอร์ สำหรับในงานวิจัยนี้ได้เลือกใช้ lex ซึ่งเป็นโปรแกรมที่ใช้สร้างโปรแกรมวิเคราะห์ศัพท์ที่นิยมใช้งานกัน โดยโปรแกรมที่สร้างขึ้นโดย lex จะเป็นโปรแกรมภาษาซี

ในการใช้งาน lex มีขั้นตอน 3 ขั้นตอน (ดูรูปที่ 2.2) ดังนี้:

- เขียนแฟ้มที่กำหนดรายละเอียดที่เรียกว่า "Lex specification" ซึ่งทำหน้าที่กำหนดโทเค็นและสิ่งที่ต้องทำเมื่อพบโทเค็นดังกล่าวในชุดคำสั่งมอดูลที่นำมาเลขชิด
- เรียกโปรแกรม lex โดยส่งแฟ้มกำหนดรายละเอียดในชื่อแรก เพื่อสร้างแฟ้มที่ชื่อว่า "lex.yy.c" ซึ่งเป็น source code ของโปรแกรมเลขชิดที่เขียนด้วยภาษาซี
- ทำการแปลโปรแกรม lex.yy.c ด้วยโปรแกรมแปลภาษาซี เพื่อให้ได้โปรแกรมที่ทำหน้าที่เลขชิด



รูปที่ 2.2 ภาพแสดงขั้นตอนในการใช้ lex ในการสร้างโปรแกรม lexical analyzer

### 2.2.1 วิธีการเขียน Lex Specification

ส่วนประกอบหลักของ Lex Specification เป็นกฎ (rule) หรือกลุ่มของกฎ (rules) ซึ่งปกติจะเก็บเป็นแฟ้มสกุล .l โดยมีรูปแบบอย่างน้อยที่สุด ดังนี้:

```
%%
```

```
rules
```

โดย %% เป็นสัญลักษณ์ที่ใช้ในการบอกจุดเริ่มต้นของกฎ และในแต่ละกฎ ประกอบด้วยส่วนที่เรียกว่า "นิพจน์ปรกติ" (Regular Expression) ซึ่งมีลักษณะเป็นคำหลัก (keyword) ซึ่งตรงกับสิ่งที่อยู่ในแฟ้มที่นำมาเลขชิด และส่วนที่เรียกว่า "Action" ซึ่งเป็นส่วนที่กำหนดสิ่งที่ต้องการให้ทำเมื่อพบ ส่วนของนิพจน์ปรกติ ปรากฏอยู่ในแฟ้มที่นำมาทำเลขชิด โดยอยู่ในรูปของชุดคำสั่งภาษาซี และถ้าในส่วนที่เป็น Action ประกอบด้วยชุดคำสั่งภาษาซี มากกว่า 1 คำสั่ง ชุดคำสั่งทั้งหมดจะต้องอยู่ภายใน "{}" ด้วย ตัวอย่างของ Lex Specification เช่น

%%

```
zippy      printf("I recognize zippy");
```

จากตัวอย่าง “zippy” เป็นนิพจน์ปรกติ โดยที่มีส่วน Action เป็นคำสั่งภาษาซี ที่ให้แสดงข้อความว่า “I recognize zippy” หากนำเพิ่มที่มีคำว่า “zippy” มาทำการเลกซ์คัลโดยโปรแกรมที่ถูกสร้างมาจาก Lex Specification ตามตัวอย่าง ก็จะได้ข้อความ “I recognize zippy” แสดงออกมาเท่ากับจำนวนคำว่า “zippy” ที่พบ ส่วนคำหรือข้อความอื่นในเพิ่มจะไม่มีการทำงานใดๆ เนื่องจากไม่ตรงกับกฎที่ตั้งไว้ใน Lex Specification

## 2.2.2 วากยสัมพันธ์ในส่วนของนิพจน์ปรกติ

นิพจน์ปรกติเป็นส่วนที่ทำหน้าที่นิยามถึง รูปแบบของข้อความที่เป็นโทเค็นที่ต้องการ โดยนิพจน์ปรกติอาจเป็นเพียงเซตว่าง (empty set), ตัวอักษร, กลุ่มหรือการเชื่อมต่อของ 2 นิพจน์ปรกติ หรือ เป็นอีกนิพจน์ปรกติก็ได้ เราสามารถแบ่งประเภทของนิพจน์ปรกติออกได้ดังนี้

- นิพจน์เดี่ยว เป็นนิพจน์ปรกติที่มีรูปแบบอย่างง่าย ซึ่งส่วนใหญ่จะเป็น โทเค็นของข้อความ อักขระเดี่ยว หรือคำเดี่ยว เช่น

[ \t]            หมายถึง โทเค็นของอักขระว่างหรือ อักขระตั้งระยะ 1 อักขระ

[ \t]+           หมายถึง โทเค็นของอักขระว่าง หรือ อักขระตั้งระยะอย่างน้อย 1 อักขระ

[int]            หมายถึง โทเค็นของข้อความ คำว่า “int”

- นิพจน์ประกอบ เป็นนิพจน์ปรกติที่ประกอบด้วยนิพจน์เดี่ยวและตัวดำเนินการ ซึ่งทำให้สามารถกำหนดรูปแบบของโทเค็นที่ซับซ้อนได้ เช่น

[0-9]+|[0-9]+\.[0-9]+|\.[0-9]+    เป็น โทเค็นของ เลขจำนวน (number)

ที่ประกอบด้วยนิพจน์เดี่ยว 3 นิพจน์ โดยนิพจน์แรกประกอบด้วยอักขระตัวเลข 0 ถึง 9 อย่างน้อย 1 อักขระ นิพจน์ที่สองประกอบด้วยอักขระตัวเลข 0 ถึง 9 อย่างน้อย 1 อักขระ ตามด้วยอักขระ “.” และอักขระตัวเลข 0 ถึง 9 อย่างน้อยอีก 1 อักขระ และนิพจน์สุดท้ายประกอบด้วยอักขระ “.” และตามด้วยอักขระตัวเลข 0 ถึง 9 อย่างน้อย

อีก 1 อักขระ โดยนิพจน์เดียวทั้ง 3 นิพจน์ถูกเชื่อมต่อกันด้วยตัวดำเนินการ “หรือ” (“|”) ซึ่งหมายความว่า หากพบข้อความใดในแฟ้มที่นำมาตรวจโทเค็น ที่สอดคล้องกับนิพจน์เดียวนิพจน์ใดนิพจน์หนึ่งข้างต้น ก็จะถือว่าตรวจพบโทเค็นของเลขจำนวน

ในนิพจน์ปรกตินอกจากอักขระหรือกลุ่มอักขระที่ต้องการตัดคำแล้ว ยังมีส่วนของตัวดำเนินการต่างๆประกอบอยู่ในนิพจน์ด้วย โดยตัวดำเนินการต่างๆที่มีในวากยสัมพันธ์ของนิพจน์ปรกติได้แสดงไว้ในตาราง 2.1

ตารางที่ 2.1 ตัวดำเนินการของนิพจน์ปรกติ

อักขระ	ความหมาย
.	แทนอักขระใดๆ 1 อักขระ ยกเว้นอักขระขึ้นบรรทัดใหม่ (new line)
\$	แทนอักขระขึ้นบรรทัดใหม่
^	แทนตำแหน่งเริ่มต้นบรรทัด
[ ]	แทนกลุ่มอักขระที่ระบุไว้ภายในตัวดำเนินการ เช่น [0-9] คู่กันได้กับ อักขระใดๆ ตั้งแต่ อักขระ “0” ถึง อักขระ “9”
-	ใช้ภายในตัวดำเนินการ [ ] เพื่อระบุเป็นช่วงอักขระ
?	ใช้กำหนดนิพจน์ปรกติก่อนหน้าว่าเป็นตัวเลือก เช่น 10?9 คู่กันได้กับ 19 หรือ 109 ก็ได้
*	ใช้ร่วมกับนิพจน์ปรกติก่อนหน้าที่เป็นอักขระตัวเลขว่ามีอย่างน้อยหนึ่ง หรือไม่มีก็ได้
+	ใช้ร่วมกับนิพจน์ปรกติก่อนหน้าที่เป็นอักขระตัวเลขว่ามีอย่างน้อยหนึ่ง
	เป็นตัวเชื่อมนิพจน์ปรกติ 2 นิพจน์ เช่น 10 11 คู่กันได้กับ 10 หรือ 11
()	ใช้ในการจัดกลุ่มของนิพจน์ปรกติ
/	ใช้กำหนดการคู่กันได้ของนิพจน์ปรกติก่อนหน้าให้เป็นไปตามการคู่กันได้ของนิพจน์ปรกติที่ตามมา เช่น 10/11 คู่กันได้กับ 10 11 แต่ไม่คู่กับ 10 12 เป็นต้น
{ }	ใช้ระบุว่าเป็นนิพจน์ปรกติที่ได้กำหนดไปแล้ว
<>	กำหนดจุดเริ่มต้นของเงื่อนไข

## 2.3 โครงสร้างและวากยสัมพันธ์ของภาษา อินฟอร์มิคซ์ โฟท์จีแอล

ภาษาอินฟอร์มิคซ์ โฟท์จีแอล (4GL ย่อมาจาก Fourth-Generation Application Development Language) เป็นภาษาที่พัฒนาขึ้นโดย บริษัท อินฟอร์มิคซ์ ซอฟต์แวร์ (Informix Software Inc.) เพื่อใช้สำหรับพัฒนาโปรแกรมประยุกต์สำหรับทำงานร่วมกับตัวจัดการฐานข้อมูลอินฟอร์มิคซ์ (Informix Database Engines) โครงสร้างภาษาคลายคลึงกับภาษาซี หรือภาษาปาสคาล สำหรับในงานวิจัยนี้จะทำการศึกษาในส่วนต่างๆ ดังนี้

### 2.3.1 โครงสร้างโปรแกรมในภาษาอินฟอร์มิคซ์ โฟท์จีแอล

ในโปรแกรมภาษาอินฟอร์มิคซ์ โฟท์จีแอล สามารถแบ่งส่วนของโปรแกรมออกเป็นส่วนใหญ่ได้ 2 ส่วน ดังนี้

- ส่วนฟังก์ชันหลัก (main function)

เป็นฟังก์ชันที่เป็นจุดเริ่มต้นการทำงานของโปรแกรม และในโปรแกรมหนึ่งจะสามารถมีฟังก์ชันหลักได้เพียงฟังก์ชันเดียวเท่านั้น จุดเริ่มต้นของฟังก์ชันหลักนั้นเริ่มที่คำสั่ง “main” และสิ้นสุดที่ “end main” และในฟังก์ชันหลักนั้นจะ ไม่มีการรับพารามิเตอร์ โดยมีวากยสัมพันธ์ดังนี้

```
main
    {จุดคำสั่งโปรแกรม}
end main
ตัวอย่างเช่น
main
    display “Hello World !”
end main
```

- ส่วนฟังก์ชัน (function)

เป็นฟังก์ชันที่เป็นส่วนประกอบของโปรแกรมที่ถูกเรียกใช้จากฟังก์ชันหลักหรือฟังก์ชันอื่น ผู้ที่พัฒนาโปรแกรมต้องกำหนดชื่อของฟังก์ชันขึ้นเอง และภายในแต่ละโปรแกรมสามารถมีฟังก์ชันได้ไม่จำกัด แต่ห้ามมีฟังก์ชันที่มีชื่อซ้ำกัน จุดเริ่มต้นของฟังก์ชันทั่วไปเริ่มที่คำสั่ง “function” และสิ้นสุดที่ “end function” โดยมีวากยสัมพันธ์ดังนี้

```
function {ชื่อฟังก์ชัน} ([ชื่อพารามิเตอร์] [ชื่อพารามิเตอร์...])
[define {ชื่อพารามิเตอร์} {ชนิดตัวแปร}]
```

```

{ชุดคำสั่ง โปรแกรม}
[return [ชื่อตัวแปร] ]
end function
ตัวอย่างเช่น
function calcvat (amt)
define amtfloat
define vat float
vat = amt * 7 / 100
return vat
end function

```

นอกจากที่กล่าวมา โปรแกรมภาษาอินฟอร์มิกซ์ โฟทจีแอล ได้ นิยาม เพิ่มที่เป็นส่วนของโปรแกรม ว่า “ชุดคำสั่งมอดูล (module)” ดังนั้นจึงเป็นไปได้ว่า ในชุดคำสั่งมอดูลอาจมีแต่เพียงส่วนฟังก์ชันเท่านั้น และเรียก ชุดคำสั่งที่มีส่วนหลักของโปรแกรมประกอบอยู่ด้วยว่า “ชุดคำสั่งมอดูลหลัก (main module)”

### 2.3.2 การกำหนดหมายเหตุ (Comments)

ในภาษาอินฟอร์มิกซ์ โฟทจีแอล มีรูปแบบในการกำหนดข้อความที่เป็น หมายเหตุเพื่อใช้ในการอธิบายโปรแกรมนั้นอยู่ 3 รูปแบบ ดังนี้

- เริ่มต้นหมายเหตุด้วยอักขระ “{” และสิ้นสุดด้วยอักขระ “}” แต่ไม่สามารถ กำหนดหมายเหตุวิธีนี้ซ้อนกันได้
- เริ่มต้นหมายเหตุด้วยอักขระ “#” จนถึงสิ้นสุดหมายเหตุที่ทำบรรทัดนั้นๆ
- เริ่มต้นหมายเหตุด้วยอักขระ “--” จนถึงสิ้นสุดหมายเหตุที่ทำบรรทัดนั้นๆ

จากรูปที่ 2.3 ในบรรทัดที่ 1 ถึง 4 เป็นตัวอย่างแสดงการกำหนดหมายเหตุ ด้วยอักขระ “#” โดยหมายเหตุที่กำหนดจะเริ่มจากอักขระ “#” ไปจนถึงสิ้นสุดบรรทัด การกำหนดหมายเหตุประเภทนี้มักใช้สำหรับแบ่งส่วนของโปรแกรมเพื่อให้ สะดวกในการตรวจสอบโปรแกรม ในบรรทัดที่ 9 และ 16 เป็นตัวอย่างแสดงการ กำหนดหมายเหตุด้วยอักขระ “--” โดยหมายเหตุที่กำหนดจะเริ่มจากอักขระ “--” ไปจนถึงสิ้นสุดบรรทัด การกำหนดหมายเหตุประเภทนี้มักใช้เพื่ออธิบายการทำงาน ของคำสั่งเฉพาะบรรทัด และในบรรทัด 11 ถึง 14 เป็นตัวอย่างแสดงการกำหนด



หมายเหตุเป็นช่วง โดยหมายเหตุที่กำหนดจะเริ่มต้นด้วยอักขระ “{” และสิ้นสุดด้วยอักขระ “}” การกำหนดหมายเหตุประเภทนี้มักใช้ในการหมายเหตุเป็นช่วงกว้าง

```
1 #-----
2 # Program Name : btp001.4gl
3 # Programmer ID: smt
4 #-----
5 globals
6 define msg1 char(100)
7 end globals
8 main
9 --define msg1 char(80)
10 define msg2 char(200)
11 {
12 let msg1 = "Hello World !"
13 display msg1
14 }
15 let msg2 = "New Hello World !"
16 display msg2 -- Display New Hello World ! message
17 end main
```

รูปที่ 2.3 ตัวอย่างการกำหนดหมายเหตุในภาษา อินฟอร์มิกซ์ โฟท์จีแอล

### 2.3.3 การกำหนดตัวแปรและชนิดข้อมูล (Variable Definition and Data Type)

ในภาษาอินฟอร์มิกซ์ โฟท์จีแอล การกำหนดตัวแปรใช้ คำสั่ง “define” โดยมีวากยสัมพันธ์ดังนี้

define (ชื่อตัวแปร) (ชนิดข้อมูล)

ตัวแปรที่กำหนดขึ้นนั้น สามารถแบ่งประเภทตามขอบเขตที่สามารถอ้างถึงตัวแปรได้ ออกเป็น 3 ประเภท ดังนี้

- ตัวแปรเฉพาะที่ (local variables) เป็นตัวแปรที่ได้รับการกำหนดขึ้นภายในบล็อกของ ส่วนหลัก หรือส่วนฟังก์ชัน การอ้างถึงตัวแปรดังกล่าวก็สามารถอ้างถึงได้เฉพาะภายในบล็อกของส่วนหลัก หรือส่วนฟังก์ชันที่ตัวแปรได้ถูกกำหนดขึ้นเท่านั้น
- ตัวแปรมอดูล (module variables) เป็นตัวแปรที่ได้รับการกำหนดขึ้นภายนอกบล็อกก่อนบล็อกแรกของชุดคำสั่งมอดูล ไม่ว่าจะบล็อกแรกของชุดคำสั่งมอดูลจะเป็น ส่วนหลัก หรือส่วนฟังก์ชัน การอ้างถึงตัวแปรดังกล่าวก็สามารถอ้างถึงได้เฉพาะภายในชุดคำสั่งมอดูลที่ตัวแปรได้ถูกกำหนดขึ้นเท่านั้น
- ตัวแปรส่วนกลาง (global variables) เป็นตัวแปรที่ได้รับการกำหนดขึ้นระหว่างคำสั่ง “globals” และ “end gobals” ภายในชุดคำสั่งมอดูลหลัก โดยต้องกำหนดก่อนบล็อกแรกของชุดคำสั่งมอดูลหลัก การอ้างถึงตัวแปรดังกล่าวก็สามารถอ้างถึงได้ตลอดทุกคำสั่งในโปรแกรม

ในการตั้งชื่อตัวแปรของภาษาอินฟอร์มิคซ์ ไฟท์จีแอลมีข้อกำหนดว่า ชื่อตัวแปรสามารถประกอบด้วยตัวอักษร, ตัวเลข และอักขระขีดกลาง (underscores) โดยอักขระตัวแรกของชื่อตัวแปรต้องเป็นตัวอักษร หรือ อักขระขีดกลางเท่านั้น ไม่สนใจขนาดตัวอักษร(case insensitive) ชื่อของตัวแปรสามารถตั้งได้ยาวไม่จำกัดจำนวนอักขระ แต่ตัวภาษาอินฟอร์มิคซ์ ไฟท์จีแอล ทำการพิจารณาเพียง 8 อักขระแรกของชื่อตัวแปรเท่านั้น และการตั้งชื่อตัวแปรต้องไม่ซ้ำกับคำสงวนในภาษาอินฟอร์มิคซ์ ไฟท์จีแอล

สำหรับการศึกษานิคข้อมูลของภาษาอินฟอร์มิคซ์ ไฟท์จีแอลในงานวิจัยนี้ เป็นการศึกษานิคข้อมูลเฉพาะชนิดข้อมูลที่กำหนดไว้ในตัวภาษาเท่านั้น ไม่ได้รวมถึงชนิดข้อมูลที่สามรถกำหนดขึ้นได้เองโดยผู้เขียน โปรแกรม และชนิดข้อมูลที่กำหนดไว้ให้ขึ้นกับชนิดข้อมูลในฐานข้อมูล โดยชนิดข้อมูลที่ได้ทำการศึกษาได้รวบรวมและแสดงไว้ดังที่แสดงในตาราง 2.2

ตารางที่ 2.2 ชนิดข้อมูลในภาษาอินฟอร์มิคซ์ โฟท์จีแอล

ชนิดข้อมูล	รายละเอียด
SMALLINT	เป็นข้อมูลเลขจำนวนเต็มที่มีค่าอยู่ในช่วง -32,767 ถึง +32,767
INT INTEGER	เป็นข้อมูลเลขจำนวนเต็มที่มีค่าอยู่ในช่วง -2,147,483,647 ถึง +2,147,483,647
DECIMAL [(m [,n])] DEC [(m [,n])] NUMERIC [(m [,n])]	เป็นข้อมูลเลขทศนิยม สามารถกำหนดความละเอียดในการเก็บได้ โดย m เป็นจำนวนหลักทั้งหมด ( $m \leq 32$ ) และ n เป็นจำนวนหลักหลังจุดทศนิยม ( $n \leq m$ ) โดยสามารถรับค่าละเอียดสุด $0.5 \times 10^m$ คงความถูกต้องที่ $10^{m-n} - 10^{-n}$ และถ้าไม่มีการกำหนดค่า n ไว้จะสามารถรองรับค่าในช่วง $10^{-128}$ ถึง $10^{126}$
SMALLFLOAT REAL	เป็นข้อมูลเลขทศนิยมที่จัดเก็บในระบบเลขฐานสอง เช่นเดียวกับชนิดข้อมูล float ในภาษาซี
FLOAT DOUBLE PRECISION	เป็นข้อมูลเลขทศนิยมที่จัดเก็บในระบบเลขฐานสอง เช่นเดียวกับชนิดข้อมูล double ในภาษาซี
MONEY [(m [,n])]	เป็นข้อมูลเลขทศนิยมที่เหมือนกับ ข้อมูลชนิด DECIMAL เพียงแต่เวลาแสดงผลข้อมูลชนิดนี้ จะมีสัญลักษณ์สกุลเงินปรากฏอยู่ด้วย
CHAR [(n)] CHARACTER [(n)]	เป็นข้อมูลสายอักขระ โดยสามารถกำหนดความยาวได้ โดยที่ $1 \leq n \leq 32,767$
DATE	เป็นข้อมูลวันที่

#### 2.3.4 การเรียกใช้ฟังก์ชัน (function call)

ฟังก์ชันภาษาอินฟอร์มิคซ์ โฟท์จีแอล สามารถส่งผ่านพารามิเตอร์ให้กับฟังก์ชัน โดยการผ่านค่าเป็นการผ่านค่าแบบทางเดียวเท่านั้น การเรียกใช้ฟังก์ชันมีวากยสัมพันธ์ดังนี้

```
call {ชื่อฟังก์ชัน ([ชื่อพารามิเตอร์] [,ชื่อพารามิเตอร์...])}
      [returning {ชื่อตัวแปร} [,ชื่อตัวแปร...]]
```

ตัวอย่างเช่น

```
call kd_bankdate(pdate,3)
      returning duedate
```

## บทที่ 3

### การวิเคราะห์และออกแบบระบบ

#### 3.1 การวิเคราะห์ระบบ

ระบบจัดการเวอร์ชันซอร์สโค้ด มีหน้าที่ในการจัดเก็บซอร์สโค้ดโปรแกรม และตรวจสอบแจ้งเตือนความสัมพันธ์ที่ถูกรบกวนเนื่องจากการแก้ไขเปลี่ยนแปลงซอร์สโค้ดโปรแกรม ดังนั้นปัญหาที่ต้องวิเคราะห์ เพื่อนำมาเป็นแนวทางในการออกแบบระบบจัดการเวอร์ชันซอร์สโค้ดได้แก่

- 3.1.1 วิธีการในการจัดเก็บซอร์สโค้ดโปรแกรม ได้เลือกใช้วิธีการจัดเก็บแบบเคลด้า เนื่องจากวิธีการจัดเก็บดังกล่าวช่วยให้ประหยัดทรัพยากรที่ใช้ในการจัดเก็บข้อมูล ซึ่งวิธีการจัดเก็บแบบเคลด้า ต้องทำการหาส่วนที่แตกต่างระหว่างซอร์สโค้ดโปรแกรมเวอร์ชันก่อนหน้าและเวอร์ชันที่ต้องการจัดเก็บ โดยในการหาส่วนที่แตกต่างดังกล่าว ต้องทำในระหว่างที่ผู้ใช้ระบบกำลังจัดเก็บซอร์สโค้ดโปรแกรมเข้าสู่ระบบ โดยระบบจะทำการจัดเก็บไว้เฉพาะในส่วนของซอร์สโค้ดที่ให้จัดเก็บไว้เป็นซอร์สโค้ดเวอร์ชันล่าสุด และจัดเก็บส่วนที่แตกต่างไว้เพื่อใช้ในการย้อนเวอร์ชัน
- 3.1.2 วิธีการหาส่วนที่แตกต่างระหว่างซอร์สโค้ดโปรแกรม ได้ใช้ขั้นตอนวิธีเดียวกับคำสั่ง diff ในระบบปฏิบัติการ UNIX ของหน่วยงาน GNU General Public License ซึ่งขั้นตอนวิธีดังกล่าวเป็นซอฟต์แวร์ของสาธารณะ
- 3.1.3 วิธีการจัดเก็บความสัมพันธ์ระหว่างซอร์สโค้ด ได้เลือกใช้ฐานข้อมูลเชิงสัมพันธ์ ในการจัดเก็บความสัมพันธ์ดังกล่าว โดยสิ่งที่จัดเก็บไว้เป็นข้อมูลได้แก่ ชื่อซอร์สโค้ด, ชื่อฟังก์ชันที่อยู่ภายในซอร์สโค้ด, พารามิเตอร์ของฟังก์ชัน, ชื่อฟังก์ชันที่มีการเรียกใช้ภายในซอร์สโค้ด และพารามิเตอร์ที่ส่งผ่านค่า
- 3.1.4 วิธีการได้มาซึ่งข้อมูลความสัมพันธ์ที่ต้องการจัดเก็บ สามารถทำได้โดยการตรวจสอบวากยสัมพันธ์ของภาษา อินฟอร์มิกซ์ โฟท์จีแอล โดยใช้ Lex ซึ่งเป็นโปรแกรม

สร้างโปรแกรมวิเคราะห์คำศัพท์ เพื่อใช้ในการหา ชื่อชุดคำสั่งมอดูล ชื่อฟังก์ชันที่อยู่ภายในชุดคำสั่งมอดูล พารามิเตอร์ของฟังก์ชัน ชื่อฟังก์ชันที่มีการเรียกใช้ภายในชุดคำสั่งมอดูล และพารามิเตอร์ที่ส่งผ่านค่า จากชุดคำสั่งที่ทำการจัดเก็บเข้าสู่ระบบ

3.1.5 หลักเกณฑ์ในการระบุว่าความสัมพันธ์ระหว่างชุดคำสั่งมอดูลเป็นความสัมพันธ์ที่มีปัญหา

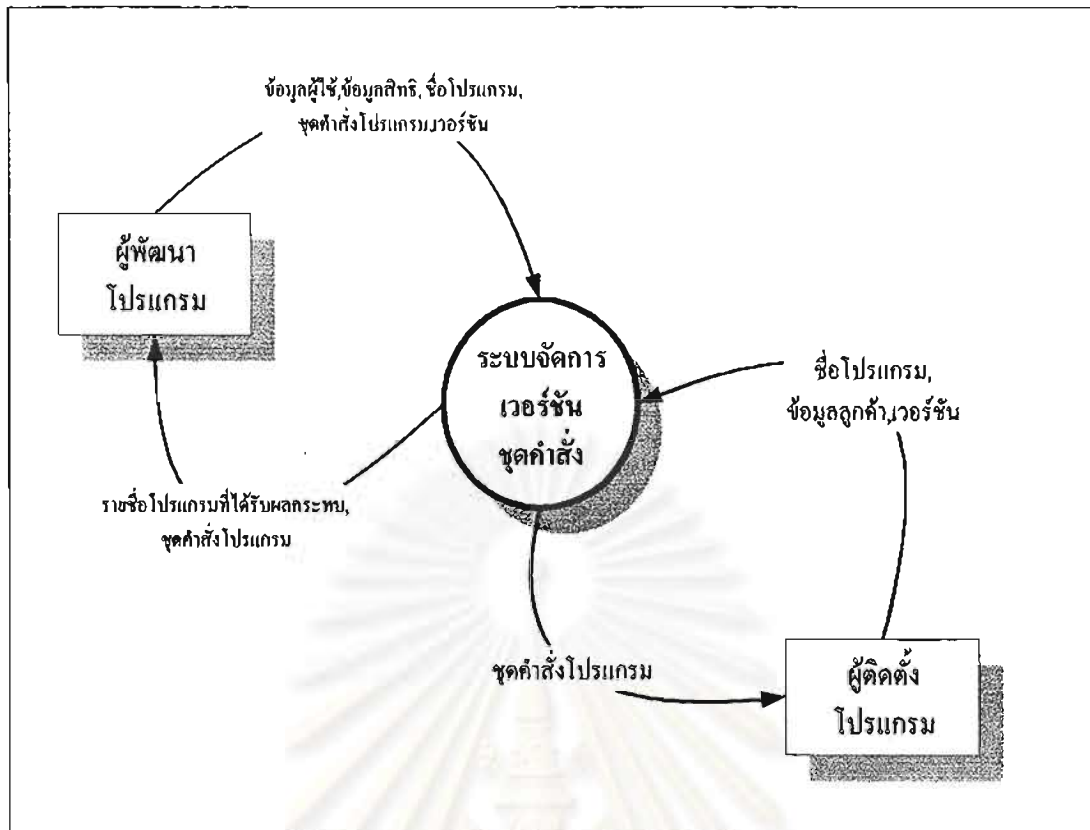
- ชุดคำสั่งมอดูลที่กำลังทำการจัดเก็บมีการเรียกใช้ฟังก์ชันที่ไม่ปรากฏว่ามีการกำหนดไว้ในชุดคำสั่งมอดูลใดเลยที่จัดเก็บอยู่ในระบบ
- ชุดคำสั่งมอดูลที่กำลังทำการจัดเก็บ มีการเรียกใช้ฟังก์ชันในชุดคำสั่งมอดูลอื่นซึ่งได้จัดเก็บอยู่ในระบบ แต่พบว่ามีการรับส่งค่าผ่านพารามิเตอร์ในจำนวนหรือชนิดข้อมูล ที่แตกต่างกัน
- ชุดคำสั่งมอดูลที่กำลังทำการจัดเก็บ มีการกำหนดฟังก์ชัน และพบว่ามีชุดคำสั่งมอดูลอื่นที่ได้จัดเก็บอยู่ในระบบ มีการเรียกใช้ฟังก์ชันดังกล่าว แต่พบว่ามีการรับส่งค่าผ่านพารามิเตอร์ในจำนวน หรือชนิดข้อมูล ที่แตกต่างกัน

### 3.2 การออกแบบระบบ

ระบบจัดการเวอร์ชันชุดคำสั่ง เป็นระบบที่มีผู้ใช้งาน 2 กลุ่ม (ดูรูปที่ 3.1) ได้แก่

- กลุ่มของผู้พัฒนาโปรแกรม โดยผู้ใช้ในกลุ่มนี้จะเป็นผู้ที่ทำการเขียนโปรแกรมหรือแก้ไขเปลี่ยนแปลงโปรแกรมแล้วนำมาจัดเก็บไว้ใน ระบบจัดการเวอร์ชันชุดคำสั่ง และเป็นผู้ที่ขอเบิกโปรแกรมออกจากระบบจัดการเวอร์ชันชุดคำสั่งเพื่อนำไปทำการแก้ไขเปลี่ยนแปลง กำหนดผู้ใช้ระบบและสิทธิในการเข้าถึง โปรแกรมที่จัดเก็บไว้ รวมทั้งเป็นผู้ที่รับรายงานผลกระทบ ต่ออันเนื่องมาจากการแก้ไขเปลี่ยนแปลงโปรแกรม

- กลุ่มของผู้ติดตั้งโปรแกรม โดยผู้ใช้กลุ่มนี้จะเป็นผู้ขอเบิกโปรแกรมออกจากระบบจัดการเวอร์ชันชุดคำสั่ง เพื่อนำไปติดตั้งให้กับลูกค้าที่ต้องการใช้งาน โปรแกรมที่พัฒนาขึ้น



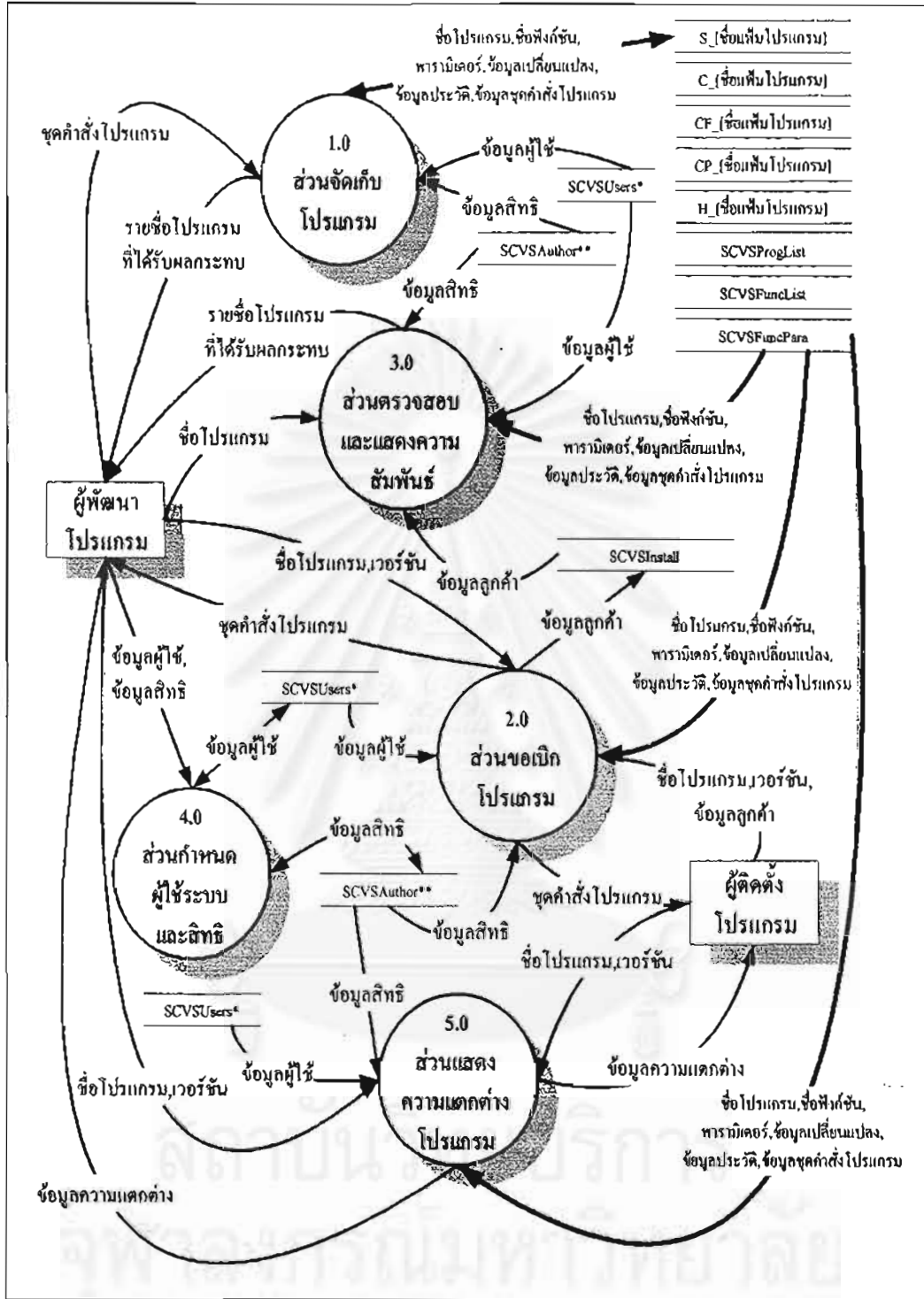
รูปที่ 3.1 แผนภาพบริบทของระบบจัดการเวอร์ชันชุดคำสั่ง

ระบบจัดการเวอร์ชันชุดคำสั่ง ประกอบด้วยส่วนที่ทำหน้าที่หลักของระบบ ดังนี้ (ดูรูปที่

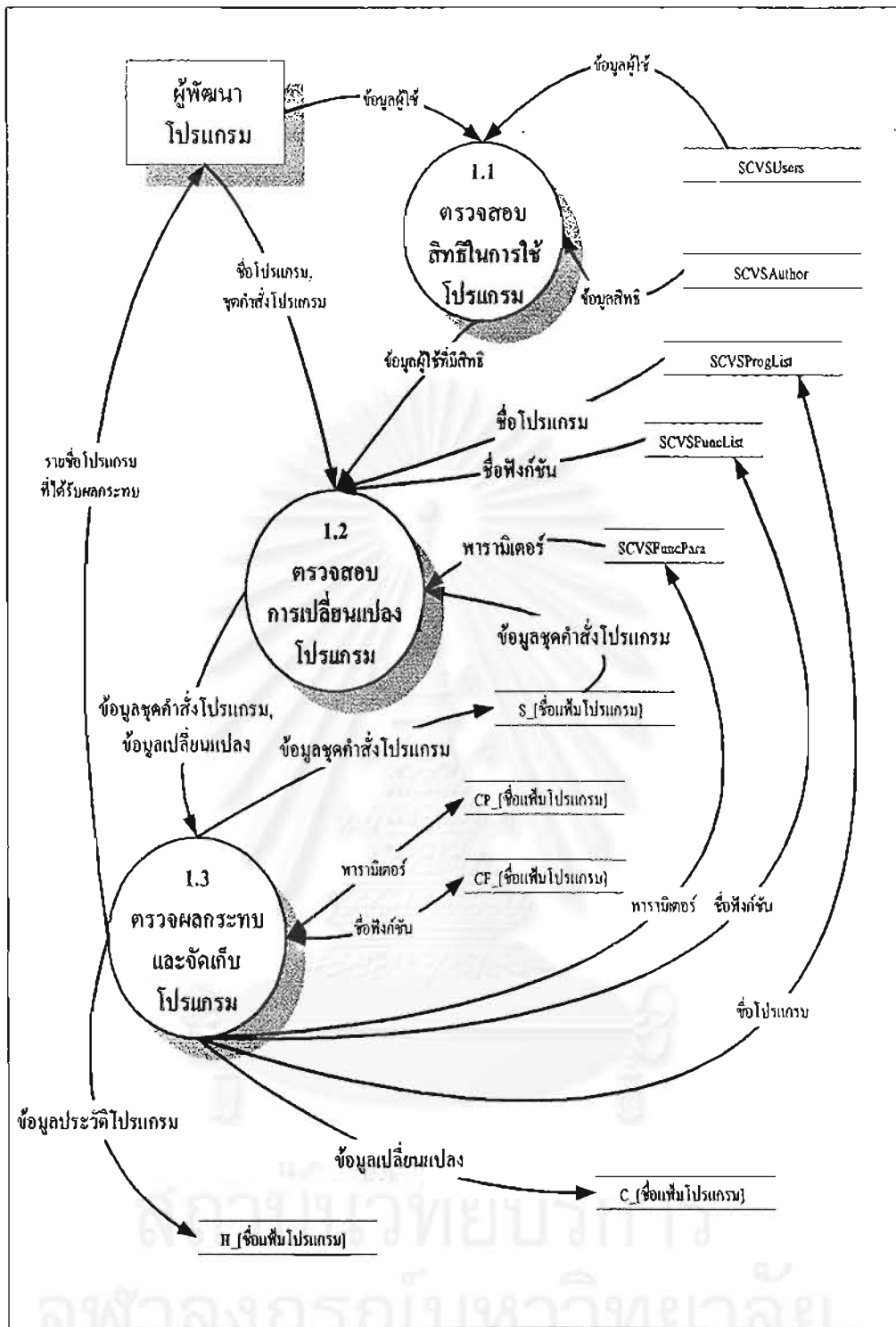
3.2)

3.2.1 ส่วนจัดเก็บโปรแกรม เป็นส่วนที่ทำหน้าที่นำชุดคำสั่ง โปรแกรมจากผู้พัฒนาโปรแกรม มาจัดเก็บในระบบจัดการเวอร์ชันชุดคำสั่ง โดยจะทำการตรวจสอบความเปลี่ยนแปลงของชุดคำสั่งโปรแกรม และจัดการการเปลี่ยนแปลงที่เกิดขึ้น (ดังรูปที่ 3.3) ประกอบด้วย 3 ส่วนดังนี้

- ส่วนตรวจสอบสิทธิในการใช้โปรแกรม โดยทำหน้าที่รับข้อมูลผู้ใช้เพื่อตรวจสอบเพิ่มข้อมูลผู้ใช้และเพิ่มข้อมูลสิทธิ
- ส่วนตรวจสอบการเปลี่ยนแปลงโปรแกรม โดยทำหน้าที่อ่านชุดคำสั่งโปรแกรมที่ต้องการจัดเก็บ มาตรวจสอบเปรียบเทียบกับเวอร์ชันล่าสุดที่เก็บไว้ในระบบ
- ส่วนตรวจสอบผลกระทบและจัดเก็บโปรแกรม โดยทำหน้าที่ตรวจสอบความสัมพันธ์ระหว่างฟังก์ชันของโปรแกรมที่กำลังจัดเก็บกับโปรแกรมอื่นที่เก็บอยู่ในระบบ และจัดเก็บ โปรแกรมเข้าสู่ระบบเป็นเวอร์ชันล่าสุด



รูปที่ 3.2 แผนภาพกระแสข้อมูลระดับ 0 ของระบบจัดการเวอร์ชันชุดคำสั่ง

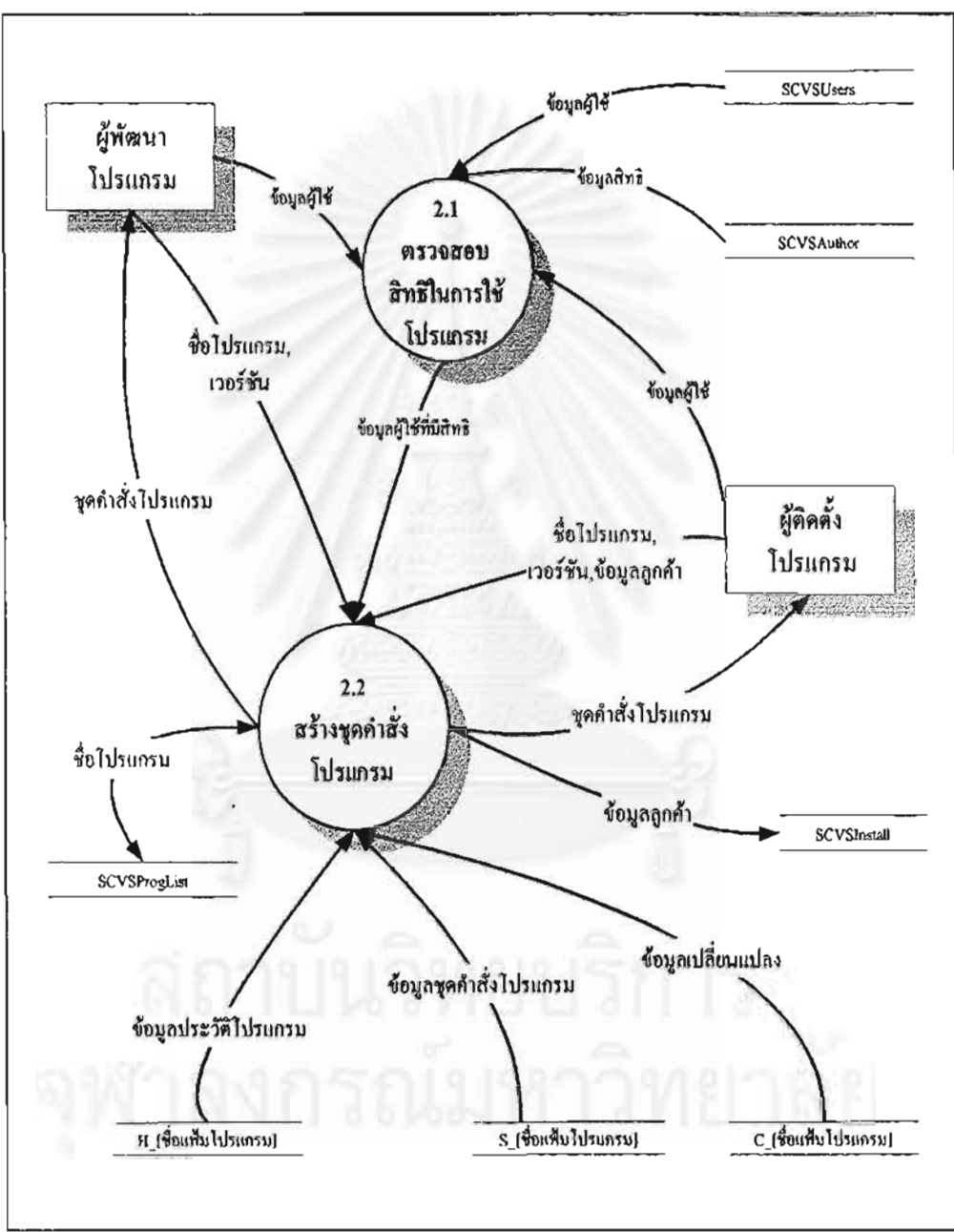


รูปที่ 3.3 แผนภาพกระแสข้อมูลระดับ 1 ของส่วนจัดเก็บโปรแกรม

3.2.2 ส่วนขอเบิกโปรแกรม เป็นส่วนที่ทำหน้าที่นำชุดคำสั่งโปรแกรมที่จัดเก็บอยู่ในระบบจัดการเวอร์ชันชุดคำสั่งออกมาเพื่อให้ผู้พัฒนาโปรแกรมได้นำไปแก้ไข หรือเพื่อให้ผู้ติดตั้งโปรแกรมได้นำไปติดตั้งให้กับลูกค้า และจะห้ามบันทึกการขอเบิกไว้ในระบบ (ดังรูปที่ 3.4) ประกอบด้วย 2 ส่วน ดังนี้



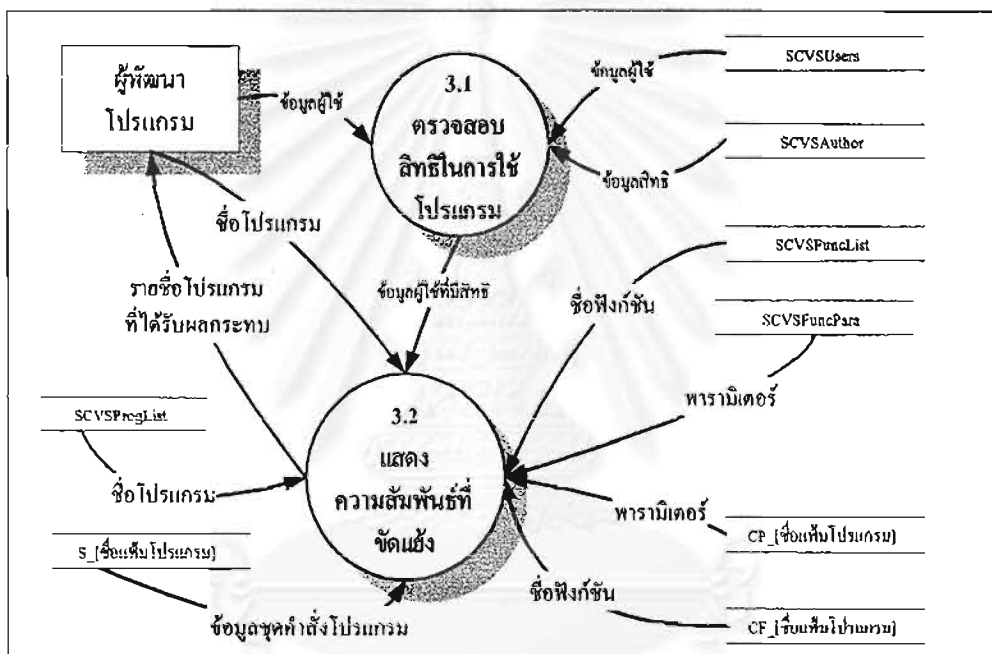
- ส่วนตรวจสอบสิทธิในการใช้โปรแกรม โดยทำหน้าที่รับข้อมูลผู้ใช้ เพื่อตรวจสอบแฟ้มข้อมูลผู้ใช้และแฟ้มข้อมูลสิทธิ
- ส่วนสร้างชุดคำสั่งโปรแกรม โดยทำหน้าที่รับชื่อโปรแกรมและเวอร์ชัน ที่ผู้ใช้ขอเบิกออกจากระบบ ทำการค้นหาและสร้างชุดคำสั่งโปรแกรมในเวอร์ชันที่ผู้ใช้ต้องการ



รูปที่ 3.4 แผนภาพกระแสข้อมูลระดับ 1 ของส่วนขอเบิกโปรแกรม

3.2.2 ส่วนตรวจสอบและแสดงความสัมพันธ์ที่ขัดแย้ง เป็นส่วนที่ทำหน้าที่นำแสดงรายการความสัมพันธ์ที่ขัดแย้งกันระหว่างโปรแกรมชุดคำสั่ง โปรแกรมที่จัดเก็บอยู่ในระบบจัดการเวอร์ชันชุดคำสั่งออกมาเพื่อให้ผู้พัฒนาโปรแกรม ได้ทำการแก้ไขในภายหลัง (ดังรูปที่ 3.5) ประกอบด้วย 2 ส่วน ดังนี้

- ส่วนตรวจสอบสิทธิในการใช้โปรแกรม โดยทำหน้าที่รับข้อมูลผู้ใช้เพื่อตรวจสอบเพิ่มข้อมูลผู้ใช้และเพิ่มข้อมูลสิทธิ
- ส่วนแสดงความสัมพันธ์ที่ขัดแย้ง โดยทำหน้าที่ตรวจสอบความสัมพันธ์ที่ขัดแย้งกันระหว่างโปรแกรมที่เก็บไว้ในระบบ

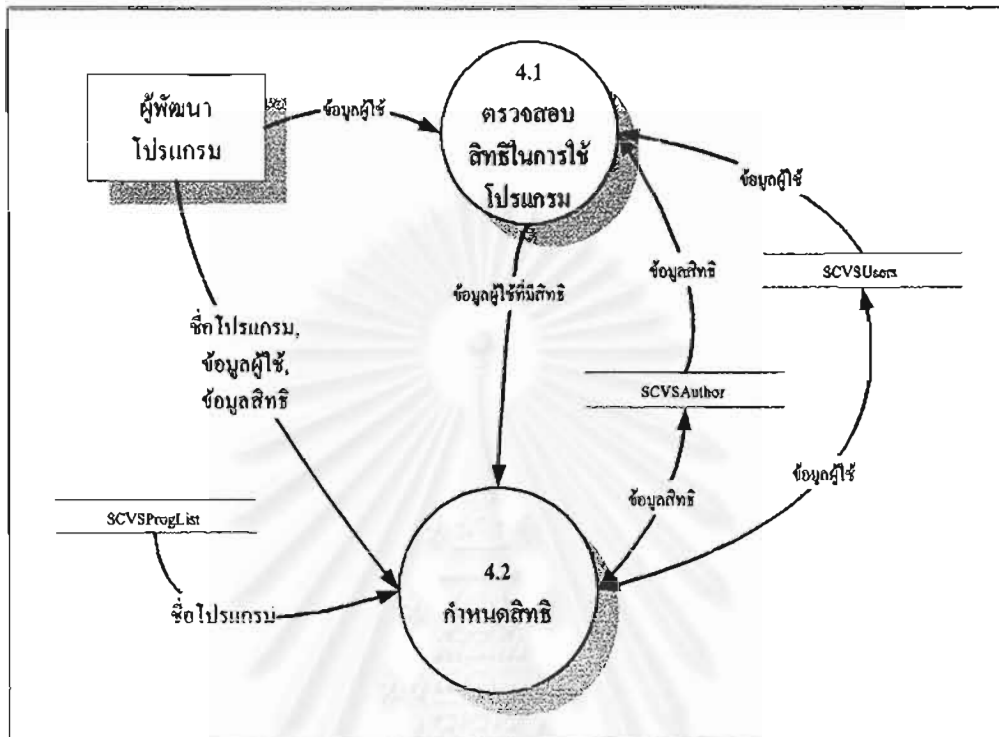


รูปที่ 3.5 แผนภาพกระแสข้อมูลระดับ 1 ของส่วนตรวจสอบและแสดงความสัมพันธ์ที่ขัดแย้ง

3.2.4 ส่วนกำหนดผู้ใช้ระบบและสิทธิในการใช้งาน เป็นส่วนที่ทำหน้าที่ในการกำหนดข้อมูลของผู้ใช้ในระบบจัดการเวอร์ชันชุดคำสั่ง และกำหนดสิทธิในการเข้าถึงชุดคำสั่ง โปรแกรมที่จัดเก็บไว้ในระบบ (ดังรูปที่ 3.6) ประกอบด้วย 2 ส่วน ดังนี้

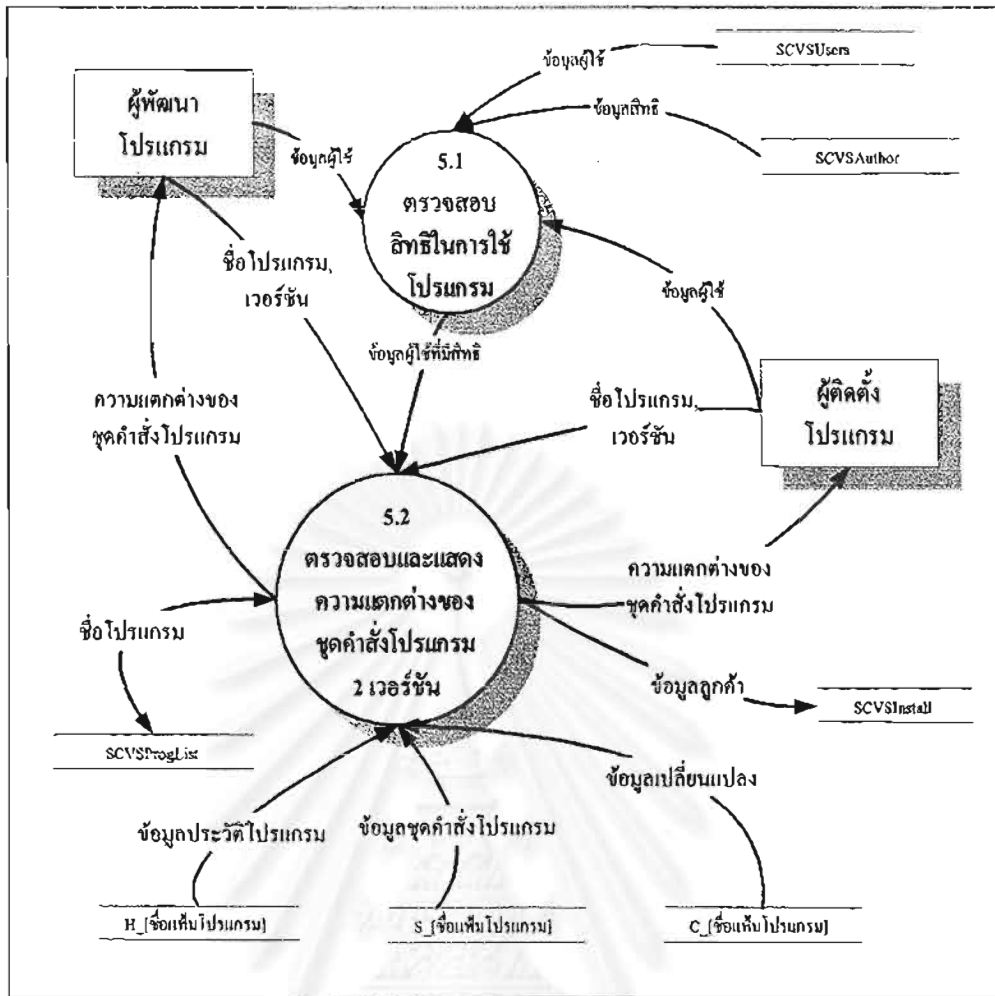
- ส่วนตรวจสอบสิทธิในการใช้โปรแกรม โดยทำหน้าที่รับข้อมูลผู้ใช้เพื่อตรวจสอบเพิ่มข้อมูลผู้ใช้และเพิ่มข้อมูลสิทธิ

- ส่วนกำหนดสิทธิ์ ทำหน้าที่ให้ผู้ใช้ใช้ในการเพิ่ม, แก้ไขข้อมูลในการ  
ผู้ใช้ กำหนดสิทธิ์ของผู้ใช้ และกำหนดสิทธิ์ในการเข้าถึง โปรแกรมที่  
จัดเก็บไว้ในระบบ



รูปที่ 3.6 แผนภาพกระแสข้อมูลระดับ 1 ของส่วนกำหนดผู้ใช้ระบบและสิทธิในการใช้งาน

- 3.2.5 ส่วนตรวจสอบและแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งโปรแกรม เป็นส่วนที่ทำหน้าที่ตรวจสอบและแสดงให้เห็นความแตกต่างของชุดคำสั่งโปรแกรม 2 เวอร์ชัน ว่ามีส่วนใดของชุดคำสั่งโปรแกรมบ้างที่ถูกแก้ไขเปลี่ยนแปลงไป (ดังรูปที่ 3.7) ประกอบด้วย 2 ส่วน ดังนี้
- ส่วนตรวจสอบสิทธิในการใช้โปรแกรม โดยทำหน้าที่รับข้อมูลผู้ใช้เพื่อตรวจสอบเพิ่มข้อมูลผู้ใช้และเพิ่มข้อมูลสิทธิ์
  - ส่วนตรวจสอบและแสดงความแตกต่างของ ชุดคำสั่งโปรแกรม 2 เวอร์ชัน เพื่อแสดงให้เห็นรายละเอียดความเปลี่ยนแปลงที่เกิดขึ้นกับชุดคำสั่งโปรแกรมที่ได้รับการเปลี่ยนแปลงเวอร์ชัน



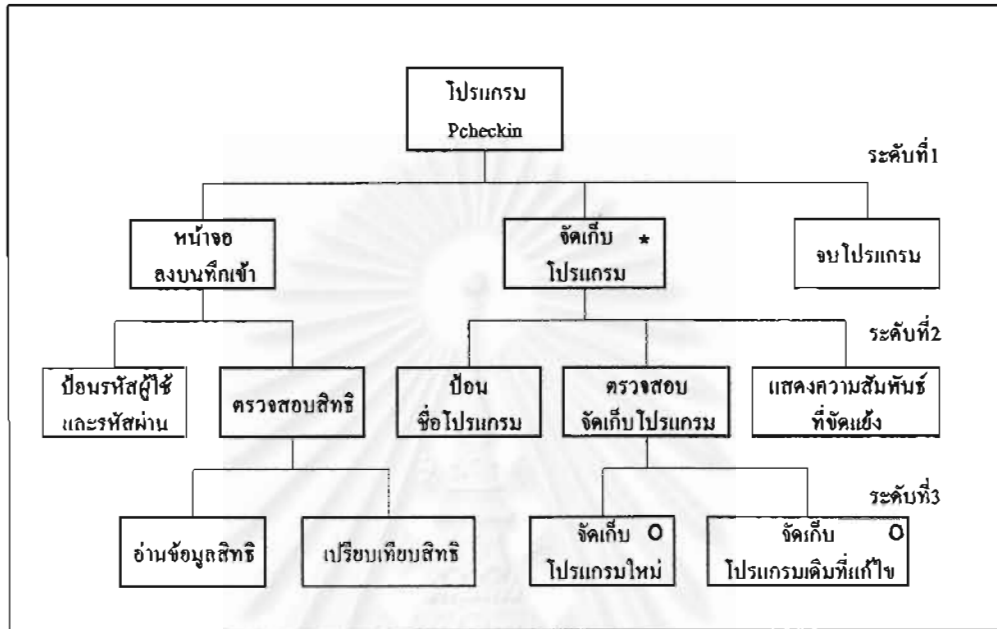
รูปที่ 3.7 แผนภาพกระแสข้อมูลระดับ 1 ของส่วนตรวจสอบและแสดง ความแตกต่างของชุดคำสั่งโปรแกรม

ระบบจัดการเวอร์ชันชุดคำสั่งในงานวิจัยนี้ ได้ออกแบบระบบให้แยกออกเป็นโปรแกรมหลายๆ โปรแกรม ตามการเรียกใช้งานของผู้ใช้ระบบ เพื่อให้โปรแกรมมีขนาดไม่ใหญ่มากนัก สามารถทำงานได้รวดเร็วขึ้น โดยแบ่งออกเป็นโปรแกรมได้ดังนี้

1. โปรแกรม Pcheckin เป็นโปรแกรมที่ทำหน้าที่ในส่วนจัดเก็บโปรแกรม โดยมีหน้าที่หลัก คือ (ดูรูปที่ 3.8)

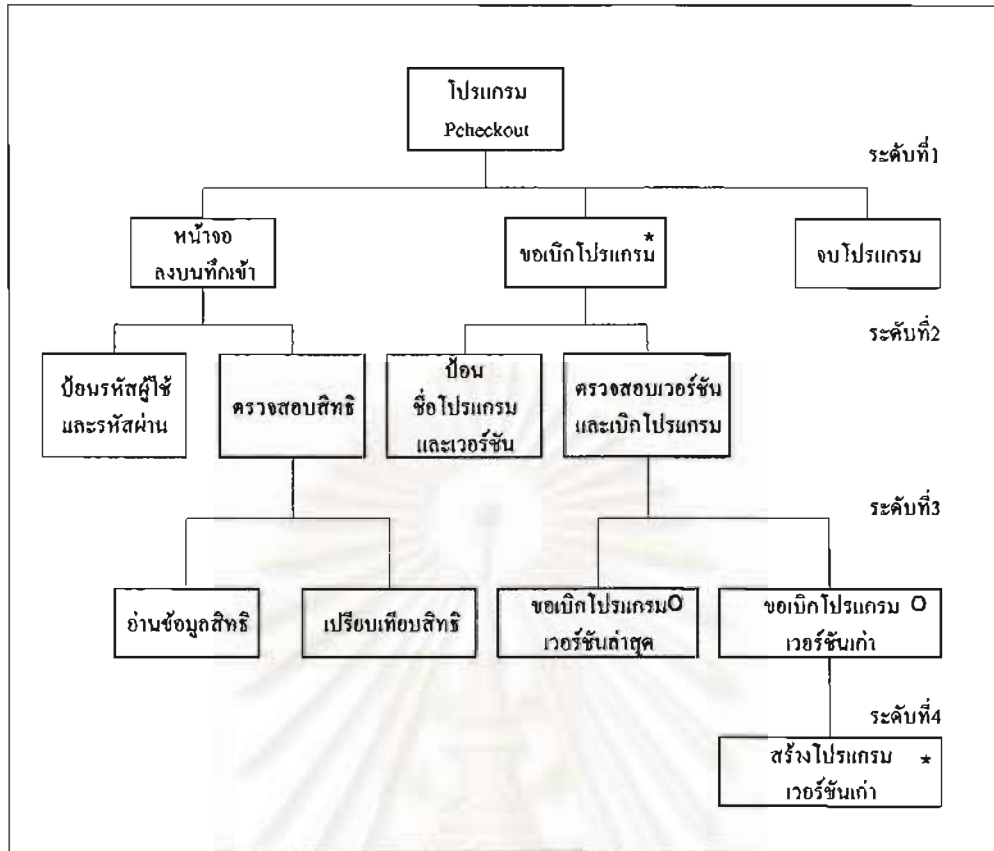
- หน้าจอลงบันทึกเข้า ซึ่งทำหน้าที่รับป้อนรหัสผู้ใช้และรหัสผ่าน เพื่อตรวจสอบสิทธิในการเข้าใช้งานโปรแกรม
- การจัดเก็บโปรแกรม ที่มีการทำงานแบบวนซ้ำจนผู้ใช้ของโปรแกรม จะทำหน้าที่รับป้อนชื่อโปรแกรมที่ต้องการทำการจัดเก็บ ทำการตรวจสอบว่าเป็นโปรแกรมใหม่ที่ไม่เคยมีการจัดเก็บไว้ หรือเป็นโปรแกรมเดิมที่ถูกนำออกจากระบบไปทำการแก้ไข ในกรณีที่โปรแกรมใหม่ก็จะทำการจัดเก็บด้วยส่วนที่

ทำหน้าที่จัดเก็บ โปรแกรมใหม่ แต่ถ้าเป็น โปรแกรมเดิมที่เคยจัดเก็บไว้ ก็จะทำการจัดเก็บ โปรแกรมด้วยส่วนที่ทำหน้าที่จัดเก็บ โปรแกรมที่แก้ไข ตรวจสอบ และแสดงความสัมพันธ์ที่ขัดแย้งระหว่าง โปรแกรมที่จัดเก็บกับ โปรแกรมที่มีอยู่ในระบบ



รูปที่ 3.8 แผนภาพโครงสร้าง ของโปรแกรม Pcheckin

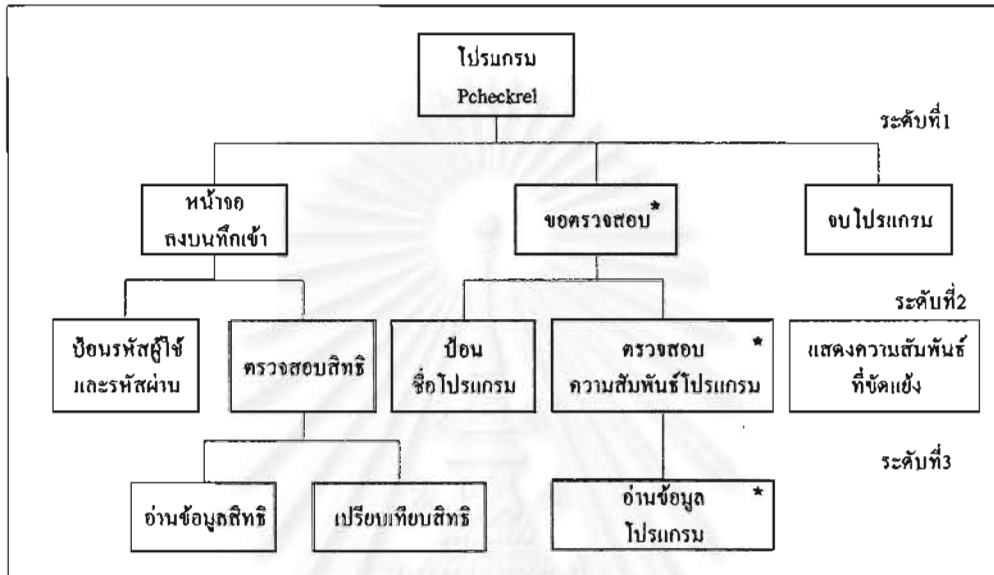
2. โปรแกรม Pcheckout เป็น โปรแกรมที่ทำหน้าที่ในส่วนขอเบิก โปรแกรม โดยมีหน้าที่หลัก คือ (ดูรูปที่ 3.9)
  - หน้าจอลงบันทึกเข้า ซึ่งทำหน้าที่รับป้อนรหัสผู้ใช้และรหัสผ่าน เพื่อตรวจสอบสิทธิในการเข้าใช้งาน โปรแกรม
  - การขอเบิกโปรแกรม ที่การทำงานแบบวนซ้ำจนผู้ใช้ของจบโปรแกรม ซึ่งทำหน้าที่รับป้อนชื่อ โปรแกรมและเวอร์ชันที่ต้องการขอเบิก ทำการตรวจสอบว่าเป็นเวอร์ชันของโปรแกรมเป็นเวอร์ชันล่าสุด หรือเป็นเวอร์ชันย้อนหลัง ในกรณีที่โปรแกรมที่ขอเบิกเป็นเวอร์ชันล่าสุดก็จะทำการนำโปรแกรมจากระบบให้กับผู้ใช้ แต่ถ้าเป็น โปรแกรมเวอร์ชันย้อนหลัง ก็จะทำการสร้างโปรแกรมเวอร์ชันย้อนหลังจากโปรแกรมเวอร์ชันล่าสุดและรายการการเปลี่ยนแปลงที่เกิดขึ้นในแต่ละเวอร์ชันที่ระบบได้จัดเก็บไว้ วนซ้ำไปจนกระทั่งได้โปรแกรมเวอร์ชันที่ผู้ใช้ต้องการ



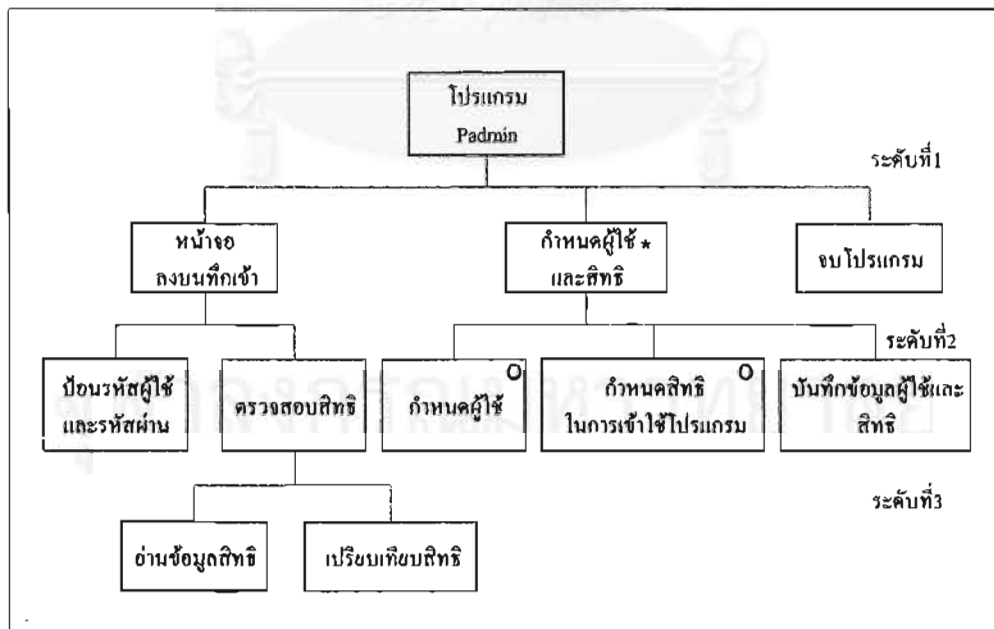
รูปที่ 3.9 แผนภาพโครงสร้าง ของโปรแกรม Pcheckout

3. โปรแกรม Pcheckrel เป็นโปรแกรมที่ทำหน้าที่ในส่วนตรวจสอบและแสดงความสัมพันธ์ที่ขัดแย้ง โดยมีหน้าที่หลัก คือ (ดูรูปที่ 3.10)
  - หน้าจอลงบันทึกเข้า ซึ่งทำหน้าที่รับป้อนรหัสผู้ใช้และรหัสผ่าน เพื่อตรวจสอบสิทธิในการเข้าใช้งานโปรแกรม
  - การตรวจสอบความสัมพันธ์ของโปรแกรม ที่มีการทำงานแบบวนซ้ำงานผู้ใช้ขอจบโปรแกรม จะทำหน้าที่รับป้อนชื่อ โปรแกรมที่ต้องการตรวจสอบ ทำการตรวจสอบว่ามีการเรียกใช้งานฟังก์ชันของโปรแกรมใดบ้าง อ่านข้อมูลของโปรแกรมเวอร์ชันล่าสุดที่มีความสัมพันธ์ ตรวจสอบความสัมพันธ์และแสดงความสัมพันธ์ที่ขัดแย้งระหว่างโปรแกรมที่จัดเก็บกับ โปรแกรมที่มีอยู่ในระบบ
4. โปรแกรม Padmin เป็นโปรแกรมที่ทำหน้าที่ในส่วนกำหนดผู้ใช้ระบบและสิทธิในการใช้งาน โดยมีหน้าที่หลัก คือ (ดูรูปที่ 3.11)

- หน้าจอลงทะเบียนที่ก่เข้า ซึ่งทำหน้าที่รับป้อนรหัสผู้ใช้และรหัสผ่าน เพื่อตรวจสอบสิทธิในการเข้าใช้งานโปรแกรม
- การกำหนดผู้ใช้และสิทธิในการใช้โปรแกรม ที่มีการทำงานแบบวนซ้ำจนผู้ใช้ขอลบโปรแกรม จะแบ่งเป็นส่วนในการกำหนดผู้ใช้งานระบบ และส่วนในการกำหนดคสิทธิในการใช้งานโปรแกรม และบันทึกข้อมูลทั้งหมดเข้าสู่ระบบ



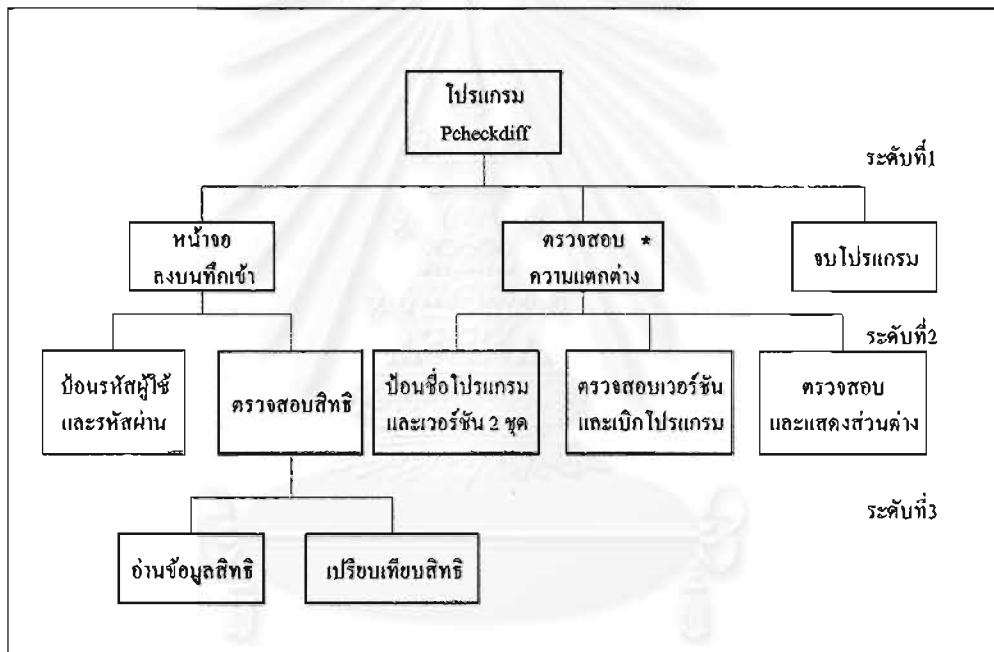
รูปที่ 3.10 แผนภาพโครงสร้าง ของโปรแกรม Pcheckrel



รูปที่ 3.11 แผนภาพโครงสร้าง ของโปรแกรม Padmin

5. โปรแกรม Pcheckdiff เป็นโปรแกรมที่ทำหน้าที่ในส่วนแสดงความแตกต่างของชุดคำสั่งมอดูลเดียวกันที่เวอร์ชันแตกต่างกัน โดยมีหน้าที่หลัก คือ (ดูรูปที่ 3.12)

- หน้าจอลงบันทึกเข้า ซึ่งทำหน้าที่รับป้อนรหัสผู้ใช้และรหัสผ่าน เพื่อตรวจสอบสิทธิในการเข้าใช้งานโปรแกรม
- การตรวจสอบความแตกต่าง ที่มีการทำงานแบบวนซ้ำจนผู้ใช้ของจบโปรแกรม จะแบ่งเป็นส่วนรับป้อนชื่อโปรแกรมที่ต้องการตรวจสอบพร้อมกับเลขเวอร์ชันทั้ง 2 ชุดที่ต้องการเปรียบเทียบความแตกต่าง ตรวจสอบเวอร์ชันที่ต้องการเปรียบเทียบและเปิดโปรแกรมออกจากระบบ มาทำการตรวจสอบความแตกต่างและแสดงผล



รูปที่ 3.12 แผนภาพโครงสร้าง ของโปรแกรม Pcheckdiff



### 3.3 การออกแบบส่วนจัดเก็บ

ในการออกแบบส่วนจัดเก็บข้อมูลชุดคำสั่งมอดูลในงานวิจัยนี้ ได้ทำการจัดเก็บข้อมูลดังกล่าวด้วยระบบฐานข้อมูลตามวิธีการจัดเก็บข้อมูลแบบเคลค้ำ โดยในการออกแบบได้แบ่งข้อมูลต่างๆออกเป็น 2 ประเภท และจัดเก็บในลักษณะของตาราง ดังนี้

#### 3.3.1 ข้อมูลสำหรับการจัดการตัวระบบ

เป็นข้อมูลที่ทำกรจัดเก็บไว้เพื่อการทำงานของตัวเอง เป็นตารางที่ได้มีการสร้างขึ้นเตรียมไว้ตั้งแต่ตอนที่ทำการติดตั้งระบบโดยมีรายละเอียดดังนี้

- 1) ตารางรายชื่อ โปรแกรม/ชุดคำสั่งมอดูล (SCVSProgList) เป็นตารางที่ใช้ในการจัดเก็บรายชื่อ โปรแกรม หรือชุดคำสั่งมอดูล เพื่อใช้ในการตรวจสอบว่าในระบบได้มีการจัดเก็บชื่อ โปรแกรม หรือชุดคำสั่งมอดูลใดบ้าง เก็บหมายเลขรุ่นล่าสุดของโปรแกรมหรือชุดคำสั่งมอดูลที่จัดเก็บอยู่ รวมทั้งรหัสผู้ใช้ระบบที่กำลังแก้ไขโปรแกรมหรือชุดคำสั่งมอดูล ในกรณีที่โปรแกรมหรือชุดคำสั่งมอดูลถูกนำออกจากระบบไปเพื่อทำการแก้ไข ลักษณะ โครงสร้างของตารางได้แสดงไว้ในตารางที่

3.1

ตารางที่ 3.1 โครงสร้างของตารางรายชื่อ โปรแกรม/ชุดคำสั่งมอดูล

ชื่อเขตข้อมูล	แบบชนิดข้อมูล	คำอธิบาย
ProgID	Integer	หมายเลขประจำ โปรแกรมหรือชุดคำสั่งมอดูล
ProgName	Text 20	ชื่อแฟ้ม โปรแกรมหรือชุดคำสั่งมอดูล
ProgDes	Text 100	คำอธิบาย โปรแกรมหรือชุดคำสั่งมอดูลอย่างย่อ
ProgLastVer1	Integer	หมายเลขรุ่นล่าสุดของ โปรแกรมหรือชุดคำสั่งมอดูล
ProgLastVer2	Integer	หมายเลขรุ่นย่อยล่าสุดของ โปรแกรมหรือชุดคำสั่งมอดูล
LockUser	Text 10	รหัสผู้ใช้ระบบที่ขอเปิดแก้ไข โปรแกรม

- 2) ตารางรายชื่อฟังก์ชันที่จัดเก็บในระบบ (SCVSPFuncList) เป็นตารางที่ใช้เก็บรายชื่อฟังก์ชันของทุก โปรแกรม หรือชุดคำสั่งมอดูลที่จัดเก็บไว้ในระบบ ซึ่งได้จากการที่ระบบได้ทำการเลิกซิคัล โปรแกรมหรือชุดคำสั่งมอดูล เพื่อค้นหาชื่อฟังก์ชันที่ได้นิยามไว้ ลักษณะ โครงสร้างของตารางได้แสดงไว้ในตารางที่ 3.2

ตารางที่ 3.2 โครงสร้างของตารางรายชื่อฟังก์ชันที่จัดเก็บในระบบ

ชื่อเขตข้อมูล	แบบชนิดข้อมูล	คำอธิบาย
ProgID	Integer	หมายเลขประจำโปรแกรมหรือชุดคำสั่งมอดูล
ProgName	Text 20	ชื่อแฟ้มโปรแกรมหรือชุดคำสั่งมอดูล
FuncName	Text 50	ชื่อฟังก์ชันที่นิยามไว้ ในแฟ้มโปรแกรม หรือชุดคำสั่งมอดูล

- 3) ตารางรายละเอียดพารามิเตอร์ของฟังก์ชัน (SCVFuncPara) เป็นตารางที่ใช้จัดเก็บรายละเอียดพารามิเตอร์แต่ละตัวในแต่ละฟังก์ชัน รวมถึงชื่อพารามิเตอร์ และชนิดของพารามิเตอร์ ซึ่งได้จากการที่ระบบได้ทำการเลือกซีคัลขณะทำการจัดเก็บโปรแกรมหรือชุดคำสั่งมอดูล เพื่อกันหานิยามของพารามิเตอร์แต่ละฟังก์ชัน ลักษณะโครงสร้างของตารางได้แสดงไว้ในตารางที่ 3.3

ตารางที่ 3.3 โครงสร้างของตารางรายละเอียดพารามิเตอร์ของฟังก์ชัน

ชื่อเขตข้อมูล	แบบชนิดข้อมูล	คำอธิบาย
FuncName	Text 50	ชื่อฟังก์ชัน
Seqno	Integer	หมายเลขแสดงลำดับของพารามิเตอร์ ในฟังก์ชัน
VarName	Text 50	ชื่อตัวแปร
VarType	Text 50	ชื่อชนิดตัวแปร
ParaType	Text 1	รหัสประเภทของพารามิเตอร์ 1 หมายถึง พารามิเตอร์ที่ส่งค่าให้ฟังก์ชัน 2 หมายถึง พารามิเตอร์ที่ฟังก์ชันส่งค่าคืนให้

- 4) ตารางรายชื่อผู้ใช้ระบบ (SCVSUsers) เป็นตารางที่ใช้ในการจัดเก็บชื่อและสิทธิในการใช้งานระบบจัดการเวอร์ชันชุดคำสั่งของผู้ใช้ระบบ โครงสร้างของตารางได้แสดงไว้ในตารางที่ 3.4

ตารางที่ 3.4 โครงสร้างของตารางรายชื่อผู้ใช้ระบบ

ชื่อเขตข้อมูล	แบบชนิดข้อมูล	คำอธิบาย
UserID	Text 10	รหัสผู้ใช้ระบบ
UserName	Text 50	ชื่อผู้ใช้ระบบ
UserSurname	Text 50	ชื่อสกุลผู้ใช้ระบบ
UserPassword	Text 10	รหัสผ่านผู้ใช้ระบบ
UserLevel	Integer	ระดับผู้ใช้ระบบ

- 5) ตารางสิทธิผู้ใช้ระบบในการเข้าถึงโปรแกรม/ชุดคำสั่งมอดูล (SCVSAuthor) เป็นตารางที่ใช้จัดเก็บสิทธิในการเข้าเรียกใช้โปรแกรมหรือชุดคำสั่งมอดูล รวมถึงชื่อพารามิเตอร์ และชนิดของพารามิเตอร์ โครงสร้างของตารางได้แสดงไว้ในตารางที่ 3.5

ตารางที่ 3.5 โครงสร้างของตารางสิทธิผู้ใช้ระบบในการเข้าถึงโปรแกรม/ชุดคำสั่งมอดูล

ชื่อเขตข้อมูล	แบบชนิดข้อมูล	คำอธิบาย
UserID	Text 10	รหัสผู้ใช้ระบบ
ProgID	Integer	หมายเลขประจำโปรแกรมหรือชุดคำสั่งมอดูล
PrivilegeCode	Text 8	รหัสสิทธิในการเข้าถึงเพิ่มโปรแกรมหรือชุดคำสั่งมอดูล

- 6) ตารางประวัติการลงโปรแกรม/ชุดคำสั่งมอดูล (SCVSInstall) เป็นตารางที่ใช้จัดเก็บบันทึกการนำโปรแกรมหรือชุดคำสั่งมอดูล จากระบบเพื่อติดตั้งใช้งานที่ถูกค้าแต่ละแห่ง ซึ่งทำให้สามารถตรวจสอบได้ว่าลูกค้าได้ติดตั้งโปรแกรมหรือชุดคำสั่งมอดูล ตัวใดไปแล้วบ้าง และเป็นเวอร์ชันใด โครงสร้างของตารางได้แสดงไว้ในตารางที่ 3.6

ตารางที่ 3.6 โครงสร้างของตารางประวัติการลงโปรแกรม/ชุดคำสั่งมอดูล

ชื่อเขตข้อมูล	แบบชนิดข้อมูล	คำอธิบาย
CustName	Text 50	ชื่อลูกค้าที่ติดตั้งโปรแกรมหรือชุดคำสั่งมอดูล

ProgID	Integer	หมายเลขประจำโปรแกรมหรือชุดคำสั่งมอดูล
ProgVer1	Integer	หมายเลขรุ่นของโปรแกรมหรือชุดคำสั่งมอดูล ที่ทำการติดตั้ง
ProgVer2	Integer	หมายเลขรุ่นย่อยของโปรแกรมหรือชุดคำสั่งมอดูล ที่ทำการติดตั้ง
UserID	Text 10	รหัสผู้ใช้ระบบที่เบกโปรแกรมหรือชุดคำสั่งมอดูล เพื่อทำการติดตั้ง
InstallDate	Date	วันที่ทำการลงโปรแกรมหรือชุดคำสั่งมอดูล

### 3.3.2 ข้อมูลชุดคำสั่งมอดูลที่ถูกจัดเก็บ

เป็นข้อมูลของโปรแกรมหรือชุดคำสั่งมอดูล ที่ระบบได้ทำการจัดเก็บไว้ในรูปของตาราง โดยระบบจะสร้างขึ้นเองภายหลังในระหว่างที่ระบบถูกใช้งานแล้ว ระบบจะทำการจัดเก็บแฟ้มโปรแกรมหรือชุดคำสั่งมอดูลที่จัดเก็บ 1 แฟ้ม แยกออกเป็นตารางย่อยๆ โดยใช้ชื่อแฟ้มเป็นส่วนหนึ่งของชื่อตารางที่สร้างขึ้น ดังนี้

- 1) ตารางเก็บโปรแกรม/ชุดคำสั่งมอดูลชุดล่าสุด เป็นตารางที่ใช้เก็บตัวโปรแกรมหรือชุดคำสั่งมอดูลชุดล่าสุด โดยชื่อของตารางที่ระบบสร้างขึ้นจะใช้กฎเกณฑ์ว่า

S\_[ชื่อของแฟ้มโปรแกรมหรือชุดคำสั่งมอดูล]

เช่น ต้องการจัดเก็บแฟ้มโปรแกรมชื่อ postcust.4gl ตารางที่ระบบสร้างขึ้นจะมีชื่อว่า S\_postcust เป็นต้น โครงสร้างของตารางที่สร้างขึ้นมีลักษณะดังแสดงในตารางที่ 3.7

ตารางที่ 3.7 โครงสร้างของตารางเก็บโปรแกรม/ชุดคำสั่งมอดูล

ชื่อเขตข้อมูล	แบบชนิดข้อมูล	คำอธิบาย
LineNo	Integer	หมายเลขบรรทัด โปรแกรมหรือชุดคำสั่งมอดูล
LineStr	Text 255	บรรทัดข้อความภายในโปรแกรมหรือชุดคำสั่งมอดูล

- 2) ตารางเก็บการเปลี่ยนแปลง โปรแกรม/ชุดคำสั่งมอดูล เป็นตารางที่ใช้เก็บส่วนที่เปลี่ยนแปลงของโปรแกรมหรือชุดคำสั่งมอดูลในแต่ละเวอร์ชัน โดยชื่อของตารางที่ระบบสร้างขึ้นจะใช้กฎเกณฑ์ว่า

C\_[ชื่อของแฟ้ม โปรแกรมหรือชุดคำสั่งมอดูล]

เช่น ต้องการจัดเก็บแฟ้มโปรแกรมชื่อ postcust.4gl ตารางที่ระบบสร้างขึ้นจะมีชื่อ  
ว่า C\_postcust เป็นต้น โครงสร้างของตารางที่สร้างขึ้นมีลักษณะดังแสดงในตาราง  
ที่ 3.8

ตารางที่ 3.8 โครงสร้างของตารางเก็บการเปลี่ยนแปลง โปรแกรม/ชุดคำสั่งมอดูล

ชื่อเขตข้อมูล	แบบชนิดข้อมูล	คำอธิบาย
ProgVer1	Integer	หมายเลขรุ่นของโปรแกรมหรือชุดคำสั่งมอดูล
ProgVer2	Integer	หมายเลขรุ่นย่อยของโปรแกรมหรือชุดคำสั่งมอดูล
LineNo	Integer	หมายเลขบรรทัด โปรแกรมหรือชุดคำสั่งมอดูล
LineStr	Text 255	บรรทัดข้อความภายในโปรแกรมหรือชุดคำสั่งมอดูล

- 3) ตารางรายชื่อฟังก์ชันที่เรียกใช้ในโปรแกรม/ชุดคำสั่งมอดูลชุดล่าสุด เป็นตารางที่  
ใช้เก็บเลขบรรทัดและชื่อฟังก์ชันที่มีการเรียกใช้งานภายในโปรแกรมหรือชุดคำสั่ง  
มอดูลชุดล่าสุด โดยชื่อของตารางที่ระบบสร้างขึ้นจะใช้กฎเกณฑ์ว่า

CF\_[ชื่อของแฟ้มโปรแกรมหรือชุดคำสั่งมอดูล]

เช่น ต้องการจัดเก็บแฟ้มโปรแกรมชื่อ postcust.4gl ตารางที่ระบบสร้างขึ้นจะมีชื่อ  
ว่า CF\_postcust เป็นต้น โครงสร้างของตารางที่สร้างขึ้น มีลักษณะดังแสดงใน  
ตารางที่ 3.9

ตารางที่ 3.9 โครงสร้างของตารางรายชื่อฟังก์ชันที่เรียกใช้ในโปรแกรม/ชุดคำสั่งมอดูลชุดล่าสุด

ชื่อเขตข้อมูล	แบบชนิดข้อมูล	คำอธิบาย
LineNo	Integer	หมายเลขบรรทัด โปรแกรมหรือชุดคำสั่งมอดูล
FuncName	Text 50	ชื่อฟังก์ชันที่โปรแกรมหรือชุดคำสั่งมอดูลเรียกใช้งาน

- 4) ตารางรายละเอียดพารามิเตอร์ของฟังก์ชันชุดล่าสุด เป็นตารางที่ใช้เก็บเลขบรรทัด  
และชื่อฟังก์ชันที่มีการเรียกใช้งานภายในโปรแกรมหรือชุดคำสั่งมอดูลชุดล่าสุด  
โดยชื่อของตารางที่ระบบสร้างขึ้นจะใช้กฎเกณฑ์ว่า

CP\_[ชื่อของแฟ้มโปรแกรมหรือชุดคำสั่งมอดูล]

เช่น ต้องการจัดเก็บแฟ้มโปรแกรมชื่อ postcust.4gl ตารางที่ระบบสร้างขึ้นจะมีชื่อ  
ว่า CP\_postcust เป็นต้น โครงสร้างของตารางที่สร้างขึ้นมีลักษณะดังแสดงใน  
ตารางที่ 3.10

ตารางที่ 3.10 โครงสร้างของตารางรายละเอียดพารามิเตอร์ของฟังก์ชันชุดล่าสุด

ชื่อเขตข้อมูล	แบบชนิดข้อมูล	คำอธิบาย
FuncName	Text 50	ชื่อฟังก์ชันที่โปรแกรมหรือชุดคำสั่งมอดูลเรียกใช้งาน
Seqno	Integer	หมายเลขแสดงลำดับของพารามิเตอร์ ในฟังก์ชันที่ถูก เรียกใช้งาน
VarName	Text 50	ชื่อตัวแปร
VarType	Text 50	ชื่อชนิดตัวแปร
ParaType	Text 1	รหัสประเภทของพารามิเตอร์ 1 หมายถึง พารามิเตอร์ที่ส่งค่าให้ฟังก์ชัน 2 หมายถึง พารามิเตอร์ที่ฟังก์ชันส่งค่าคืนให้

- 5) ตารางประวัติหมายเหตุการแก้ไข โปรแกรม/ชุดคำสั่งมอดูล เป็นตารางที่ใช้เก็บข้อ  
ความหมายเหตุ ที่ทำการแก้ไข โปรแกรมหรือชุดคำสั่งมอดูลชุดล่าสุด โดยชื่อของ  
ตารางที่ระบบสร้างขึ้นจะใช้กฎเกณฑ์ว่า

H\_[ชื่อของแฟ้มโปรแกรมหรือชุดคำสั่งมอดูล]

เช่น ต้องการจัดเก็บแฟ้มโปรแกรมชื่อ postcust.4gl ตารางที่ระบบสร้างขึ้นจะมีชื่อ  
ว่า H\_postcust เป็นต้น โครงสร้างของตารางที่สร้างขึ้นมีลักษณะดังแสดงในตาราง  
ที่ 3.11

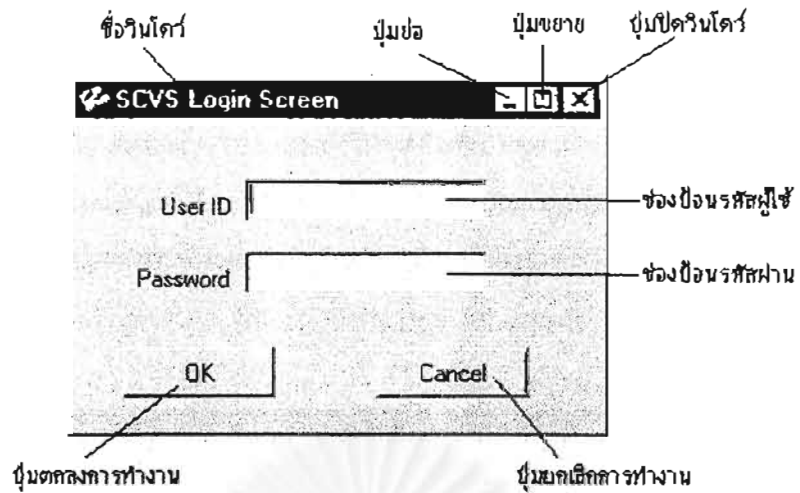
ตารางที่ 3.11 โครงสร้างของตารางประวัติหมายเหตุการแก้ไข โปรแกรม/ชุดคำสั่งมอดูล

ชื่อเขตข้อมูล	แบบชนิดข้อมูล	คำอธิบาย
ProgVer1	Integer	หมายเลขรุ่นของโปรแกรมหรือชุดคำสั่งมอดูล
ProgVer2	Integer	หมายเลขรุ่นย่อยของโปรแกรมหรือชุดคำสั่งมอดูล
ModiComment	Memo	ข้อความหมายเหตุของการแก้ไข โปรแกรมหรือชุดคำสั่ง มอดูล

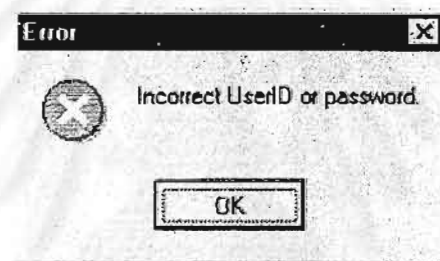
### 3.4 การออกแบบตัวประสานผู้ใช้

กำหนดให้ตัวประสานกับผู้ใช้ของโปรแกรมทุกตัว เริ่มต้นด้วยหน้าจอลงบันทึกเข้า (Login) เพื่อให้สามารถตรวจสอบสิทธิในการเรียกใช้โปรแกรม และสิทธิในการเข้าถึง โปรแกรมหรือชุดคำสั่งมอดูล โดยเมื่อเรียกใช้โปรแกรมจะมีลักษณะหน้าจอลงบันทึกเข้า (ดังรูปที่ 3.13) โดยมีรายละเอียดดังนี้

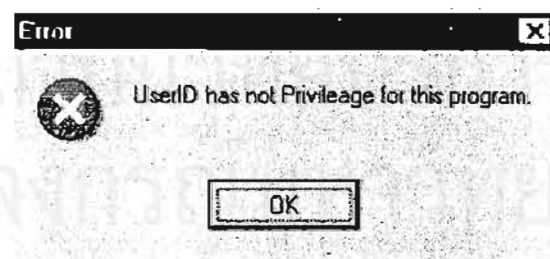
- ชื่อวินโดวส์ เมื่อกดปุ่มเมาส์ขวาจะปรากฏรายการคำสั่งสำหรับควบคุมวินโดวส์ ซึ่งประกอบด้วยคำสั่งเปลี่ยนขนาดวินโดวส์ให้มีขนาดเท่าเดิมก่อนขยายให้มีขนาดเต็มจอภาพหรือลดขนาดเป็นสัญรูป คำสั่งเคลื่อนย้ายวินโดวส์ คำสั่งกำหนดขนาดวินโดวส์ คำสั่งลดขนาดวินโดวส์ให้มีขนาดเล็กสุด คำสั่งขยายขนาดวินโดวส์ให้เต็มจอภาพ คำสั่งปิดวินโดวส์
- ปุ่มย่อ เมื่อกดปุ่มนี้จะลดขนาดวินโดวส์ให้มีขนาดเล็กสุด โดยจะแสดงวินโดวส์ในรูปของสัญรูป
- ปุ่มขยาย เมื่อกดปุ่มนี้จะขยายวินโดวส์ให้เต็มจอภาพ
- ปุ่มปิดวินโดวส์ เมื่อกดปุ่มนี้จะเป็นการปิดวินโดวส์
- ช่องป้อนรหัสผู้ใช้ ใช้สำหรับป้อนรหัสผู้ใช้ เพื่อขอเข้าใช้งานระบบจัดการเวอร์ชันชุดคำสั่ง
- ช่องป้อนรหัสผ่าน ใช้สำหรับป้อนรหัสผ่านของผู้ใช้ เพื่อยืนยันสถานะของผู้ใช้
- ปุ่มตกลงการทำงาน เมื่อกดปุ่มนี้หลังจากป้อนรหัสผู้ใช้และรหัสผ่าน ระบบจะทำการตรวจสอบชื่อผู้ใช้และรหัสผ่านกับที่เก็บไว้ในระบบ ในกรณีที่รหัสผู้ใช้หรือรหัสผ่านไม่ถูกต้องจะปรากฏหน้าจอแสดงข้อความผิดพลาด (ดังรูปที่ 3.14) หรือในกรณีที่รหัสผู้ใช้ไม่มีสิทธิในการใช้งาน โปรแกรมที่ต้องการเรียกใช้หน้าจอจะแสดงข้อความผิดพลาด (ดังรูปที่ 3.15)
- ปุ่มยกเลิกการทำงาน เมื่อกดปุ่มนี้จะเป็นการยกเลิกการทำงานและออกจากโปรแกรม



รูปที่ 3.13 หน้าจอฉบับที่กรอกข้อมูล



รูปที่ 3.14 หน้าจอแสดงข้อความผิดพลาดของรหัสผู้ใช้หรือรหัสผ่านของผู้ใช้



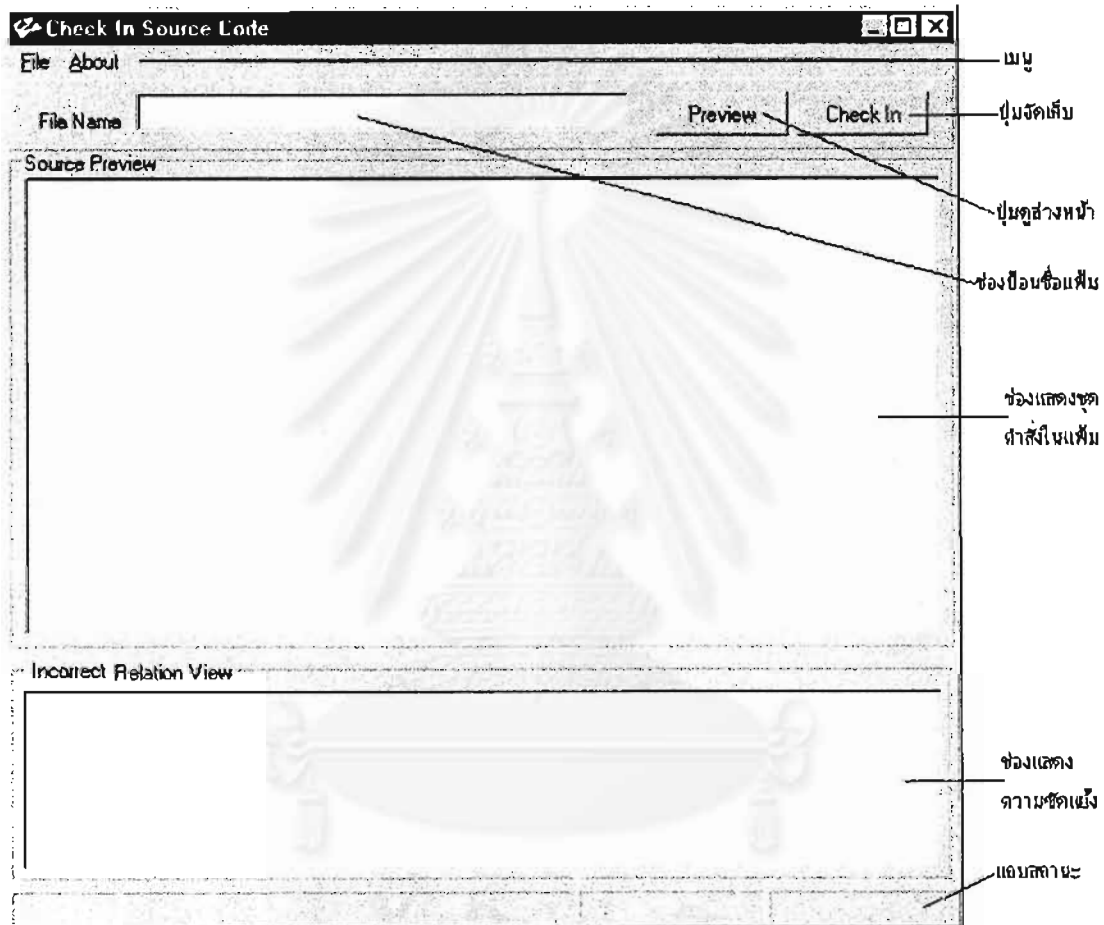
รูปที่ 3.15 หน้าจอแสดงข้อความผิดพลาดเนื่องจากผู้ใช้ไม่มีสิทธิในการใช้



สำหรับการออกแบบตัวประสานผู้ใช้ส่วนอื่น สามารถแยกออกแบบตาม โปรแกรมใช้งาน ดังนี้

### 3.4.1 โปรแกรมจัดเก็บ โปรแกรมหรือชุดคำสั่งมอดูล

เป็นโปรแกรมที่ทำหน้าที่นำโปรแกรมหรือชุดคำสั่งมอดูลใหม่ หรือแก้ไขเปลี่ยนแปลง เข้าไปจัดเก็บในระบบจัดการเวอร์ชันชุดคำสั่ง โดยมีหน้าจอโปรแกรมดังแสดงในรูปที่ 3.16 และมีรายละเอียดดังนี้



รูปที่ 3.16 หน้าจอโปรแกรมจัดเก็บโปรแกรมหรือชุดคำสั่งมอดูล

- เมนู ประกอบด้วยรายการคำสั่งต่างๆ โดยจะเรียงอยู่ในแนวนอนด้านบนของวินโดวส์ รายการคำสั่งโปรแกรมนี้มีดังต่อไปนี้
  - ◆ File เป็นรายการคำสั่งเกี่ยวกับแฟ้มข้อมูล โดยมีเมนูย่อยดังต่อไปนี้
    - Open เป็นรายการเลือกสำหรับการเปิดแฟ้มของ โปรแกรมหรือชุดคำสั่งมอดูล ที่ต้องการจัดเก็บเข้าสู่ระบบ

- Exit เป็นรายการเลือกสำหรับสั่งเลิกงาน

◆ About เป็นรายการคำสั่งสำหรับแสดงรายละเอียดของโปรแกรม

- ช่องป้อนชื่อแฟ้ม เป็นช่องที่ใช้สำหรับป้อนชื่อแฟ้ม โปรแกรมหรือชุดคำสั่งมอดูลที่ต้องการทำการจัดเก็บ เพื่อเป็นการสะดวกและรวดเร็วแทนการใช้ผ่านเมนูย่อย Open
- ปุ่มดูล่วงหน้า เมื่อกดปุ่มหลังจากที่ได้ป้อนชื่อแฟ้มที่ต้องการจัดเก็บแล้ว โปรแกรมจะทำการนำชุดคำสั่งที่อยู่ในแฟ้มมาแสดงในช่องแสดงชุดคำสั่งในแฟ้ม เพื่อให้ผู้ใช้ได้ตรวจทานก่อนจัดเก็บจริง
- ปุ่มจัดเก็บ เมื่อกดปุ่มหลังจากที่ได้ป้อนชื่อแฟ้มที่ต้องการจัดเก็บแล้ว โปรแกรมจะทำการนำชุดคำสั่งที่อยู่ในแฟ้มจัดเก็บลงในฐานะข้อมูลของระบบ
- ช่องแสดงชุดคำสั่งในแฟ้ม เป็นช่องที่ใช้แสดงชุดคำสั่งที่อยู่ในแฟ้มที่ต้องการทำการจัดเก็บไว้ในระบบ
- ช่องแสดงความขัดแย้ง เป็นช่องที่ใช้แสดงความสัมพันธ์ที่ขัดแย้งระหว่างโปรแกรมหรือชุดคำสั่งมอดูลที่ทำการจัดเก็บ กับ โปรแกรมหรือชุดคำสั่งมอดูลที่จัดเก็บไว้แล้วในระบบ
- แถบสถานะ จะแสดงข้อมูลสถานะของ โปรแกรมหรือชุดคำสั่งมอดูลที่กำลังทำการจัดเก็บ เช่น จำนวนบรรทัดในแฟ้ม หรือแสดงข้อความสถานะการทำงานของโปรแกรมจัดเก็บ โปรแกรมหรือชุดคำสั่งมอดูล เช่น “Please wait while creating new source table” ในขณะที่โปรแกรมจัดเก็บกำลังสร้างตารางสำหรับจัดเก็บ โปรแกรมหรือชุดคำสั่งมอดูล เป็นต้น

นอกจากหน้าจอโปรแกรมจัดเก็บ โปรแกรมหรือชุดคำสั่งมอดูล ที่เป็นหน้าจอหลักแล้ว ในการทำการจัดเก็บ โปรแกรมหรือชุดคำสั่งมอดูล กรณีที่แตกต่างกันจะปรากฏหน้าจอแตกต่างกันดังนี้

- กรณีที่แฟ้ม โปรแกรมหรือชุดคำสั่งมอดูล ที่จัดเก็บเป็น โปรแกรมหรือชุดคำสั่งมอดูลที่ไม่เคยมีการจัดเก็บในระบบมาก่อน จะมีวินโดวส์สำหรับป้อนคำอธิบายโปรแกรมหรือชุดคำสั่งมอดูลอย่างย่อ ปรากฏดังรูปที่ 3.17

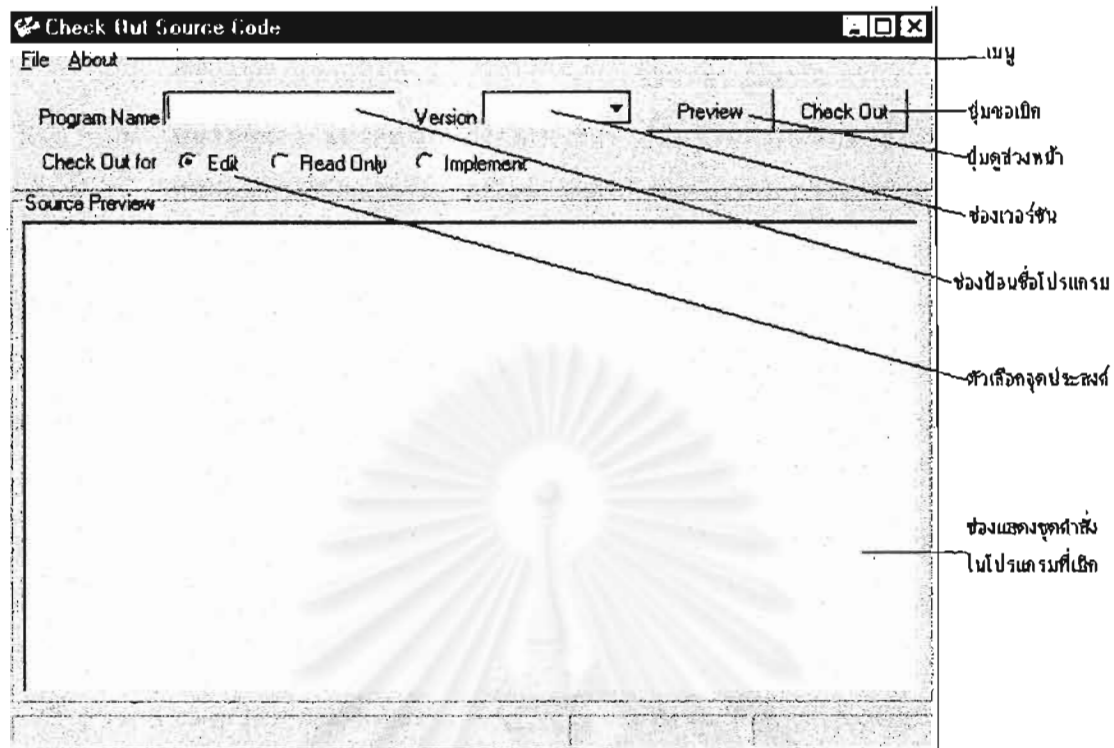
รูปที่ 3.17 หน้าจอป้อนคำอธิบายอย่างย่อสำหรับโปรแกรมหรือชุดคำสั่งมอดูล

- กรณีที่เพิ่ม โปรแกรมหรือชุดคำสั่งมอดูล ที่จัดเก็บเป็น โปรแกรมหรือชุดคำสั่งมอดูลที่เคยมีการจัดเก็บในระบบมาก่อน จะมีวินโดวส์สำหรับป้อนหมายเหตุการแก้ไขสำหรับแต่ละเวอร์ชันของ โปรแกรมหรือชุดคำสั่งมอดูล ปรากฏดังรูปที่ 3.18

รูปที่ 3.18 หน้าจอป้อนหมายเหตุการแก้ไขสำหรับโปรแกรมหรือชุดคำสั่งมอดูล

### 3.4.2 โปรแกรมขอเปิดโปรแกรมหรือชุดคำสั่งมอดูล

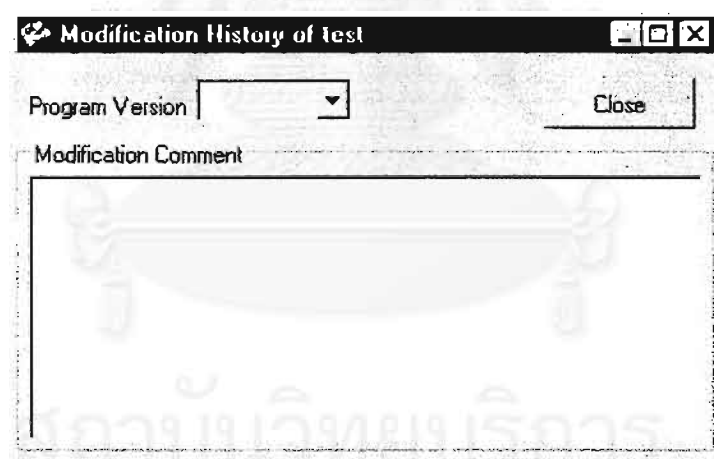
เป็นโปรแกรมที่ทำหน้าที่นำโปรแกรมหรือชุดคำสั่งมอดูลที่จัดเก็บไว้ในระบบจัดการเวอร์ชันชุดคำสั่ง สำหรับเวอร์ชันที่ต้องการออกมาเป็นแฟ้ม เพื่อการนำไปแก้ไขเปลี่ยนแปลง หรือเพื่อการนำไปติดตั้งให้กับลูกค้า เป็นต้น โดยมีหน้าจอโปรแกรมดังแสดงในรูปที่ 3.19 และมีรายละเอียดดังนี้



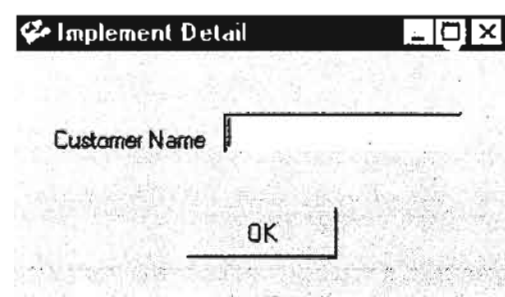
รูปที่ 3.19 หน้าจอโปรแกรมขอเปิดโปรแกรมหรือชุดคำสั่งมอดูล

- เมนู ประกอบด้วยรายการคำสั่งต่างๆ โดยจะเรียงอยู่ในแนวนอนด้านบนของวินโดวส์ รายการคำสั่งโปรแกรมนี้มีดังต่อไปนี้
  - ◆ File เป็นรายการคำสั่งเกี่ยวกับเพิ่มข้อมูล โดยมีเมนูย่อยดังต่อไปนี้
    - Save เป็นรายการเลือกสำหรับการจัดเก็บเพิ่มของ โปรแกรมหรือชุดคำสั่งมอดูล ที่ต้องการขอเปิดจากระบบ
    - History เป็นรายการเลือกสำหรับเปิดหน้าจอแสดงประวัติการแก้ไข ดังที่แสดงในรูปที่ 3.20
    - Exit เป็นรายการเลือกสำหรับสั่งเลิกงาน
  - ◆ About เป็นรายการคำสั่งสำหรับแสดงรายละเอียดของโปรแกรม
- ช่องป้อนชื่อโปรแกรม เป็นช่องที่ใช้สำหรับป้อนชื่อ โปรแกรมหรือชุดคำสั่งมอดูล ที่ต้องการขอเปิดออกจากระบบ
- ช่องเวอร์ชัน เป็นช่องที่ใช้สำหรับเลือกเวอร์ชันของ โปรแกรมหรือชุดคำสั่งมอดูลที่ ต้องการขอเปิดออกจากระบบ โดยจะต้องป้อนชื่อ โปรแกรมหรือชุดคำสั่งมอดูลในช่องป้อนชื่อโปรแกรมก่อน

- ปุ่มดูล่วงหน้า เมื่อกดปุ่มหลังจากที่ได้ป้อนชื่อโปรแกรมหรือชุดคำสั่งมอดูลที่ต้องการขอเบิกแล้ว โปรแกรมจะทำการนำชุดคำสั่งที่อยู่ในระบบ มาแสดงในช่องแสดงชุดคำสั่งในโปรแกรมที่เบิก เพื่อให้ผู้ใช้ได้ตรวจทานก่อนเบิกจริง
- ปุ่มขอเบิก เมื่อกดปุ่มหลังจากที่ได้ป้อนชื่อโปรแกรมหรือชุดคำสั่งมอดูล ที่ต้องการขอเบิกแล้ว โปรแกรมจะทำการนำชุดคำสั่งที่อยู่ในระบบ มาจัดเก็บลงในแฟ้ม
- ตัวเลือกจุดประสงค์ เป็นตัวเลือกเพื่อระบุจุดประสงค์ที่ขอเบิกโปรแกรมหรือชุดคำสั่งโปรแกรมไปใช้งาน มีตัวเลือกดังต่อไปนี้
  - ◆ Edit เป็นตัวเลือกที่ระบุจุดประสงค์ว่าต้องการเบิกโปรแกรมไปทำการแก้ไข
  - ◆ Read Only เป็นตัวเลือกที่ระบุจุดประสงค์ว่าต้องการเบิกโปรแกรมไปเพียงเพื่ออ้างอิงหรือตรวจสอบเท่านั้น ไม่มีแก้ไข
  - ◆ Install เป็นตัวเลือกที่ระบุจุดประสงค์ว่าต้องการเบิกโปรแกรมไปทำการติดตั้งเพื่อใช้งานที่ลูกค้า ซึ่งจะปรากฏหน้าจอให้ป้อนรายละเอียดเกี่ยวกับลูกค้าที่จะนำไปลง ดังรูปที่ 3.21
- ช่องแสดงชุดคำสั่งในโปรแกรมที่เบิก เป็นช่องที่ใช้แสดงชุดคำสั่งในโปรแกรมหรือชุดคำสั่งมอดูลในระบบที่ต้องการนำมาสร้างเป็นแฟ้ม

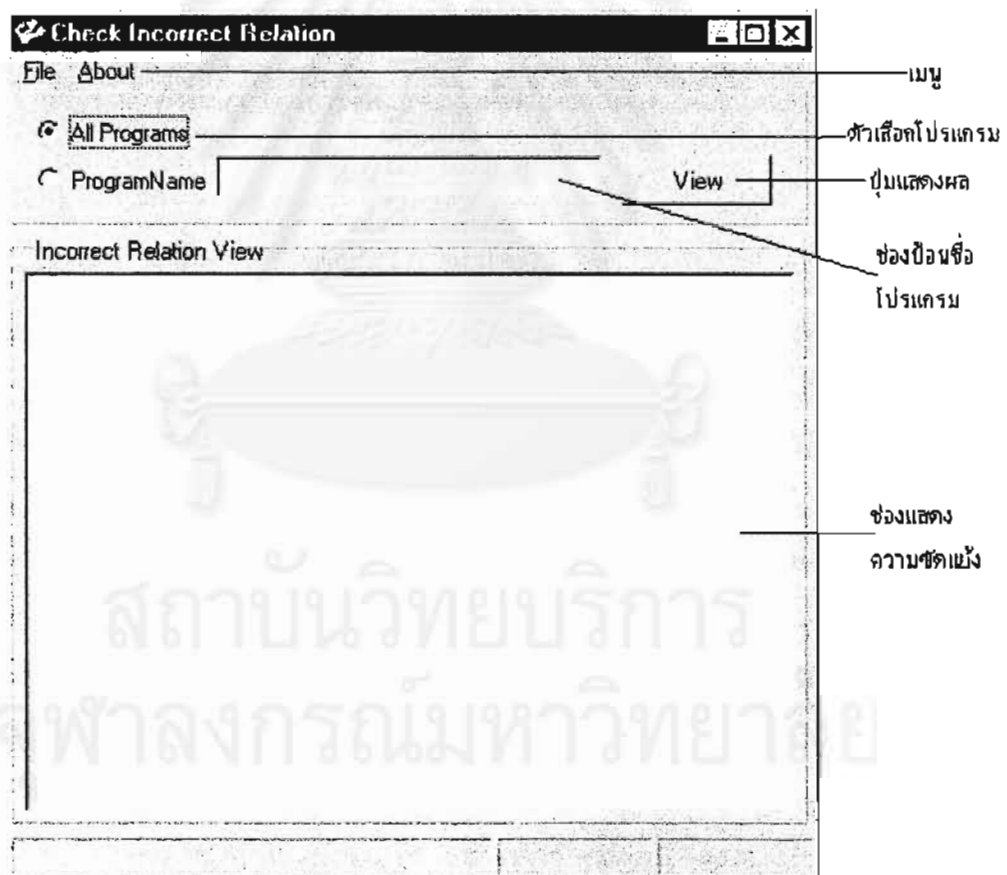


รูปที่ 3.20 หน้าจอแสดงประวัติการแก้ไขของโปรแกรมหรือชุดคำสั่งมอดูล



รูปที่ 3.21 หน้าจอป้อนรายละเอียดเกี่ยวกับลูกค้า

3.4.3 โปรแกรมแสดงความสัมพันธ์ที่ขัดแย้งระหว่างโปรแกรมหรือชุดคำสั่งมอดูล เป็นโปรแกรมที่ทำหน้าที่แสดงความสัมพันธ์ที่ขัดแย้งของโปรแกรมหรือชุดคำสั่งมอดูลที่จัดเก็บไว้ในระบบจัดการเวอร์ชันชุดคำสั่ง เพื่อให้สามารถตรวจสอบและแก้ไขเปลี่ยนแปลงให้ถูกต้อง โดยมีหน้าจอโปรแกรมดังแสดงในรูปที่ 3.22 และมีรายละเอียดดังนี้



รูปที่ 3.22 หน้าจอแสดงความสัมพันธ์ที่ขัดแย้งระหว่างโปรแกรมหรือชุดคำสั่งมอดูล

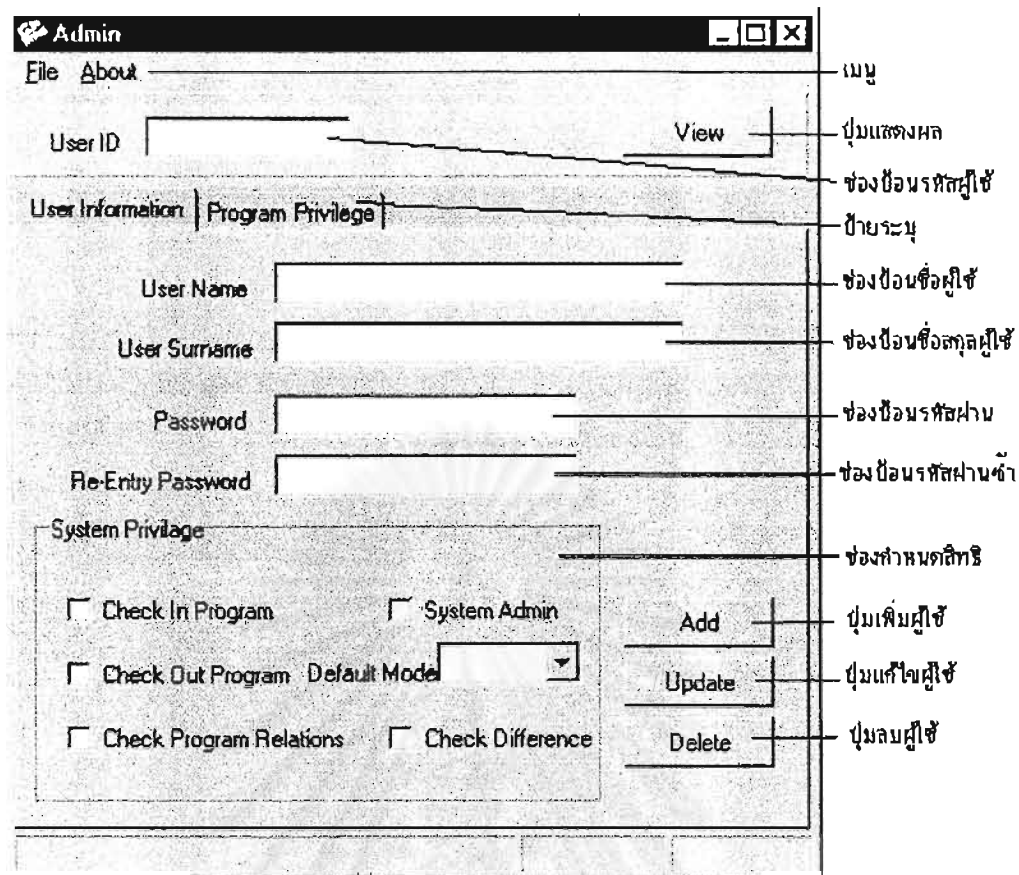
- เมนู ประกอบด้วยรายการคำสั่งต่างๆ โดยจะเรียงอยู่ในแนวนอนด้านบนของวินโดวส์ รายการคำสั่ง โปรแกรมนี้มีดังต่อไปนี้
  - ◆ File เป็นรายการคำสั่งเกี่ยวกับเพิ่มข้อมูล โดยมีเมนูย่อยดังต่อไปนี้
    - Exit เป็นรายการเลือกสำหรับสั่งเลิกงาน
  - ◆ About เป็นรายการคำสั่งสำหรับแสดงรายละเอียดของโปรแกรม
- ตัวเลือกโปรแกรม เป็นตัวเลือกที่มีเพื่อให้เลือกแสดงความสัมพันธ์ที่ขัดแย้งทั้งหมดในระบบ หรือให้แสดงเฉพาะความสัมพันธ์ที่ขัดแย้งของโปรแกรมที่ป้อนไว้ในช่องชื่อโปรแกรม
- ช่องป้อนชื่อโปรแกรม เป็นช่องที่ใช้สำหรับป้อนชื่อโปรแกรมหรือชุดคำสั่งมอดูล ที่ต้องการแสดงความสัมพันธ์ที่ขัดแย้งกันในระบบ
- ปุ่มแสดงผล เมื่อกดปุ่มหลังจากที่ได้ป้อนชื่อโปรแกรมหรือชุดคำสั่งมอดูลที่ต้องการแสดงความสัมพันธ์ที่ขัดแย้งแล้ว โปรแกรมจะนำรายการความสัมพันธ์ที่ขัดแย้งกันในระบบ มาแสดงในช่องแสดงความขัดแย้ง
- ช่องแสดงความขัดแย้ง เป็นช่องที่ใช้แสดงความสัมพันธ์ที่ขัดแย้งระหว่างโปรแกรมหรือชุดคำสั่งมอดูลต้องการให้แสดง กับ โปรแกรมหรือชุดคำสั่งมอดูลอื่นที่จัดเก็บไว้ในระบบ

#### 3.4.4 โปรแกรมกำหนดผู้ใช้ระบบและสิทธิในการทำงาน

เป็นโปรแกรมที่ใช้ในการกำหนดผู้ใช้ระบบและสิทธิในการทำงาน โดยสามารถกำหนดได้ว่าผู้ใช้ระบบสามารถใช้งานโปรแกรมใดของระบบได้ และสามารถกำหนดได้ว่าผู้ใช้ระบบรายใดสามารถเข้าถึงโปรแกรมหรือชุดคำสั่งมอดูลใดที่จัดเก็บไว้ในระบบได้บ้าง โดยมีหน้าจอโปรแกรมดังแสดงในรูปที่ 3.23 และมีรายละเอียดดังนี้

- เมนู ประกอบด้วยรายการคำสั่งต่างๆ โดยจะเรียงอยู่ในแนวนอนด้านบนของวินโดวส์ รายการคำสั่ง โปรแกรมนี้มีดังต่อไปนี้
  - ◆ File เป็นรายการคำสั่งเกี่ยวกับเพิ่มข้อมูล โดยมีเมนูย่อยดังต่อไปนี้
    - Exit เป็นรายการเลือกสำหรับสั่งเลิกงาน
  - ◆ About เป็นรายการคำสั่งสำหรับแสดงรายละเอียดของโปรแกรม
- ช่องป้อนรหัสผู้ใช้ เป็นช่องที่ใช้สำหรับรหัสผู้ใช้ระบบที่ต้องการเพิ่ม, แก้ไขสิทธิ หรือลบออกจากระบบ
- ปุ่มแสดงผล เมื่อกดปุ่มหลังจากที่ได้ป้อนรหัสผู้ใช้ที่ต้องการที่ต้องการเพิ่ม, แก้ไขสิทธิ หรือลบออกจากระบบ มาแสดง



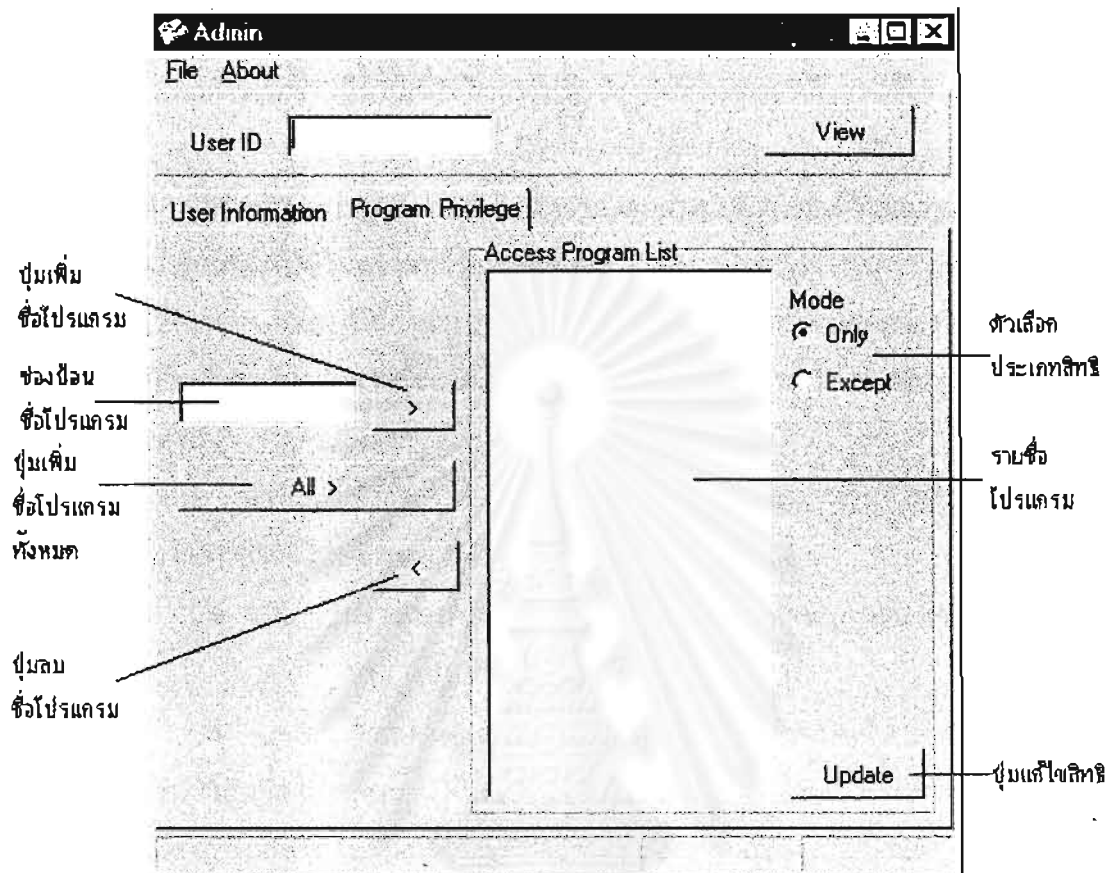


รูปที่ 3.23 หน้าจอโปรแกรมกำหนดผู้ใช้ระบบ ส่วนสิทธิในการทำงาน

- ป้ายระบุ ใช้แบ่งการแก้ไขออกเป็น ส่วนข้อมูลผู้ใช้ระบบ และส่วนสิทธิในการเข้าถึงชุดคำสั่ง โปรแกรมที่จัดเก็บในระบบ
- ช่องป้อนชื่อผู้ใช้ สำหรับป้อนชื่อของผู้ใช้งานระบบ
- ช่องป้อนชื่อสกุลผู้ใช้ สำหรับป้อนชื่อสกุลของผู้ใช้งานระบบ
- ช่องป้อนรหัสผ่าน สำหรับป้อนกำหนดรหัสผ่านของผู้ใช้งานระบบ
- ช่องป้อนรหัสผ่านซ้ำ สำหรับป้อนกำหนดรหัสผ่านของผู้ใช้งานระบบ ทวนอีกรอบ
- ช่องกำหนดสิทธิ ใช้กำหนดสิทธิการเรียกใช้โปรแกรมของ ระบบจัดการเวอร์ชันชุดคำสั่ง
- ปุ่มเพิ่มผู้ใช้ ใช้ในกรณีเพิ่มผู้ใช้ในระบบ
- ปุ่มแก้ไขผู้ใช้ ใช้ในกรณีแก้ไขข้อมูลผู้ใช้ในระบบ
- ปุ่มลบผู้ใช้ ใช้ในกรณีลบผู้ใช้ออกจากระบบ



สำหรับหน้าจอของส่วนสิทธิในการเข้าถึงชุดคำสั่ง โปรแกรมที่จัดเก็บในระบบ มีหน้าจอโปรแกรมดังแสดงในรูปที่ 3.24 และมีรายละเอียดดังนี้



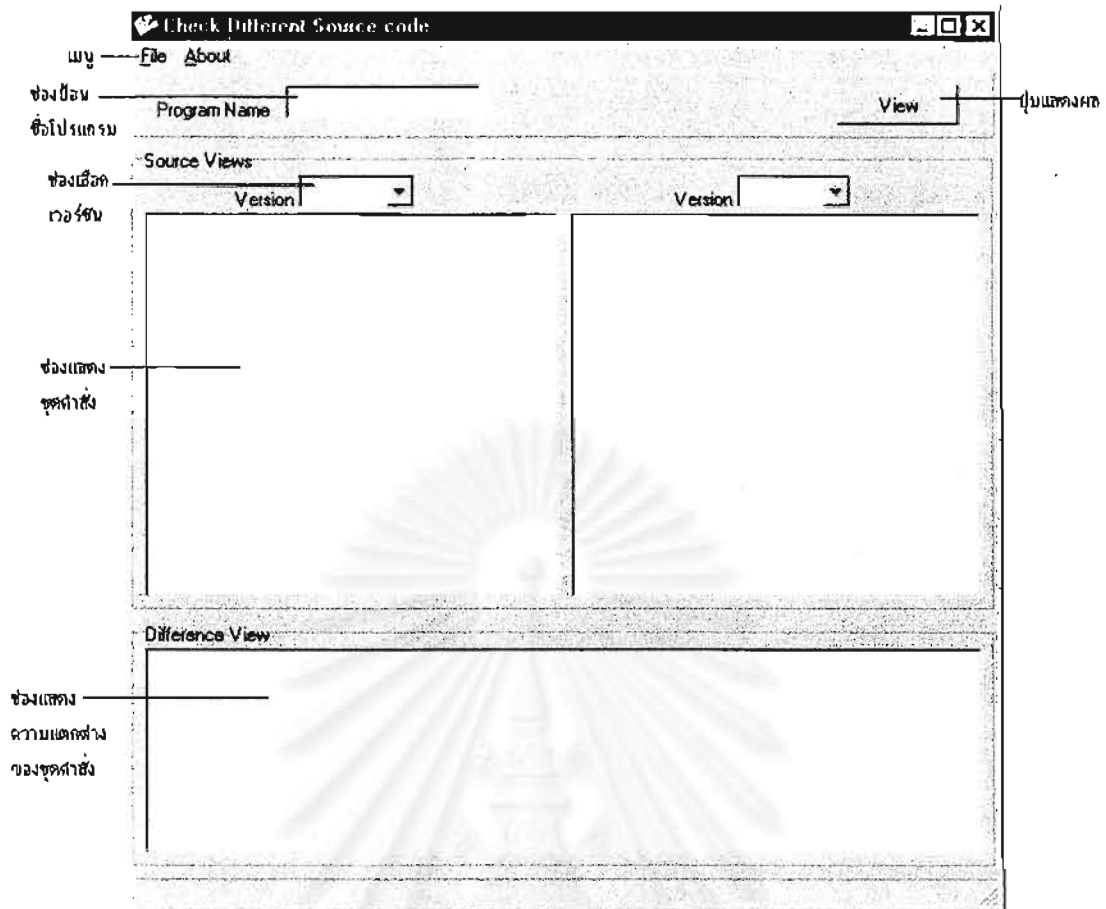
รูปที่ 3.24 หน้าจอโปรแกรมกำหนดผู้ใช้ระบบ ส่วนสิทธิในการเข้าถึงชุดคำสั่งที่จัดเก็บ

- ปุ่มเพิ่มชื่อโปรแกรม ใช้ในกรณีเพิ่มชื่อ โปรแกรมไว้ในรายชื่อ โปรแกรมที่ต้องการ กำหนดสิทธิ
- ช่องป้อนชื่อโปรแกรม ใช้ในกรณีป้อนชื่อโปรแกรมที่ ต้องการเพิ่มไว้ในรายชื่อ โปรแกรมที่ต้องการกำหนดสิทธิ
- ปุ่มเพิ่มชื่อ โปรแกรมทั้งหมด ใช้ในกรณีเพิ่มชื่อ โปรแกรมทั้งหมดที่จัดเก็บอยู่ในระบบ ไว้ในรายชื่อ โปรแกรมที่ต้องการกำหนดสิทธิ
- ปุ่มลบชื่อ โปรแกรม ใช้ในกรณีลบชื่อ โปรแกรมออกจากรายชื่อ โปรแกรมที่ต้องการ กำหนดสิทธิ
- รายชื่อ โปรแกรม เป็นช่องที่แสดงรายชื่อ โปรแกรมที่ต้องการกำหนดสิทธิ

- ตัวเลือกประเภทสิทธิ เป็นตัวเลือกที่ใช้กำหนดว่ารายชื่อโปรแกรมที่ปรากฏเป็นโปรแกรมที่ผู้ใช้มีสิทธิในการเข้าใช้งาน (ตัวเลือก only) หรือผู้ใช้มีสิทธิในการเข้าถึงโปรแกรมทั้งหมดในระบบยกเว้น โปรแกรมที่ปรากฏในรายชื่อโปรแกรม (ตัวเลือก Except)
- ปุ่มแก้ไขสิทธิ เป็นปุ่มที่ใช้ในการปรับปรุงสิทธิในการเข้าถึงของผู้ใช้

3.4.5 โปรแกรมตรวจสอบและแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูล เป็นโปรแกรมที่ใช้ในตรวจสอบและแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูลที่จัดเก็บไว้ในระบบ โดยผู้ใช้สามารถกำหนดว่าต้องการให้แสดงความแตกต่างระหว่างเวอร์ชันหนึ่งเปรียบเทียบกับเวอร์ชันอีกเวอร์ชันหนึ่งได้ โดยมีหน้าจอ โปรแกรมดังแสดงในรูปที่ 3.25 และ มีรายละเอียดดังนี้

- เมนู ประกอบด้วยรายการคำสั่งต่างๆ โดยจะเรียงอยู่ในแนวนอนด้านบนของวินโดวส์ รายการคำสั่งโปรแกรมนี้ มีดังต่อไปนี้
  - ◆ File เป็นรายการคำสั่งเกี่ยวกับเพิ่มข้อมูล โดยมีเมนูย่อยดังต่อไปนี้
    - Exit เป็นรายการเลือกสำหรับสั่งเลิกงาน
    - ◆ About เป็นรายการคำสั่งสำหรับแสดงรายละเอียดของโปรแกรม
- ช่องป้อนชื่อโปรแกรม เป็นช่องที่ใช้สำหรับป้อนชื่อโปรแกรมหรือชุดคำสั่งมอดูล ที่ต้องการแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูลที่จัดเก็บในระบบ
- ช่องเลือกเวอร์ชัน เป็นช่องที่ใช้สำหรับเลือกเวอร์ชันของโปรแกรมหรือชุดคำสั่งมอดูล ที่ต้องการตรวจสอบความแตกต่าง โดยต้องเลือกเวอร์ชัน โปรแกรมหรือชุดคำสั่งมอดูล ทั้ง 2 ชุด
- ช่องแสดงชุดคำสั่ง เป็นช่องที่ใช้แสดงชุดคำสั่งในโปรแกรมหรือชุดคำสั่งมอดูล ในระบบที่ต้องการตรวจสอบความแตกต่าง
- ช่องแสดงความแตกต่างของชุดคำสั่ง เป็นช่องที่ใช้แสดงความแตกต่างระหว่างชุดคำสั่ง 2 ชุดที่ต้องการให้แสดงเปรียบเทียบ
- ปุ่มแสดงผล เมื่อกดปุ่มหลังจากที่ได้ป้อนชื่อ โปรแกรมและเวอร์ชันของชุดคำสั่ง โปรแกรมครบถ้วนทั้ง 2 ชุด โปรแกรมจะทำการตรวจสอบและแสดงความแตกต่างในช่องแสดงความแตกต่างของชุดคำสั่ง



รูปที่ 3.25 หน้าจอโปรแกรมตรวจสอบและแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูล

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 4

### การทดสอบโปรแกรม

จากการศึกษาและออกแบบส่วนประสานกับผู้ใช้ และการออกแบบฐานข้อมูลสำหรับจัดเก็บ เมื่อนำมาสร้างเป็น ระบบจัดการเวอร์ชันชุดคำสั่ง แล้วระบบที่ได้ประกอบด้วย โปรแกรมดังนี้

- โปรแกรมจัดเก็บแฟ้ม โปรแกรมหรือชุดคำสั่งมอดูล (Pcheckin.exe)
- โปรแกรมขอเบิก โปรแกรมหรือชุดคำสั่งมอดูล (Pcheckout.exe)
- โปรแกรมแสดงความสัมพันธ์ที่ขัดแย้งระหว่าง โปรแกรมหรือชุดคำสั่งมอดูล (Pcheckrel.exe)
- โปรแกรมกำหนดผู้ใช้ระบบและสิทธิในการทำงาน (Padmin.exe)
- โปรแกรมแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูล (Pcheckdiff.exe)

ซึ่งโปรแกรมดังกล่าว เขียนขึ้นด้วย เคลไฟ 4 และใช้ ไมโครซอฟต์แอกเซส 97 เป็นฐานข้อมูลสำหรับจัดเก็บโปรแกรมหรือชุดคำสั่งมอดูลในการทดสอบระบบนี้

#### 4.1 สภาพแวดล้อมในการทดสอบโปรแกรม

ระบบจัดการเวอร์ชันชุดคำสั่ง ได้ทำการทดสอบโปรแกรม ภายใต้สภาพแวดล้อมดังนี้

##### 4.1.1 เครื่องคอมพิวเตอร์ที่ใช้ในการทดสอบ

- หน่วยประมวลผลกลาง เพนเทียมทู 233 MHz
- หน่วยความจำ 32 MB
- งานบันทึกแบบแข็งความจุ 6 GB
- ซีดีรอม 40X

##### 4.1.2 ระบบปฏิบัติการวิน โควส์ 95 Thai Edition

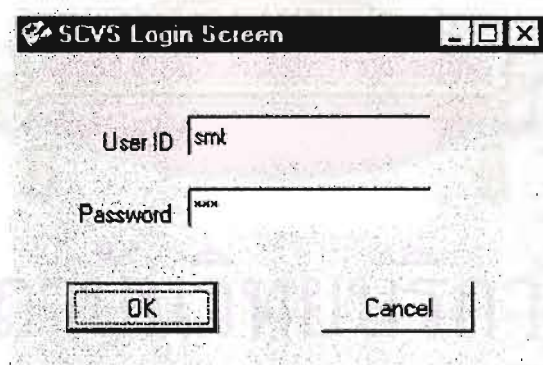
##### 4.1.3 ระบบจัดการฐานข้อมูล Microsoft Access 97

- 4.1.4 โปรแกรมจัดการการเชื่อมต่อฐานข้อมูล BDE Administrator รุ่น 5.00 สำหรับ  
โปรแกรมประยุกต์ที่เขียนขึ้น โดย Delphi4

## 4.2 การทดสอบโปรแกรม

ในการทดสอบโปรแกรมที่สร้างขึ้นนี้ เพื่อให้สามารถทดสอบได้สะดวกขึ้น จึงได้แบ่งการทดสอบออกเป็น 2 ส่วนดังนี้

- 4.2.1 การทดสอบการจัดเก็บและขอเบิกโปรแกรม ได้ทำการทดสอบจัดเก็บชุดคำสั่งโปรแกรมภาษาอินฟอร์มิกซ์ โฟท์จีแอล โดยการทดสอบโปรแกรมมีขั้นตอนดังต่อไปนี้
- 4.2.1.1 ทำการเรียกโปรแกรมจัดเก็บเพิ่มโปรแกรมหรือชุดคำสั่งมอดูล ซึ่งจะปรากฏหน้าต่างสำหรับป้อนรหัสผู้ใช้และรหัสผ่าน ทำการป้อนรหัสผู้ใช้และรหัสผ่านดังแสดงในรูปที่ 4.1
- 4.2.1.2 เลื่อนเมาส์ไปที่ปุ่มตกลงการทำงาน กดปุ่มเมาส์ด้านซ้าย ซึ่งโปรแกรมจะทำการตรวจสอบสิทธิของผู้ใช้ระบบจากระหัสผู้ใช้และรหัสผ่านกับตารางรายชื่อผู้ใช้ระบบ ก่อนเข้าสู่หน้าต่างหลัก ของโปรแกรมจัดเก็บเพิ่มโปรแกรมหรือชุดคำสั่งมอดูล



รูปที่ 4.1 หน้าต่างป้อนรหัสผู้ใช้และรหัสผ่าน ขณะโปรแกรมทำงาน

- 4.2.1.3 เตรียมเพิ่มชุดคำสั่งโปรแกรมภาษาอินฟอร์มิกซ์ โฟท์จีแอล ชื่อ Calcomm.4gl ดังในรูปที่ 4.2

```

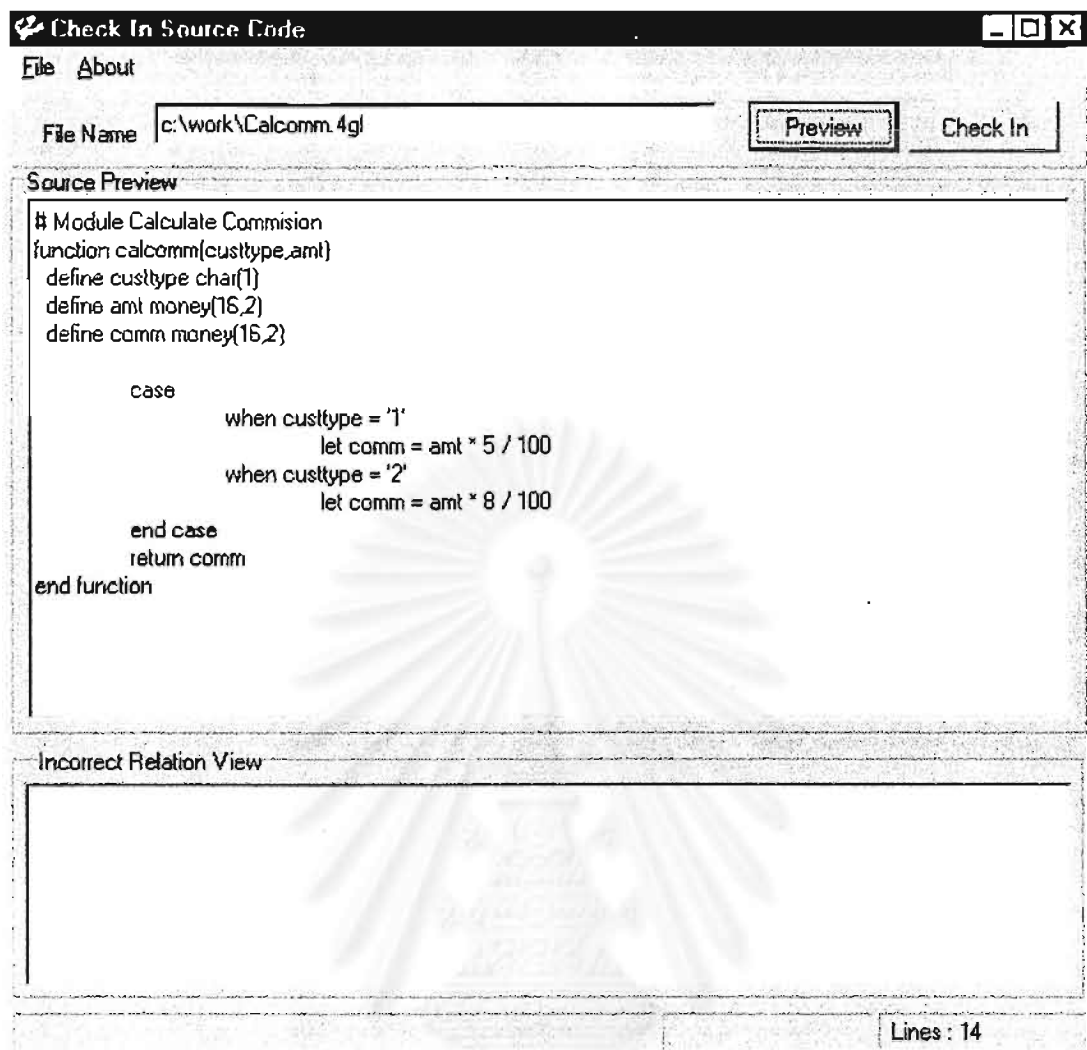
# Module Calculate Commision
function calcomm(custtype,amt)
  define custtype char(1)
  define amt money(16,2)
  define comm money(16,2)

  case
    when custtype = '1'
      let comm = amt * 5 / 100
    when custtype = '2'
      let comm = amt * 8 / 100
  end case
  return comm
end function

```

รูปที่ 4.2 เพิ่มชุดคำสั่งโปรแกรม Calcomm.4gl ที่ต้องการจัดเก็บ

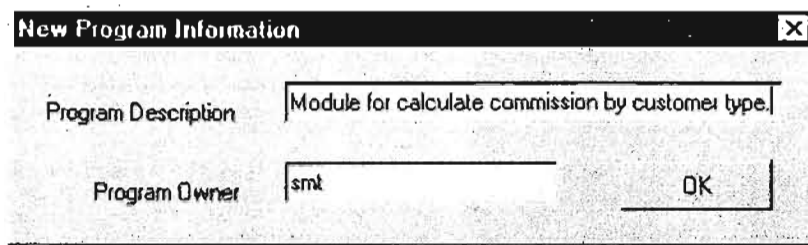
- 4.2.1.4 ป้อนชื่อเพิ่มชุดคำสั่งโปรแกรม Calcomm.4gl ลงในช่องป้อนชื่อเพิ่ม  
 เลื่อนเมาส์ไปที่ปุ่มดูล่วงหน้า กดปุ่มเมาส์ด้านซ้าย เพื่อตรวจสอบว่าเป็น  
 เพิ่มชุดคำสั่งโปรแกรมที่ต้องการจัดเก็บเข้าสู่ระบบ ซึ่ง โปรแกรมจะนำเอา  
 ชุดคำสั่งโปรแกรมในแฟ้ม มาแสดงในช่องแสดงชุดคำสั่งในแฟ้มดังในรูป  
 ที่ 4.3



รูปที่ 4.3 หน้าจอแสดงการจับเก็บชุดคำสั่งโปรแกรม Calcomm.4gl

4.2.1.5 ทำการจับเก็บโปรแกรม โดยเลื่อนเมาส์ไปที่ปุ่มจับเก็บ กดปุ่มเมาส์ด้านซ้าย เพื่อให้โปรแกรมเริ่มทำการจับเก็บ ซึ่งในกรณีนี้ เป็นการจับเก็บชุดคำสั่งโปรแกรม Calcomm.4gl เข้าสู่ระบบเป็นครั้งแรก โปรแกรมจะทำการแสดงหน้าต่างให้ป้อนคำอธิบายอย่างย่อ เพื่อให้ป้อนคำอธิบาย ดังในรูปที่

4.4

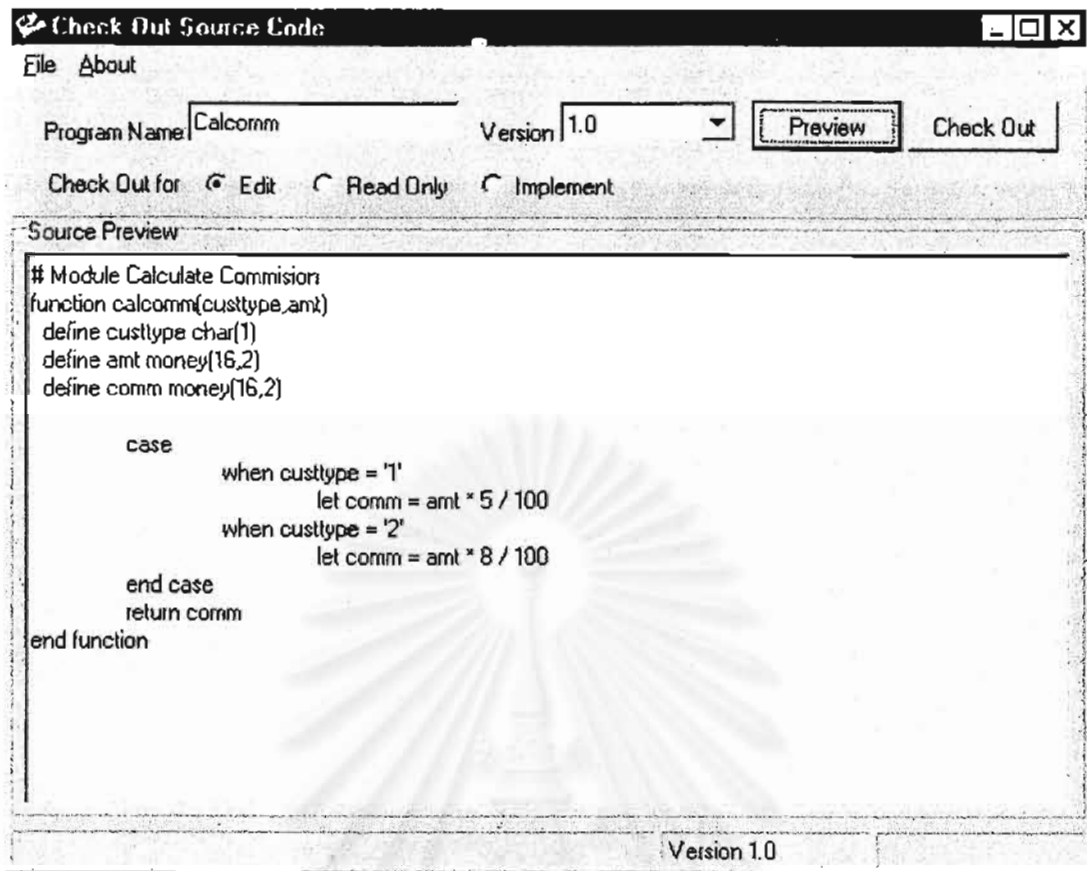


รูปที่ 4.4 แสดงการป้อนหน้าต่างป้อนคำอธิบายอย่างย่อ

- 4.2.1.6 เลื่อนเมาส์ไปที่ปุ่มตกลงการทำงาน กดปุ่มเมาส์ด้านซ้าย เมื่อทำการป้อนคำอธิบายอย่างย่อเสร็จเรียบร้อยแล้ว หลังจากนั้นโปรแกรมก็จะจัดเก็บชุดคำสั่งโปรแกรม Calcomm.4gl เข้าไปในระบบ เมื่อเสร็จเรียบร้อยแล้วให้ออกจากโปรแกรมด้วยการเลื่อนเมาส์ไปที่ปุ่มปิดวินโดวส์ กดปุ่มเมาส์ด้านซ้าย
- 4.2.1.7 ทำการเรียกโปรแกรมขอเบิกโปรแกรมหรือชุดคำสั่งมอดูล ซึ่งจะปรากฏหน้าต่างสำหรับป้อนรหัสผู้ใช้และรหัสผ่าน ทำการป้อนรหัสผู้ใช้และรหัสผ่าน ป้อนชื่อโปรแกรมที่ต้องการขอเบิกออกจากระบบในช่องป้อนชื่อโปรแกรม เลือกเวอร์ชันที่ต้องการขอเบิกจากช่องเวอร์ชัน เลื่อนเมาส์ไปที่ปุ่มดูล่วงหน้า กดปุ่มเมาส์ด้านซ้าย เพื่อตรวจสอบว่าเป็นชุดคำสั่งโปรแกรมที่ต้องการขอเบิก โปรแกรมจะนำชุดคำสั่งโปรแกรมออกมาแสดงในช่องแสดงชุดคำสั่ง ดังรูปที่ 4.5
- 4.2.1.8 เลื่อนเมาส์ไปที่ปุ่มขอเบิก กดปุ่มเมาส์ด้านซ้าย จะปรากฏหน้าต่างให้ป้อนชื่อแฟ้มที่ต้องการใช้เก็บชุดคำสั่งที่ขอเบิกจากระบบ ดังรูปที่ 4.6 ให้ทำการป้อนชื่อแฟ้มแล้วเลื่อนเมาส์ไปที่ปุ่มจัดเก็บแฟ้ม (Save) กดปุ่มเมาส์ด้านซ้าย โปรแกรมขอเบิกก็จะทำการจัดเก็บชุดคำสั่งโปรแกรมที่ขอเบิกไว้จากระบบลงแฟ้มที่ได้ระบุไว้

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



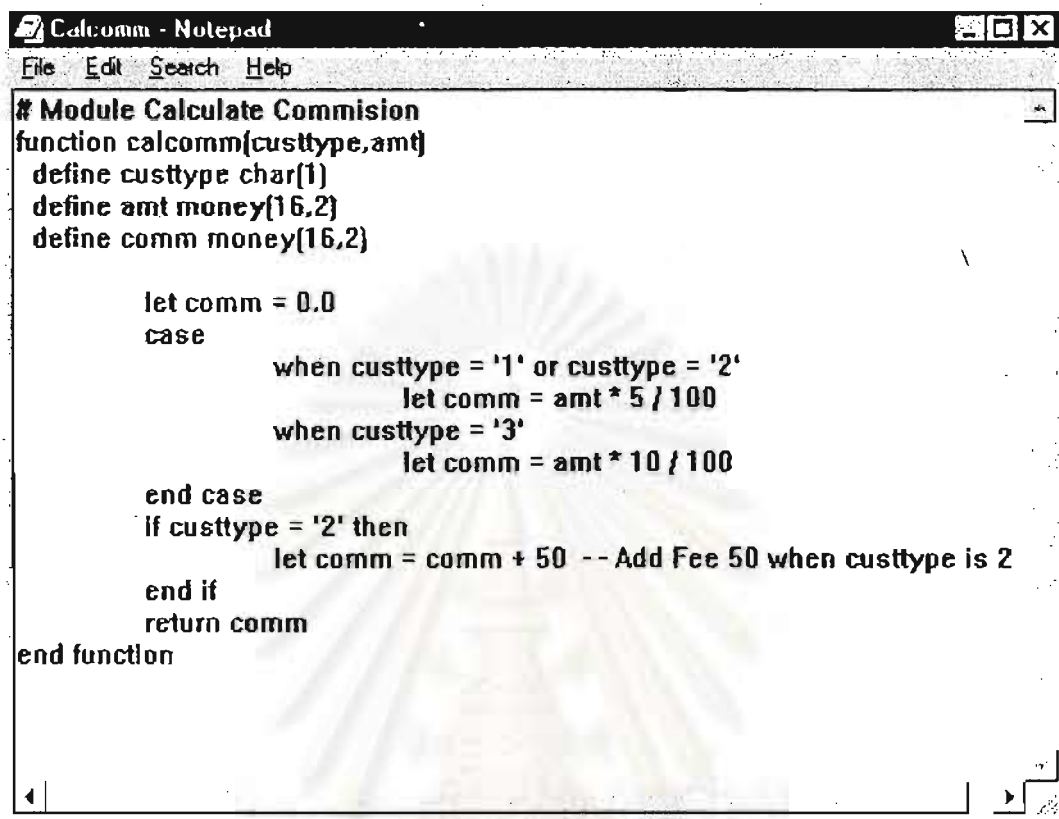


รูปที่ 4.5 หน้าจอโปรแกรมขอเบ็กโปรแกรม Calcomm เพื่อทำการแก้ไข



รูปที่ 4.6 หน้าต่างป้อนชื่อแฟ้มสำหรับจัดเก็บชุดคำสั่งที่ขอเบ็กจากระบบ

- 4.2.1.9 ทำแก้ไขชุดคำสั่งโปรแกรม Calcomm.4gl โดยโปรแกรมบรรณาธิกรณ  
เช่น Notepad เป็นต้น ดังในรูปที่ 4.7

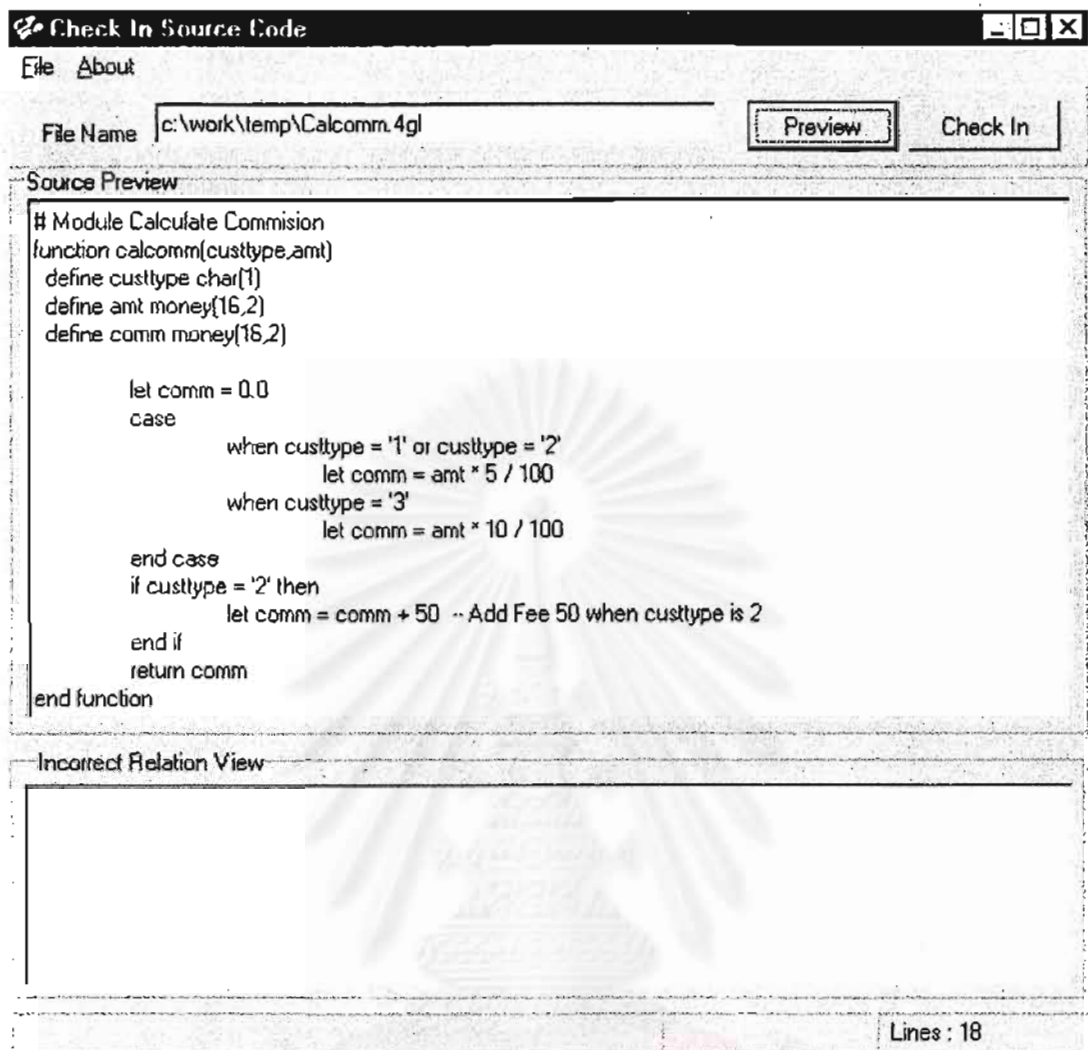


```
# Module Calculate Commision
function calcomm{custtype,amt}
define custtype char(1)
define amt money(16,2)
define comm money(16,2)

let comm = 0.0
case
when custtype = '1' or custtype = '2'
let comm = amt * 5 / 100
when custtype = '3'
let comm = amt * 10 / 100
end case
if custtype = '2' then
let comm = comm + 50 -- Add Fee 50 when custtype is 2
end if
return comm
end function
```

รูปที่ 4.7 ชุดคำสั่งโปรแกรม Calcomm.4gl ที่ถูกแก้ไขโดยใช้ Notepad

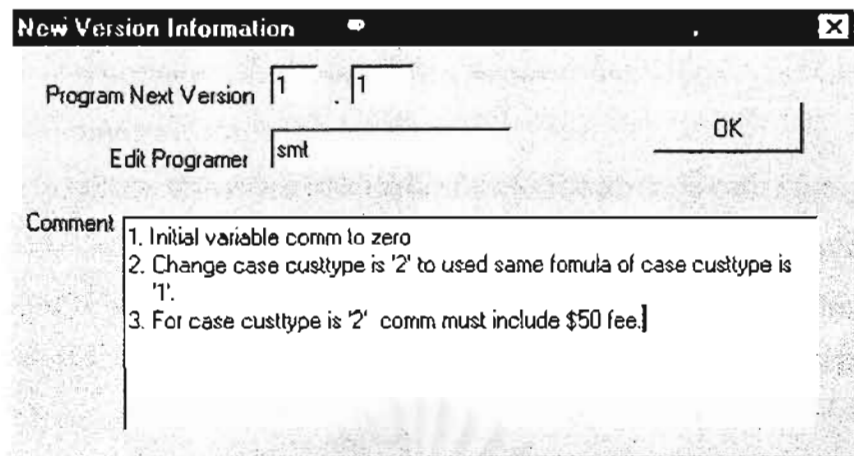
- 4.2.1.10 ทำการเรียกโปรแกรมจัดเก็บแฟ้ม โปรแกรมหรือชุดคำสั่งมอดูลอีกครั้ง  
เพื่อทำการจัดเก็บชุดคำสั่งโปรแกรม Calcomm.4gl ที่ได้รับการแก้ไขเพิ่ม  
เติม โดยเริ่มจากทำการป้อนรหัสผู้ใช้และรหัสผ่าน ลงในหน้าต่างสำหรับ  
ป้อนรหัสผู้ใช้และรหัสผ่าน เลื่อนเมาส์ไปที่ปุ่มตกลงการทำงาน กดปุ่ม  
เมาส์ด้านซ้าย เข้าสู่หน้าต่างหลัก ของ โปรแกรมจัดเก็บแฟ้มโปรแกรมหรือ  
ชุดคำสั่งมอดูล ป้อนชื่อแฟ้มชุดคำสั่งโปรแกรม Calcomm.4gl ลงในช่อง  
ป้อนชื่อแฟ้ม เลื่อนเมาส์ไปที่ปุ่มดูล่วงหน้า กดปุ่มเมาส์ด้านซ้าย เพื่อตรวจ  
สอบว่าเป็นแฟ้มชุดคำสั่ง โปรแกรมที่ต้องการจัดเก็บเข้าสู่ระบบ ซึ่ง  
โปรแกรมจะนำเอาชุดคำสั่งโปรแกรมในแฟ้ม มาแสดงในช่องแสดงชุดคำ  
สั่งในแฟ้ม ดังในรูปที่ 4.8



รูปที่ 4.8 หน้าจอจัดเก็บชุดคำสั่งโปรแกรม Calcomm.4gl ที่ได้รับการแก้ไขแล้ว

4.2.1.11 ทำการจัดเก็บโปรแกรม โดยเลื่อนเมาส์ไปที่ปุ่มจัดเก็บ กดปุ่มเมาส์ด้านซ้าย เพื่อให้โปรแกรมเริ่มทำการจัดเก็บ ซึ่งในกรณีนี้ เป็นการจัดเก็บชุดคำสั่งโปรแกรม Calcomm.4gl เข้าสู่ระบบซึ่งเดิมมีจัดเก็บอยู่แล้ว โปรแกรมจะทำการแสดงหน้าต่างให้ป้อนหมายเหตุในการแก้ไข เพื่อให้ป้อนคำอธิบาย พร้อมทั้งแสดงเลขเวอร์ชันใหม่ของชุดคำสั่งโปรแกรมที่กำลังทำการจัดเก็บ ดังในรูปที่ 4.9

4.2.1.12 เมื่อทำการป้อนหมายเหตุในการแก้ไขเสร็จเรียบร้อยแล้ว ให้เลื่อนเมาส์ไปที่ปุ่มตกลงการทำงาน กดปุ่มเมาส์ด้านซ้าย เพื่อทำการจัดเก็บ เสร็จแล้วออกจากโปรแกรม



รูปที่ 4.9 แสดงการป้อนหน้าต่างหมายเหตุการแก้ไขโปรแกรมหรือชุดคำสั่งมอดูล

4.2.1.13 ทำการเรียกใช้โปรแกรมขอเบิกโปรแกรมหรือชุดคำสั่งมอดูลอีกครั้ง โดยทำการขอเบิกชุดคำสั่งโปรแกรม Calcomm ย้อนหลังเป็นเวอร์ชัน 1.0 ซึ่งจะได้เพิ่มชุดคำสั่งโปรแกรม Calcomm.4gl ดังรูปที่ 4.10

```
# Module Calculate Commision
function calcomm(custtype,amt)
  define custtype char(1)
  define amt money(16,2)
  define comm money(16,2)

  case
    when custtype = '1'
      let comm = amt * 5 / 100
    when custtype = '2'
      let comm = amt * 8 / 100
  end case
  return comm
end function
```

รูปที่ 4.10 เพิ่มชุดคำสั่งโปรแกรม Calcomm.4gl เวอร์ชัน 1.0

4.2.2 การทดสอบการตรวจสอบและแสดงความสัมพันธ์ระหว่างฟังก์ชันที่ขัดแย้งกัน ได้ทำการทดสอบจัดเก็บชุดคำสั่งโปรแกรมภาษาอินฟอร์มิทซ์ โฟท์จีแอล โดยการทดสอบโปรแกรมมีขั้นตอนดังต่อไปนี้

4.2.2.1 ทำการเรียกโปรแกรมจัดเก็บเพิ่มโปรแกรมหรือชุดคำสั่งมอดูล ซึ่งจะปรากฏหน้าต่างสำหรับป้อนรหัสผู้ใช้และรหัสผ่าน ทำการป้อนรหัสผู้ใช้และรหัสผ่าน เลื่อนเมาส์ไปที่ปุ่มตกลงการทำงาน กดปุ่มเมาส์ด้านซ้ายระบบ เพื่อเข้าสู่หน้าต่างหลักของโปรแกรมจัดเก็บเพิ่มโปรแกรมหรือชุดคำสั่งมอดูล

4.2.2.2 เตรียมเพิ่มชุดคำสั่งโปรแกรมภาษาอินฟอร์มิทซ์ โฟท์จีแอล ชื่อ CustOrder.4gl ดังในรูปที่ 4.11

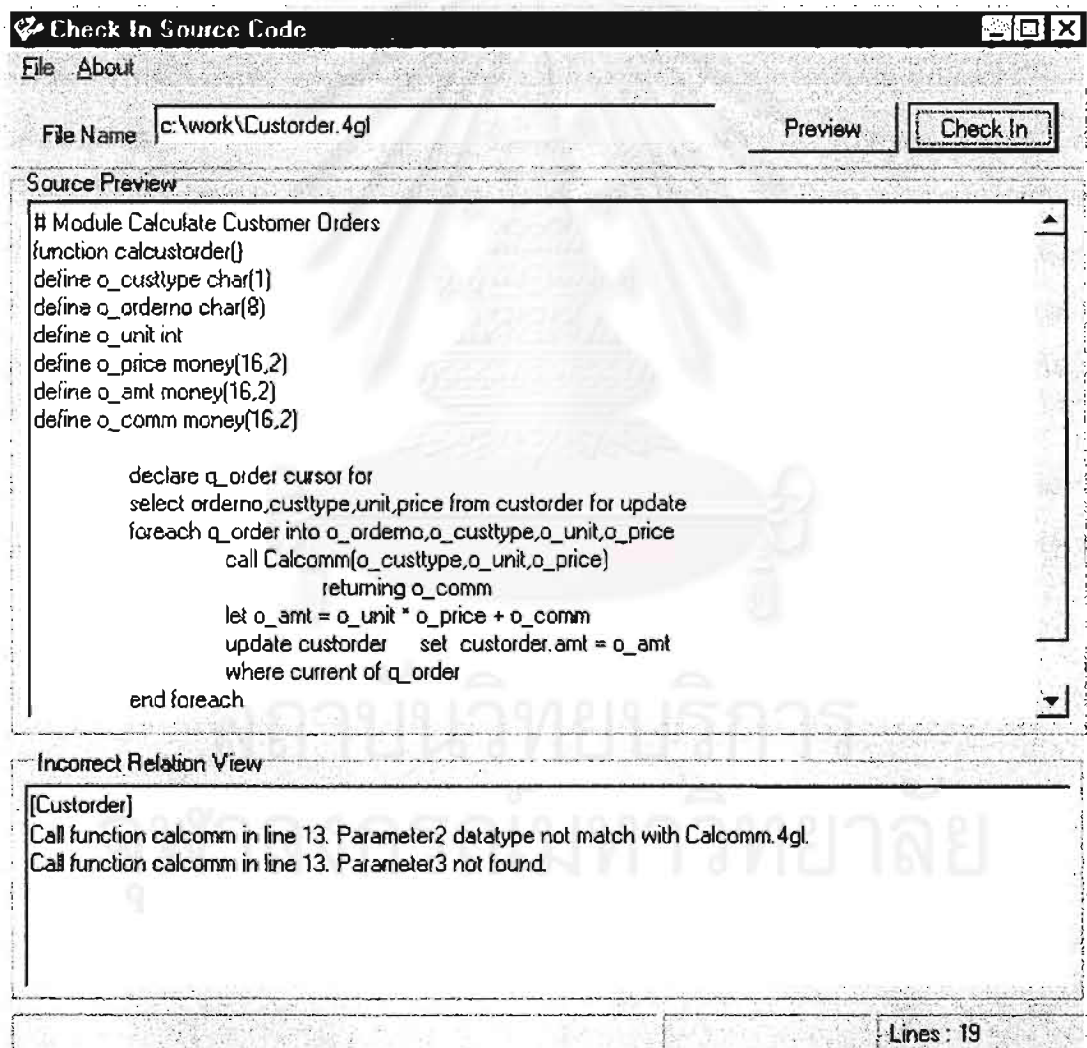
```
# Module Calculate Customer Orders
function calcustorder()
define o_custtype char(1)
define o_orderno char(8)
define o_unit int
define o_price money(16,2)
define o_amt money(16,2)
define o_comm money(16,2)

declare q_order cursor for
select orderno,custtype,unit,price from custorder for update
foreach q_order into o_orderno,o_custtype,o_unit,o_price
call Calcomm(o_custtype,o_unit,o_price)
returning o_comm
let o_amt = o_unit * o_price + o_comm
update custorder set custorder.amt = o_amt
where current of q_order
end foreach
end function
```

รูปที่ 4.11 เพิ่มชุดคำสั่งโปรแกรม CustOrder.4gl ที่ต้องการจัดเก็บ



4.2.2.3 ทำการจัดเก็บโปรแกรม โดยเลื่อนเมาส์ไปที่ปุ่มจัดเก็บ กดปุ่มเมาส์ด้านซ้าย เพื่อให้โปรแกรมเริ่มทำการจัดเก็บ ซึ่งในกรณีนี้ เป็นการจัดเก็บชุดคำสั่งโปรแกรม Custorder.4gl เข้าสู่ระบบเป็นครั้งแรก โปรแกรมจะทำการแสดงหน้าต่างสำหรับป้อนคำอธิบายอย่างย่อ ทำการป้อนคำอธิบายอย่างย่อ เมื่อเสร็จเรียบร้อยแล้วให้เลื่อนเมาส์ไปที่ปุ่มตกลงการทำงาน กดปุ่มเมาส์ด้านซ้าย หลังจากนั้นโปรแกรมก็จะจัดเก็บชุดคำสั่งโปรแกรม Custorder.4gl เข้าไปในระบบ เมื่อตรวจพบว่ามีความสับสนระหว่างฟังก์ชันที่มีความขัดแย้ง โปรแกรมก็จะนำมาแสดงในช่องแสดงความขัดแย้ง ดังแสดงในรูปที่ 4.12



รูปที่ 4.12 หน้าจอแสดงการจัดเก็บชุดคำสั่งโปรแกรม Custorder.4gl

## บทที่ 5

### สรุปผลการวิจัยและเสนอแนะ

#### 5.1 สรุปผลการวิจัย

ระบบจัดการเวอร์ชันชุดคำสั่งนี้ ถูกสร้างขึ้น เพื่อเป็นระบบที่ใช้จัดเก็บชุดคำสั่งมอดูล หรือโปรแกรม ของโครงการการพัฒนาโปรแกรม โดยจัดเก็บในลักษณะเป็นเวอร์ชัน ช่วยให้ลดทรัพยากรที่ใช้ในการจัดเก็บชุดคำสั่งมอดูลหรือโปรแกรม ด้วยวิธีการจัดเก็บแบบเคลด้า รวมทั้งเพิ่มเติมความสามารถในการตรวจสอบความสัมพันธ์ระหว่างฟังก์ชันในส่วนของการกำหนด และเรียกใช้ฟังก์ชัน ซึ่งสามารถทำได้จากการวิเคราะห์ศัพท์ ในชุดคำสั่งโปรแกรมที่ทำการจัดเก็บเข้าสู่ระบบ โดยไม่ต้องอาศัยส่วนอื่นใดเพิ่มเติม ซึ่งนับเป็นข้อดีที่เหนือกว่าระบบจัดการเวอร์ชันชุดคำสั่งทั่วไป เพราะความสามารถดังกล่าว มีส่วนช่วยในการลดความผิดพลาด ในเรื่องของความครบถ้วนในการแก้ไขโปรแกรม โดยเฉพาะอย่างยิ่ง การแก้ไขโปรแกรมที่ส่งผลกระทบต่อฟังก์ชันโปรแกรม ที่มีใช้งานร่วมกันมากกว่าหนึ่งโปรแกรม

การทำวิจัยครั้งนี้ ได้ทำการพัฒนา ระบบจัดการเวอร์ชันชุดคำสั่ง โดยระบบนี้ทำงานภายใต้วินโดวส์ ซึ่งมีตัวประสานงานกับผู้ใช้ที่ง่ายต่อการใช้งาน โดยระบบที่พัฒนาได้นำมาทำการทดสอบสามารถสรุปผลได้ดังนี้

##### 5.1.1 เป็นระบบที่ทำงานภายใต้

- เครื่องไมโครคอมพิวเตอร์ ที่มีซีพียู เพนเทียมทู ขึ้นไป และมีหน่วยความจำอย่างน้อย 32 เมกะไบต์
- ระบบปฏิบัติการ ไมโครซอฟต์วินโดวส์ 95

##### 5.1.2 สามารถจัดเก็บแฟ้ม โปรแกรมหรือชุดคำสั่งมอดูลที่เขียนขึ้นด้วยภาษา อินฟอร์มิกซ์ โฟทจีแอล ลงในระบบจัดการเวอร์ชันชุดคำสั่ง โดยจัดเก็บในลักษณะของเวอร์ชัน

##### 5.1.3 สามารถนำโปรแกรมหรือชุดคำสั่งมอดูลเวอร์ชันที่ต้องการ จากระบบจัดการเวอร์ชันชุดคำสั่ง ออกมาเป็นแฟ้มสำหรับใช้งาน

- 5.1.4 สามารถจัดเก็บความสัมพันธ์ระหว่างฟังก์ชัน ในส่วนของการกำหนดและเรียกใช้งานฟังก์ชัน ในแฟ้มโปรแกรมหรือชุดคำสั่งมอดูล
- 5.1.5 สามารถตรวจสอบและแสดงความสัมพันธ์ระหว่างฟังก์ชันที่ขัดแย้งกัน ของโปรแกรมหรือชุดคำสั่งมอดูล เวอร์ชันล่าสุด โดยระบบจะทำการเตือนการเรียกใช้ฟังก์ชันที่มีการส่งพารามิเตอร์ที่มีจำนวน หรือชนิดข้อมูลแตกต่างจากที่กำหนดฟังก์ชัน

## 5.2 ข้อจำกัดของระบบ

ในส่วนของการพิจารณาความสัมพันธ์ระหว่างชุดคำสั่งมอดูล ว่าชุดคำสั่งมอดูล 2 ชุดมีความสัมพันธ์กันนั้น ระบบจะทำการตรวจสอบให้เฉพาะชุดคำสั่งเวอร์ชันล่าสุดเท่านั้น และได้ใช้หลักเกณฑ์ในการพิจารณาจากการที่ชุดคำสั่งมอดูลหนึ่ง เรียกใช้งานฟังก์ชันในอีกชุดคำสั่งมอดูลหนึ่ง ดังนั้นจึงเกิดข้อจำกัดที่ว่า หากมีชุดคำสั่งมอดูลใดที่มีความสัมพันธ์กันในลักษณะอื่น เช่น มีการเรียกใช้ตัวแปรส่วนกลางร่วมกัน เป็นต้น ระบบจัดการเวอร์ชันชุดคำสั่งจะไม่สามารถตรวจพบความสัมพันธ์ดังกล่าวได้

ในการพิจารณาความสัมพันธ์ระหว่างชุดคำสั่งมอดูลนั้น จำเป็นต้องอาศัยการพิจารณาจากความสัมพันธ์ของตัวภาษา อินฟอร์มิกซ์ โฟทจีแอล ดังนั้นชุดคำสั่งมอดูลที่ต้องการทำการจัดเก็บเข้าสู่ระบบ จำเป็นที่จะต้องมีความสัมพันธ์ที่ถูกต้องตามหลักของภาษา อินฟอร์มิกซ์ โฟทจีแอล เพื่อให้การพิจารณาความสัมพันธ์ระหว่างชุดคำสั่งมอดูลสามารถทำได้ถูกต้อง

ข้อจำกัดอีกสิ่งหนึ่งของระบบก็คือ การพิจารณาความสัมพันธ์ระหว่างชุดคำสั่งมอดูล โดยอาศัยการวิเคราะห์จากชนิดข้อมูลของตัวแปรพารามิเตอร์ที่ส่งผ่านระหว่างชุดคำสั่งที่เรียกใช้ฟังก์ชัน กับชุดคำสั่งที่กำหนดฟังก์ชันนั้น มีข้อจำกัดที่ไม่สามารถวิเคราะห์พารามิเตอร์ที่ไม่ใช่ตัวแปรที่มีชนิดข้อมูลเป็นชนิดข้อมูลภายในของภาษา อินฟอร์มิกซ์ โฟทจีแอล ได้ถูกต้องทั้งหมด เช่น มีการผ่านค่าพารามิเตอร์เป็นค่าคงที่ตัวเลข ซึ่งสามารถวิเคราะห์เป็นชนิดข้อมูลได้มากกว่าหนึ่งชนิดข้อมูลได้

## 5.3 ข้อเสนอแนะในการพัฒนาเพิ่มเติม

เนื่องจากได้กำหนดขอบเขตในการวิจัยนี้ ให้ตรวจสอบเฉพาะชนิดข้อมูลภายในตัวภาษา 4GL เท่านั้น จึงมีส่วนสามารถพัฒนาต่อได้ ในส่วนการทำเลกซิคัล ให้สามารถตรวจสอบชนิดข้อ



มุลที่กำหนดขึ้นเอง โดยผู้เขียน โปรแกรม และชนิดข้อมูลที่อ้างอิงกับชนิดข้อมูลในฐานข้อมูลที่โปรแกรมใช้งาน โดยการอ่านพจนานุกรมข้อมูลของฐานข้อมูล เพื่อค้นหาว่าชนิดข้อมูลของเขตข้อมูลในตารางที่อ้างอิงถึงเป็นข้อมูลชนิดใด

นอกจากนี้ยังสามารถพัฒนาในส่วนของการแจ้งเตือนเมื่อระบบตรวจพบความสัมพันธ์ระหว่างฟังก์ชันที่ขัดแย้งกัน โดยให้สามารถส่งเอกสารเตือนแจ้งผู้ที่เกี่ยวข้องในรูปแบบของ Email ได้

#### 5.4 ผลงานที่เกี่ยวข้องกับการวิจัย

งานวิจัยนี้นอกจากได้จัดทำเป็นรูปเล่มของวิทยานิพนธ์ฉบับนี้แล้ว ยังได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวข้อของวิศวกรรมซอฟต์แวร์ ชื่อเรื่อง “A Development of Source Code Control System with Data Coupling Dependency Alert” ในหนังสือ “The 1999 National Computer Science and Engineering Conference” (ดูรายละเอียดในภาคผนวก ค.) และนำเสนอในงาน NCSEC'99 [6] ซึ่งจัดขึ้นระหว่างวันที่ 16-17 ธันวาคม 1999 เวลา 8.30-17.00 น. ที่โรงแรม LandMark กรุงเทพฯ

## รายการอ้างอิง

- [1] Mikkelsen, T., and Pherigo, S. Practical Software Configuration Management.  
U.S.A. : Prentice-Hall, Inc., 1997.
- [2] Aho Alfred, V., and Ullman Jeffrey, D. Principles of Compiler Design.  
U.S.A. : Addison-Wesley Publishing Company, 1977.
- [3] Mason T., and Brown D. Lex & Yacc.  
Sebastopol: O'Reilly & Associates, Inc., 1990.
- [4] Informix Software, Inc. Informix-4GL Reference Manual Volume One.  
Canada: Informix Software, Inc., 1990.
- [5] Informix Software, Inc. Informix-4GL Reference Manual Volume Two.  
Canada: Informix Software, Inc., 1990.
- [6] Assumption University. The 1999 National Computer Science and Engineering Conference.  
Bangkok: Assumption University, 1999. 369-372.



ภาคผนวก ก

## การใช้งานระบบจัดการเวอร์ชันซุคคำสั่ง

ระบบจัดการเวอร์ชันซุคคำสั่ง เป็นเครื่องมือประเภท Software Configuration Management (SCM) ที่พัฒนาขึ้นเพื่อใช้จัดเก็บซุคคำสั่ง โปรแกรมภาษา อินฟอร์มิกซ์ โฟทจีแอล ไว้ในรูปแบบที่เป็นเวอร์ชัน และแจ้งเตือนผลกระทบที่เกิดขึ้นกับซุคคำสั่ง โปรแกรมที่จัดเก็บไว้ในระบบ ซึ่งเกิดจากการเปลี่ยนแปลงแก้ไขซุคคำสั่ง โปรแกรมที่มีความสัมพันธ์กัน

ระบบจัดการเวอร์ชันซุคคำสั่ง ประกอบด้วย โปรแกรมสำหรับใช้งานดังนี้

- โปรแกรมจัดเก็บเพิ่มซุคคำสั่งมอดูล (Pcheckin.exe)
- โปรแกรมเบิกซุคคำสั่งมอดูล (Pcheckout.exe)
- โปรแกรมแสดงความสัมพันธ์ที่ขัดแย้งระหว่างซุคคำสั่งมอดูล (Pcheckrel.exe)
- โปรแกรมกำหนดผู้ใช้ระบบและสิทธิในการใช้งาน (Padmin.exe)
- โปรแกรมแสดงความแตกต่างระหว่างเวอร์ชันของซุคคำสั่งมอดูล (Pcheckdiff.exe)

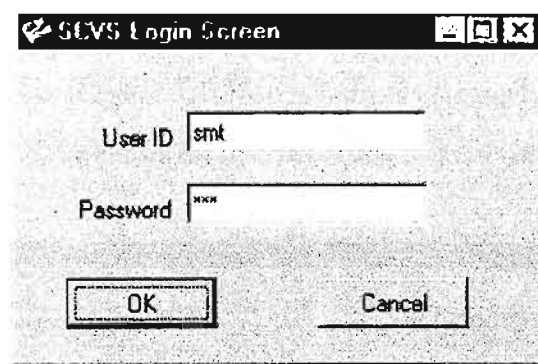
### โปรแกรมจัดเก็บเพิ่มซุคคำสั่งมอดูล (Pcheckin.exe)

เป็นโปรแกรมที่ใช้สำหรับจัดเก็บเพิ่มซุคคำสั่งมอดูล เข้าสู่ระบบจัดการเวอร์ชันซุคคำสั่ง ซึ่งในกรณีที่เพิ่มซุคคำสั่งมอดูลที่ต้องการจัดเก็บ เป็นซุคคำสั่งมอดูลที่เคยมีการจัดเก็บไว้ในระบบแล้ว โปรแกรมจะทำการตรวจหาส่วนที่แตกต่างระหว่างซุคคำสั่งมอดูลที่อยู่ในเพิ่มที่ต้องการจัดเก็บและซุคคำสั่งมอดูลเวอร์ชันล่าสุดที่เก็บอยู่ในระบบ เพื่อจัดเก็บไว้เป็นส่วนต่างที่ใช้สำหรับการย้อนเวอร์ชันซุคคำสั่งมอดูล ตามวิธีการจัดเก็บข้อมูลแบบเคลด้า และนำซุคคำสั่งมอดูลในเพิ่มจัดเก็บเข้าสู่ระบบเป็นซุคคำสั่งเวอร์ชันล่าสุด

นอกจากการหน้าที่ที่กล่าวมาแล้ว โปรแกรมยังทำหน้าที่วิเคราะห์ความสัมพันธ์ของซุคคำสั่งมอดูลที่นำมาจัดเก็บ เพื่อตรวจหาและสร้างความสัมพันธ์ของซุคคำสั่งมอดูลที่นำมาจัดเก็บ กับซุคคำสั่งมอดูลเวอร์ชันล่าสุดอื่นที่จัดเก็บอยู่ในระบบ พร้อมทั้งรายงานความสัมพันธ์ระหว่างซุคคำสั่งมอดูลที่ได้รับผลกระทบจากการแก้ไข

ในการใช้งาน โปรแกรมจัดเก็บเพิ่มซุคคำสั่งมอดูล มีลำดับขั้นตอนการใช้งานดังนี้

1. เมื่อเรียกใช้โปรแกรมจัดเก็บเพิ่มซุคคำสั่งมอดูล โปรแกรมจะแสดงหน้าต่างลงบันทึกเข้า เพื่อให้ผู้ใช้งานป้อนรหัสผู้ใช้และรหัสผ่าน ดังแสดงในรูปที่ ก.1



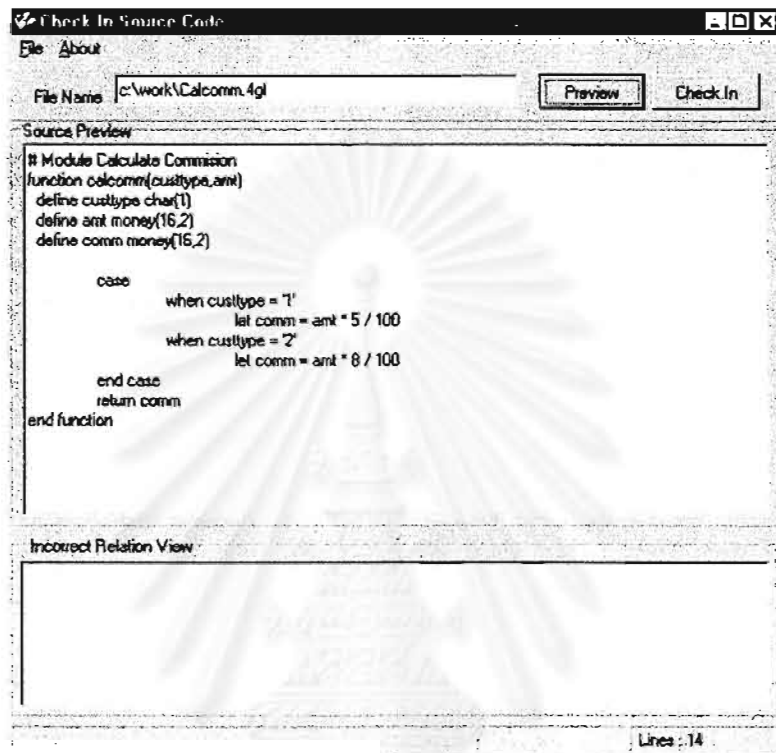
รูปที่ ก.1 หน้าต่างลงบันทึกเข้า

- หลังจากที่ผู้ใช้งานป้อนรหัสผู้ใช้และรหัสผ่าน และกดปุ่ม OK แล้ว โปรแกรมจะเข้าสู่หน้าต่างหลัก ทำการป้อนชื่อเพิ่มชุดคำสั่งที่ต้องการจัดเก็บเข้าสู่ระบบ ดังแสดงในรูปที่ ก.2



รูปที่ ก.2 หน้าต่างหลัก โปรแกรมจัดเก็บเพิ่มข้อมูลชุดคำสั่งมอดูล

3. ผู้ใช้งานสามารถตรวจสอบความถูกต้องชุดคำสั่งมอดูลในเพิ่มชุดคำสั่งมอดูลที่ต้องการจัดเก็บ ก่อนจัดเก็บลงระบบ โดยการกดปุ่ม Preview หลังจากที่ได้ป้อนชื่อเพิ่มชุดคำสั่งที่ต้องการจัดเก็บเข้าสู่ระบบเรียบร้อยแล้ว ซึ่งโปรแกรมจะนำชุดคำสั่งมอดูลในเพิ่ม มาแสดงในช่อง Source Preview ดังแสดงในรูปที่ ก.3



รูปที่ ก.3 โปรแกรมทำการแสดงชุดคำสั่งในเพิ่มข้อมูลชุดคำสั่งมอดูลที่ต้องการจัดเก็บ

4. ทำการกดปุ่ม Check In เพื่อให้โปรแกรมเริ่มทำการจัดเก็บ ซึ่งในขั้นตอนนี้โปรแกรมจะทำการตรวจสอบว่าเป็นชุดคำสั่งมอดูลที่เคยมีการเก็บอยู่ในระบบแล้วหรือไม่ ถ้าเป็นกรณีชุดคำสั่งที่ต้องการจัดเก็บยังไม่เคยมีการจัดเก็บไว้ในระบบมาก่อน โปรแกรมจะแสดงหน้าจอให้ป้อนรายละเอียดของโปรแกรม ดังแสดงในรูปที่ ก.4 แต่ถ้าเป็นกรณีที่ชุดคำสั่งที่ต้องการจัดเก็บเคยมีการจัดเก็บไว้ในระบบมาก่อน โปรแกรมจะแสดงหมายเลขเวอร์ชันที่กำลังจะทำการจัดเก็บ และให้ผู้ใช้งานป้อนหมายเหตุของการเปลี่ยนแปลง ดังแสดงในรูปที่ ก.5

**New Program Information**

Program Description: Module for calculate commission by customer type.

Program Owner: smt

OK

รูปที่ ก.4 แสดงการป้อนเมื่อชุดคำสั่งมอดูลที่ต้องการจัดเก็บที่ไม่เคยมีการจัดเก็บมาก่อน

**New Version Information**

Program Next Version: 1.1

Edit Programmer: smt

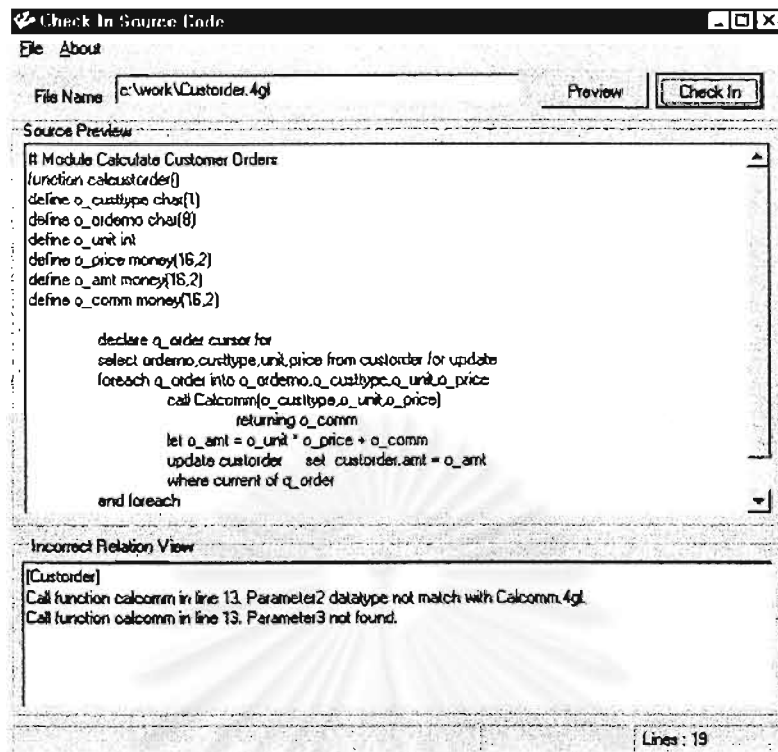
Comment:

1. Initial variable comm to zero
2. Change case custtype is '2' to used same fomula of case custtype is '1'.
3. For case custtype is '2' comm must include \$50 fee.

OK

รูปที่ ก.5 แสดงการป้อนเมื่อชุดคำสั่งมอดูลที่ต้องการจัดเก็บที่เคยมีการจัดเก็บมาก่อน

5. หลังจากผ่านขั้นตอนการป้อนในข้อ 4 โปรแกรมจะตรวจสอบวากยสัมพันธ์ของชุดคำสั่งที่ต้องการจัดเก็บ ทำการบันทึกความสัมพันธ์ ตรวจสอบและแสดงผลกระทบที่เกิดขึ้นเนื่องจากการแก้ไขเปลี่ยนแปลงชุดคำสั่งมอดูลที่จัดเก็บ ดังแสดงในรูปที่ ก.6
6. ในกรณีที่ผู้ใช้มีเพิ่มชุดคำสั่งที่ต้องการจัดเก็บหลายเพิ่ม ผู้ใช้งานสามารถจัดเก็บเพิ่มชุดคำสั่งเพิ่มต่อไปได้ โดยการป้อนชื่อเพิ่มชุดคำสั่งได้
7. สำหรับการจัดเก็บเพิ่มชุดคำสั่งมอดูลที่เคยจัดเก็บไว้ในระบบแล้ว โปรแกรมจะทำการตรวจสอบว่า ชุดคำสั่งมอดูลต้องถูกเบิกออกจากระบบในแบบ “การเบิกเพื่อแก้ไข” โดยโปรแกรมเบิกเพิ่มชุดคำสั่งมอดูล และผู้ใช้ที่ทำการจัดเก็บต้องเป็นเดียวกันกับผู้ใช้ที่ขอเบิกชุดคำสั่งดังกล่าวไปแก้ไขเท่านั้น



รูปที่ ก.6 แสดงการแจ้งเตือนผลกระทบเมื่อทำการจัดเก็บชุดคำสั่งมอดูล

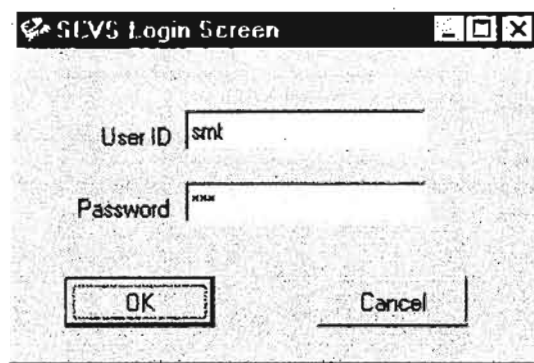
## โปรแกรมเบิกเพิ่มชุดคำสั่งมอดูล (Pcheckout.exe)

เป็นโปรแกรมที่ใช้สำหรับนำชุดคำสั่งมอดูลที่เก็บไว้ในระบบจัดการเวอร์ชันชุดคำสั่ง ออกมาเป็นเพิ่มชุดคำสั่งมอดูล โดยโปรแกรมสามารถประมวลผลเพื่อสร้างเพิ่มชุดคำสั่งมอดูลเวอร์ชันย้อนหลัง จากชุดคำสั่งมอดูลเวอร์ชันล่าสุดและความแตกต่างของการแก้ไขเปลี่ยนแปลงชุดคำสั่งมอดูลในแต่ละเวอร์ชันที่เก็บอยู่ในระบบ ในการเบิกชุดคำสั่งมอดูลออกจากระบบทุกครั้ง ระบบต้องให้ระบุจุดประสงค์ของการเบิก เพื่อให้สามารถตรวจสอบและป้องกันการแก้ไขชุดคำสั่งมอดูลซ้ำซ้อนกันในช่วงเวลาเดียวกันโดยผู้ใช้นี้มากกว่าหนึ่งคน

ในการใช้งาน โปรแกรมเบิกเพิ่มชุดคำสั่งมอดูล มีลำดับขั้นตอนการใช้งานดังนี้

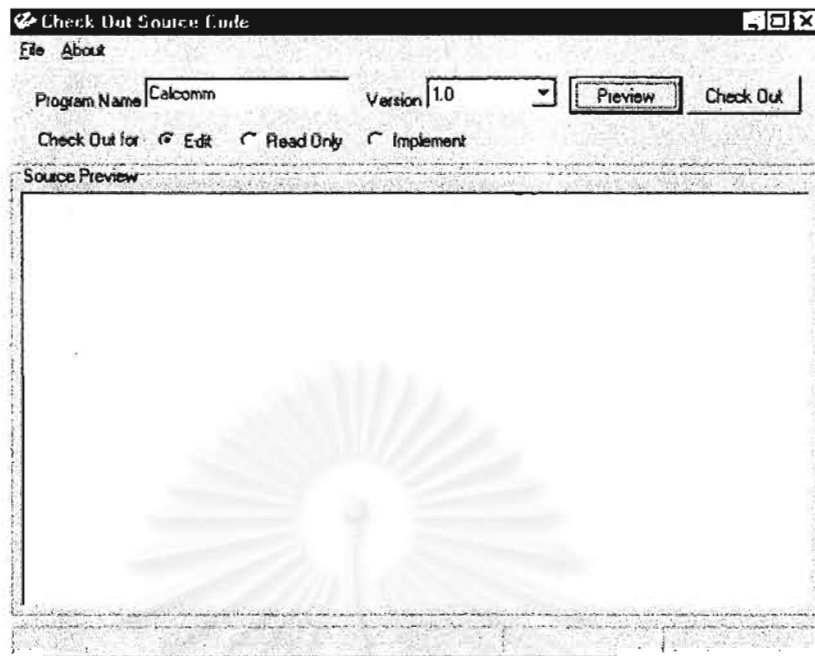
1. เมื่อเรียกใช้โปรแกรมจัดเก็บเพิ่มชุดคำสั่งมอดูล โปรแกรมจะแสดงหน้าต่างลงบันทึกเข้า เพื่อให้ผู้ใช้งานป้อนรหัสผู้ใช้และรหัสผ่าน ดังแสดงในรูปที่ ก.7



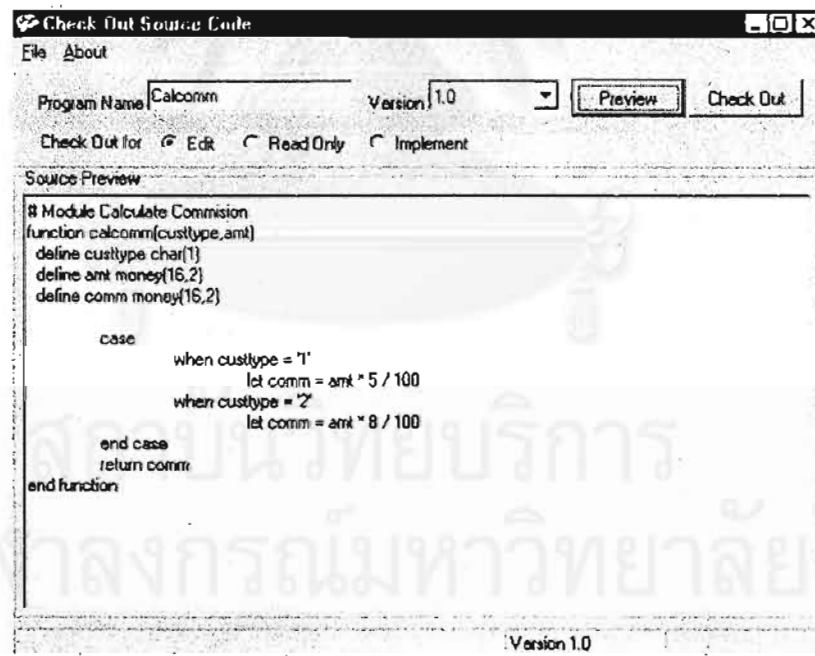


รูปที่ ก.7 หน้าต่างลงบันทึกเข้า

2. หลังจากที่คุณใช้งานป้อนรหัสผู้ใช้และรหัสผ่าน และกดปุ่ม OK แล้ว โปรแกรมจะเข้าสู่หน้าต่างหลัก ทำการป้อนชื่อชุดคำสั่งมอดูลและเวอร์ชันที่ต้องการเบิกจากระบบ พร้อมทั้งเลือกรูปแบบของการเบิก ดังแสดงในรูปที่ ก.8 ซึ่งรูปแบบของการเบิกชุดคำสั่งมอดูล สามารถแบ่งออกเป็น
  - Edit เป็นการเบิกของผู้พัฒนาโปรแกรมเพื่อนำเอาชุดคำสั่งมอดูล มาแก้ไขเปลี่ยนแปลงเพิ่ม โดยระบบจะทำการบันทึกที่รหัสผู้ใช้ของผู้ที่เบิกชุดคำสั่งมอดูล ไปแก้ไข และปิดกั้นไม่ให้ผู้อื่นมาเบิกชุดคำสั่งดังกล่าวไปแก้ไขซ้ำซ้อนกัน
  - Read Only เป็นการเบิกชุดคำสั่งเพื่อนำไปอ้างอิงหรือตรวจสอบ โดยไม่มีการแก้ไขเปลี่ยนแปลงชุดคำสั่งมอดูลดังกล่าว
  - Implement เป็นการเบิกชุดคำสั่งเพื่อนำชุดคำสั่งมอดูลไปทำการติดตั้งที่ลูกค้าที่ได้ชื่อชุดคำสั่งมอดูลดังกล่าว โปรแกรมจะให้ป้อนรายละเอียดเกี่ยวกับลูกค้าที่จะนำไปติดตั้ง เพื่อเก็บไว้เป็นประวัติอ้างอิงในภายหลัง
3. ผู้ใช้งานสามารถตรวจสอบความถูกต้องชุดคำสั่งมอดูลในชุดคำสั่งมอดูลที่ต้องการเบิก ก่อนจัดสร้างเป็นแพ้มชุดคำสั่งมอดูล โดยการกดปุ่ม Preview หลังจากที่ได้ป้อนชื่อชุดคำสั่งและเวอร์ชันที่ต้องการเบิกจากระบบเรียบร้อยแล้ว ซึ่งโปรแกรมจะนำชุดคำสั่งมอดูลในระบบ มาแสดงในช่อง Source Preview ดังแสดงในรูปที่ ก.9

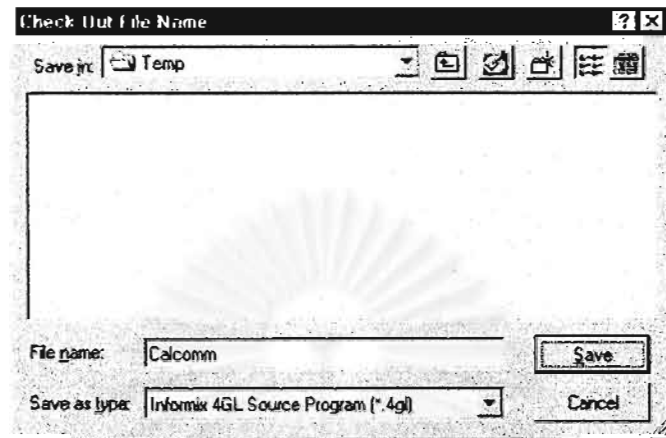


รูปที่ ก.8 หน้าต่างหลัก โปรแกรมเบิกเพิ่มข้อมูลชุดคำสั่งมอดูล



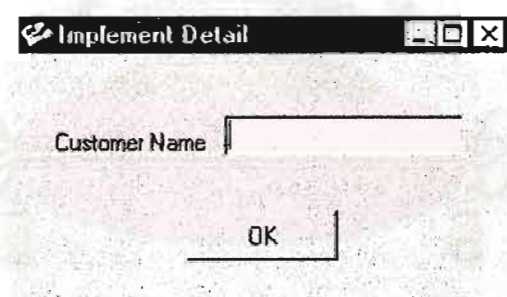
รูปที่ ก.9 แสดงการเบิกชุดคำสั่งมอดูลที่ต้องการแก้ไข

4. ทำการกดปุ่ม Check Out เพื่อให้โปรแกรมเริ่มทำจัดเก็บชุดคำสั่งลงโมดูล ลงในแฟ้ม ซึ่งโปรแกรมแสดงหน้าต่างป้อนชื่อแฟ้มที่ต้องการจัดเก็บ ดังแสดงในรูปที่ ก.10



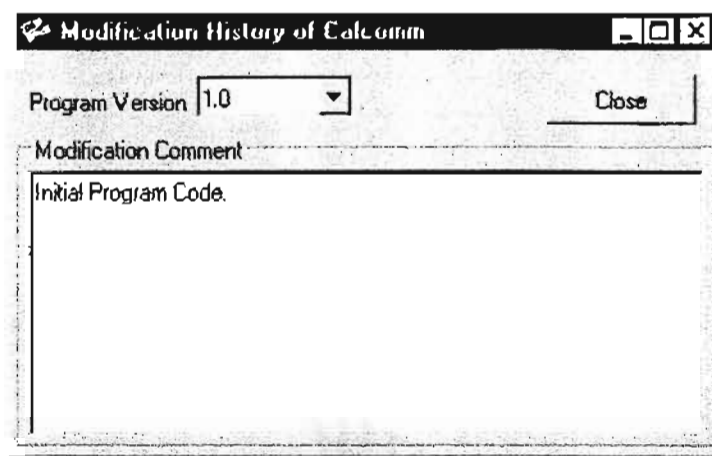
รูปที่ ก.10 หน้าต่างสำหรับป้อนชื่อแฟ้มชุดคำสั่งโมดูลที่เบิกจากระบบ

5. ในกรณีการเบิกเพื่อติดตั้ง โปรแกรมจะแสดงหน้าต่างสำหรับป้อนรายละเอียดลูกค้าที่นำไปติดตั้ง ดังแสดงในรูปที่ ก.11



รูปที่ ก.11 หน้าต่างป้อนรายละเอียดเกี่ยวกับลูกค้า

6. นอกจากหน้าที่ในการเบิกชุดคำสั่งโมดูลจากระบบแล้ว โปรแกรมนี้ยังสามารถแสดงประวัติการแก้ไขของชุดคำสั่งโมดูล ซึ่งเป็นข้อความหมายเหตุในการแก้ไขชุดคำสั่งโมดูล ที่ป้อนไว้ในขณะทำการจัดเก็บชุดคำสั่งโมดูลเข้าสู่ระบบ โดยเลือกรายการเลือกแบบดิ่งลง หัวข้อ File และหัวข้อย่อย History หลังจากที่ได้ป้อนชื่อชุดคำสั่งและเวอร์ชันเรียบร้อยแล้ว โปรแกรมจะแสดงหน้าต่างประวัติการแก้ไข ดังแสดงในรูปที่ ก.12



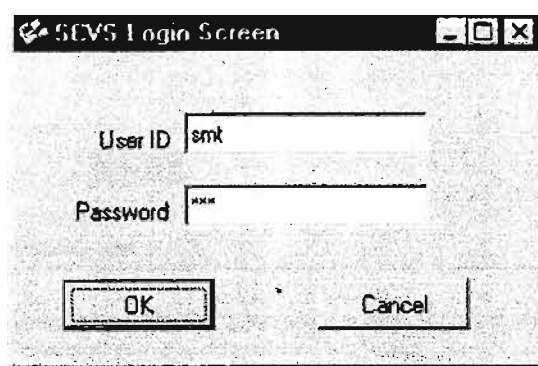
รูปที่ ก.12 หน้าต่างแสดงประวัติการแก้ไขของชุดคำสั่งมอดูล

### โปรแกรมแสดงความสัมพันธ์ที่ขัดแย้งระหว่างชุดคำสั่งมอดูล (Pcheckrel.exe)

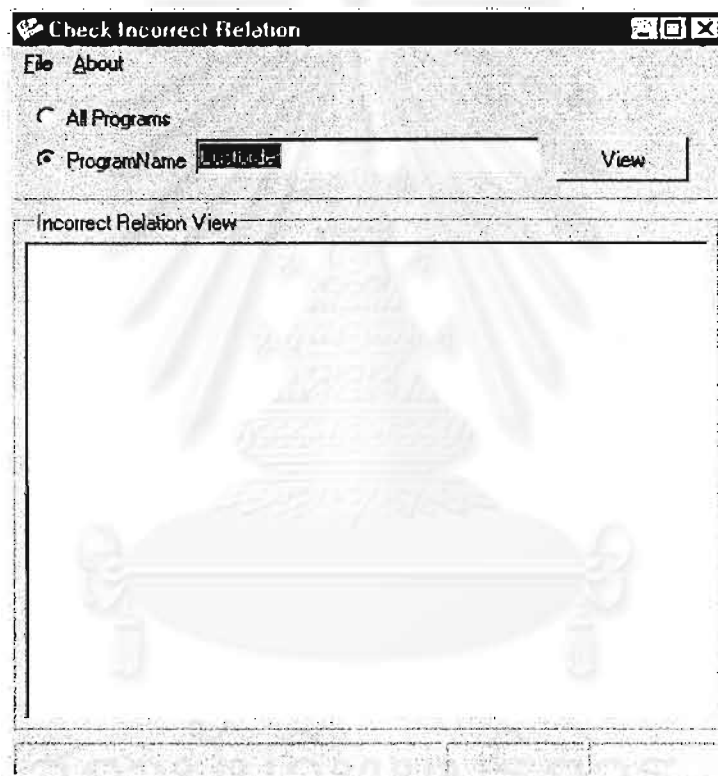
เป็นโปรแกรมที่ใช้สำหรับตรวจสอบความสัมพันธ์ของชุดคำสั่งมอดูลที่เก็บไว้ในระบบจัดการเวอร์ชันชุดคำสั่ง เพื่อให้ผู้พัฒนาโปรแกรมสามารถตรวจสอบว่าได้ทำการแก้ไขชุดคำสั่งมอดูลที่ได้รับผลกระทบเนื่องจากการแก้ไขเปลี่ยนแปลงครบถ้วนหรือไม่ โดยโปรแกรมจะทำการตรวจสอบความสัมพันธ์ของชุดคำสั่งเฉพาะเวอร์ชันล่าสุด สามารถเลือกให้แสดงความสัมพันธ์ที่ขัดแย้งของชุดคำสั่งมอดูลทั้งหมดที่จัดเก็บอยู่ในระบบ หรือเลือกแสดงความสัมพันธ์ที่ขัดแย้งของชุดคำสั่งมอดูลเฉพาะเจาะจงมอดูลใดมอดูลหนึ่ง

ในการใช้งาน โปรแกรมแสดงความสัมพันธ์ที่ขัดแย้งระหว่างชุดคำสั่งมอดูล มีลำดับขั้นตอนการใช้งานดังนี้

1. เมื่อเรียกใช้แสดงความสัมพันธ์ที่ขัดแย้งระหว่างชุดคำสั่งมอดูล โปรแกรมจะแสดงหน้าต่างลงบันทึกเข้า เพื่อให้ผู้ใช้งานป้อนรหัสผู้ใช้และรหัสผ่าน ดังแสดงในรูปที่ ก.13
2. หลังจากป้อนรหัสผู้ใช้และรหัสผ่าน และกดปุ่ม OK แล้ว โปรแกรมจะเข้าสู่หน้าต่างหลัก ทำการเลือกว่าต้องการแสดงความสัมพันธ์ที่ขัดแย้งของชุดคำสั่งมอดูลทั้งหมดที่จัดเก็บอยู่ในระบบ (All Programs) หรือเลือกแสดงความสัมพันธ์ที่ขัดแย้งของชุดคำสั่งมอดูลใดมอดูลหนึ่ง โดยการป้อนชื่อชุดคำสั่งมอดูลที่เก็บอยู่ในระบบ ดังแสดงในรูปที่ ก.14

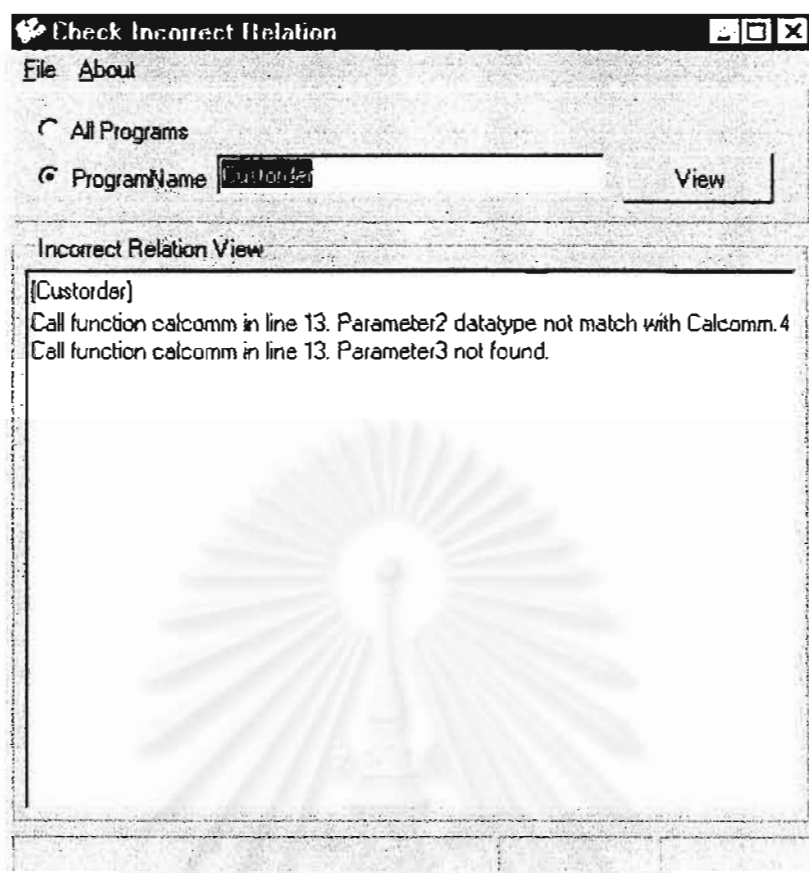


รูปที่ ก.13 หน้าต่างลงบันทึกเข้า



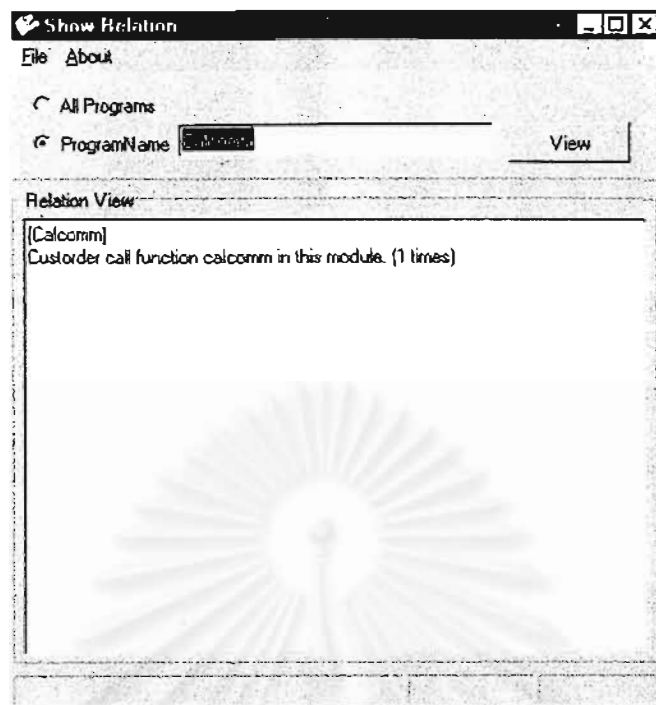
รูปที่ ก.14 หน้าต่างหลักโปรแกรมแสดงความสัมพันธ์ที่ขัดแย้งระหว่างชุดคำสั่งมอดูล

3. ทำการกดปุ่ม View เพื่อให้โปรแกรมเริ่มตรวจสอบความสัมพันธ์ที่ขัดแย้งระหว่างชุดคำสั่งมอดูล ซึ่งโปรแกรมแสดงผลการตรวจสอบความสัมพันธ์ในช่อง "Incorrect Relation View" ดังแสดงในรูปที่ ก.15



รูปที่ ก.15 แสดงความสัมพันธ์ที่ขัดแย้งระหว่างชุดคำสั่งมอดูล

- นอกจากการแสดงความสัมพันธ์ที่ขัดแย้งระหว่างชุดคำสั่งมอดูลแล้ว ในการเลือกแสดงความสัมพันธ์เฉพาะชุดคำสั่งมอดูลใดมอดูลหนึ่ง โปรแกรมยังสามารถแสดงรายชื่อชุดคำสั่งมอดูลอื่นที่เรียกใช้ฟังก์ชันของชุดคำสั่งนั้นๆ โดยกลุ่มซ้ายของเมาส์ 2 ครั้ง ที่ข้อความ "Incorrect Relation View" ซึ่งข้อความดังกล่าวจะเปลี่ยนเป็น "Relation View" เมื่อคลิก View โปรแกรมจะแสดงรายชื่อชุดคำสั่งมอดูลอื่นที่เรียกใช้ฟังก์ชันของชุดคำสั่งนั้น ดังแสดงในรูปที่ ก.16



รูปที่ ก.16 แสดงรายชื่อชุดคำสั่งมอดูลอื่นที่เรียกใช้ฟังก์ชันที่กำหนดภายในชุดคำสั่งมอดูล

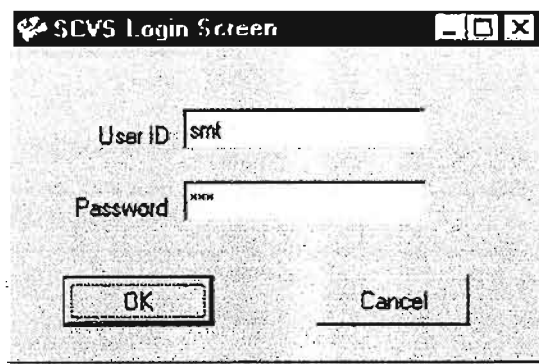
## โปรแกรมกำหนดผู้ใช้ระบบและสิทธิในการทำงาน (Padmin.exe)

เป็นโปรแกรมที่ใช้สำหรับกำหนดผู้ใช้ระบบและสิทธิในการทำงานระบบจัดการเวอร์ชันชุดคำสั่ง เพื่อความปลอดภัยของชุดคำสั่งที่จัดเก็บไว้ในระบบ

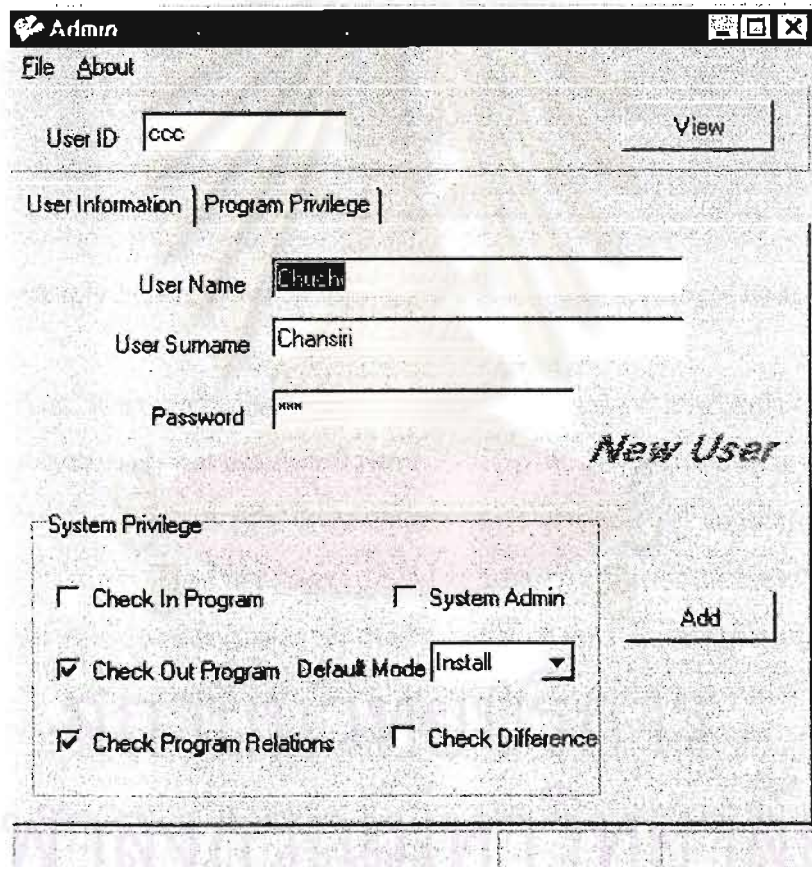
ในการทำงาน โปรแกรมกำหนดผู้ใช้ระบบและสิทธิในการทำงาน มีลำดับขั้นตอนการใช้งานดังนี้

1. เมื่อเรียกใช้โปรแกรมจัดเก็บแฟ้มชุดคำสั่งมอดูล โปรแกรมจะแสดงหน้าต่างลงบันทึกเข้า เพื่อให้ผู้ใช้งานป้อนรหัสผู้ใช้และรหัสผ่าน ดังแสดงในรูปที่ ก.17
2. หลังจากที่ผู้ใช้งานป้อนรหัสผู้ใช้และรหัสผ่าน และกดปุ่ม OK แล้ว โปรแกรมจะเข้าสู่หน้าต่างหลัก เพื่อให้ป้อนรหัสผู้ใช้ ในกรณีที่รหัสผู้ใช้ไม่เคยมีมาก่อนในระบบ โปรแกรมจะถือว่าเป็นการเพิ่มรหัสผู้ใช้ใหม่เข้าในระบบ ดังแสดงในรูปที่ ก.18 แต่ถ้ารหัสผู้ใช้ที่ป้อนเป็นผู้ใช้ที่มีอยู่ในระบบ โปรแกรมจะนำข้อมูลเกี่ยวกับผู้ใช้นั้น มาแสดงเพื่อให้ทำการแก้ไข ดังแสดงในรูปที่ ก.19



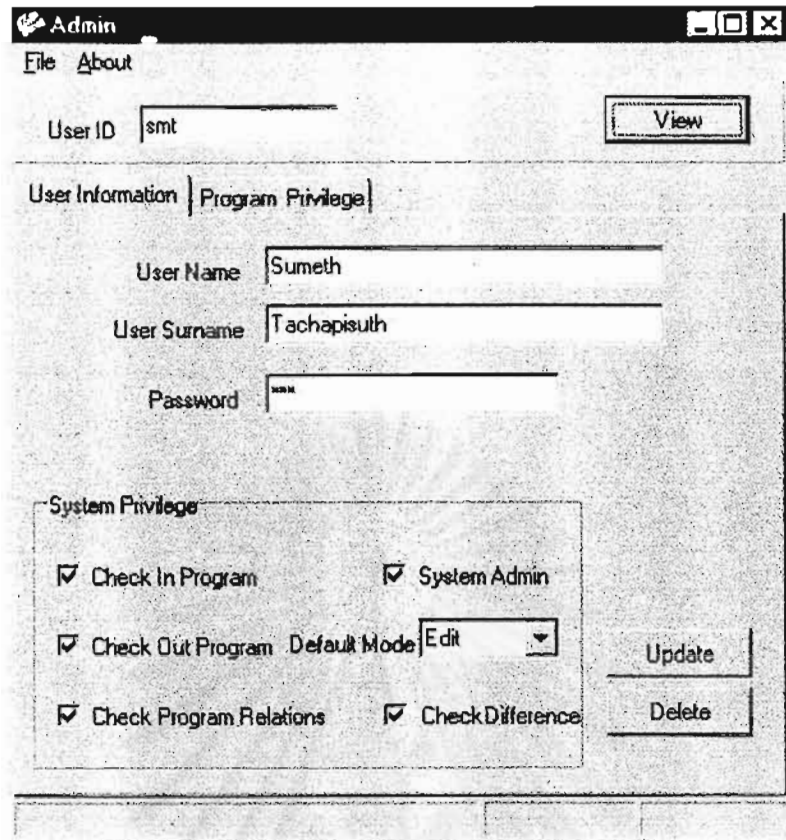


รูปที่ ก.17 หน้าต่างลงบันทึกเข้า



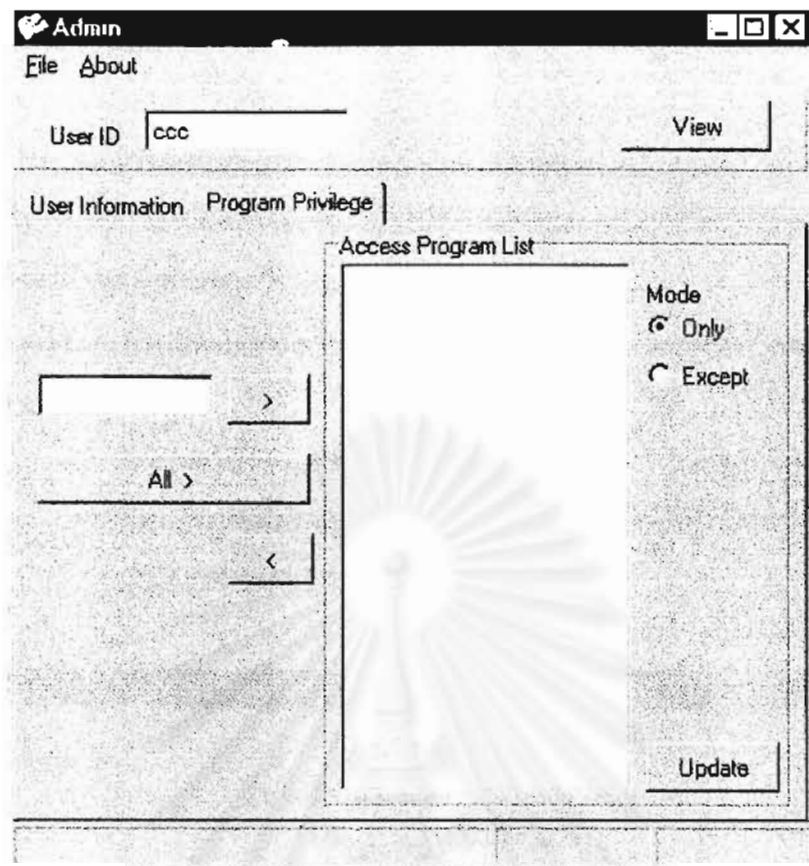
รูปที่ ก.18 โปรแกรมกำหนดผู้ใช้และสิทธิในการทำงาน ขณะกำหนดผู้ใช้งานใหม่ลงสู่ระบบ





รูปที่ ก.19 โปรแกรมกำหนดผู้ใช้และสิทธิในการทำงาน ขณะแก้ไขข้อมูลผู้ใช้ระบบ

3. หลังจากได้ทำการเพิ่มผู้ใช้งานใหม่ลงในระบบแล้ว สามารถกำหนดสิทธิในการเข้าถึงชุดคำสั่งมอดูลที่เกี่ยวข้องในระบบได้ โดยกดปุ่มแถบป้ายระบุ "Program Privilege" แล้ว โปรแกรมจะเข้าสู่หน้าต่างหลัก สำหรับกำหนดสิทธิในการเข้าถึงชุดคำสั่งมอดูล โดยผู้ใช้งานสามารถเพิ่มหรือลดชื่อชุดคำสั่งในรายการชื่อชุดคำสั่งมอดูล รวมทั้งกำหนดได้ว่ารายการชื่อชุดคำสั่งมอดูลเป็นชื่อชุดคำสั่งมอดูลที่ผู้ใช้นั้นมีสิทธิในการเข้าถึง หรือเป็นรายชื่อชุดคำสั่งมอดูลที่ถูกยกเว้นไม่ให้มีสิทธิในการเข้าถึง เพื่อให้สะดวกในกรณีที่ผู้ใช้นั้นมีสิทธิในการเข้าถึงชุดคำสั่งมอดูลเกือบทั้งระบบยกเว้นชุดคำสั่งมอดูลเพียงบางชุดเท่านั้น ดังแสดงในรูปที่ ก.20



รูปที่ ก.20 แสดงการกำหนดสิทธิในการเข้าถึงชุดคำสั่งมอดูลของผู้ใช้ระบบ

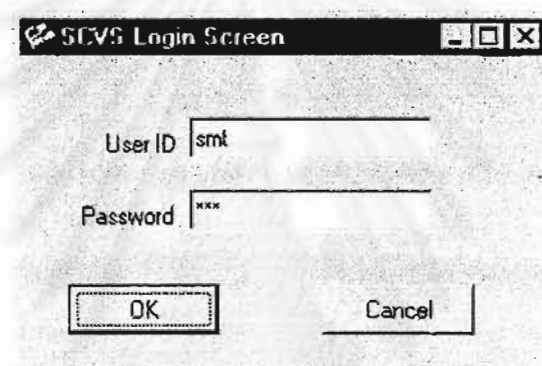
4. ในการกำหนดสิทธิในการเข้าถึงชุดคำสั่งมอดูล โปรแกรมจะทำการบันทึกสิทธิในการเข้าถึงตามรายชื่อชุดคำสั่งมอดูลที่ปรากฏในช่อง Access Program List ผู้ใช้สามารถเพิ่มชื่อชุดคำสั่งมอดูลที่ต้องการเข้าใน Access Program List โดยการป้อนชื่อชุดคำสั่งมอดูลในช่องหน้าปุ่ม > แล้วกดปุ่ม > ส่วนการลบชื่อออกจาก Access Program List ทำได้โดยการเลือกชื่อชุดคำสั่งมอดูลแล้วกดปุ่ม < ในกรณีที่ต้องการเพิ่มชื่อชุดคำสั่งมอดูลทั้งหมดลงใน Access Program List สามารถทำได้โดยกดปุ่ม All > สำหรับรายชื่อชุดคำสั่งที่อยู่ในช่อง Access Program List ผู้ใช้สามารถกำหนดได้ว่าให้เป็นรายชื่อที่กำหนดสิทธิในการเข้าถึงได้ โดยเลือกโหมดเป็น Only หรือกำหนดให้เป็นรายชื่อที่ยกเว้นการเข้าถึง โดยเลือกโหมดเป็น Except ซึ่งหมายถึงผู้ใช้งานสามารถเข้าถึงชุดคำสั่งมอดูลทั้งหมดยกเว้นชุดคำสั่งมอดูลที่มีชื่อปรากฏใน Access Program List

## โปรแกรมแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูล (Pcheckdiff.exe)

เป็นโปรแกรมที่ใช้สำหรับเปรียบเทียบความแตกต่างที่เปลี่ยนแปลงไป ของชุดคำสั่งมอดูล 2 เวอร์ชันที่จัดเก็บในระบบจัดการเวอร์ชัน เพื่อให้ผู้พัฒนาโปรแกรมสามารถตรวจสอบและติดตามได้ว่าเคยมีการแก้ไขชุดคำสั่งมอดูลไปอย่างไรบ้าง

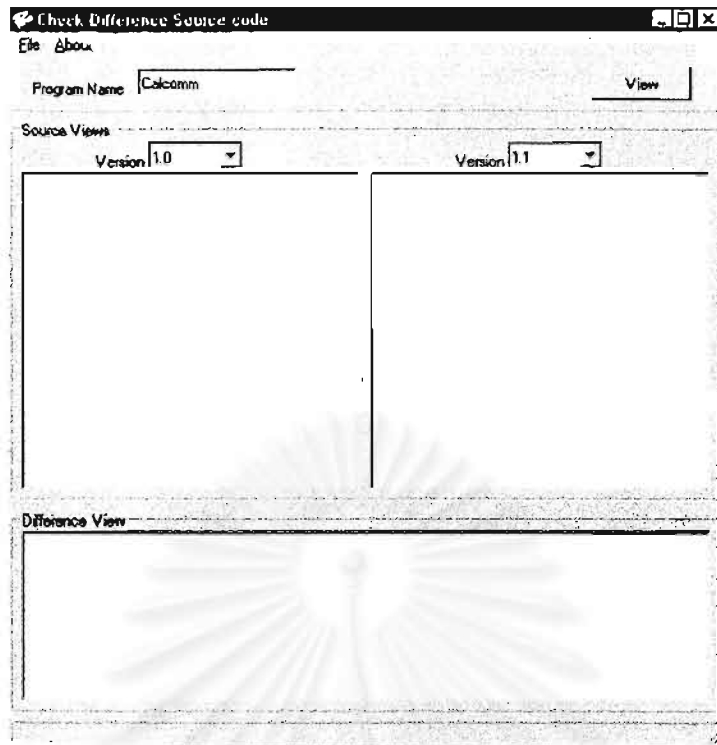
ในการใช้งานโปรแกรมแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูล มีลำดับขั้นตอนการใช้งานดังนี้

1. เมื่อเรียกใช้โปรแกรมแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูล โปรแกรมจะแสดงหน้าต่างลงบันทึกเข้า เพื่อให้ผู้ใช้งานป้อนรหัสผู้ใช้และรหัสผ่าน ดังแสดงในรูปที่ ก.21

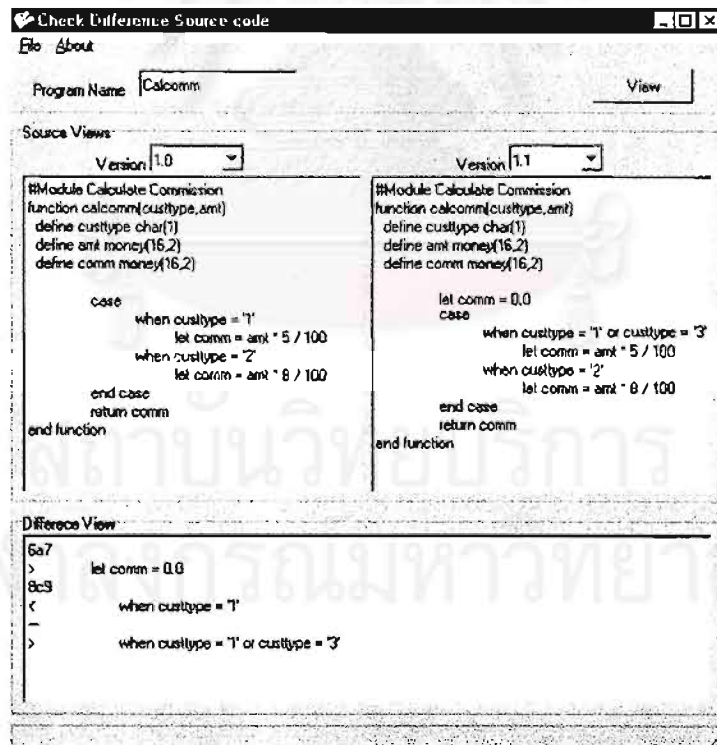


รูปที่ ก.21 หน้าต่างลงบันทึกเข้า

2. หลังจากที่คุณใช้งานป้อนรหัสผู้ใช้และรหัสผ่าน และกดปุ่ม OK แล้ว โปรแกรมจะเข้าสู่หน้าต่างหลัก เพื่อให้ป้อนชื่อชุดคำสั่งมอดูลและเวอร์ชัน 2 ชุด ที่ต้องการเปรียบเทียบความแตกต่าง ดังแสดงในรูปที่ ก.22
3. กดปุ่ม View เพื่อให้โปรแกรมเริ่มทำการนำชุดคำสั่งมอดูลเวอร์ชันที่ต้องการเปรียบเทียบ ออกมาแสดงและทำการตรวจสอบความแตกต่างระหว่างเวอร์ชัน เพื่อนำมาแสดงดังแสดงในรูปที่ ก.23



รูปที่ ก.22 หน้าต่างหลักโปรแกรมแสดงความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูล



รูปที่ ก.23 แสดงการเปรียบเทียบความแตกต่างระหว่างเวอร์ชันของชุดคำสั่งมอดูล



ภาคผนวก ข



**ผลงานดีพิมพ์ในงาน NCSEC'99**



# A Development of Source Code Control System with Data Coupling Dependency Alert

Sumeth Tachapisuth<sup>1</sup> and Wiwat Vatanawood<sup>2</sup>  
Department of Computer Engineering  
Faculty of Engineering  
Chulalongkorn University  
Phyathai Road, Bangkok 10330  
Tel: (662) 218-6975 Fax: (662) 218-6955  
Email: smt@kernel.co.th<sup>1</sup>, wiwat@chula.ac.th<sup>2</sup>

**Abstract:** In this research, the source code control system is developed using delta storage technique as to save storage. It means that source codes common to more than one version is not duplicated. The developer is capable to retrieve previous multiple versions of program source codes, if needed. Moreover, an extended feature is proposed during the source code control task as to alert the functional dependency among modules of program source codes. In our approach, data coupling metric is exploited to identify the functional dependency between two modules. The data coupling dependency checking feature requires only program source codes as input alone without any information from design document. It should be one of the practical software configuration management tools to handle and control the existing uncompleted documented software system. The proposed system has been tested using synthetic test source codes of medium-scale 4GL application. The results are satisfactory. The information on how many related modules of program source codes will be affected as the consequence of the changes of source codes, is necessary in the early stage. The developer is able to manage the risk of the changes of source codes and to reduce the ripple effect that means when errors occur at one location and propagate through a system.

**Key words:** data coupling, source code control system, functional dependency, 4GL.

## 1. Introduction

Software configuration management is a group of activities to manage systematically the changes to the software configuration items during the lifecycle of software system. Four basic activities of software configuration management are identification, change control, status accounting and auditing [1]. Software configuration items include program codes and documents. It is necessary to ensure that the original program codes are documented during the software system project – especially in large-scale project that the changes of configuration items always occur.

Source code control system is a UNIX based source code revision control system [2]. It is the first generation of software configuration management tools surveyed in [3] that provides version control to source code. However, the source code control system does not provide the recommendation of the related modules to be inspected as the consequence of the changes of any version of source codes. Any error may occur in the related modules and needed to be indicated in the early stage.

In our approach, data coupling dependency checking feature is added to the source code control system as to enhance the capability of developer to identify a risk of the changes of the source code.

This paper is organized as follows. Section 1 is the introduction. Section 2 describes briefly on the design of our approach - to attach coupling dependency alert to the source code control system. Section 3 describes the experiments and results and section 4 is the conclusion.

## 2. Design of Source Code Control System with Coupling Dependency Alert

In general, version number is used to label all modules of program source codes. A version number is consecutively added as each time an original program module is changed. The latest version of all modules of program source codes will be identified as the most up-to-date document. Moreover, developer is capable of tracking back to previous version of program source codes as needed.

In our design, the concept of delta storage [3] is implemented as to save the storage. It means that source codes common to more than one version is not duplicated. The latest versions of all modules of program source codes are stored with a set of script files describing how to convert the available version back to previous versions. Figure 1 shows that source codes of version 1.2 is finally extracted

after source codes version 1.3 has been converted from the latest version of source code, version 1.4. With this design, the source code control system is expected to manage unlimited numbers of previous versions of program source codes without any complication of the file management.

In addition, data coupling dependency checking is another distinct feature for developer. Our basic idea is how to provide a developer a set of coupling dependent modules to be concerned during the changes of the program source codes. Data coupling dependency [4] among program modules is detected.

In our approach, the changes of 'Numbers of parameters' and 'Types of parameters' are considered. Figure 2 shows a sample of the change of 'Numbers of parameters' of function 'findshare\_commtype(scode)'. The new function 'findshare\_commtype(scode,pdate)' has two parameters. In this situation, the data coupling dependency between function 'createtrade()' in module 'posttrade.4gl' and function 'findshare\_commtype()' in module 'calcomm.4gl' is concerned. The developer should be alerted and informed that module 'posttrade.4gl' needed to be inspected and correct as a new version, if possible.

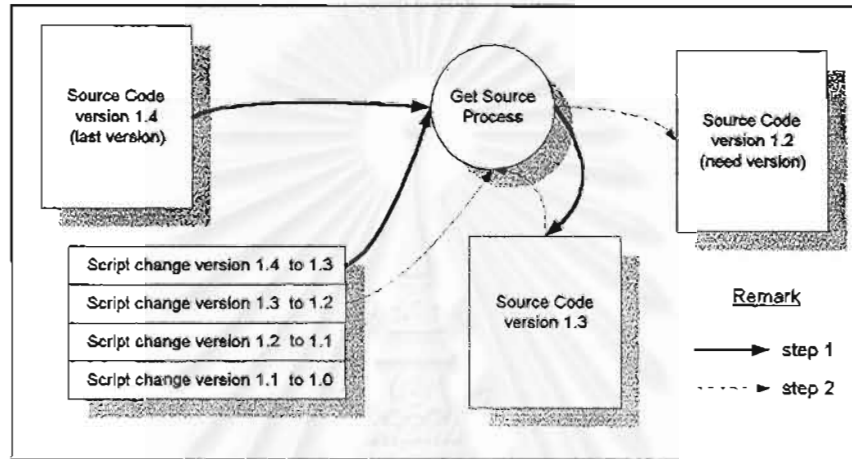


Figure 1: Sample of previous version of program source codes using delta storage

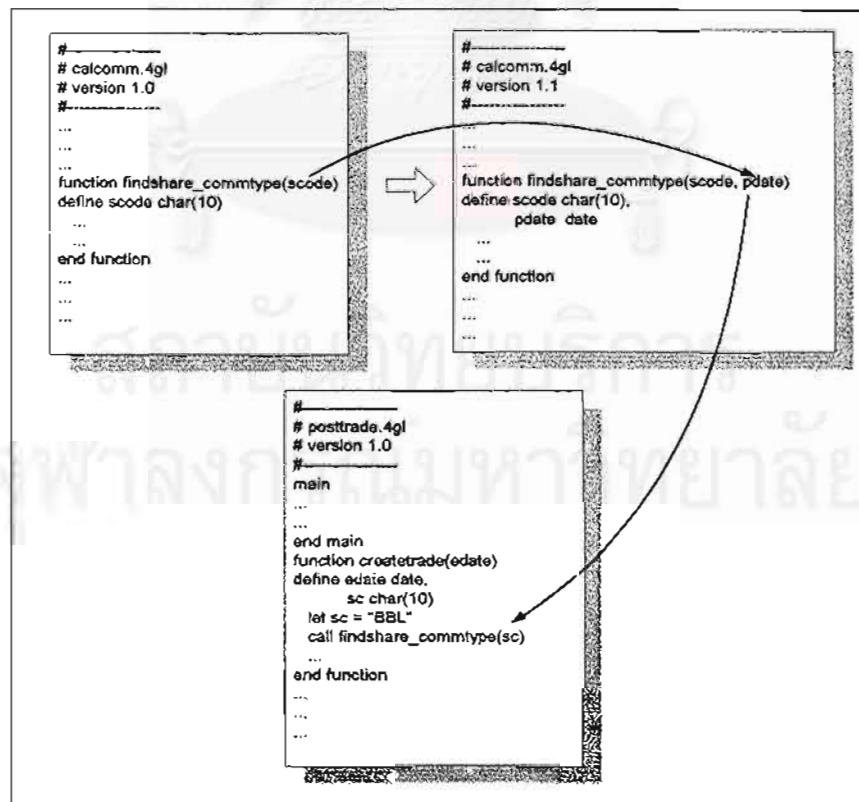


Figure 2: Sample of the change of 'Numbers of parameters'



A sample of the change of 'Types of parameters' is shown in figure 3. As the data coupling dependency between module 'posttrade.4gl' and module 'calcomm.4gl' has been concerned, the change of parameter of function 'findshare\_commttype()' from 'scode as char(10)' to 'shareid as integer' will alert the developer that function 'createtrade()' in module 'posttrade.4gl' should be inspected consequently.

### 3. Experiments

In our experiment, we investigate at least a medium-scale of software system - a software system developed by using 4GL INFORMIX. The

software system consists of 60 modules of program source codes. Each module contains approximately 30 functions. Each function is written at least 10 lines of 4GL instructions. We synthetically generate test source codes with at least 1,800 functions. 10 percents of 1,800 functions are subroutines in library modules. It means that more than 360 subroutine invoking will occur if each subroutine is invoked at least twice.

The results are correctly reported in [5] as the changes of program source codes occur. Figure 4 shows a sample of the script files to convert the latest version 1.1 back to version 1.0.

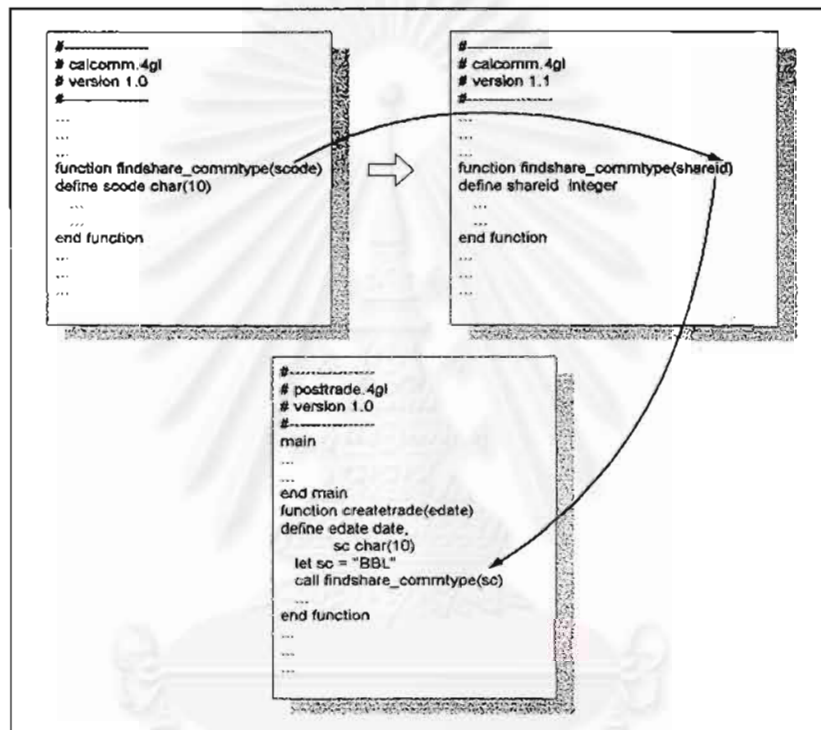


Figure 3: Sample of the change of 'Types of parameters'

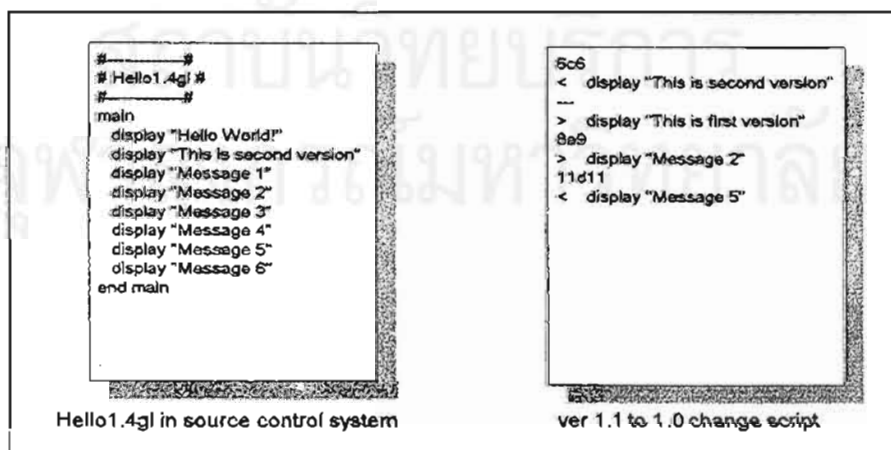


Figure 4: Sample of script files

#### 4. Conclusion

In our approach, developer will be provided with an alternative to manage program source codes during software system lifecycle. With an extended feature of detecting the functional dependency among modules of source codes, any violation to changes of source codes is alerted. The proposed system is capable of automatic controlling the version of the program source codes and maintaining the data coupling dependency among the modules of source codes at the same time.

The contribution of this development is indicated as the automatic tool to practically support developer's activity during program source code changes. The information on how much the risk of the changes will occur to the rest of the related modules is available in the early stage.

#### 5. References

[1] K.J. Choi, K.B. Kim, S.I. Jin, "Software Configuration Management Tools", IEEE

Pacific Rim Conference on Communications, Computers and Signal Processing, pp.430-433, 1997.

- [2] Rochkind M., "The Source Code Control System", IEEE Transactions on Software Engineering, December, 1975.
- [3] A.K.F. Chan, S. Hung, "A Modelling Method of Software Configuration Change Control", Proceedings of Eighth IEEE International Workshop on Software Technology and Engineering Practice, pp.238-250, 1997.
- [4] R.S. Pressman, "Software Engineering - A Practitioner's Approach", Fourth Edition, pp.359-361, McGraw-Hill, 1997.
- [5] S. Tachapisuth, "A Development of Source Code Version Management", Technical Report, Software Engineering Laboratory, Department of Computer Engineering, Chulalongkorn University, 1999.



## ประวัติผู้เขียน

นายสุเมธ เตชาพิสุทธิ์ เกิดวันที่ 6 สิงหาคม พ.ศ. 2512 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับเตรียมอุดมจาก โรงเรียน สวนกุหลาบวิทยาลัย เมื่อปีการศึกษา 2529 และ สำเร็จการศึกษาปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาคณิตศาสตร์ประยุกต์ จากสถาบันเทคโนโลยี พระจอมเกล้า ธนบุรี เมื่อปีการศึกษา 2533 จากนั้นเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2539

