รายการอ้างอิง

[1] Shari Lawrence Pfleeger and Shawn A. Bohner. A Framework for Software Maintenance Metrics. IEEE Transactions on Software Engineering, 1990.

[2] R. S. Arnold and S. A. Bohner. Impact Analysis – Towards A Framework for Comparison. Proceedings of the Conference on Software Maintenance, 1993.

[3] S. Yau and J. Collofello. Some stability measures for software maintenance. IEEE Transactions on Software Engineering, 1980.

[4] Boehm, Barry W. Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[5] S. Yau, J. Collofello and T. MacGregor. Ripple effect analysis of software maintenance. Computer Software and Applications Conference, 1978.

[6] Michelle L. Lee. Change Impact Analysis of Object-Oriented Software. Technical Report, George Mason University, 1998.

[7] Mahmoud O. Elish and David Rine. Investigation of Metrics for Object-Oriented Design Logical Stability. Proceedings of the Seventh European Conference On Software Maintenance And Reengineering, 2003.

[8] James Rumbaugh, Ivar Jacobson and Grady Booch. The Unified Modeling Language Reference Manual. Massachusetts:Addison-Wesley , 1999.

[9] กัลยา วานิชย์บัญชา. การวิเคราะห์สถิติ : สถิติสำหรับการบริหารและวิจัย. พิมพ์ครั้งที่ 3. กรุงเทพมหานคร : ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย, 2539.

[10] Conte, S.D., Dunsmore, H.E., and Shen, V.Y. Software engineering metrics and models. Benjamin/Cummings Publishing Company, 1986.

[11] L. Li and J. Offutt. Algorithmic Analysis of the Impact of Changes to Object-Oriented Software. Proceedings of the International Conference on Software Maintenance, 1996.

[12] D. Kung, J. Gao, P. Hsia, F. Wen, Y. Toyoshima, and C. Chen. Change Impact Identification in Object Oriented Software Maintenance. Proceedings of the International Conference on Software Maintenance, 1994.

[13] รายการรีแฟคเตอริง แหล่งที่มา: http://www.refactoring.com [2004, January 6]

[14] SDMetrics Home Page. http://www.sdmetrics.com

[15] MagicDraw UML Home Page. http://www.magicdraw.com

[16] Java source code, Available from : http://sourceforge.net [2007 ,January 16]

[17] Java source code. Available from : http://www.planet-source-code.com [2007, January 16]

[18] ธีรเดช แซ่ตั้น. การใช้มาตรวัดเชิงวัตถุทำนายเสถียรภาพของเมทธอด. วิทยานิพนธ์ปริญญามหาบัณฑิต, ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2547.

ภาคผนวก

ภาคผนวก ก

อัลกอริทึมหาค่าความเสถียรเชิงตรรกะของคลาส

ในส่วนนี้จะแสดงถึงอัลกอริทึมที่ใช้ในการหาค่าความเสถียรเชิงตรรกะของคลาสจากซอร์สโคด โดยมีรายละเอียดดังแสดงในตารางที่ ก-1

ตารางที่ ก-1 อัลกอริทึมหาค่าความเสถียรเชิงตรรกะของคลาส

```
BEGIN
    TNC = 0
    FOR each class C DO
        FOR each type of System level change I DO
            IF I can be applied to C THEN
                IC = {set of classes impacted by applying I to C}
                FOR each class I in IC DO
                    NTE_I = NTE_I + 1
                ENDIF
                TNC = TNC + 1
            ENDIF
        ENDFOR
    ENDFOR
    FOR each attribute A in C DO
        FOR each type of attribute change J DO
            IF J can be applied to A THEN
                IC = {set of classes impacted by applying J to A}
                FOR each class I in IC DO
                    NTE_I = NTE_I + 1
                ENDIF
                TNC = TNC + 1
            ENDIF
```

ตารางที่ ก-1 อัลกอริทึมหาค่าความเสถียรเชิงตรรกะของคลาส (ต่อ)

```
            ENDFOR
        ENDFOR
        FOR each method M in C DO
            FOR each type of method change K DO
                IF K can be applied to M THEN
                    IC = {set of classes impacted by applying K to M}
                    FOR each class I in IC DO
                        NTE_I = NTE_I + 1
                    ENDIF
                    TNC = TNC + 1
                ENDIF
            ENDFOR
        ENDFOR
    ENDFOR
    FOR each class I DO
        CLS_I = 1- (NTE_I / TNC)
    ENDFOR
END
```

สำหรับความหมายของตัวแปรในอัลกอริทึมมีดังนี้

- C คือ คลาสที่สนใจ
- A คือ แอททริบิวต์ของคลาสที่สนใจ
- M คือ เมทธอดของคลาสที่สนใจ
- I คือ รูปแบบการเปลี่ยนแปลงระดับระบบ
- J คือ รูปแบบการเปลี่ยนแปลงระดับคลาสในส่วนของแอททริบิวต์
- K คือ รูปแบบการเปลี่ยนแปลงระดับในส่วนของเมทธอด

- TNC คือ จำนวนครั้งของการเปลี่ยนแปลงที่เป็นไปได้ทั้งหมด (Total Number of Changes)
- NTE_i คือ จำนวนครั้งที่แต่ละคลาสได้รับผลกระทบ (Number of Time class recieved Effects)

ภาคผนวก ข

ความสัมพันธ์ระหว่างการทำรีแฟคเตอริงกับการเปลี่ยนแปลง

ในส่วนนี้จะแสดงความสัมพันธ์ระหว่างรูปแบบการเปลี่ยนแปลงในตารางที่ 3.1 กับรูปแบบการทำรีแฟคเตอริงต่างๆ

การทำรีแฟคเตอริง คือการจัดรูปแบบใหม่เพื่อให้รูปแบบโครงสร้างมีความถูกต้องเหมาะสม การจัดรูปแบบใหม่ทำได้โดยการเปลี่ยนแปลงแก้ไขส่วนหนึ่งของโปรแกรม ดังนั้นกระบวนการในการทำรีแฟคเตอริงจึงเกิดจากการประกอบกันของรูปแบบการเปลี่ยนแปลงในตารางที่ 3.1 เช่น การย้ายเมทธอด (Move Method) คือการลบเมทธอดจากคลาสหนึ่ง และเพิ่มเมทธอดในอีกคลาสหนึ่ง หรือการย้ายเมทธอดประกอบไปด้วยรูปแบบการเปลี่ยนแปลง 2 รูปแบบ คือ รูปแบบที่ 22 ลบเมทธอด และรูปแบบที่ 21 เพิ่มเมทธอด รายละเอียดแสดงความสัมพันธ์ทั้งหมดระหว่างรูปแบบการทำรีแฟคเตอริงกับรูปแบบการเปลี่ยนแปลงแสดงดังตารางที่ ข-1 โดยด้านซ้ายคือรูปแบบการทำรีแฟคเตอริง และด้านขวาคือรูปแบบการเปลี่ยนแปลงที่ใช้ในการทำรีแฟคเตอริงนั้นๆ

ตารางที่ ข-1 ความสัมพันธ์ระหว่างการทำรีแฟคเตอริงกับการเปลี่ยนแปลง

รีแฟคเตอริง	การเปลี่ยนแปลง
Add Parameter	24
Change Bidirectional Association to Unidirectional	10
Change Reference to Value	10,9
Change Unidirectional Association to Bidirectional	9
Change Value to Reference	9,10
Collapse Hierarchy	4,14
Consolidate Conditional Expression	26
Consolidate Duplicate Conditional Fragments	26
Convert Dynamic to Static Construction by Gerard M. Davison	26
Convert Static to Dynamic Construction by Gerard M. Davison	26

ตารางที่ ข-1 ความสัมพันธ์ระหว่างการทำรีแฟคเตอริงกับการเปลี่ยนแปลง (ต่อ)

รีแฟคเตอริง	การเปลี่ยนแปลง
Decompose Conditional	26,21
Duplicate Observed Data	22,13,9
Eliminate Inter-Entity Bean Communication	1
Encapsulate Collection	26,21
Encapsulate Downcast	26
Encapsulate Field	18,21
Extract Class	16,22,13,9
Extract Interface	1
Extract Method	26,21
Extract Package by Gerard M. Davison	*
Extract Subclass	16,22,3
Extract Superclass	16,22,1
Form Template Method	26,21,22
Hide Delegate	22,21,10,9
Hide Method	27
Hide presentation tier-specific details from the business tier	*
Inline Class	16,22,14,15,21
Inline Method	26,22
Inline Temp	16
Introduce A Controller (Link only)	26,21,13
Introduce Assertion	26
Introduce Business Delegate (Link only)	3

ตารางที่ ข-1 ความสัมพันธ์ระหว่างการทำรีแฟคเตอริงกับการเปลี่ยนแปลง (ต่อ)

รีแฟคเตอริง	การเปลี่ยนแปลง
Introduce Explaining Variable	26,15
Introduce Foreign Method	21
Introduce Local Extension	13 หรือ 3
Introduce Null Object	3
Introduce Parameter Object	13,24,26
Introduce Synchronizer Token (Link only)	11
Localize Disparate Logic (Link only)	13
Merge Session Beans (Link only)	14,15,21
Move Business Logic to Session (Link only)	
Move Class by Gerard M. Davison	*
Move Field	16,15
Move Method	22,21
Parameterize Method	22,21
Preserve Whole Object	26
Pull Up Constructor Body	21,26
Pull Up Field	16,15
Pull Up Method	22,21
Push Down Field	16,15
Push Down Method	22,21
Reduce Scope of Variable by Mats Henricson	26
Refactor Architecture by Tiers	*
Remove Assignments to Parameters	26

ตารางที่ ข-1 ความสัมพันธ์ระหว่างการทำรีแฟคเตอริงกับการเปลี่ยนแปลง (ต่อ)

รีแฟคเตอริง	การเปลี่ยนแปลง
Remove Control Flag	26
Remove Double Negative by Ashley Frieze and Martin Fowler	26,21
Remove Middle Man	22,21,10,9
Remove Parameter	24
Remove Setting Method	22
Rename Method	23
Replace Array with Object	26 หรือ 13
Replace Assignment with Initialization by Mats Henricson	26
Replace Conditional with Polymorphism	26
Replace Conditional with Visitor by Ivan Mitrovic	26,13
Replace Constructor with Factory Method	26,22,21
Replace Data Value with Object	16,5
Replace Delegation with Inheritance	10,3
Replace Error Code with Exception	26
Replace Exception with Test	26
Replace Inheritance with Delegation	4,9
Replace Iteration with Recursion by Dave Whipp	26
Replace Magic Number with Symbolic Constant	26,15
Replace Nested Conditional with Guard Clauses	26
Replace Parameter with Explicit Methods	26,21
Replace Parameter with Method	26
Replace Record with Data Class	13,26

ตารางที่ ข-1 ความสัมพันธ์ระหว่างการทำรีแฟคเตอริงกับการเปลี่ยนแปลง (ต่อ)

รีแฟคเตอริง	การเปลี่ยนแปลง
Replace Recursion with Iteration by Ivan Mitrovic	26
Replace Static Variable with Parameter by Marian Vittek	24,26
Replace Subclass with Fields	22,15,26
Replace Temp with Query	26,21
Replace Type Code with State/Strategy	16,9
Replace Type Code with Subclasses	16,3
Reverse Conditional by Bill Murphy and Martin Fowler	26
Self Encapsulate Field	26,21
Separate Data Access Code (Link only)	26,9
Separate Query from Modifier	22,21
Split Loop by Martin Fowler	26
Split Temporary Variable	26
Substitute Algorithm	26
Use a Connection Pool (Link only)	*
Wrap entities with session (Link only)	*

* ไม่สามารถระบุได้

ภาคผนวก ค

ข้อมูลที่ใช้ในการประเมินผลโมเดล

ในส่วนนี้จะแสดงข้อมูลที่ใช้ในการประเมินผลโมเดลการประมาณค่าความเสถียรเชิงตรรกะของทั้ง 3 กลุ่มซอฟต์แวร์

ตารางที่ ค-1 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการคำนวณ

Class Name	NumOps	NumDesc	NumAnc	AttrInh	ClassifInst	MsgSent	Class Stability	Estimated Class Stability	MRE
JCalc	1	0	0	0	1	3	0.954386	0.979128	0.024742
JCalcBuffer	20	1	0	0	47	8	0.8	0.916824	0.116824
JCalcList	12	0	1	1	21	16	0.629825	0.955721	0.325896
JCalcMenuBar	6	0	0	0	8	1	0.738597	0.982673	0.244077
JCalcStandardFrame	23	0	0	0	26	34	0.498246	0.973232	0.474987
CalcMachineNumber	0	0	0	0	20	4	0.919964	0.911018	0.008946
_func	3	0	0	0	18	56	0.967626	0.87933	0.088296
CalculatorException	0	0	0	0	34	29	0.968525	0.839752	0.128773
CalculatorTester	0	0	0	0	12	14	0.938849	0.927039	0.01181

ตารางที่ ค-1 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการคำนวณ (ต่อ)

Class Name	NumOps	NumDesc	NumAnc	AttrInh	ClassifInst	MsgSent	Class Stability	Estimated Class Stability	MRE
CalculatorTest	0	0	0	0	6	4	1	0.956855	0.043145
E	0	0	0	0	5	1	0.952338	0.963181	0.010843
Entries	0	0	0	0	16	9	0.948741	0.919029	0.029712
Entry	0	0	0	0	2	1	1	0.973003	0.026997
GuiCommandLine	1	0	0	0	9	6	0.940648	0.949884	0.009237
jcalc	1	0	0	0	7	7	0.958633	0.955415	0.003218
jcalc_applet	0	0	0	0	1	2	0.980216	0.97526	0.004956
jcalc_math	0	0	0	0	13	4	0.936151	0.933937	0.002214
jcalc_trig	2	0	0	0	17	5	0.869604	0.929595	0.059991
operatorChecker	0	0	0	0	1	0	0.94964	0.977294	0.027654
OperatorControlCenter	3	0	0	0	19	17	0.94964	0.915726	0.033914
PI	0	0	0	0	13	2	0.928957	0.935971	0.007014
ResultsList	1	0	0	0	13	9	0.923561	0.933737	0.010175

ตารางที่ ค-1 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการคำนวณ (ต่อ)

ResultList_MouseListener	1	0	0	0	6	0	0.958633	0.96581	0.007177
VariableTable	0	0	0	0	6	6	0.954137	0.954821	0.000684

ตารางที่ ค-2 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการจัดการข้อความ

Class Name	NumOps	NumPubOps	Class Stability	Estimated Class Stability	MRE
TWedit	2	2	0.9502165	0.984849	0.034632
TWedit_Frame1	25	15	0.32467532	0.559524	0.234849
TWedit_Frame1_AboutBox	5	2	0.8095238	0.969963	0.160439
MyTextArea	9	9	0.8773424	0.782546	0.094797
AboutDialog	5	1	0.9199319	0.993901	0.07397
FindReplaceDialog	5	1	0.8994889	0.993901	0.094413
MainWindow	28	9	0.49403745	0.688269	0.194232
TextDocumentListener	6	5	0.9131175	0.893186	0.019932
WindowSizer	6	5	0.91141397	0.893186	0.018228

ตารางที่ ค-3 ข้อมูลที่ใช้ในการประเมินผลโมเดลด้านการจัดการรูปภาพ

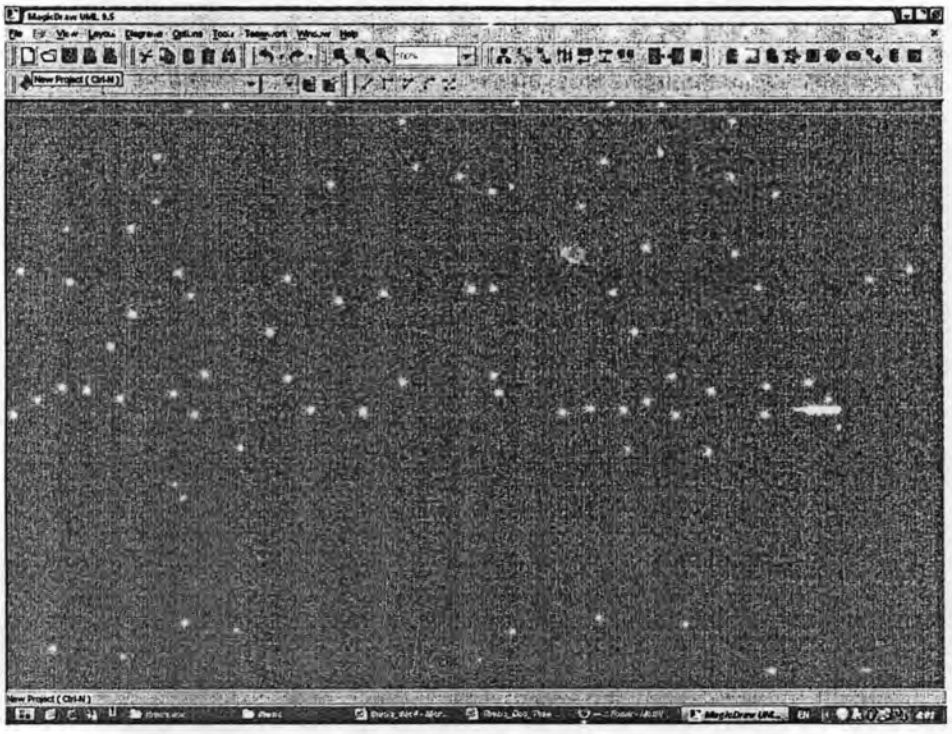
Class Name	NumAttr	NumOps	IfImpl	NumAnc	CLD	OpsInh	Class Stability	Estimated Class Stability	MRE
filterObject	0	8	0	0	0	0	0.901709	0.974056	0.072347
graphicsfile	0	2	0	0	0	0	0.965812	0.99035	0.024538
graphicswindow	9	5	0	0	0	0	0.489316	0.922232	0.432916
JDrawerPane	0	2	0	0	0	0	0.963675	0.99035	0.026674
jgraphicsframe	0	16	0	0	0	0	0.75	0.952332	0.202332
IMGCanvas	0	0	0	1	0	13	0.791339	0.919826	0.128487
IMGCircle	0	0	0	2	0	14	0.937008	0.977229	0.040221
IMGEllipse	0	0	0	1	1	13	0.811024	0.891765	0.080741
IMGen	0	0	0	0	0	0	0.852362	0.995781	0.143419
IMGImage	0	0	1	0	0	0	0.830709	0.92061	0.089901
IMGLine	0	0	0	1	0	13	0.78937	0.919826	0.130456
IMGRectangle	0	0	0	1	1	13	0.78937	0.891765	0.102395
IMGShape	0	0	1	0	2	0	0.751969	0.864488	0.11252
IMGSquare	0	0	0	2	0	16	0.913386	0.955003	0.041617
IMGText	0	0	1	0	0	0	0.814961	0.92061	0.105649

ภาคผนวก ง

ตัวอย่างวิธีการแปลงกลับ

การแปลงกลับแผนภาพจากซอร์สโคด เพื่อสร้างแผนภาพคลาสและแผนภาพซีเควนซ์เพื่อใช้ในงานวิจัยนี้ ได้อาศัยโปรแกรม MagicDraw UML เวอร์ชัน 9.5 ซึ่งเป็นโปรแกรมสำหรับการวาดแผนภาพยูเอ็มแอล และมีระบบสนับสนุนการทำงานเพื่อช่วยในการออกแบบซอฟต์แวร์ และมีคุณสมบัติในการแปลงกลับแผนภาพจากซอร์สโคดอีกด้วย สำหรับรายละเอียดทั้งหมดของโปรแกรม MagicDraw UML สามารถหาได้ที่ www.magicdraw.com โดยในที่นี้จะแสดงการทำงานในส่วนของการแปลงกลับแผนภาพคลาสและแผนภาพซีเควนซ์จากซอร์สโคดเท่านั้น ซึ่งมีการทำงานดังนี้

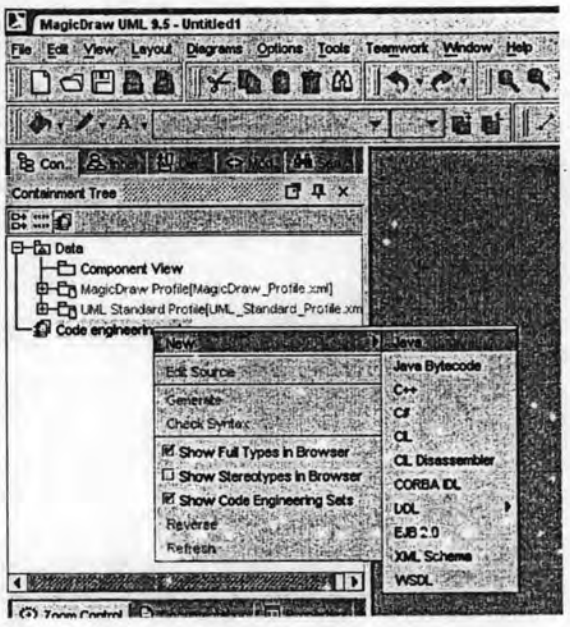
เมื่อเปิดโปรแกรม MagicDraw UML ขึ้นจะปรากฏหน้าต่างการทำงานหลักมีลักษณะดังรูปที่ ง-1 โดยจะแบ่งออกเป็น 2 ส่วนคือ ส่วนของเมนูการทำงานด้านบน และส่วนของพื้นที่การทำงาน



รูปที่ ง-1 หน้าต่างการทำงานหลักของโปรแกรม MagicDraw UML

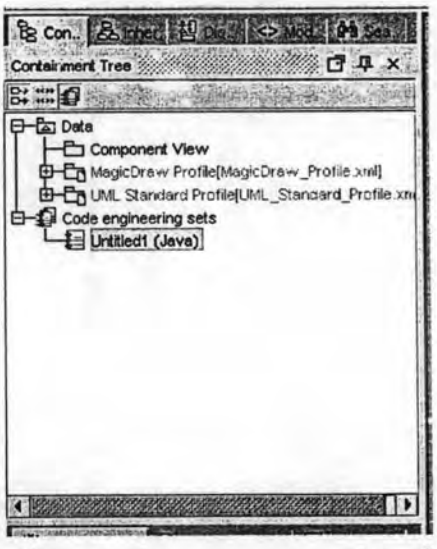
เริ่มการทำงานโดยเลือกคำสั่ง File->New Project เพื่อสร้างโปรเจคต์ใหม่ โดยเมื่อทำการสร้างโปรเจคต์ใหม่จะปรากฏหน้าต่างย่อยทางด้านซ้ายมือของพื้นที่การทำงานหลัก ซึ่งแสดงรายละเอียดโครงสร้างและองค์ประกอบของโปรเจคต์ที่สร้างขึ้น องค์ประกอบที่สำคัญหนึ่งของทุก

โปรเจคต์คือ Code Engineering Set ซึ่งเป็นองค์ประกอบสำหรับสร้าง ระบุและทำงานต่างๆ ที่เกี่ยวข้องกับซอร์สโคด รวมไปถึงการแปลงกลับแผนภาพด้วย ในการแปลงกลับแผนภาพต้องทำการสร้าง Code Engineering Set เสียก่อน โดยคลิกขวาที่ Code Engineering Set แล้วเลือกคำสั่ง new ซึ่งจะปรากฏรายการของประเภทไฟล์ซอร์สโคดของ code engineering set ในที่นี้เลือกรายการเป็นจาวา แสดงดังรูปที่ ง-2



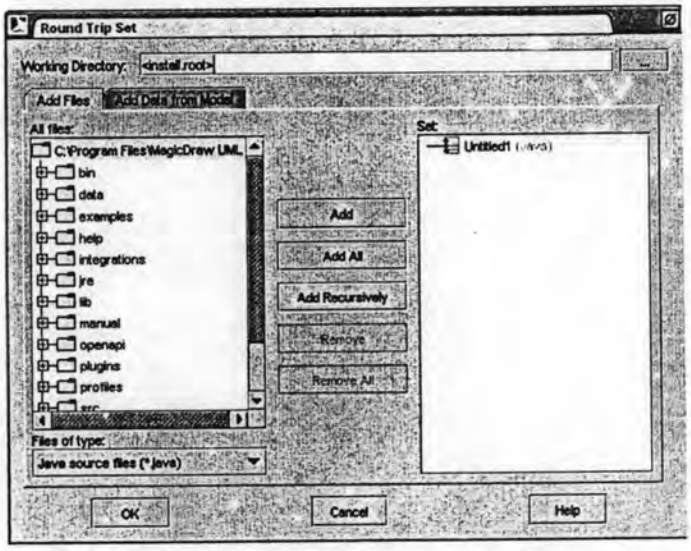
รูปที่ ง-2 การสร้าง code engineering set

เมื่อทำการเลือกประเภทของไฟล์ซอร์สโคดที่ต้องการนำเข้าแล้ว จะปรากฏ code engineering set ที่สร้างขึ้นในหน้าต่างแสดงองค์ประกอบดังรูปที่ ง-3



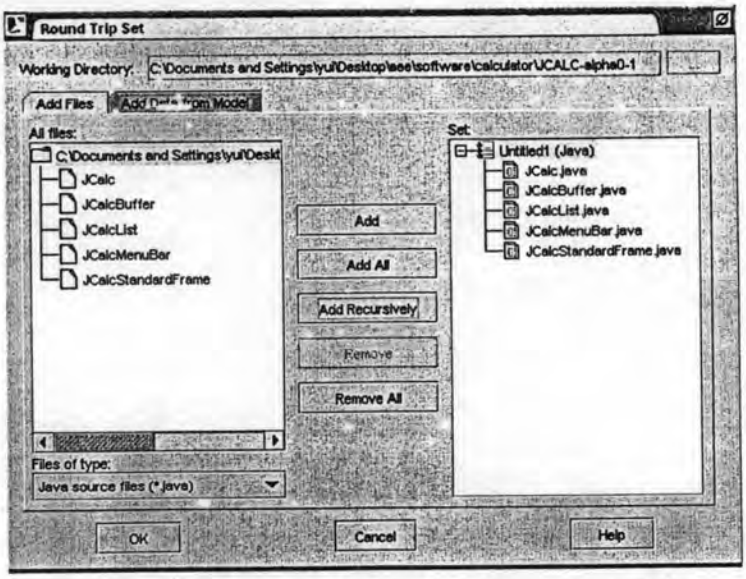
รูปที่ ง-3 code engineering set ที่สร้างขึ้น

เมื่อสร้าง code enginnering set ที่ต้องการแล้วต้องทำการนำเข้าไฟล์ซอร์สโคดของซอฟต์แวร์ที่ต้องการแปลงกลับ โดยคลิกขวาที่ code engineering set ที่สร้างขึ้นแล้วเลือกคำสั่ง edit จากนั้นโปรแกรมจะแสดงหน้าต่างสำหรับเลือกรายการของไฟล์ซอร์สโคดที่ต้องการ ดังรูปที่ ง-4



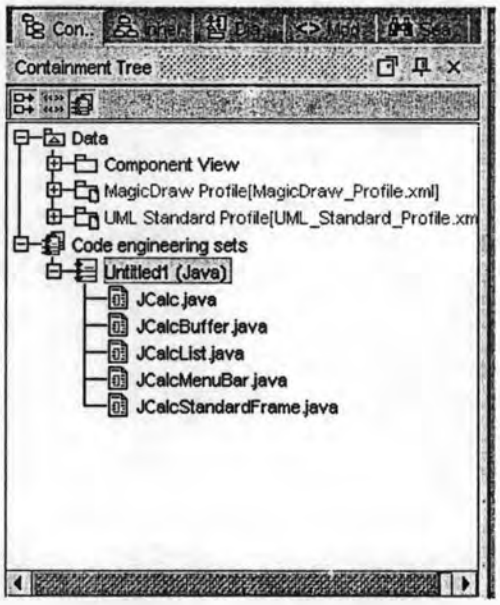
รูปที่ ง-4 หน้าจอสำหรับเลือกไฟล์ซอร์สโคด

ในการเลือกไฟล์สามารถเลือกได้ 3 แบบด้วยกันคือ เลือกไฟล์เดียว (Add) เลือกทั้งหมด (Add All) หรือเลือกแบบวนกลับ (Add Recursively) โดยการเลือกแบบวนกลับ คือการเลือกไฟล์ซอร์สโคดตามประเภทที่กำหนดข้างต้นจากไดเรกทอรีปัจจุบัน และไดเรกทอรีย่อยทั้งหมดโดยอัตโนมัติ เมื่อทำการเลือกไฟล์ที่ต้องการนำเข้าเสร็จสิ้นแล้วจะปรากฏหน้าจอดังรูปที่ ง-5



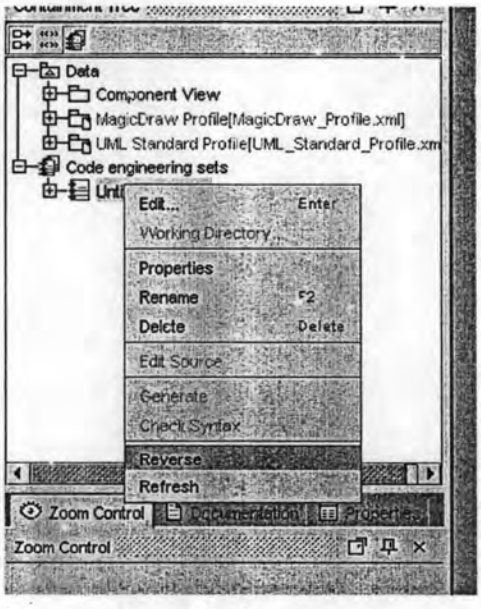
รูปที่ ง-5 หน้าจอสำหรับเลือกไฟล์ซอร์สโคดเมื่อได้ทำการเลือกไฟล์ที่ต้องการแล้ว

หลังจากทำการเลือกไฟล์ที่ต้องการเข้าสู่โปรเจคต์แล้ว องค์ประกอบของ code engineering set จะเปลี่ยนแปลงไปโดยจะปรากฏรายการของไฟล์ซอร์สโคดที่ได้ทำการนำเข้าอยู่ภายใน code engineering set ดังรูปที่ ง-6

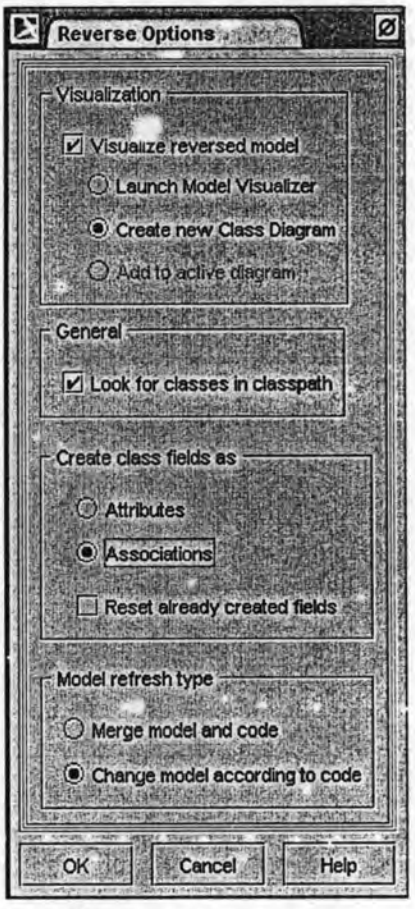


รูปที่ ง-6 รายการของไฟล์ซอร์สโคด

หลังจากทำการนำเข้าไฟล์ซอร์สโคดแล้วต้องทำการแปลงกลับซอร์สโคดที่เลือกเข้ามาให้อยู่ในรูปแบบข้อมูลที่สามารถนำไปใช้ในการสร้างแผนภาพเสียก่อน โดยการคลิกขวาที่ code engineering set ที่สร้างขึ้นแล้วเลือกคำสั่ง Reverse ดังรูปที่ ง-7 จากนั้นจะปรากฏหน้าต่างสำหรับการเลือกตัวเลือกในการแปลงกลับดังรูปที่ ง-8



รูปที่ ง-7 การเลือกคำสั่งเพื่อทำการแปลงกลับข้อมูล

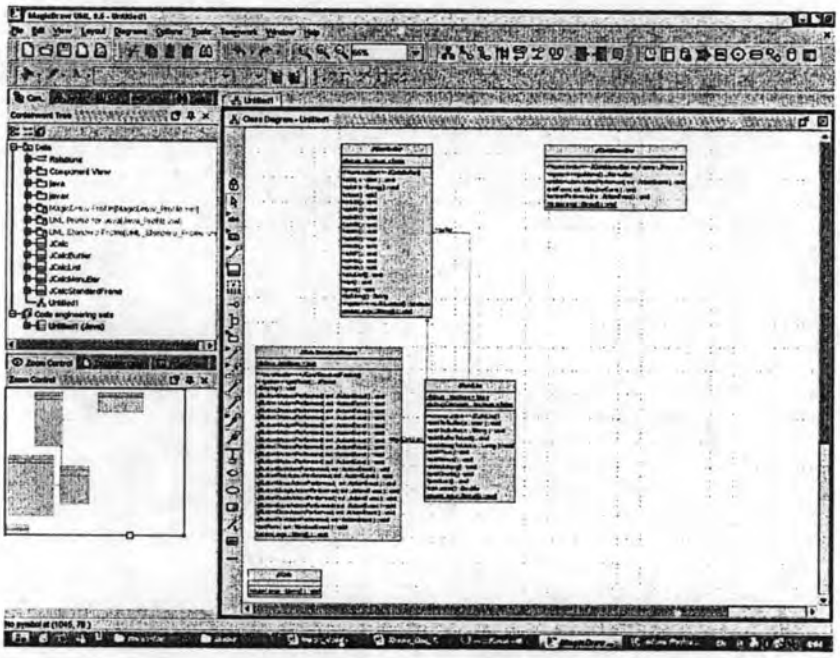


รูปที่ ง-8 ตัวเลือกการแปลงกลับ

ตัวเลือกการแปลงกลับมี 4 ตัวเลือกด้วยกัน ดังนี้

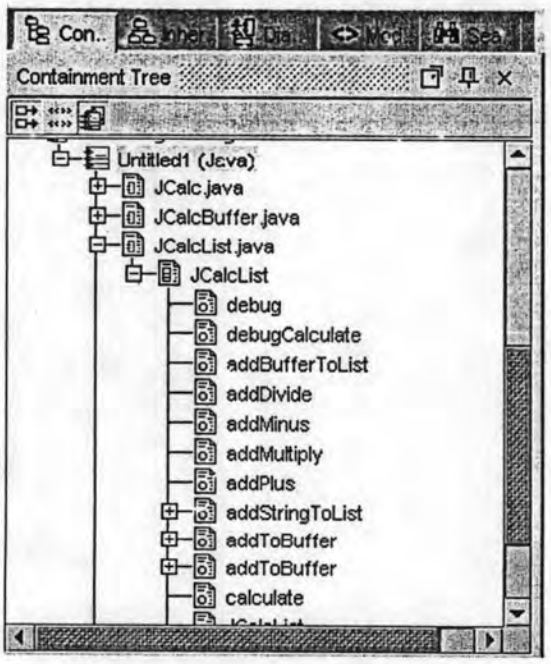
- ตัวเลือก Visualization คือตัวเลือกในการสร้างแผนภาพที่ได้จากการแปลงกลับ โดยสามารถเลือกให้แสดงตัวช่วยสร้างแผนภาพ หรือสร้างแผนภาพคลาสโดยอัตโนมัติ
- ตัวเลือก General คือตัวเลือกให้โปรแกรมทำการมองหาคลาสเพิ่มเติมในคลาสพาธหรือไม่
- ตัวเลือก Create class fields as คือตัวเลือกการแสดงผลความสัมพันธ์แอสโซซิเอชันระหว่างคลาส ให้อยู่ในรูปแบบของแอททริบิวต์ของอีกคลาส หรือแสดงเป็นความสัมพันธ์แอสโซซิเอชัน
- ตัวเลือก Model refresh type คือตัวเลือกเงื่อนไขการทำให้โมเดลและซอร์สโคดสอดคล้องกัน โดยมีตัวเลือกคือ ให้ทั้งโมเดลและซอร์สโคดทำการแก้ไขให้สอดคล้องกันเมื่อมีการแก้ไขฝ่ายหนึ่งฝ่ายใด หรือให้โมเดลทำการแก้ไขให้สอดคล้องกับซอร์สโคดเพียงฝ่ายเดียว

เมื่อทำการแปลงกลับข้อมูลเสร็จสิ้นแล้วหากมีการเลือกให้โปรแกรมทำการสร้างแผนภาพคลาสโดยอัตโนมัติ จะปรากฏแผนภาพคลาสที่สร้างขึ้นอยู่ในหน้าต่างการทำงานดังรูปที่ ง-9



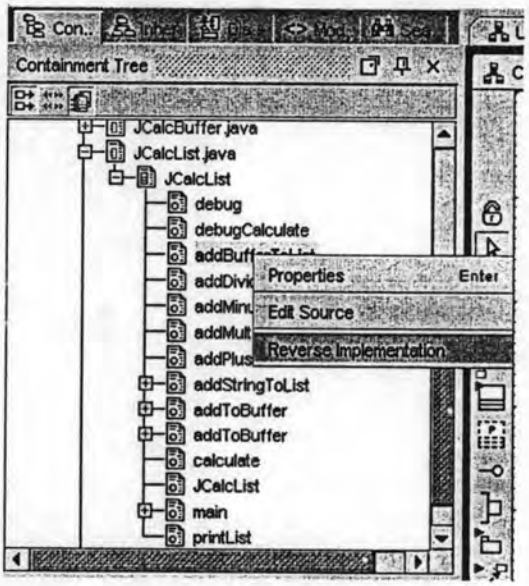
รูปที่ ง-9 แผนภาพคลาสที่ได้จากการแปลงกลับ

เมื่อทำการแปลงกลับข้อมูลจากซอร์สโคดพร้อมทั้งแผนภาพคลาสเสร็จสิ้นแล้ว จะปรากฏการเปลี่ยนแปลงที่รายการของไฟล์ซอร์สโคด โดยจะมีรายการของคลาส แอททริบิวต์ และเมทธอดที่อยู่ในแต่ละไฟล์ปรากฏขึ้นมา ดังรูปที่ ง-10



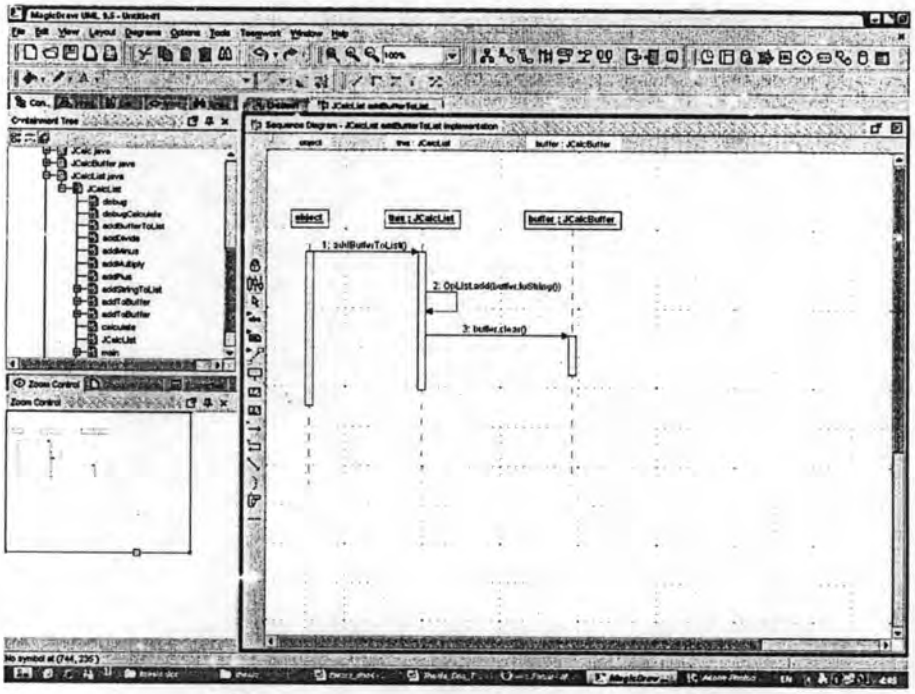
รูปที่ ง-10 รายการคลาสภายในไฟล์ซอร์สโคด

การแปลงกลับแผนภาพซีเควนซ์สามารถทำได้โดย ทำการเลือกเมทธอดที่ต้องการจากนั้นคลิกขวาที่เมทธอดแล้วเลือกคำสั่ง Reverse Implementation ดังรูปที่ ง-11



รูปที่ ง-11 การแปลงกลับแผนภาพซีเควนซ์

เมื่อทำการแปลงกลับแผนภาพซีเควนซ์เสร็จสิ้นแล้วจะปรากฏแผนภาพซีเควนซ์ขึ้นบริเวณหน้าต่างการทำงานดังรูปที่ ง-12 ทั้งนี้ในการทำงานวิจัยนี้ต้องทำการแปลงกลับแผนภาพซีเควนซ์ของทุกๆ เมทธอด ในทุกๆ คลาสเพื่อให้ได้รายละเอียดมากที่สุด

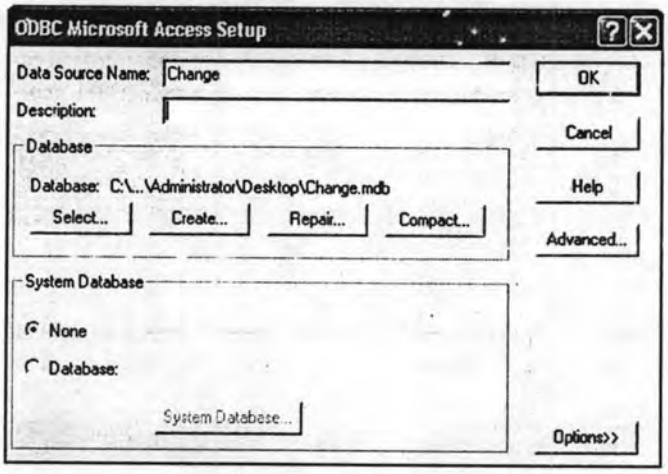


รูปที่ ง-12 แผนภาพซีเควนซ์ที่ได้จากการแปลงกลับ

ภาคผนวก จ

การใช้งานโปรแกรมคำนวณความเสถียรเชิงตรรกะ

โปรแกรมคำนวณความเสถียรเชิงตรรกะที่พัฒนาขึ้นจะทำการคำนวณความเสถียรเชิงตรรกะจากไฟล์ซอร์สโคดที่ผู้ใช้เลือก แล้วบันทึกค่าลงในฐานข้อมูลที่กำหนดไว้ ดังนั้นก่อนการใช้งานโปรแกรมคำนวณความเสถียรเชิงตรรกะ ต้องทำการกำหนดช่องทางติดต่อฐานข้อมูลผ่านโอดีบีซี โดยใช้ ไดรเวอร์ของไมโครซอฟต์แอคเซสและกำหนดดาต้าซอร์สเนม ชื่อ "Change" และไฟล์ดาต้าเบสชื่อ "change.mdb" เสียก่อนดังแสดงในรูปที่ จ-1



รูปที่ จ-1 การตั้งค่าโอดีบีซี

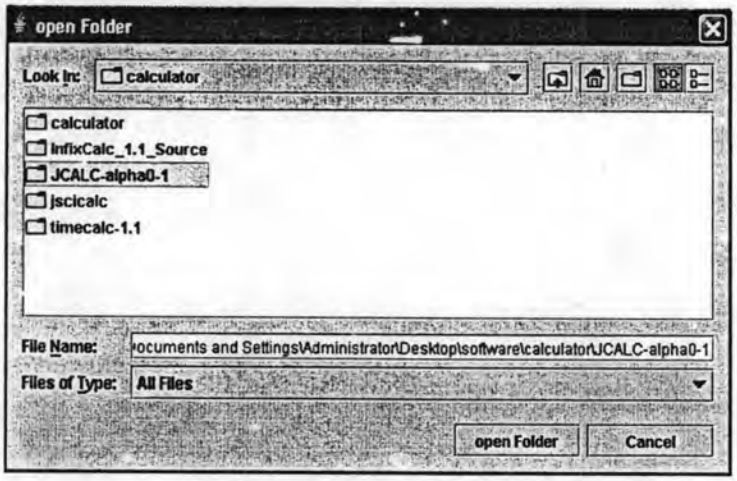
โปรแกรมคำนวณความเสถียรเชิงตรรกะมีการทำงานผ่านทางส่วนต่อประสานผู้ใช้ โดยเมื่อทำการเปิดโปรแกรมจะปรากฏหน้าจอการทำงานหลักดังรูปที่ จ-2

รูปที่ จ-2 หน้าจอการทำงานหลักของโปรแกรมคำนวณความเสถียรเชิงตรรกะ

โดยในหน้าจอการทำงานหลักจะมีปุ่มคำสั่ง 3 ปุ่มบริเวณด้านล่างดังนี้

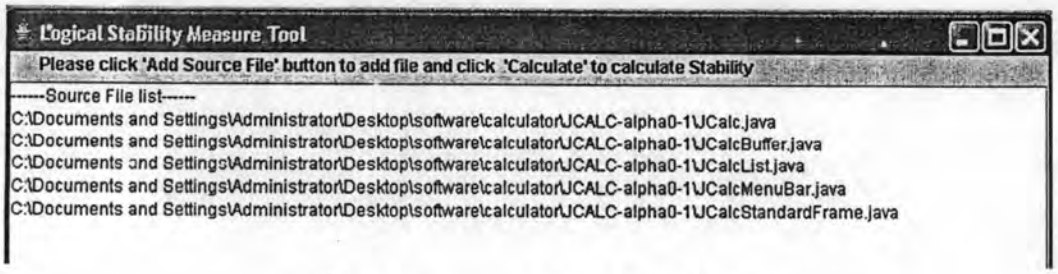
- ปุ่มคำสั่ง Add file สำหรับทำการเลือกไฟล์ซอร์สโคด
- ปุ่มคำสั่ง Calculate สำหรับสั่งให้โปรแกรมทำการคำนวณ
- ปุ่มคำสั่ง Reset สำหรับสั่งให้ทำการล้างข้อมูลทั้งหมด

การเลือกไฟล์ซอร์สโคดเข้าสู่โปรแกรมสามารถทำได้โดยการคลิกปุ่มคำสั่ง Add file และเมื่อทำการคลิกปุ่มคำสั่งแล้ว โปรแกรมจะแสดงหน้าจอสำหรับเลือกไฟล์ซอร์สโคดที่ต้องการ ทั้งนี้การเลือกไฟล์ซอร์สโคดของโปรแกรมคำนวณความเสถียรเชิงตรรกะ ผู้ใช้ต้องทำการเลือกโฟลเดอร์ที่เก็บไฟล์ซอร์สโคดที่ต้องการไว้ จากนั้นโปรแกรมจะทำการอ่านค่าไฟล์ซอร์สโคดทั้งหมดในโฟลเดอร์นั้น และโฟลเดอร์ย่อย หน้าจอการเลือกโฟลเดอร์แสดงดังรูปที่ จ-3

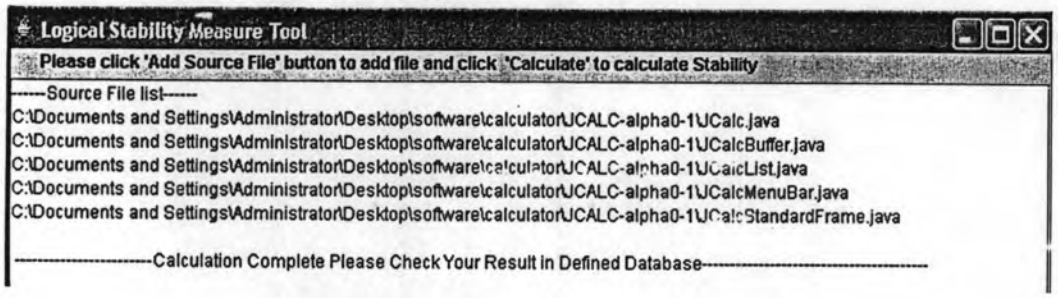


รูปที่ จ-3 การเลือกไฟล์ซอร์สโคด

เมื่อผู้ใช้ทำการเลือกไฟล์ซอร์สโคดเสร็จสิ้น โปรแกรมจะแสดงรายการของไฟล์ที่เลือกทั้งหมดให้ผู้ใช้ได้รับทราบ ดังแสดงในรูปที่ จ-4 จากนั้นผู้ใช้สามารถทำการคำนวณค่าความเสถียรเชิงตรรกะได้โดยคลิกที่ปุ่ม Calculate และโปรแกรมจะทำการคำนวณความเสถียรเชิงตรรกะพร้อมทั้งบันทึกผลลงในฐานข้อมูลที่ระบุไว้โดยอัตโนมัติ แล้วจึงทำการแจ้งให้ผู้ใช้รับทราบดังรูปที่ จ-5 ตัวอย่างข้อมูลที่บันทึกลงในฐานข้อมูลแสดงในรูปที่ จ-6



รูปที่ จ-4 รายการของไฟล์ซอร์สโคดที่เลือก



รูปที่ จ-5 หน้าจอการทำงานเมื่อโปรแกรมคำนวณเสร็จ

ClassTable : ตาราง

ID	ProjName	PackageName	ClassName	ClassStability
1618	null	None	C1	.26086956
1619	null	None	JCalc	.9514019
1620	null	null	WindowAdapter	1
1621	null	None	JCalcBuffer	.7905542
1622	null	None	JCalcList	.6130841
1623	null	None	JCalcMenuBar	.7738318

รูปที่ จ-6 ตัวอย่างข้อมูลในฐานข้อมูล

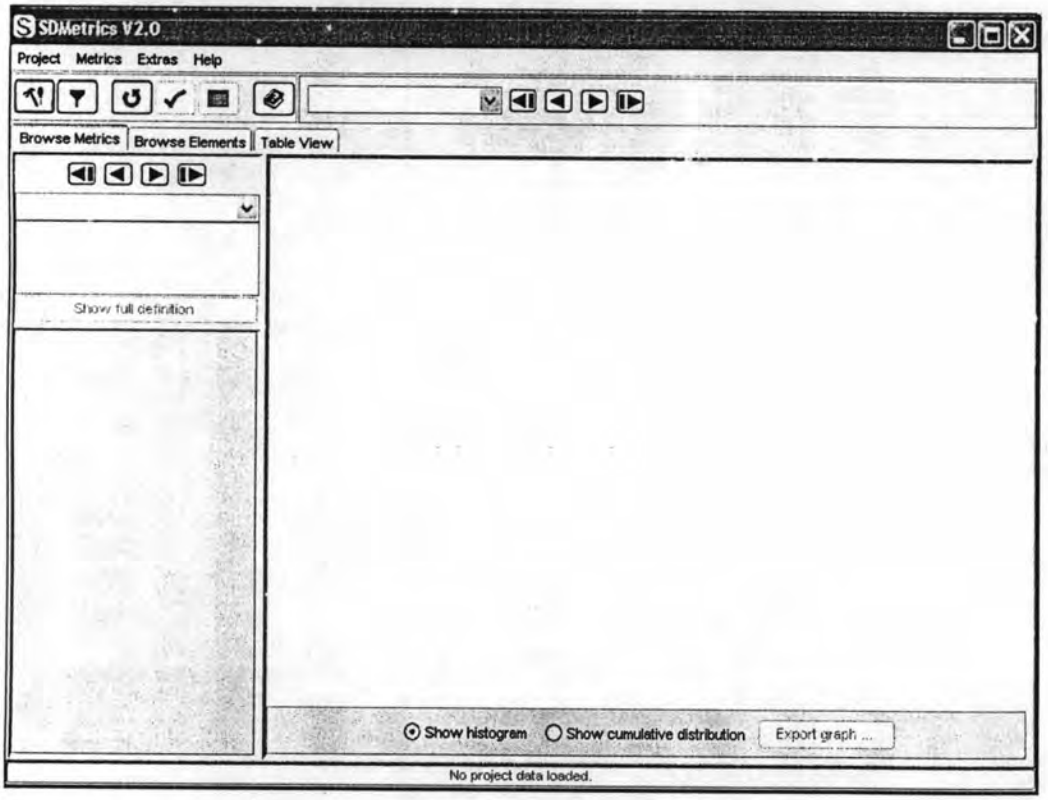
ภาคผนวก ฉ

การใช้งานโปรแกรม SDMetrics

ในที่นี้จะกล่าวถึงการใช้งานโปรแกรม SDMetrics ในส่วนของการใช้งานหลักเพื่อทำการคำนวณค่าตัววัดแผนภาพเท่านั้น สำหรับรายละเอียดของโปรแกรมโดยละเอียดสามารถศึกษาเพิ่มเติมได้ในเวบไซต์ของโปรแกรมในรายการอ้างอิง

โปรแกรม SDMetric มีความสามารถในการคำนวณค่าตัววัดแผนภาพ โดยรับค่าแผนภาพที่อยู่ในรูปแบบไฟล์เอกซ์เอมแอล หรือเอกซ์เอมไอ แล้วทำการอ่านค่าแผนภาพที่ต้องการตามที่มีการระบุไว้ในไฟล์ Metamodel definition โดยวิธีการในการอ่านค่าแผนภาพจากไฟล์ข้อมูลแผนภาพต้องระบุไว้ในไฟล์ XMI Transformation จากนั้นเมื่อได้ข้อมูลแผนภาพที่ต้องการแล้วจึงทำการคำนวณค่าตัววัดแผนภาพต่างๆ ตามที่มีการระบุไว้ในไฟล์ Metrics definition

หน้าจอการทำงานหลักของโปรแกรม SDMetrics แสดงดังรูปที่ ฉ-1 โดยจะมีเมนูคำสั่งและปุ่มคำสั่งต่างๆ อยู่ด้านบน และมีการแสดงผลการคำนวณตัววัดต่างๆ อยู่ด้านล่าง

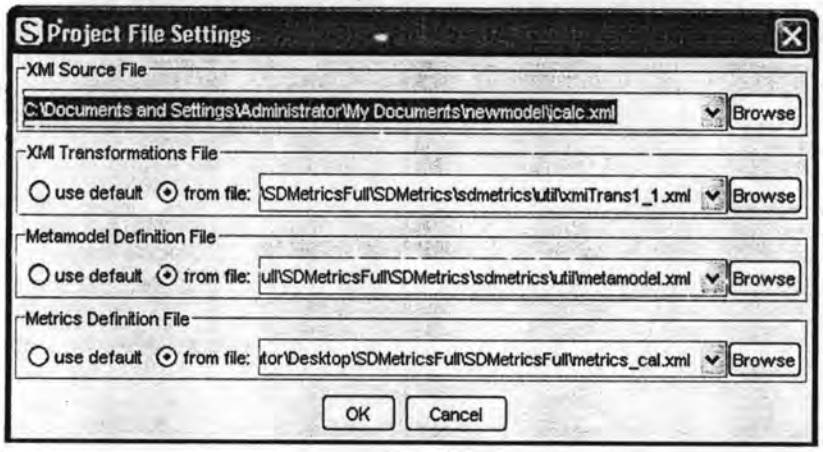


รูปที่ ฉ-1 หน้าจอการทำงานหลักของโปรแกรม SDMetrics

ในการคำนวณตัววัดแผนภาพจะต้องทำการตั้งค่าโปรเจคต์เพื่อระบุข้อมูลที่จำเป็นดังนี้

1. ไฟล์ของข้อมูลแผนภาพ
2. ไฟล์ Metamodel definition ที่ต้องการใช้
3. ไฟล์ XMI transformation ที่ต้องการใช้
4. ไฟล์ Metrics definition ที่ต้องการใช้

การตั้งค่าโปรเจคต์สามารถทำได้โดยการเลือกคำสั่ง Project->Project File Setting บนแถบเมนูด้านบน หรือคลิกปุ่ม บนแถบเครื่องมือด้านบน จากนั้นโปรแกรมจะแสดงหน้าจอสำหรับการตั้งค่าโปรเจคต์ดังรูปที่ ฉ-2

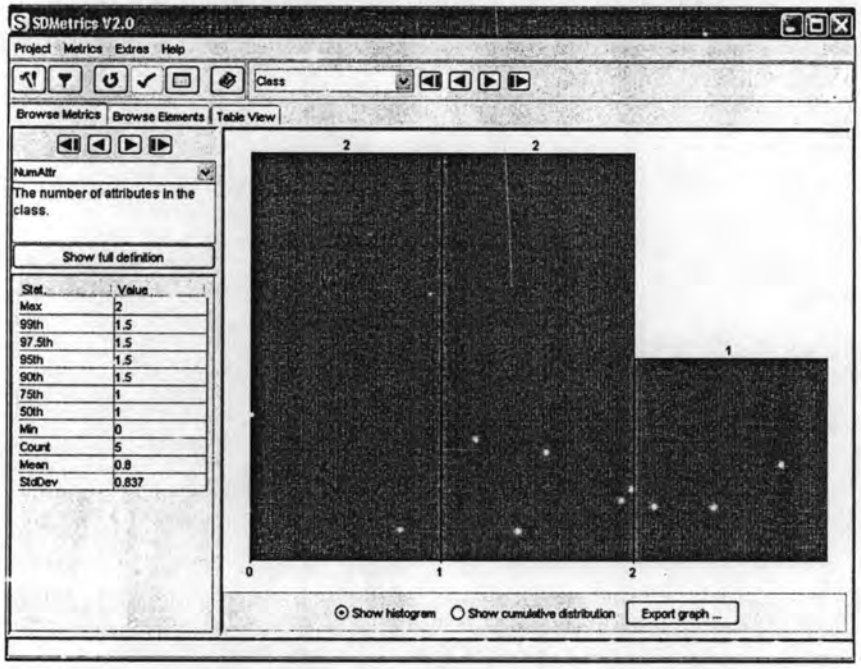


รูปที่ ฉ-2 การตั้งค่าโปรเจคต์

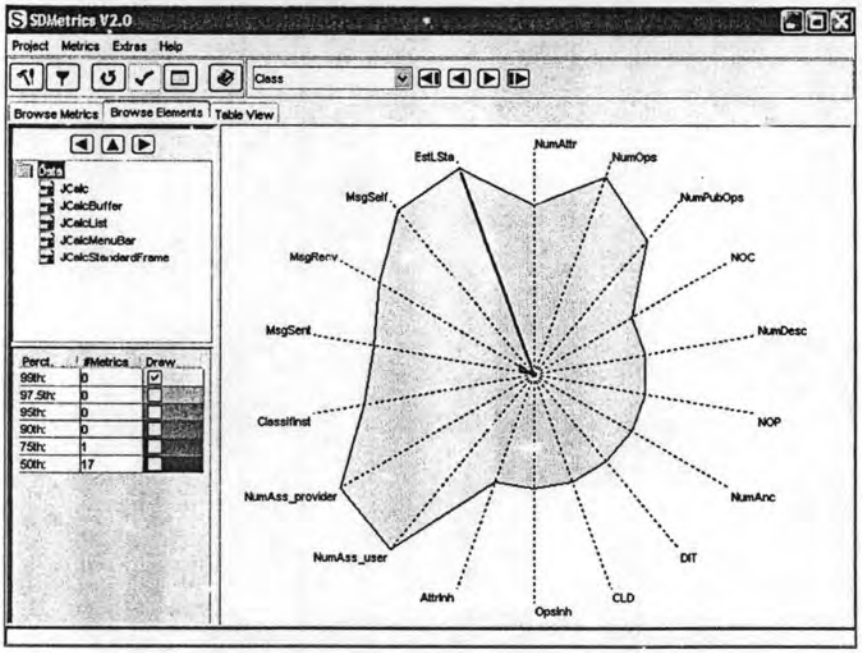
จากรูปที่ ฉ-2 ผู้ใช้ต้องทำการเลือกไฟล์ข้อมูลแผนภาพที่ต้องการ โดยการคลิกปุ่ม Browse เพื่อเลือกไฟล์ที่ต้องการ ในขณะที่ไฟล์ Metamodel definition ไฟล์ XMI Transformation และไฟล์ Metrics definition ผู้ใช้สามารถเลือกไฟล์ที่ต้องการโดยการคลิกปุ่ม Browse หรือเลือกใช้ค่ามาตรฐานของโปรแกรมก็ได้ ทั้งนี้ในงานวิจัยนี้ผู้ใช้ต้องทำการเลือก ไฟล์ Metamodel definition และไฟล์ XMI Transformation ที่กำหนดไว้สำหรับงานวิจัยนี้โดยเฉพาะ เช่นเดียวกับไฟล์ Metrics definition ซึ่งต้องเลือกใช้ไฟล์ที่กำหนดไว้เช่นกัน ยกเว้นเพียงแต่ไฟล์ Metrics definition ที่ใช้ในงานวิจัยนี้จะมี 3 ไฟล์ด้วยกันตามประเภทของซอฟต์แวร์ที่ต้องการ โดยผู้ใช้ต้องทำการเลือกให้ถูกต้อง

หลังจากทำการตั้งค่าโปรเจคต์เสร็จสิ้นแล้วสามารถทำการคำนวณตัววัดแผนภาพได้โดยเลือกคำสั่ง Metrics->Calculate Metrics บนแถบเมนู หรือคลิกปุ่ม บนแถบเครื่องมือ จากนั้นโปรแกรมจะทำการคำนวณค่าตัววัดแผนภาพโดยอัตโนมัติแล้วจึงแสดงผลเมื่อทำการคำนวณเสร็จสิ้น โดยการแสดงผลจะแบ่งออกเป็น 3 แบบ คือ การแสดงผลข้อมูลสถิติตัววัด การแสดงผลข้อมูล

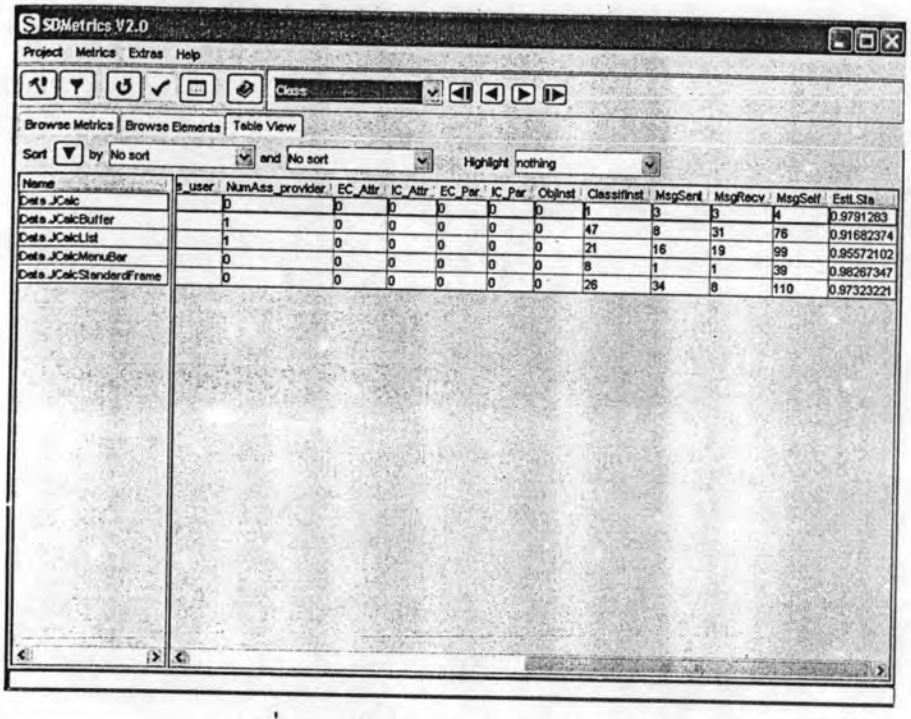
แผนภาพตัววัดขององค์ประกอบ และการแสดงผลแบบตาราง โดยผู้ใช้สามารถเลือกรูปแบบการแสดงผลที่ต้องการได้จากแทบรูปแบบทั้ง 3 คือ Browse Metrics Browse Element และ Table View ภาพหน้าจอการแสดงผลทั้ง 3 แบบแสดงในรูปที่ ฉ-3 รูปที่ ฉ-4 และรูปที่ ฉ-5 ตามลำดับ



รูปที่ ฉ-3 หน้าจอการแสดงผลข้อมูลสถิติตัววัด

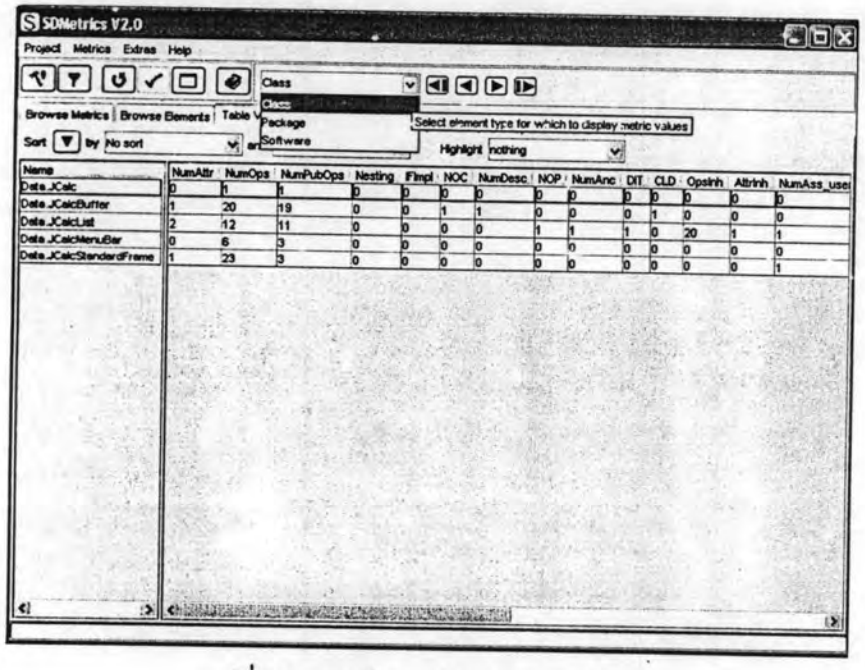


รูปที่ ฉ-4 หน้าจอการแสดงผลข้อมูลแผนภาพตัววัดขององค์ประกอบ



รูปที่ ฉ-5 หน้าจอการแสดงผลแบบตาราง

นอกจากนี้ในการแสดงผลแต่ละรูปแบบยังแบ่งระดับของข้อมูลแผนภาพที่นำมาแสดงผลได้อีกด้วยตามแต่ที่มีการกำหนดไว้ในไฟล์ Metamodel definition โดยในงานวิจัยนี้จะมี 3 ระดับด้วยกัน คือ ระดับคลาส ระดับแพ็คเกจ และระดับซอฟต์แวร์ โดยสามารถเลือกดูข้อมูลในแต่ละระดับได้จากตัวเลือกบนแถบเครื่องมือดังรูปที่ ฉ-6



รูปที่ ฉ-6 การเลือกระดับของการแสดงผล

ประวัติผู้เขียนวิทยานิพนธ์

นาย ศุภวัชร์ รังสิยวัฒน์ เกิดเมื่อวันที่ 3 กันยายน พ.ศ. 2524 จบการศึกษาระดับมัธยมศึกษาจากโรงเรียนสวนกุหลาบวิทยาลัย จากนั้นได้เข้าศึกษาต่อในระดับปริญญาตรี สาขาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี และสำเร็จการศึกษาในปี พ.ศ. 2546 และเข้าศึกษาต่อในระดับปริญญามหาบัณฑิตที่จุฬาลงกรณ์มหาวิทยาลัย ในหลักสูตรวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ในปีเดียวกัน