

รายการอ้างอิง

ภาษาไทย

- กิตติ ภัคดีวัฒนะกุล. 2545. JAVA ฉบับพื้นฐาน. พิมพ์ครั้งที่ 4. กรุงเทพมหานคร: สำนักพิมพ์เคทีพี คอมพ์ แอนด์ คอนซัลท์.
- ชาติ วรรณพิพัฒน์ และ เทพฤทธิ์ บัณฑิตวัฒนาวงศ์. 2544. UML ภาษามาตรฐานเพื่อผู้พัฒนาซอฟต์แวร์. กรุงเทพมหานคร: สำนักพิมพ์ซีเอ็ดยูเคชั่น.
- ธีรเดช แซ่ตัน. 2545. การใช้มาตรฐานวัดเชิงวัตถุทำนายเสถียรภาพของเมทรูด. วิทยานิพนธ์ปริญญา มหาบัณฑิต. สาขาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย กรุงเทพมหานคร.
- วีระศักดิ์ ชิงถาวร. 2547. Java Programming Volume III. กรุงเทพมหานคร: สำนักพิมพ์ซีเอ็ดยูเคชั่น.
- วีระศักดิ์ ชิงถาวร. 2549. Java Programming Volume I (JavaSE 5.0). กรุงเทพมหานคร: สำนักพิมพ์ ซีเอ็ดยูเคชั่น.
- ศิริชัย พงษ์วิชัย. 2546. การวิเคราะห์ข้อมูลทางสถิติด้วยคอมพิวเตอร์. พิมพ์ครั้งที่ 12. กรุงเทพมหานคร: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.
- ศิริวรรณ เสรีรัตน์, สมชาย หิรัญกิตติ, จิรศักดิ์ จิยะจันทร์, ชวลิต ประภวานนท์, ณาชา จันทร์สม และ วลัยลักษณ์ อัดธีรวงศ์. 2541. การวิจัยธุรกิจ. กรุงเทพมหานคร: สำนักพิมพ์เพชรจรัสแสง แห่งโลกธุรกิจ.
- ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ. แผนแม่บทเทคโนโลยีสารสนเทศและการ สื่อสารของประเทศไทย. ที่มา : http://www.nitc.go.th/ict_masterplan
- โอภาส เอี่ยมสิริวงศ์. 2545. การวิเคราะห์และออกแบบระบบ. กรุงเทพมหานคร: สำนักพิมพ์ซีเอ็ด ยูเคชั่น.
- อำนาจ วังจิ้น และ พรรณี บุญสุยา 2548. สถิติทั่วไป. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร: สำนัก วิชาการศึกษาทั่วไป มหาลัยศรีปทุม.

ภาษาอังกฤษ

- Baker, M., and Shih, W. 1992. Performance Prototyping : A Simulation Methodology for Software Performance Engineering. New York : IBM T. J. Watson Research Center.
- Bass, L., et al. 1998. Software Architecture in Practice. United States of America: Addison-Wesley.
- Cartwright, M., and Shepperd, M. 1998. An Empirical View of Inheritance. Information and Software Technology 40 : 795 – 799.
- Chidamber, S.R., and Kemerer, C.F. 1994. A Metrics Suite for Object – Oriented Design. IEEE Transactions on Software Engineering 20 (June 1994) : 476 – 493.
- Daly, J., Brooks, A., Miller, J., Roper, M., and Wood, M. 1995. The Effect of Inheritance on the Maintainability of Object – Oriented Software : An Empirical Study. Proceedings of the International Conference on Software Maintenance : 20 – 29.
- Elish, M. O., and Rine, D. 2003. Investigation of Metrics for Object – Oriented Design Logical Stability. Proceedings of the Seventh European Conference of Software Maintenance and Reengineering (March 2003) : 193 – 200.
- Gerard O. R. 2002. A Practical Approach to Software Quality. New York: Springer – Verlag.
- Grady, R. B., and Caswell, D. L. 1987. Software Metrics : Establishing a Company – Wide Program. United States of America: Prentice-Hall.
- Grassle, P., Baumann, H., and Baumann, P. 2005. UML 2.0 in Action – A Project Bases Tutorial. Birmingham United Kingdom: Packt Publishing.
- Harison, R., Counsell, S., and Nithi R. 2000. Experimental Assessment of the Effect of Inheritance on the Maintainability of Object-Oriented Systems. Proceedings of the International Conference on Empirical Assessment and Evaluation in Software Engineering 52 (June 2000) : 173 – 179.
- Holzner, S. 2004. Eclipse – A Java Developer’ s Guide. United States of America: O'Reilly Media.
- International Standard. ISO/IEC 9126-1. 2000. Information Technology – Software Product Quality – Part 1 : Quality Model. International Organization for Standardization.

- Kan, S. H. 2000. Metrics and Models in Software Quality Engineering. United States of America: Addison-Wesley.
- Kung, D., et al. 1994. Change Impact Identification Object – Oriented Software Maintenance. Proceedings of the International Conference on Software Maintenance (September 1994) : 202 – 211.
- Lakos, C., and Lewis, G. 2000. Behavior Inheritance for Object Lifecycles. Technology of Object-Oriented Languages, Proceedings International Conference 33 (June 2000) : 262 – 273.
- Larman, C. 2002. Applying UML And Patterns – An Introduction to Object – Oriented Analysis and Design and Unified Process. 2nd ed. United States of America: Prentice-Hall International.
- Li, L. 1998. Change Impact Analysis of Object – Oriented Software. Doctoral dissertation. Department of Information Technology, George Mason University.
- Li, L., and Offutt, A. J. Nov, 1996. Algorithm Analysis of the Impact of Changes to Object – Oriented Software. Proceedings of the International Conference on Software Maintenance (November 1996) : 171 – 184.
- Li, W., and Henry, S. 1993. Object- Oriented Metrics that Predict Maintainability. Journal of Systems and Software 23 (November 1993) : 111– 122.
- Martin L.S. 1983. Software Engineering Design, Reliability and Management. New York: McGraw – Hill.
- McCall, J. A. 1977. An Introduction to Software Quality Metrics. New York: A Petrocelli Book.
- Munson, C. J. 1992. Measuring Dynamic Program Complexity. IEEE Transactions on Software Engineering 9 (November 1992) : 48 – 55.
- Perchelt, et al. 2003. A Controlled Experiment on Inheritance Depth as a Cost Factor for Code Maintenance. . Journal of Systems and Software 65 (February 2003) : 115 – 126.
- Pressman, R. S. 2001. Software Engineering a practitioner's Approach. 4th ed. United States of America: McGraw-Hill.
- Pressman, R. S. 2005. Software Engineering a practitioner's Approach. 6th ed. United States of America: McGraw-Hill.

- Rajlish, V. 1997. A Model for Change propagation Based on Graph Rewriting. Proceedings of IEEE international Conference on Software Maintenance (October 1997) : 84 – 91.
- Sarker, M. 2005. An Overview of Object Oriented Design Metrics. Master' s Thesis. Department of Computer Science, Umea University.
- Sommerville, I. 2001. Software Engineering. 6th ed. United States of America: Addison-Wesley.
- Unger, B., and Prechelt, L. 1998. The Impact of Inheritance Depth on Maintenance tasks – Detailed description and Evaluation of two experiment replications. Technical Report of Karlsruhe University 18 (July 1998) : 1 – 177.
- Young, L., and Kai, H. C. 2000. Reusability and Maintainability Metrics for Object – Oriented Software. ACM Southeast Regional Conference on Software Maintenance (April 2000) : 88 – 94.
- Eclipse (Integrated Development Environment - IDE). Available from : <http://www.eclipse.org>
- Java Runtime Environment. Available from : <http://www.java.sun.com>
- Java Source Code. Available from : <http://www.planet-source-code.com>
- Java Source Code. Available from : <http://www.programmersheaven.com>

ภาคผนวก

ภาคผนวก ก

เอกสารแสดงคลาสไดอะแกรมของซอฟต์แวร์หน่วยตัวอย่างที่ใช้ในงานวิจัย

ในส่วนนี้จะกล่าวถึงหน่วยตัวอย่างที่ใช้สำหรับทดสอบงานวิจัยนี้ โดยจะแบ่งออกเป็น 2 ส่วน คือ

- ส่วนที่ 1 คือ หน่วยตัวอย่างที่ใช้เพื่อเปรียบเทียบประสิทธิภาพของโครงสร้างคลาสของซอฟต์แวร์ขณะประมวลผล ซึ่งผู้วิจัยกำหนดให้หน่วยตัวอย่างเป็นซอฟต์แวร์ที่ออกแบบโดยใช้หลักการเชิงวัตถุจำนวน 5 หน่วยตัวอย่าง
- ส่วนที่ 2 คือ หน่วยตัวอย่างที่ใช้เพื่อเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชัน ซึ่งผู้วิจัยเลือกซอฟต์แวร์กรณีศึกษาที่ออกแบบโดยใช้หลักการเชิงวัตถุจำนวน 1 ระบบ และกำหนดให้หน่วยตัวอย่างที่ใช้ในการทดลองเป็นความต้องการเชิงฟังก์ชันของระบบซอฟต์แวร์ที่เลือกจำนวน 30 หน่วยตัวอย่าง

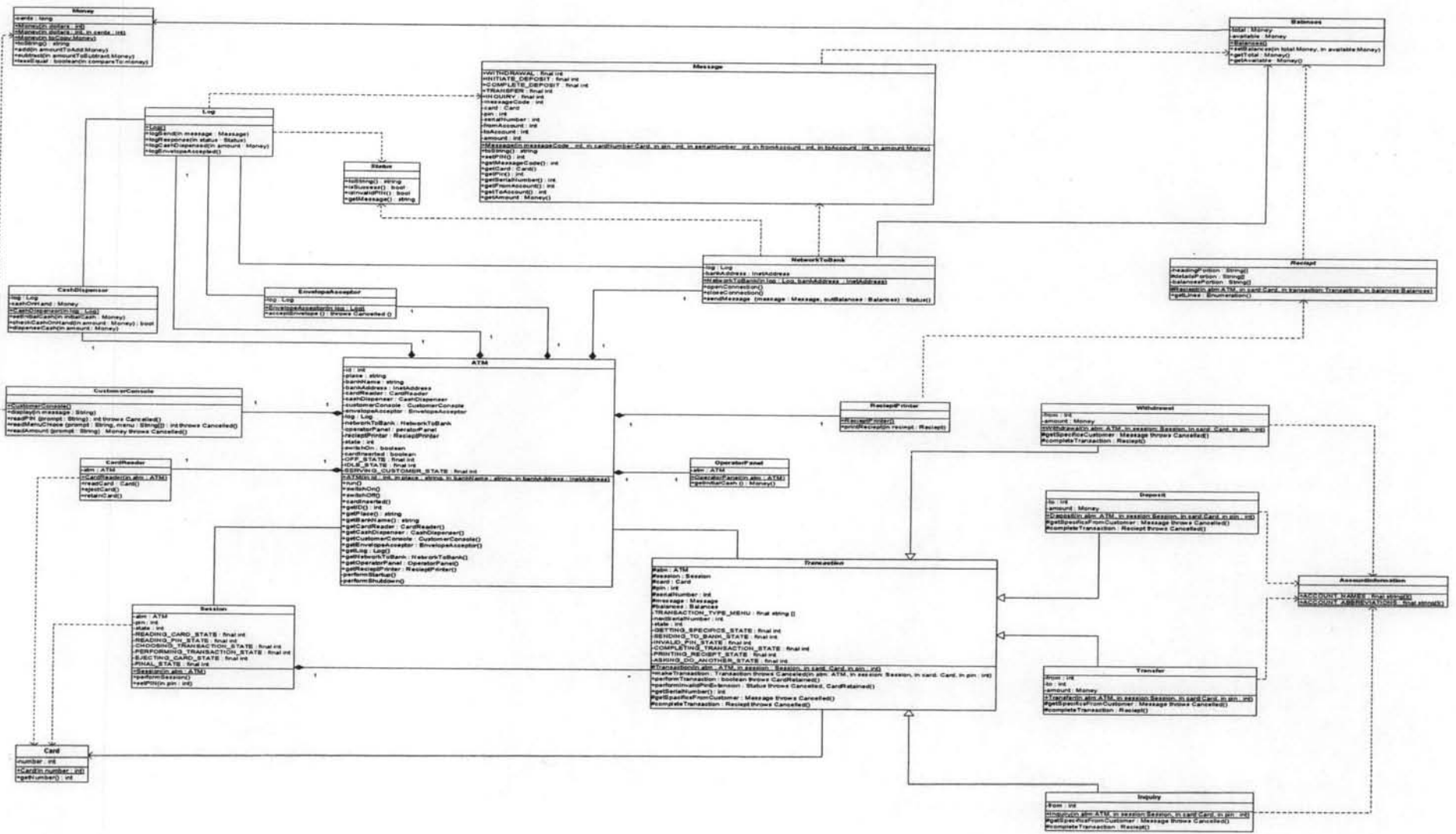
โดยรายละเอียดของโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันและโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันแต่ละหน่วยตัวอย่างในแต่ละส่วนสามารถแสดงได้ ดังนี้

1. คลาสไดอะแกรมของซอฟต์แวร์หน่วยตัวอย่างที่ใช้เพื่อเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผล

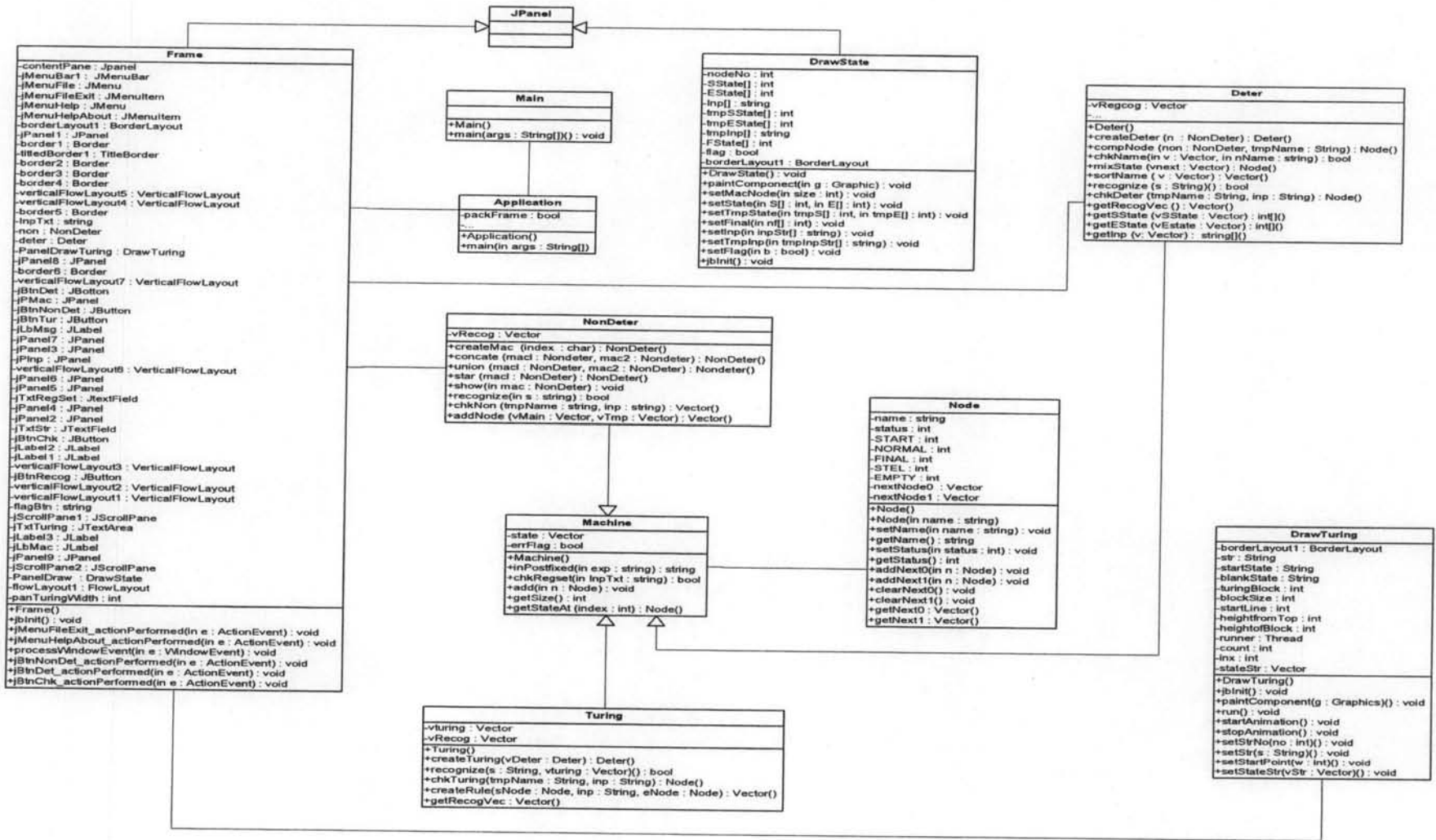
หน่วยตัวอย่างที่ใช้ในการทดสอบการวิจัยเพื่อเปรียบเทียบประสิทธิภาพของโครงสร้างคลาสของซอฟต์แวร์ขณะประมวลผล ซึ่งผู้วิจัยกำหนดให้หน่วยตัวอย่างเป็นซอฟต์แวร์ที่ออกแบบโดยใช้หลักการเชิงวัตถุจำนวน 5 หน่วยตัวอย่าง ซึ่งประกอบด้วยซอฟต์แวร์

- หน่วยตัวอย่างที่ 1: ซอฟต์แวร์เอทีเอ็มออนไลน์ (Online ATM System)
- หน่วยตัวอย่างที่ 2: ซอฟต์แวร์ช่วยสอนออโตเมตา (Learning Software - Autometa)
- หน่วยตัวอย่างที่ 3: ซอฟต์แวร์อนุมัติบัตรเครดิต (Credit Card Approval System)
- หน่วยตัวอย่างที่ 4: ซอฟต์แวร์จัดการเรียนการสอน (Short Course Management System)
- หน่วยตัวอย่างที่ 5: ซอฟต์แวร์หน้าร้านขายหนังสือ (Vijay Bookshop System Ver. 1.04)

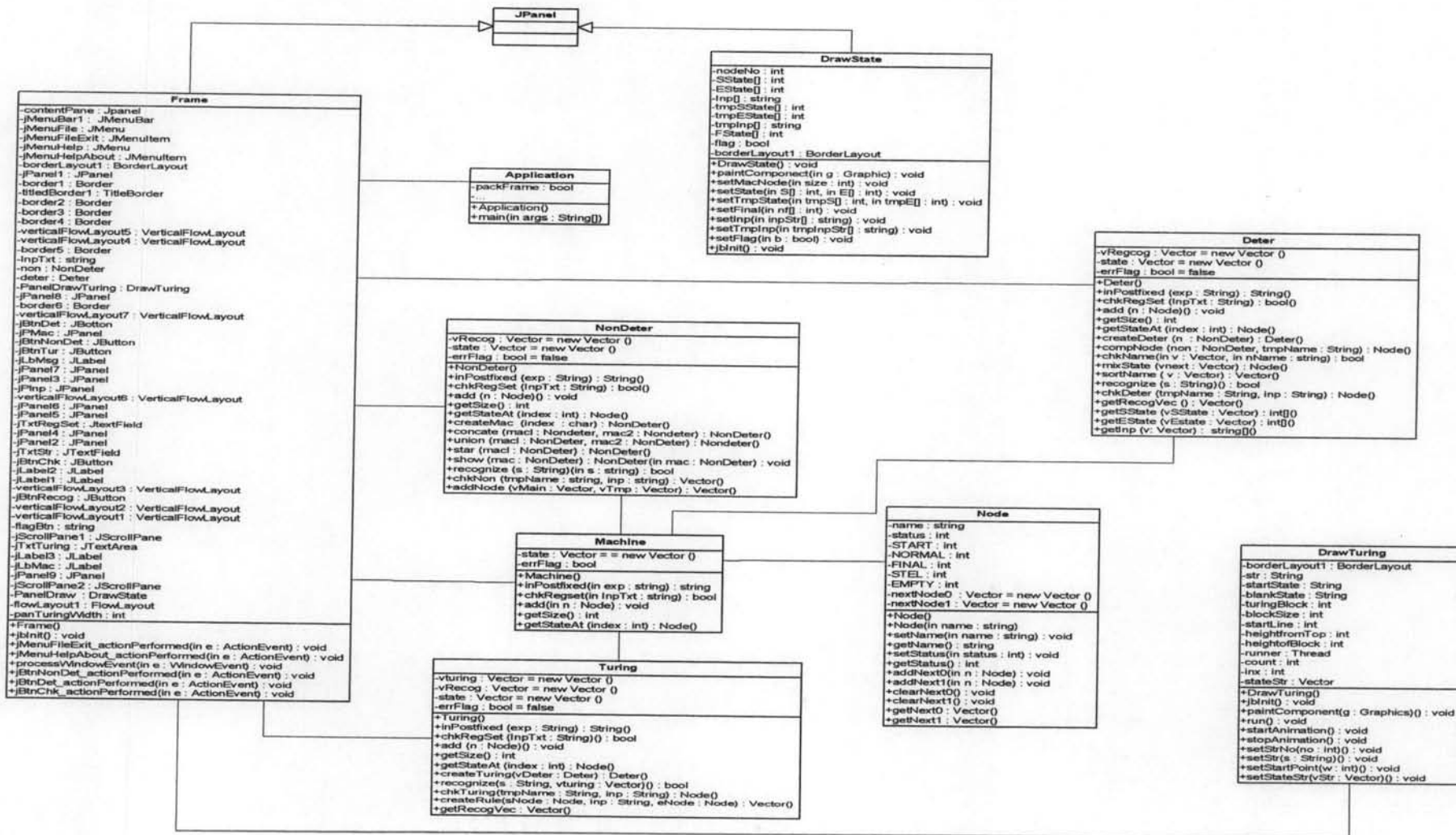
โดยแต่ละหน่วยตัวอย่างมีข้อกำหนดว่าจะต้องมีจำนวนคลาสการทำงานในเชิงธุรกิจ (Business Class Diagram) อย่างน้อย 5 คลาส และเป็นซอฟต์แวร์ที่พัฒนาภายใต้โครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันที่มีการสืบทอดคุณสมบัติ (Inheritance) จากซูเปอร์คลาส (Super Class) และมีซับคลาส (Subclass) ที่รับการถ่ายทอดคุณสมบัติจากซูเปอร์คลาสอย่างน้อยหนึ่งซับคลาสในโครงสร้างคลาสของซอฟต์แวร์ ซึ่งโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันและโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันของแต่ละหน่วยตัวอย่าง แสดงได้ดังรูปที่ ก-1 ถึงรูปที่ ก-10



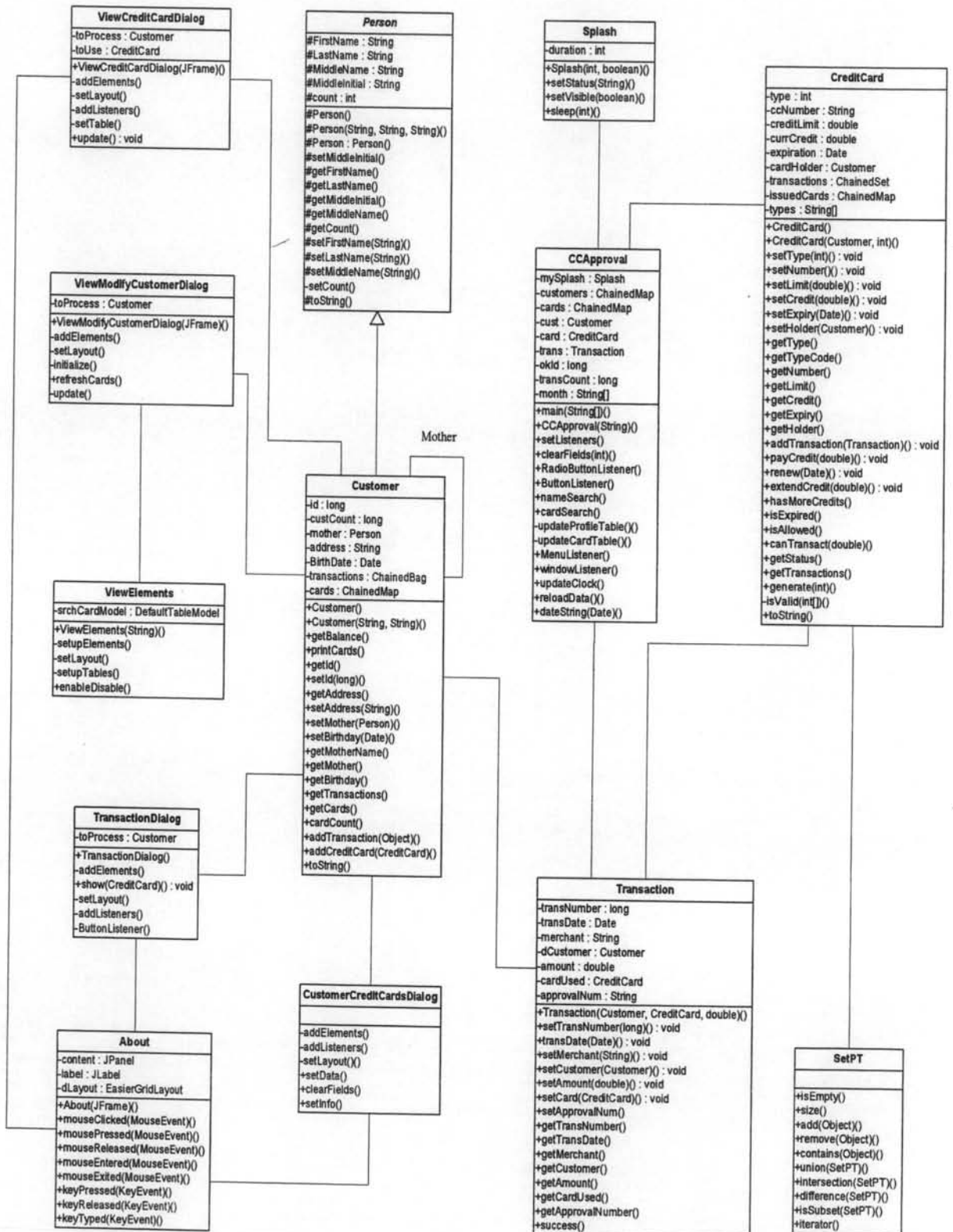
รูปที่ ก-1 โครงสร้างคลาสของซอฟต์แวร์เอทีเอ็มออนไลน์ในความสัมพันธ์แบบเจนอรัลไลเซชัน



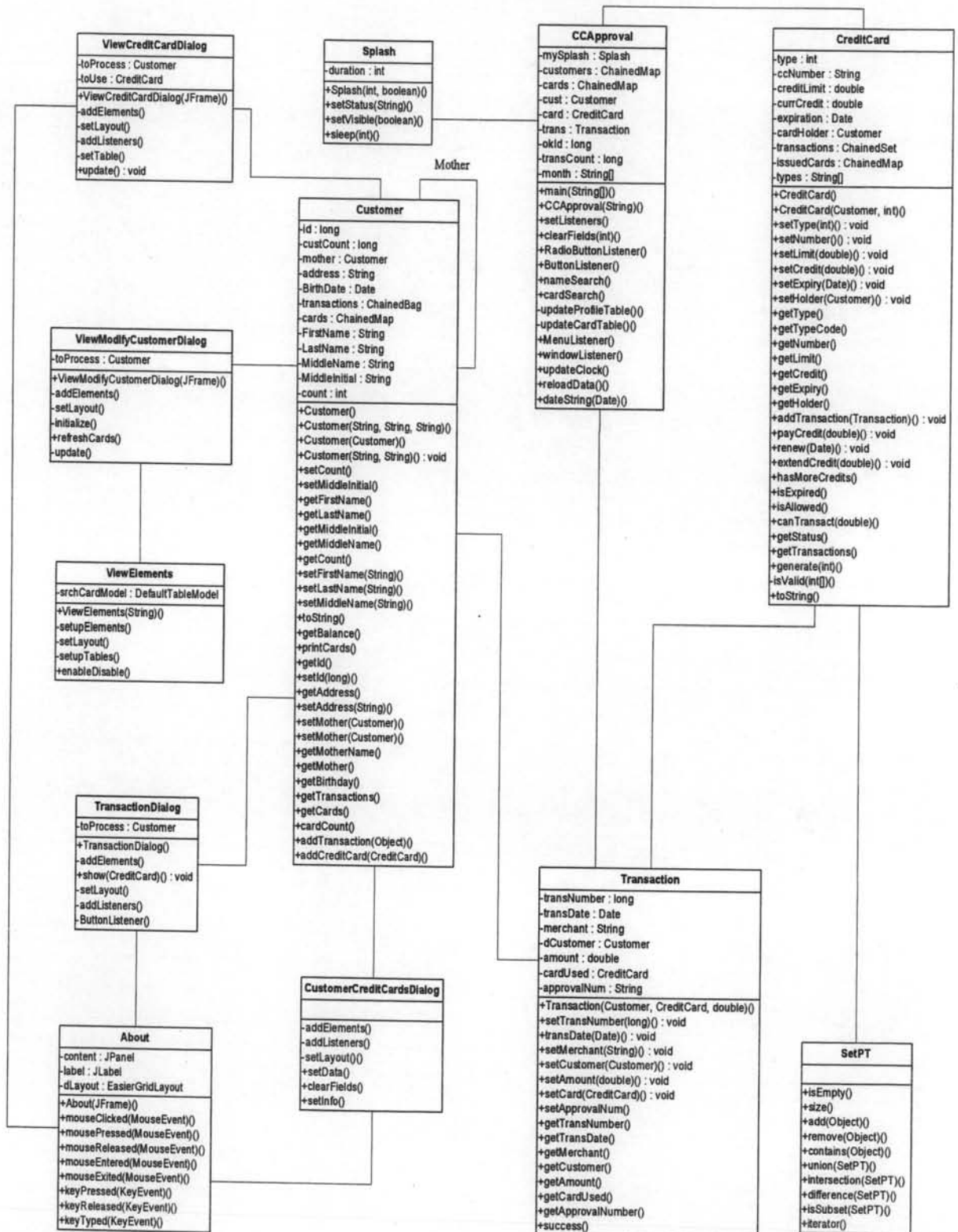
รูปที่ ก-3 โครงสร้างคลาสของซอฟต์แวร์ช่วยสอนออโตเมตาในความสัมพันธ์แบบเจนอรัลไลเซชัน



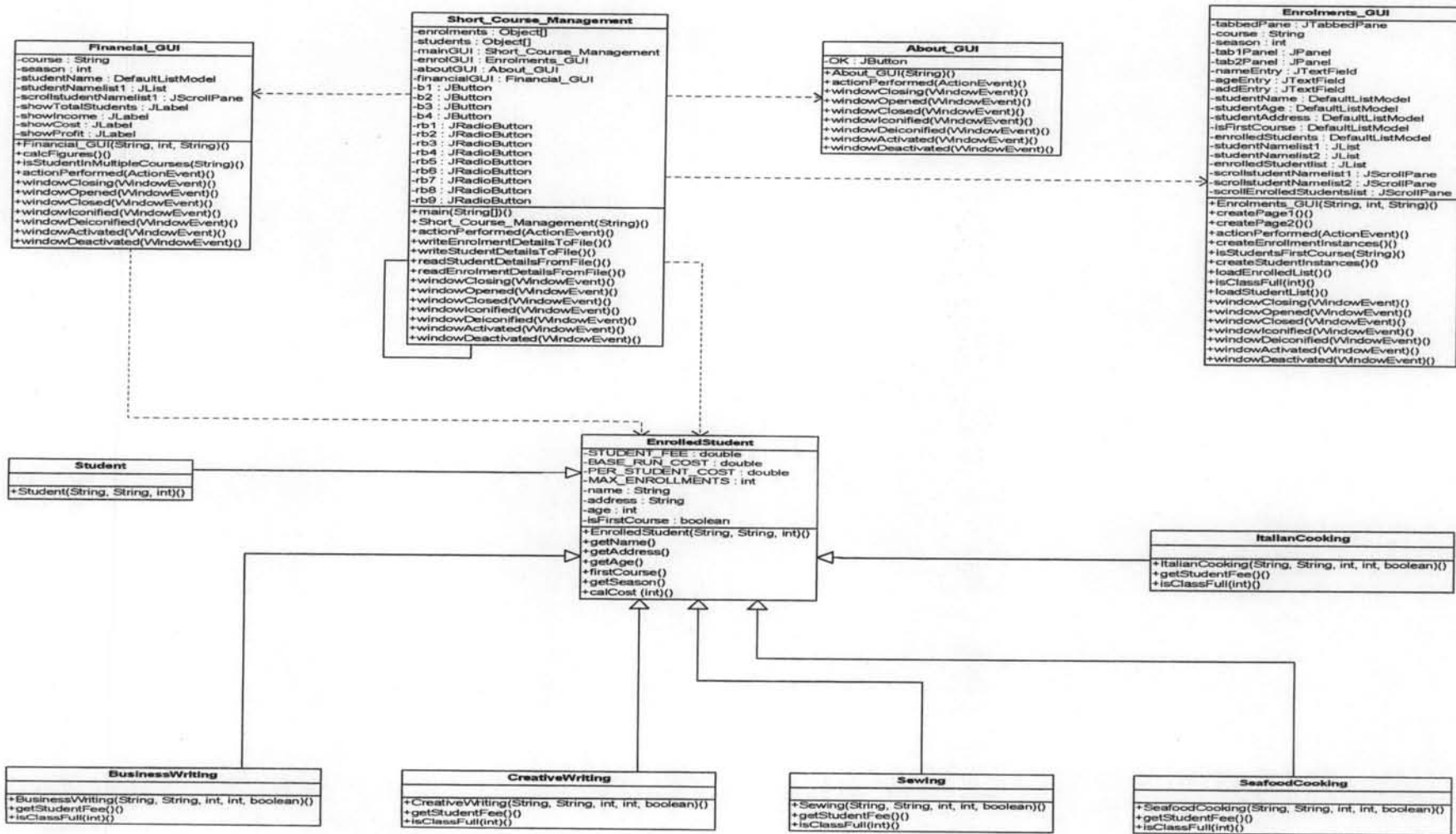
รูปที่ ก-4 โครงสร้างคลาสของซอฟต์แวร์ช่วยสอนอัตโนมัติในความสัมพันธ์แบบแอโซซิเอชัน



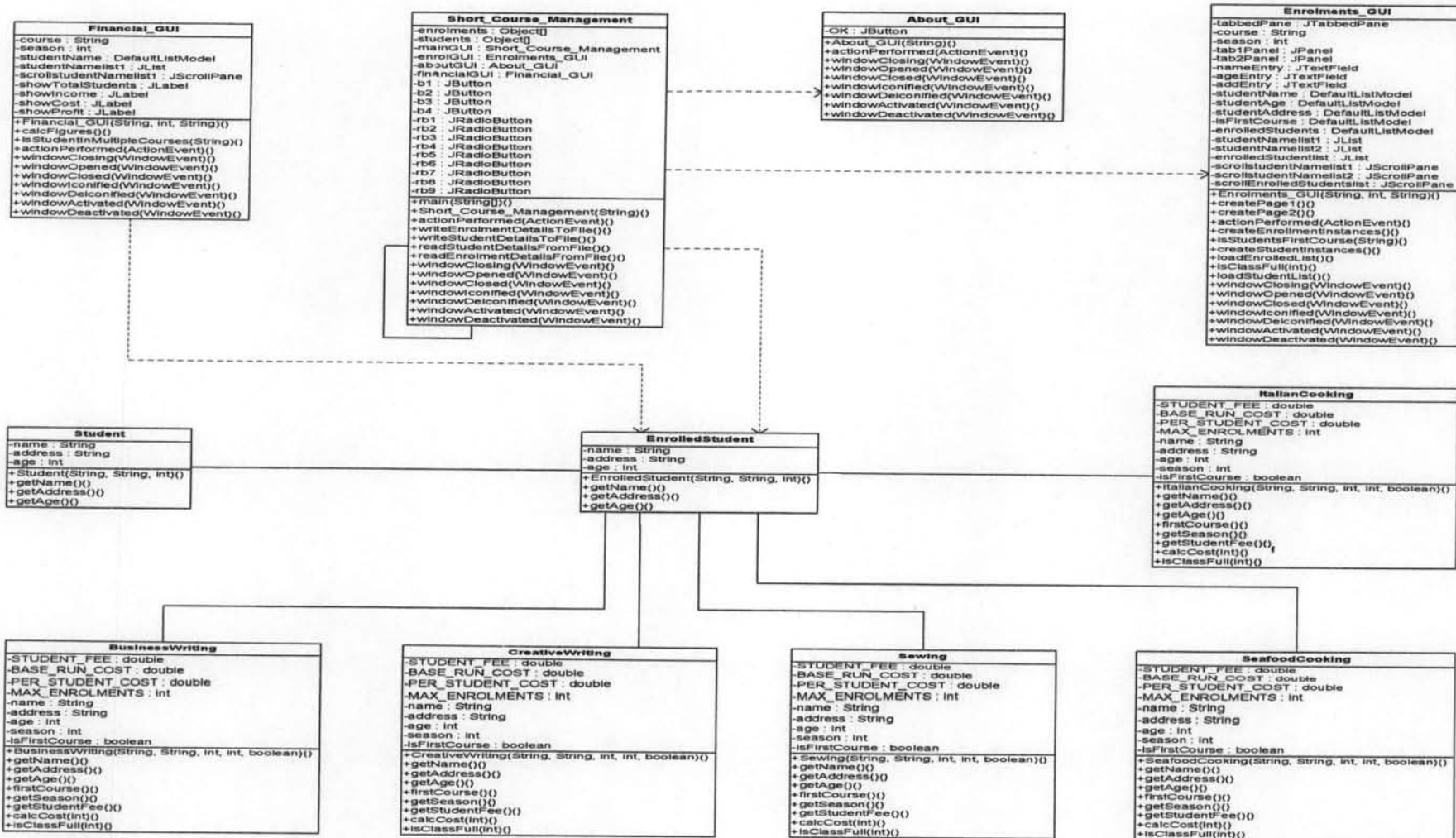
รูปที่ ก-5 โครงสร้างคลาสของซอฟต์แวร์อนุมัติบัตรเครดิตในความสัมพันธ์แบบเจนอรัลไลเซชัน



รูปที่ ก-6 โครงสร้างคลาสของซอฟต์แวร์คอมพิวเตอร์คิด ในความสัมพันธ์แบบแอตโซซิเอชัน



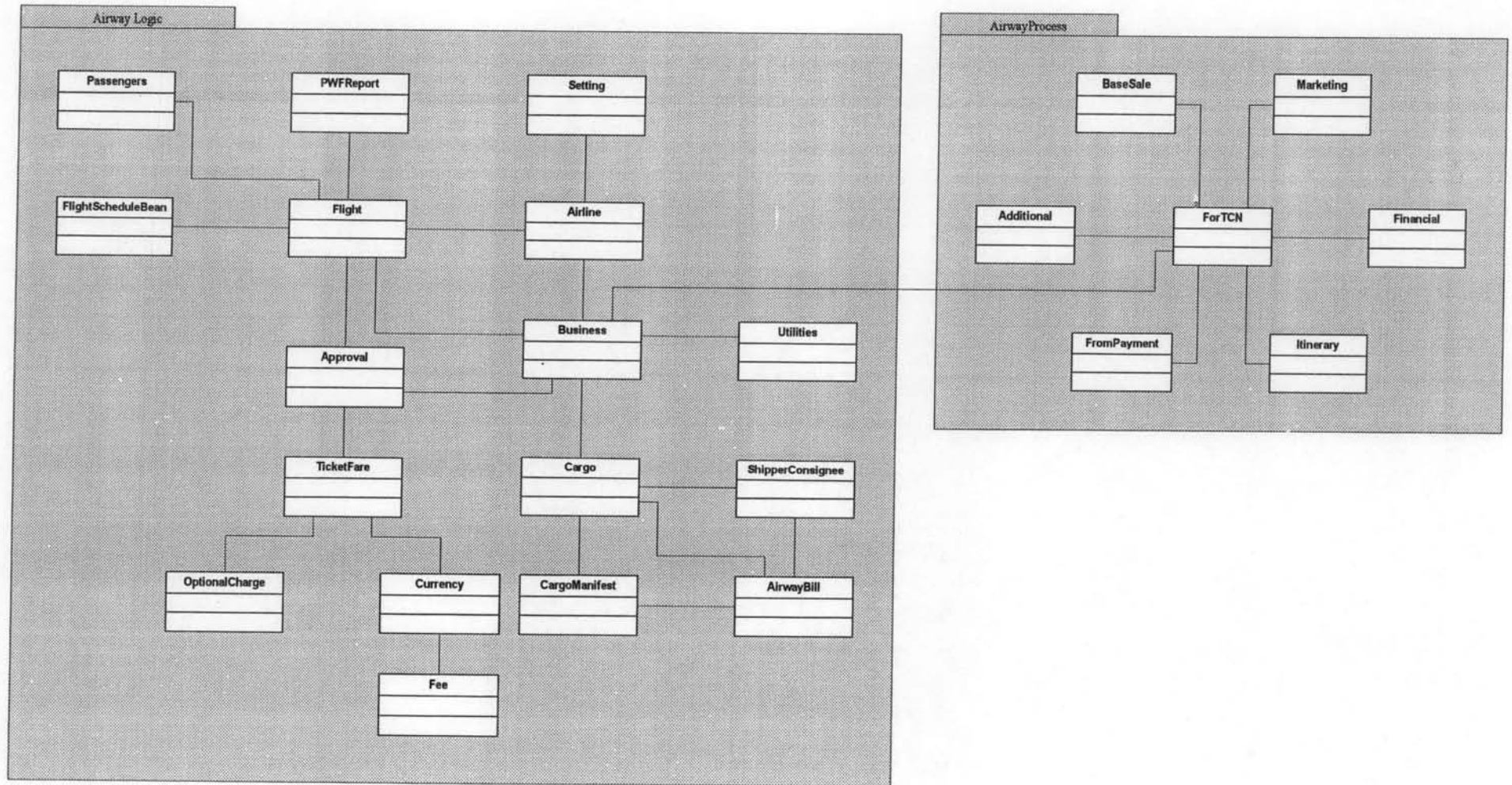
รูปที่ ก-7 โครงสร้างคลาสของซอฟต์แวร์การจัดการเรียนการสอนในความสัมพันธ์แบบเจนอรัลไลเซชัน



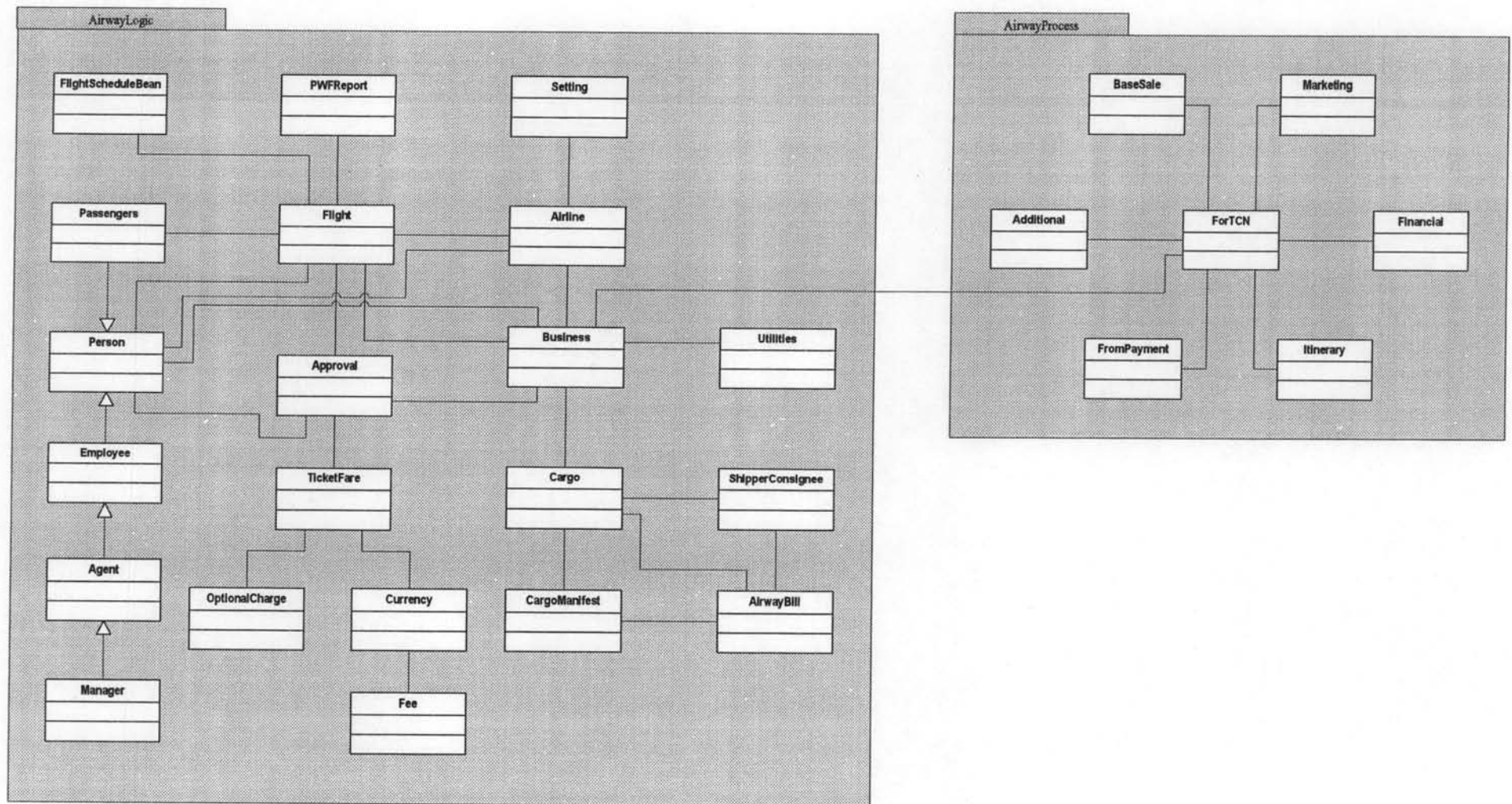
รูปที่ ก-8 โครงสร้างคลาสของซอฟต์แวร์การจัดการเรียนการสอนในความสัมพันธ์แบบแอสโซซิเอชัน

2. คลาสไดอะแกรมของซอฟต์แวร์หน่วยตัวอย่างที่ใช้เพื่อเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการ

หน่วยตัวอย่างที่ใช้ในการทดสอบการวิจัยเพื่อเปรียบเทียบผลกระทบจากการเปลี่ยนแปลงซอฟต์แวร์ตามความต้องการเชิงฟังก์ชัน ผู้วิจัยได้เลือกซอฟต์แวร์กรณีศึกษา คือซอฟต์แวร์การจัดการสายการบินและการบริหารการบิน (Airway and Flight Management System) ขึ้นมาเป็นซอฟต์แวร์กรณีศึกษา ซึ่งเป็นซอฟต์แวร์ประเภทเว็บแอปพลิเคชัน (Web Application Programming Language) โดยมีข้อกำหนดว่าซอฟต์แวร์กรณีศึกษาจะต้องมีจำนวนคลาสการทำงานในเชิงธุรกิจ (Business Class Diagram) ในระบบอย่างน้อย 5 คลาสขึ้นไปและโครงสร้างคลาสของซอฟต์แวร์กรณีศึกษาที่ได้มาอยู่ในลักษณะโครงสร้างคลาสในความสัมพันธ์แบบแอสโซซิเอชันและสามารถนำมาปรับเปลี่ยนให้เป็นโครงสร้างคลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน โดยจะแสดงโครงสร้างคลาสดิอะแกรมในส่วนที่เกี่ยวข้องกับกระบวนการทำงานทางธุรกิจของซอฟต์แวร์การจัดการสายการบินและการบริหารการบินเท่านั้น ซึ่งแสดงได้ดังรูปที่ ข-11 และรูปที่ ข-12



รูปที่ ก-11 โครงสร้างคลาสของซอฟต์แวร์การจัดการสายการบินและการบริหารการบินในความสัมพันธ์แบบแอโซซิเอชัน



รูปที่ ก-12 โครงสร้างคลาสของซอฟต์แวร์การจัดการสายการบินและการบริหารการบินในความสัมพันธ์แบบเจเนอรัลไลเซชัน

ภาคผนวก ข

เอกสารแสดงรายละเอียดการทำทรานแซกชันของซอฟต์แวร์หน่วยตัวอย่าง

ในส่วนนี้จะกล่าวถึงรายละเอียดของการทำทรานแซกชันของซอฟต์แวร์หน่วยตัวอย่างที่ใช้สำหรับทดสอบงานวิจัยในส่วนของเปรียบเทียบประสิทธิภาพของซอฟต์แวร์ขณะประมวลผลระหว่างโครงสร้างคลาสิกในความสัมพันธ์แบบแอตโซซิเอชันและโครงสร้างคลาสิกในความสัมพันธ์แบบเจเนอร์ลไลเซชัน โดยมีหน่วยตัวอย่างที่ใช้ทำการทดลองจำนวน 5 หน่วยตัวอย่าง ดังนี้

หน่วยตัวอย่างที่ 1: ซอฟต์แวร์เอทีเอ็มออนไลน์ (Online ATM System)

หน่วยตัวอย่างที่ 2: ซอฟต์แวร์ช่วยสอนออโตเมตา (Learning Software - Autometa)

หน่วยตัวอย่างที่ 3: ซอฟต์แวร์อนุมัติบัตรเครดิต (Credit Card Approval System)

หน่วยตัวอย่างที่ 4: ซอฟต์แวร์จัดการเรียนการสอน (Short Course Management System)

หน่วยตัวอย่างที่ 5: ซอฟต์แวร์หน้าร้านขายหนังสือ (Vijay Bookshop System Ver. 1.04)

ซึ่งการจัดเก็บข้อมูลสำหรับงานวิจัยนี้ ดำเนินการโดยซอฟต์แวร์หน่วยตัวอย่างแต่ละหน่วย ผู้วิจัยจะประมวลผลทรานแซกชันจำนวน 3 แบบ โดยรายละเอียดของการทำทรานแซกชันในแต่ละซอฟต์แวร์หน่วยตัวอย่างแสดงในตารางที่ ข-1 และทรานแซกชันในแต่ละแบบมีลักษณะของการทำรายการ ดังนี้

- ทรานแซกชันที่ 1 คือ การทำรายการซอฟต์แวร์หน่วยตัวอย่างโดยยังไม่มี การเปลี่ยนแปลงข้อมูล (แอททริบิวต์) และการดำเนินการ (เมธอด) ภายในโครงสร้างคลาสิก ในความสัมพันธ์แบบเจเนอร์ลไลเซชัน (ซูเปอร์คลาสิกและซับคลาสิกที่มีการสืบทอดหรือ ถ่ายทอดคุณสมบัติกัน)
- ทรานแซกชันที่ 2 คือ การทำรายการซอฟต์แวร์หน่วยตัวอย่างเพื่อให้มีการเปลี่ยนแปลง ข้อมูล (แอททริบิวต์) และการดำเนินการ (เมธอด) ภายในคลาสิก โดยกำหนดให้ โครงสร้างคลาสิกในความสัมพันธ์แบบเจเนอร์ลไลเซชันมีการเปลี่ยนแปลงข้อมูล (แอททริ บิวต์) และการดำเนินการ (เมธอด) ที่กระทบกับซับคลาสิกใดซับคลาสิกหนึ่งที่มีการสื บทอดคุณสมบัติในโครงสร้างคลาสิกในความสัมพันธ์แบบเจเนอร์ลไลเซชัน
- ทรานแซกชันที่ 3 คือ การทำรายการซอฟต์แวร์หน่วยตัวอย่างเพื่อให้มีการเปลี่ยนแปลง ข้อมูล (แอททริบิวต์) และการดำเนินการ (เมธอด) ภายในคลาสิก โดยกำหนดให้ โครงสร้างคลาสิกในความสัมพันธ์แบบเจเนอร์ลไลเซชันมีการเปลี่ยนแปลงข้อมูล

(แอททริบิวต์) และการดำเนินการ (เมธอด) ที่กระทบกับหลาย ๆ ชั้นคลาสที่มีการสืบทอดคุณสมบัติในโครงสร้างคลาสนี้ความสัมพันธ์แบบเจเนอรัลเซชัน

ทรานแซกชันแบบที่	รายละเอียดของการทำทรานแซกชัน	คลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ที่ถูกเปลี่ยนแปลงค่าในแอททริบิวต์และเมธอด
หน่วยตัวอย่างที่ 1: ซอฟต์แวร์เอทีเอ็มออนไลน์ (Online ATM System)		
ทรานแซกชันแบบที่ 1	ลูกค้าติดต่อกับระบบเอทีเอ็ม โดยการ สอดบัตรเอทีเอ็ม กรอกรหัสผ่าน (PIN Code) ระบบตรวจสอบรหัสผ่านและอนุมัติการใช้งาน ลูกค้าต้องการตรวจสอบข้อมูลการทำรายการเอทีเอ็มที่ผ่านมา	-
ทรานแซกชันแบบที่ 2	ลูกค้าติดต่อกับระบบเอทีเอ็ม โดยการ สอดบัตรเอทีเอ็ม กรอกรหัสผ่าน ระบบตรวจสอบรหัสผ่านและอนุมัติการใช้งาน ลูกค้าทำการโอนเงินระหว่างบัญชี (Transfer) โดยกรอกชื่อบัญชีปลายทางที่ต้องการ โอนเงิน และจำนวนเงินที่ต้องการ โอน ระบบบันทึกรายการและและออกสลิปให้ลูกค้า	Transaction และ Transfer
ทรานแซกชันแบบที่ 3	ลูกค้าติดต่อกับระบบเอทีเอ็ม โดยการ สอดบัตรเอทีเอ็ม กรอกรหัสผ่าน ระบบตรวจสอบรหัสผ่านและอนุมัติการใช้งาน ลูกค้าทำการฝากเงิน (Deposit) โดยเลือกประเภทบัญชีเงินฝากกระแสรายวัน (Checking Account) และใส่จำนวนเงินที่ต้องการฝาก ระบบบันทึกรายการ ลูกค้าต้องการทำการถอน โดยเลือกการทำรายการถอนเงิน (Withdrawal) โดยเลือกประเภทบัญชีเงินฝากกระแสรายวันและกรอกจำนวนเงินที่ต้องการถอน ระบบตรวจสอบยอดเงินคงเหลือว่ามีเพียงพอกับจำนวนที่ลูกค้าต้องการถอนหรือไม่ เพียงพอก็บันทึกรายการ และออกสลิปให้ลูกค้า	Transaction, Deposit และ Withdrawal

ตารางที่ ข-1 แสดงรายละเอียดการทำทรานแซกชันของซอฟต์แวร์หน่วยตัวอย่าง

ทราชนแซคชันแบบที่	รายละเอียดของการทำทราชนแซคชัน	คลาสในความสัมพันธ์แบบเจเนอรัลไลเซชันที่ถูกเปลี่ยนแปลงค่าในแอททริบิวต์และเมธอด
หน่วยตัวอย่างที่ 2: ซอฟต์แวร์ช่วยสอนอัตโนมัติ (Learning Software - Autometa)		
ทราชนแซคชันแบบที่ 1	รับข้อมูลในรูปแบบของเรกกิวเลเซท (Regular Set) โดยไม่ให้มีการแปลงข้อมูลเพื่อสร้างมะชีน (Machine)	-
ทราชนแซคชันแบบที่ 2	นำเข้าเรกกิวเลเซท [1^*U00] เพื่อจะนำไปสร้างเป็นโหนด (Node) โดยการนำเรกกิวเลเซท เข้ามาสร้างทีละหนึ่งแคริคเทอะ (Character) แล้วนำโหนดที่ได้จากการสร้างเรกกิวเลเซท ไปสร้างเป็นนอนดิเทอร์มินิซติคมะชีน (Nondeterministic Machine) และรับข้อมูลทดสอบการรู้จำ (Recognition) เข้ามาทดสอบการรู้จำกับนอนดิเทอร์มินิซติคมะชีน แล้วแสดงผลการรู้จำ	Machine และ NonDeter
ทราชนแซคชันแบบที่ 3	นำเข้าเรกกิวเลเซท [$((10)^*U(101)^*)^*1$] เพื่อจะนำไปสร้างเป็นโหนด โดยการนำเรกกิวเลเซท เข้ามาสร้างทีละหนึ่งแคริคเทอะ แล้วนำโหนดที่ได้จากการสร้างเรกกิวเลเซท ไปสร้างเป็นดิเทอร์มินิซติคมะชีน (Deterministic Machine) และรับข้อมูลทดสอบการรู้จำเข้ามาทดสอบการรู้จำกับดิเทอร์มินิซติคมะชีน แล้วแสดงผลการรู้จำ จากนั้นนำดิเทอร์มินิซติคมะชีน ไปสร้างเป็นเทอร์ริงมะชีน (Turing Machine) และรับข้อมูลทดสอบการรู้จำเข้ามาทดสอบการรู้จำกับเทอร์ริงมะชีน แสดงผลการรู้จำ	Machine, Deter และ Turing

ตารางที่ ข-1 แสดงรายละเอียดการทำทราชนแซคชันของซอฟต์แวร์หน่วยตัวอย่าง (ต่อ)

ทรานแซกชันแบบที่	รายละเอียดของการทำทรานแซกชัน	คลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ที่ถูกเปลี่ยนแปลงค่าในแอททริบิวต์และเมธอด
หน่วยตัวอย่างที่ 3: ซอฟต์แวร์อนุมัติบัตรเครดิต (Credit Card Approval System)		
ทรานแซกชันแบบที่ 1	บริษัทต้องการตรวจสอบรายชื่อลูกค้าบัตรเครดิต	-
ทรานแซกชันแบบที่ 2	ลูกค้าต้องการยื่นคำร้องขอเปลี่ยนแปลงที่อยู่บริษัท และขอเพิ่มวงเงินในบัตรเครดิต	Person และ Customer
ทรานแซกชันแบบที่ 3	บริษัทเพิ่มรายชื่อลูกค้าบัตรเครดิต โดยการกรอกรายละเอียดข้อมูลลูกค้า และอนุมัติการใช้บัตรเครดิต จากนั้นลูกค้านำบัตรเครดิตไปทำรายการซื้อที่ห้างสรรพสินค้า ระบบอนุมัติยอดเงินใช้จ่าย	Person และ Customer (เพิ่มการเปลี่ยนแปลงที่เมธอด <code>getTransactions</code> , <code>getCards</code> และ <code>addTransaction</code>)
หน่วยตัวอย่างที่ 4: ซอฟต์แวร์จัดการเรียนการสอน (Short Course Management System)		
ทรานแซกชันแบบที่ 1	นิสิตต้องการตรวจสอบการประเมินผลการเรียน และตรวจสอบรายวิชาที่ต้องลงทะเบียนเรียนในแต่ละภาคการศึกษา	-
ทรานแซกชันแบบที่ 2	นิสิตต้องการเพิกถอนรายวิชาการทำอาหารทะเล (Seafood Cooking) ที่ได้มีกลงทะเบียนเรียน ไปเรียบร้อยแล้วในภาคการศึกษา จากนั้นระบบออกใบยืนยันการเพิกถอนรายวิชา	EnrolledStudent และ SeafoodCooking
ทรานแซกชันแบบที่ 3	นิสิตต้องการเพิ่มการลงทะเบียนเรียนในรายวิชาการเขียนทางธุรกิจ (Business Writing) และการเขียนอย่างสร้างสรรค์ (Creative Writing) ระบบออกใบบันทึกการลงทะเบียน	EnrolledStudent, BusinessWriting และ CreativeWriting

ตารางที่ ข-1 แสดงรายละเอียดการทำทรานแซกชันของซอฟต์แวร์หน่วยตัวอย่าง (ต่อ)

ทรานแซกชันแบบที่	รายละเอียดของการทำทรานแซกชัน	คลาสในความสัมพันธ์แบบเจเนอรัลไลเซชัน ที่ถูกเปลี่ยนแปลงค่าในแอททริบิวต์และเมธอด
หน่วยตัวอย่างที่ 5: ซอฟต์แวร์หน้าร้านขายหนังสือ (Vijay Bookshop System Ver. 1.04)		
ทรานแซกชันแบบที่ 1	พนักงานประจำร้าน ค้นหารายการหนังสือตามหมวดหมู่วิทยาศาสตร์ (Science) ระบบ แสดงผลรายงานการค้นหาหนังสือตามหมวดหมู่	-
ทรานแซกชันแบบที่ 2	พนักงานประจำร้าน (Clerk) เปิดบิล (Bill) รายการขายหนังสือ และพิมพ์ออกใบเสร็จให้ ลูกค้า	BookshopStaff และ Clerk
ทรานแซกชันแบบที่ 3	ผู้จัดการร้าน (Administrator) ทดสอบเข้าสู่ระบบโดยการกรอกชื่อผู้ใช้และรหัสผ่าน ระบบ มีการตรวจสอบสิทธิ์การใช้งาน ผู้จัดการเพิ่มจำนวนรายการหนังสือเข้าร้านขายหนังสือ และตรวจสอบยอดคงเหลือของรายการสินค้า และพนักงานประจำร้านออกรายงานการ ขายและยอดขายหนังสือ	BookshopStaff, Administrator และ Clerk

ตารางที่ ข-1 แสดงรายละเอียดการทำทรานแซกชันของซอฟต์แวร์หน่วยตัวอย่าง (ต่อ)

ประวัติผู้เขียนวิทยานิพนธ์

นางสาวสุพรรณษา เนคมานุรักษ์ เกิดเมื่อวันที่ 13 ตุลาคม พ.ศ. 2524 สำเร็จการศึกษาระดับปริญญาตรีวิทยาศาสตร์บัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ จากภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา ในปีการศึกษา 2547

จากนั้นได้เข้าศึกษาต่อในระดับปริญญาโท หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาการพัฒนาซอฟต์แวร์ด้านธุรกิจ ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย