

วิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการแก้ปัญหาของอ็อบเจกต์โมเดลที่สามารถปรับเปลี่ยนได้หรือเอโอเอ็ม ซึ่งประกอบด้วยดีไซน์แพทเทิร์นต่างๆ ได้แก่ ไทป์อ็อบเจกต์ พรอบเพอร์ตี ไทป์สแควร์ แอ็คเคาน์ทะบิลิตีแพทเทิร์น ปัญหาของเอโอเอ็มคือ อ็อบเจกต์โมเดลอยู่ในรูปแบบของเมต้า ทำให้ยากต่อการทำความเข้าใจ ส่งผลให้ซอฟต์แวร์มีคุณลักษณะที่ไม่ดีคือ บำรุงรักษาได้ยาก ดังนั้นงานวิจัยจึงได้นำเสนอวิธีการแปลงเอโอเอ็มให้อยู่ในรูปของยูนิตโมเดล จุดประสงค์เพื่อช่วยให้นักพัฒนาสามารถทำความเข้าใจโมเดลได้ง่ายขึ้น และจะส่งผลให้ความสามารถในด้านการบำรุงรักษาซอฟต์แวร์เพิ่มขึ้นด้วย

ผลการทดลองพบว่า การแปลงอ็อบเจกต์โมเดลแบบเอโอเอ็มให้อยู่ในรูปของยูนิตโมเดลนั้น ทำให้ความซับซ้อนของแผนภาพคลาสนั้นลดลงเล็กน้อย นอกจากนั้น ด้วยคุณสมบัติที่ได้จากการแปลงให้อยู่ในรูปของยูนิตโมเดล ทำให้สถาปัตยกรรมของระบบมีภาพที่เป็นระดับสูงขึ้น และสามารถแบ่งเป็นระดับชั้นได้ ซึ่งจะช่วยในเรื่องของความสามารถในการทำความเข้าใจระบบให้นักพัฒนา อีกทั้งยูนิตโมเดลมีภาษาและเครื่องมือที่ช่วยให้ผู้ออกแบบสถาปัตยกรรมของซอฟต์แวร์สามารถตรวจสอบความถูกต้องได้ง่ายกว่าการใช้ในระดับแพกเกจ

This thesis proposes a method for solving a problem inherently found in Adaptive Object Model (AOM). AOM basically consists of many design patterns such as Type Object, Property, Type-Square, and Accountability. The problem of AOM is that object model is presented in meta-form, not in business-object form. As a result, it is difficult to understand and maintain the model. To address the problem, we propose a method for converting the model from AOM to a unit model, which is easy to understand and maintain.

The experimental results from this research show that the complexity of class diagrams in a sample application is reduced slightly. In addition with unit models, software has a high-level architecture and layering components which increase understandability for developer. Moreover, the software architects can verify correctness of architecture easier than normal package, since the unit model has both language and tools for development.