

การทดสอบแบบบูรณาการเป็นการหาข้อผิดพลาดในการทำงานร่วมกันระหว่างส่วนประกอบของซอฟต์แวร์ ในการทดสอบแบบบูรณาการสำหรับซอฟต์แวร์เชิงวัตถุ จำเป็นต้องกำหนดลำดับคลาสที่นำมาทดสอบ แต่ปัญหาของการเรียงลำดับคลาสคือ การเรียงลำดับคลาสโดยที่ความสัมพันธ์ระหว่างคลาสมีการเรียกที่ขึ้นต่อกันแบบมีวง สำหรับวิธีการเรียงลำดับคลาสวิธีอื่นๆ จะทำการลบความสัมพันธ์ระหว่างคลาสเพื่อจัดการเรียกที่ขึ้นต่อกันแบบมีวง โดยจะต้องสร้างสแต็คขึ้นเพื่อใช้ทดสอบคลาสที่ถูกลบความสัมพันธ์ งานวิจัยนี้นำเสนอวิธีการเรียงลำดับคลาสสำหรับการทดสอบแบบบูรณาการ วิธีการนี้ใช้การตัดส่วนเชิงวัตถุในการจัดการเรียกที่ขึ้นต่อกันแบบมีวง เพื่อลดจำนวนสแต็คในการเรียงลำดับคลาสสำหรับทำการทดสอบ

วิธีการเรียงลำดับคลาสสำหรับการทดสอบแบบบูรณาการที่เสนอในวิทยานิพนธ์นี้ประกอบด้วยขั้นตอนหลัก 3 ขั้นตอนได้แก่ ขั้นตอนการวิเคราะห์ความสัมพันธ์ระหว่างคลาส ขั้นตอนการจัดการเรียกที่ขึ้นต่อกันแบบมีวง และขั้นตอนการเรียงลำดับคลาส โดยในขั้นตอนการวิเคราะห์ความสัมพันธ์ระหว่างคลาส วิทยานิพนธ์นี้ใช้ทาร์จันอัลกอริทึมในการหากลุ่มคลาสที่ความสัมพันธ์ระหว่างคลาสมีการเรียกที่ขึ้นต่อกันแบบมีวง จากนั้นจึงทำการแบ่งประเภทกลุ่มคลาสที่มีการเรียกที่ขึ้นต่อกันแบบมีวง สำหรับการจัดการเรียกที่ขึ้นต่อกันแบบมีวงซึ่งเป็นขั้นตอนถัดไปนั้น ใช้การตัดส่วนเชิงวัตถุในการจัดการเรียกที่ขึ้นต่อกันตามประเภทของกลุ่มคลาสที่ได้แบ่งประเภทไว้แล้ว เมื่อการเรียกที่ขึ้นต่อกันแบบมีวงถูกขจัดจนหมด จึงเรียงลำดับคลาสเป็นขั้นตอนสุดท้าย

ในงานวิทยานิพนธ์นี้ได้พัฒนาเครื่องมือเพื่อทดสอบวิธีการที่นำเสนอ และทดลองเรียงลำดับคลาสสำหรับการทดสอบแบบบูรณาการจากโปรแกรมเชิงวัตถุ ผลที่ได้จากการทดลองกับกรณีศึกษา 3 ระบบ แสดงให้เห็นว่า วิธีการเรียงลำดับการบูรณาการคลาสโดยใช้เทคนิคการตัดส่วนเชิงวัตถุ มีการใช้สแต็คน้อยกว่าวิธีการเรียงลำดับคลาสวิธีอื่นๆ

KEYWORDS : INTEGRATION TESTING / TEST ORDERS / OBJECT-ORIENTED PROGRAM SLICING / SOFTWARE TESTING

JUTARAT JAROENPIBOONKIT : AN APPROACH OF ORDERING CLASS FOR INTEGRATION TESTING USING OBJECT-ORIENTED SLICING TECHNIQUE.
ADVISOR : ASSOC.PROF. TARATIP SUWANNASART, PH.D., 90 pp.

Integration testing is to find component faults that cause inter-component failures. Integration testing for object-oriented software needs ordering classes to be tested. However, a problem of ordering classes is the determination of cyclic dependency calls. Most approaches proposed ordering classes by removing relationships to break cycles. Removing a relationship would require a stub to be created for testing classes at the end of the relationship. This thesis presents an approach of ordering class for integration testing using object-oriented slicing technique to break cyclic dependencies. The objective is to minimize a number of test stubs in ordering classes.

The proposed approach consists of three steps: analyzing relationship step, breaking cycle step, and sorting class step. In the analyzing relationship step, this research uses tarjan algorithm for detecting cycle, then classifies classes with cyclic dependencies to cycle type. Breaking cycles, which is the next step, uses object-oriented slicing to break cycles by cycle type. After no cycle remains, the final step is ordering classes.

This thesis develops a tool to test this approach and, to demonstrate ordering classes for integration testing from object-oriented software. The result of our experiments with three case studies indicates that an approach for ordering class integration using object-oriented slicing technique use less test stubs than other approaches.