

ปัจจุบัน ซอฟต์แวร์ทั่วไปมีการใช้ส่วนต่อประสานกราฟิกกับผู้ใช้หรือจียูไอกันมาก เนื่องจากสะดวกและใช้งานง่ายกว่าการใช้งานแบบบรรทัดคำสั่ง การทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้เป็นสิ่งสำคัญ เนื่องจากหากส่วนต่อประสานกราฟิกกับผู้ใช้มีข้อผิดพลาดก็เหมือนระบบทำงานผิดพลาด แม้ว่าแอปพลิเคชันซอฟต์แวร์จะทำงานได้ถูกต้อง อย่างไรก็ตามการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้ยังคงเป็นงานหวัข้อวิจัย เนื่องจากถ้าทำการทดสอบจากโครงสร้างของส่วนต่อประสานกราฟิกกับผู้ใช้ ซึ่งปกติสามารถแทนได้ด้วยเครื่องสถานะจำกัด จะได้กรณีทดสอบจำนวนมหาศาล ซึ่งไม่เหมาะสมในทางปฏิบัติหรือใช้ทดสอบจริง ดังนั้นวิทยานิพนธ์นี้จึงได้นำเสนอแนวทางการสร้างกรณีทดสอบจากแผนภาพยูเอ็มแอล โดยใช้วิธีการสร้างกรณีทดสอบในระดับของงานซึ่งมีแนวคิดคล้ายคลึงกับแผนงานปัญญาประดิษฐ์ ชุดกรณีทดสอบจะครอบคลุมทุกหน้าที่การทำงานที่มีปฏิสัมพันธ์กับผู้ใช้ดังปรากฏในเอกสารการออกแบบระบบ ผลที่ได้คือจำนวนกรณีทดสอบที่เป็นไปได้ในทางปฏิบัติของการทดสอบซอฟต์แวร์จียูไอนอกจากนี้ผู้วิจัยยังได้พัฒนาเครื่องมือกึ่งอัตโนมัติสำหรับสร้างกรณีทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้จากเอกสารการออกแบบยูเอ็มแอล เพื่อให้สามารถสร้างกรณีทดสอบได้ทันทีหลังจากทราบข้อกำหนดซอฟต์แวร์ วิธีการและเครื่องมือดังกล่าวจะช่วยประหยัดทรัพยากรที่ใช้ในขั้นตอนการทดสอบ รวมทั้งสนับสนุนการนำกรณีทดสอบกลับมาใช้งานได้อีก

Graphical User Interfaces (GUIs) are ubiquitous as the user interfaces of most of software today. For users, GUIs provide easier access to the systems compared to using command lines. GUI testing is essential to ensure that the application software correctly functions. However, GUI testing has been an ongoing research topic. According to the GUI structure, which can be represented by Finite State Machine, the number of test cases generated is enormous and it is not practical for real-world testing. This thesis, therefore, proposes an approach to generating GUI test cases from UML diagrams. The method generates test cases at the level of task that is similar to the concept of AI planning. The test suite will cover every function that has interactions with a user as described in the system design document. As a result, the number of test cases generated from our approach is practical to serve for testing purposes. We also developed a tool to semi-automate GUI test case generation from UML design documents. This would enable the generation of test cases once the software specification had been obtained. The proposed method and the implementation of the tool in this research would save the resources during the testing phase as well as enable the reusability of GUI test cases.