

วิทยานิพนธ์ฉบับนี้กล่าวถึงการแปลงแผนภาพซีควเอนซ์หลายแผนภาพไปเป็นพฤติกรรมในระดับปฏิบัติการของรหัสคำสั่งภาษาจาวา ซึ่งเป็นการสนับสนุนการทำการแปลงข้อมูลในระยะของการออกแบบซอฟต์แวร์ไปเป็นข้อมูลในระยะของการอิมพลีเมนต์โดยอัตโนมัติ อันจะทำให้สามารถลดค่าใช้จ่ายและเวลาในกระบวนการพัฒนาซอฟต์แวร์ อีกทั้งยังช่วยเพิ่มความถูกต้องให้กับซอฟต์แวร์ได้เป็นอย่างดี

ในการวิจัยได้มีการออกแบบขั้นตอนและกฎในการทำการแปลงดังกล่าว โดยผู้วิจัยให้ความสำคัญในการจัดข้อจำกัดของงานวิจัยที่มีอยู่เดิมอันเป็นอุปสรรคต่อการนำไปใช้งานจริง อันได้แก่การไม่สามารถทำการแปลงแผนภาพซีควเอนซ์ที่มีการส่งเมลเสจซ้อนกันหลายระดับและไม่สามารถรวมพฤติกรรมของแต่ละโอเปอเรชันซึ่งอาจปรากฏอยู่ในแผนภาพซีควเอนซ์หลายแผนภาพตามแต่ละสถานการณ์ที่แตกต่างกัน จากนั้นจึงได้ทำการอิมพลีเมนต์ขั้นตอนและกฎดังกล่าวด้วยภาษาเอ็กซ์เอสแอลที ซึ่งทำให้ได้เครื่องมือที่สามารถทำงานได้บนหลายแพลตฟอร์ม และได้ทำการทดสอบเครื่องมือดังกล่าวกับกรณีศึกษาสองกรณีด้วยการเปรียบเทียบรหัสคำสั่งที่ได้จากเครื่องมือกับรหัสคำสั่งที่ได้จากการทำการแปลงด้วยตนเอง พบว่าเครื่องมือสามารถให้ผลการทำการแปลง ๙ ที่ถูกต้อง ซึ่งเมื่อคิดเป็นร้อยละของขนาดของรหัสคำสั่งเชิงพฤติกรรมที่เครื่องมือสร้างได้จากกรณีศึกษาทั้งสองแล้ว จะคิดได้เป็นร้อยละ 100 สำหรับโอเปอเรชันที่ไม่มีพฤติกรรมที่แตกต่างกันตามแต่ละสถานการณ์ และคิดได้เป็นร้อยละตั้งแต่ 72.73 ถึง 82.61 สำหรับโอเปอเรชันที่มีพฤติกรรมแตกต่างกันตามแต่ละสถานการณ์การณซึ่งผู้ใช้จะต้องทำการเพิ่มเติมรหัสคำสั่งส่วนของการเลือกสถานการณ์ด้วยตนเอง

This thesis describes a transformation of multiple sequence diagrams into operation-level behavior of Java code, which promotes the automatic transformation of information in software design phase into information in implementation phase. As a result, it significantly reduces time and cost in the software development process, and also improves correctness of software.

In this research, steps and rules for the transformation are designed to overcome the major limitations of the previous research, which are obstacles for real situation usage. These limitations include the lack of ability to transform sequence diagrams that have multiple levels of call nesting, and to merge behavior from different scenarios. Then these steps and rules are implemented as a tool using XSLT language, which enables the tool to be applied in multiple platforms. The tool is tested with two case studies by comparing its transformation results to the manual transformation results. The comparison shows that the tool can give the correct results. 100 percent of code is generated for operations that do not have different behavior for each scenario. 72.73 to 82.61 percent of code is generated for operations that have different behavior for each scenario in which users have to add the code for selecting scenarios by themselves.