

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 บทสรุป

วิทยานิพนธ์ฉบับนี้ได้นำเสนอวิธีการ ค้นหาร่องรอยที่ไม่ดีสำหรับซอร์สโค้ดภาษาจาวา 4 ประเภท คือ Feature Envy, Message Chains, Middle Man, และ Inappropriate Intimacy (General Form) โดยจะพิจารณาจากองค์ประกอบของโปรแกรม และ ใช้วิธีการของโปรล็อกในการค้นหา ผู้วิจัยได้เลือกวิธีการแสดงความรู้ โดยใช้ตรรกะเพรดิคเตค ซึ่งจะออกแบบกฎโปรล็อกจากร่องรอยที่ไม่ดี และข้อเท็จจริงได้มาจากซอร์สโค้ดต้นฉบับภาษาจาวา จากนั้นได้ประเมินความสามารถของวิธีการ โดยการทดสอบกับโปรแกรมต้นฉบับภาษาจาวา 3 โปรแกรม แล้วเปรียบเทียบค่าที่ได้จากการใช้เครื่องมือ กับค่าที่ได้จากการค้นหาด้วยบุคคล พร้อมทั้งพัฒนาเครื่องมือตามวิธีการที่ได้นำเสนอ

ผลการทดสอบพบว่า วิธีการนี้สามารถค้นหาร่องรอยที่ไม่ดีประเภท Feature Envy (Attribute), Message Chains, Middle Man, และ Inappropriate Intimacy (General Form) ได้ 100% และ Feature Envy (Method) ได้ 82.61%

6.2 ข้อเสนอแนะ

1. วิธีการค้นหาร่องรอยที่ไม่ดีที่ผู้วิจัยได้นำเสนอใช้ได้กับ ร่องรอยที่ไม่ดี 4 ประเภท ดังรายละเอียดข้างต้น ซึ่งสามารถนำวิธีการออกแบบ และ สร้างกฎโปรล็อก ที่มีในงานวิจัยนี้ เพื่อเป็นแนวทางในการออกแบบสำหรับร่องรอยที่ไม่ดีประเภทอื่นๆ หรือ วิธีการทำรีเฟคทอริงต่อไป
2. เครื่องมือที่พัฒนาขึ้นในงานวิจัยนี้มีการทำงานที่ล่าช้า ดังนั้นถ้าต้องนำไปใช้งานจริง อาจต้องมีการปรับปรุงให้สามารถทำงานได้เร็วขึ้นมากกว่านี้
3. การค้นหาร่องรอยที่ไม่ดี ตามวิธีการที่ผู้วิจัยได้นำเสนอนอกจากเรื่องของกฎการสร้างกฎแล้วยังมีการสร้างข้อเท็จจริง ซึ่งมีความสำคัญมาก ดังนั้น ถ้าสามารถเก็บรายละเอียดเกี่ยวกับโปรแกรมต้นฉบับเพื่อมาสร้างเป็นข้อเท็จจริงได้ครอบคลุมกับซอร์สโค้ดต้นฉบับมากเท่าใด การค้นหาก็จะมีความถูกต้องมากเท่านั้น

6.3 ข้อจำกัดของงานวิจัย

1. การค้นหาร่องรอยที่ไม่ดีประเภท Inappropriate Intimacy (General Form) สามารถค้นหาได้เฉพาะในกรณีที่เกิดขึ้นจากการเรียกใช้คุณลักษณะเท่านั้น
2. การค้นหาร่องรอยที่ไม่ดีประเภท Message Chains สามารถค้นหาได้เฉพาะที่เกิดขึ้นกับ 3 คลาสเท่านั้น
3. เครื่องมือไม่สามารถสร้างข้อเท็จจริงของบางคำสั่งได้ เช่น คำสั่ง "String result = delegate.get(i).toString();" เครื่องมือยังไม่รองรับการเก็บ get(i) ได้ เพราะ ใน Statement นี้ เครื่องมือจะมองว่าเป็นการเรียกใช้ เมธอด toString() ของคลาส String โดยเรียกผ่าน อินสแตนซ์ของคลาสอื่นอีกที เป็นต้น
4. ในงานวิจัยนี้ผู้วิจัยยังไม่ได้คำนึงถึงเรื่องของวัตถุประสงค์ในการเรียกใช้เมธอดระหว่างคลาส เช่น ถ้าเมธอด "m1" เรียกใช้เมธอดที่อยู่ในคลาส "C" แล้วมีการนำค่าที่ได้มาคำนวณหรือ มีการเปลี่ยนแปลงค่า ก่อนที่จะส่งไปให้คลาสอื่น การเกิดกรณีนี้ขึ้นอาจจะไม่เป็น Middle Man ซึ่งถ้าจะพิจารณาตรงจุดนี้ อาจต้องทำในเชิงลึกมากกว่านี้