

การใช้ AOP เพื่อสร้างกรอบงานการจำลองระบบจราจรแบบยืดหยุ่น
Using AOP to Build the Flexible Traffic System Simulation Framework
มนัส โชคมาเสริมกุล, ทรงศักดิ์ ร่องวิริยะพานิช

บทคัดย่อ

คุณลักษณะสำคัญประการหนึ่งของการโปรแกรมเชิงวัตถุ คือ การสร้างคลาสใหม่ที่สามารถสืบทอดคุณสมบัติจากคลาสเดิมได้หลายระดับชั้น และการนำฟังก์ชันการทำงานที่มีอยู่เดิมกลับมาใช้งานได้ใหม่ ทำให้สามารถขยายการทำงานของโปรแกรมได้ง่าย สะดวก และรวดเร็วขึ้น แม้ว่ากลไกการสืบทอดคุณสมบัติให้ประโยชน์ต่อการพัฒนาโปรแกรมมาก แต่บ่อยครั้งที่การเพิ่มคุณสมบัติใหม่จำเป็นต้องเปลี่ยนแปลง แก้ไขคลาสเดิมที่มีให้สามารถรองรับการทำงานใหม่ได้ การเปลี่ยนแปลงโปรแกรมมักมีผลกระทบต่อคลาสเดิมทำให้ต้องปรับปรุงแก้ไข

AspectJ เป็นภาษาโปรแกรมที่สร้างขึ้นตามแนวคิดของ Aspect-Oriented Programming ซึ่งเป็นเทคนิคการโปรแกรมใหม่ที่ขยายความสามารถของการโปรแกรมเชิงวัตถุ โดยแยกฟังก์ชันการทำงานรองที่แฝงอยู่กับฟังก์ชันการทำงานหลักของเมธอด สร้างเป็นหน่วยการทำงานใหม่ เรียกว่า aspect โดยใช้กลไกโมเดลจุดเชื่อมสำหรับรวม aspect เข้ากับฟังก์ชันการทำงานหลักของเมธอด ณ ตำแหน่งที่ต้องการฟังก์ชันการทำงานของ aspect นั้น กลไกนี้ช่วยขยายความสามารถของโปรแกรมโดยไม่ต้องแก้ไขโครงสร้างคำสั่งของโปรแกรมเดิม รวมทั้งยังช่วยให้ผู้เขียนโปรแกรมสามารถเพิ่มฟังก์ชันการทำงานใหม่ให้กับโปรแกรมได้อีกด้วย จึงเหมาะสำหรับการประยุกต์กับระบบที่มีพฤติกรรมซับซ้อนและพลวัต เช่น การจำลองระบบจราจร เป็นต้น

งานวิจัยของวิทยานิพนธ์นี้ได้แสดงการใช้ AspectJ เพื่อเปลี่ยนแปลงและขยายความสามารถของกรอบงานการจำลองระบบจราจรแบบพลวัต โดยจำลองการจราจรโครงข่ายการเดินรถสองทิศทางบนโครงข่ายการเดินรถทิศทางเดียว และเปลี่ยนเป็นโครงข่ายการจราจรที่มีจุดกลับรถบนโครงข่ายการจราจรเดินรถสองทิศทาง ผลการทดลองแสดงให้เห็นว่า AspectJ สามารถเปลี่ยนแปลงโครงสร้างของโครงข่ายและพฤติกรรมเคลื่อนที่ของรถแบบพลวัตได้ รวมทั้งสามารถนำคำสั่งเดิมกลับมาใช้งานใหม่ได้ทั้งหมดโดยไม่ต้องแก้ไขคลาสเดิม และยังคงทำงานได้เช่นเดียวกับกรอบงานที่สร้างด้วยภาษาจาวาและดีไซด์แพทเทิร์น ในขณะที่การจำลองโดยกรอบงานที่ใช้ภาษาจาวาร่วมกับดีไซด์แพทเทิร์น การเปลี่ยนแบบโครงข่ายไม่สามารถนำคลาสเดิมกลับมาใช้งานใหม่ได้ทั้งหมด ต้องปรับปรุงแก้ไขคลาสเดิมบางคลาสเพื่อรองรับฟังก์ชันการทำงานใหม่ ดังนั้นในระบบอื่นซึ่งเป็นระบบที่มีพลวัตสามารถใช้ AspectJ เปลี่ยนฟังก์ชันการทำงานของโปรแกรม ณ เวลาประมวลผลได้เช่นเดียวกัน เช่น การเปลี่ยนจำนวนเลน การเปลี่ยนทิศทางของเลน หรือการเพิ่มเฟสของสัญญาณควบคุมไฟจราจร ณ เวลาประมวลผล เป็นต้น

ABSTRACT

Inheritance is one of the important properties of Object-Oriented Programming (OOP). The new class, which inherits from its parent, can use the properties and methods of its ancestors in the hierarchy. The reusable property leads to the more easily and rapidly software development..

Although the inheritance mechanism is useful, but, more often, the addition needs to modify some old classes of the program for supporting the new function. Thus, the new version affects the structure of the old.

AspectJ is the general propose Aspect-Oriented Programming, which is the new technology that expands the old paradigm as OOP. Its idea is used to separate the spaghetti codes, the same function which scatter in any method, and builds the cross-cutting concern or aspect. It relies on the join point model (JPM) and weaving its advice to the pointcut, which is the specific point in the sequence of program. This mechanism allows the programmer to change the ability of the program with no modification of its structure. Therefore, it is suitable to apply to the complex and dynamic system, like the traffic system simulation.

This thesis intends to present the use of AspectJ to build the flexible traffic system simulation framework, which simulate the twoway traffic on the oneway traffic and the u-turn traffic with the same results as the framework which builds on Java and design pattern. The result, in the scope of this thesis, shows that, the framework with AspectJ completely reuses all classes of the oneway traffic. The change to the twoway traffic and the u-turn traffic needs no modify classes. And it run in the same way as the version of Java and design pattern does. In the other hand, the other framework which builds on Java and design pattern needs to modify some classes of the oneway traffic for supporting the twoway and u-turn traffic. It can conclude that AspectJ makes change to the program dynamically and is flexibility to add or remove an aspect without any effect to the program. Otherwise, AspectJ can have the same effect to the other dynamic system also. For example, we can use it to change the number of lane, the number of traffic light's phase and the path of vehicle at runtime.