

시나리오 생성기의 구조 확장을 위한 Visitor 패턴과 Composite 패턴의 병합 기법

엄재학*,곽정훈*,성연식*

*계명대학교 게임모바일콘텐츠학과

{jaehak, jeonghoon, yunsick}@kmu.ac.kr

Integration Method of Composite Pattern and Visitor Pattern for Expanding the Structure of Scenario Generator

Jaehak Uam*, Jeonghoon Kwak*, Yunsick Sung*

*Department of Game Mobile Contents, Keimyung University

{jaehak, jeonghoon, yunsick}@kmu.ac.kr

요 약

Visitor 패턴은 구조 문제로 새로운 ConcreteElement 클래스가 추가될 때 Visitor 클래스 및 ConcreteVisitor 클래스를 수정해야 한다. 이 논문에서는 Visitor 패턴을 적용한 시나리오 생성기가 유연한 구조를 가지기 위해서 Visitor 패턴과 Composite 패턴을 병합하여 적용하는 기법을 제안한다. 그래서 시나리오 생성기에 ConcreteElement 클래스를 추가할 때마다 ConcreteVisitor 클래스의 멤버 함수를 유연하게 추가 및 삭제할 수 있는 클래스 구조를 제공한다.

1. 서론

시나리오를 자동으로 생성하는 연구에는 페이지안 확률을 도입한 연구가 있다[1]. 측정된 센서로 정의한 시나리오를 수집하고 페이지안 확률을 도입하여 새로운 시나리오를 생성한다.

시나리오 생성기는 시나리오 생성에 필요한 다양한 설정 과정 및 값을 처리하기 위해서 디자인 패턴의 Visitor 패턴[2]을 적용한다. 그래서 Visitor 패턴의 ConcreteVisitor 클래스에 멤버 함수 및 변수로 정의한다. 하지만 Visitor 패턴은 Visitor 패턴의 ConcreteElement 클래스와 ConcreteVisitor 클래스 간의 결합도를 증가시킨다. 그래서 ConcreteElement 클래스를 추가하기 어렵다. 이를 해결하기 위해서는 ConcreteVisitor 클래스 내의 멤버 함수들을 관련 있는 함수끼리 묶고 나누어 관리하는 방법이 필요하다.

이 논문에서는 Visitor 패턴의 ConcreteElement 클래스를 유연하게 추가하기 위해서 Visitor 패턴과 Composite 패턴을 병합하는 방법을 제안한다. Visitor 클래스 및 ConcreteVisitor 클래스 멤버 함수들을 Composite 패턴 기반으로 나누어 관리한다. 이와 같은 구조는 ConcreteVisitor 클래스와 ConcreteElement 클래스 간의 의존성을 낮출 수 있다.

2. 관련 연구

소프트웨어의 구조 개선을 위해서 디자인 패턴[2]을 적용해 볼 수 있다. 이 장은 디자인 패턴으로 구조 문제를 해결한 사례를 소개한다.

관점지향 소프트웨어를 개발하기 위한 방법론과 디자인 패턴을 통합한 연구가 있다[3]. 소프트웨어의 기능을 명확하게 정의하고 구현하기 위해서 관점지향적인 개발 방법이 필요하다. 이 연구는 관점 지향적인 개발 방법에 디자인 패턴을 적용하는 방법을 소개한다. 그리고 관점지향적인 개발 방법과 디자인 패턴을 출입보안 시스템에 적용하는 과정을 설명한다.

LBS 기반의 모바일 시스템을 구현하기 위해서 디자인 패턴 중에서 Abstract Factory 패턴, Prototype 패턴, Facade 패턴, Proxy 패턴, Command 패턴, 그리고 Mediator 패턴을 적용한 연구가 있다[4]. 디자인 패턴을 적용함으로써 모바일 장치의 메모리 한계 극복하고 시스템의 안정성을 높일 수 있다.

디자인 패턴의 확장 지점을 식별하는 연구도 있다[5]. 디자인 패턴은 레퍼런스의 흐름에 따라 확장된다. 이 확장 흐름을 파악하면 소프트웨어의 유지 보수를 용이하게 할 수 있다.

서비스 지향 컴퓨팅을 위한 GoF 디자인 패턴의 적용 기법 연구가 있다[6]. 디자인 패턴은 소프트웨어 설계할 때 발생하는 문제들을 해결하기 위한 범용성과 재사용성을 토대로 한 방법들이며, 이 연구에서는 객체들 간의 상호작용 및 연관 관계를 보여준다.

3. Visitor 패턴의 구조 문제 해결 기법

제한한 방법에서는 ConcreteVisitor 클래스의 멤버 함수를 유연하게 추가 및 삭제하기 위해서 서로 관련된 멤버 함수를 묶어 새로운 ConcreteVisitor 클래스로 정의한다. 그리고 정의한 클래스는 계층 구조로 관리한다.

계층 구조로 관리하기 위해서 디자인 패턴 중 Composite 패턴을 Visitor 패턴에 적용한다. (그림 1)은 제안한 패턴을 클래스 다이어그램으로 표현한 것이다.

Visitor 패턴의 Visitor 클래스를 Composite 패턴의 Leaf 클래스처럼 계층적 관리하기 위해 Component 클래스를 상속하는 LeafVisitor 클래스를 정의하고 Client 클래스는 Visitor 클래스를 직접 포함하지 않고 Component 클래스를 통해서 LeafVisitor 클래스를 관리한다. Visitor 패턴의 ConcreteVisitor 클래스는 ConcreteLeafVisitor 클래스로 명칭을 변경한다.

ConcreteElement 클래스에 대응되는 Visitor 클래스와 ConcreteVisitor 클래스의 멤버 함수는 서로 관련된 것을 묶어 ConcreteLeafVisitor 클래스를 추가 정의한다. ConcreteLeafVisitor 클래스는 LeafVisitor 클래스를 상속하고 다수 개 정의가 가능하다.

Element 클래스는 ConcreteElement 클래스가 상속하는 부모 클래스이자 추상 클래스이다. Component 클래스를 인자로 받는 Accept 함수를 추상 함수로 정의한다. ConcreteElement 클래스는 Accept 함수를 오버라이딩으로 구현한다. Accept 함수는 인자로 받는 Component 클래스를 통해 등록된 모든 ConcreteLeafVisitor 클래스를 참조해서 VisitorConcreteElement 클래스의 멤버 함수를 호출한다. 제안한 패턴에서 Element 클래스와 ConcreteElement 클래스는 기존의 Visitor 패턴을 그대로 따른다.

4. 결론

Visitor 패턴은 ConcreteElement 클래스를 추가하거나 삭제할 때 ConcreteVisitor 클래스의 수정이 발생한다. 이 논문에서는 ConcreteElement 클래스의 추가 및 삭제시

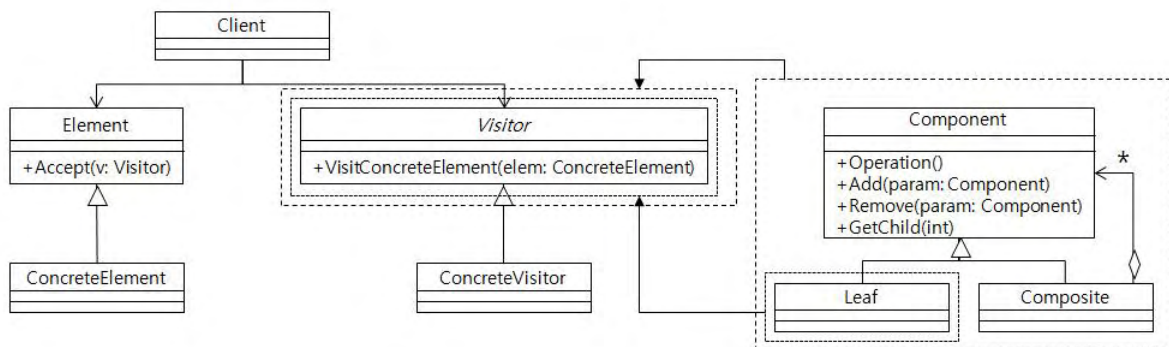
ConcreteVisitor 클래스의 변경이 없도록 Composite 패턴과 Visitor 패턴을 병합하는 방법을 제안하였다.

시스템의 문제를 해결하기 위해서 각각의 패턴을 따로 적용하는 것도 가능하지만 패턴이 가지고 있는 구조적인 문제가 해결이 안 되는 경우가 있다. 제안한 방법처럼 두 개 이상의 패턴을 통합해서 사용함으로써 패턴이 가지고 있는 근본 문제를 해결할 수 있다.

이 논문은 계명대학교 공학교육혁신센터에서 지원하는 2014년 특화동아리 지원 사업의 지원을 받아 수행되었으며 이에 감사드립니다.

참고문헌

- [1] Sung, Y., Helal, A., Lee, J. W., Cho, K., "Bayesian-based Scenario Generation Method for Human Activities," 2013 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (PADS), pp. 147-157, 19-20 May 2013, Montreal, Quebec, Canada.
- [2] Erich, G., Helm, R., Johnson, R., Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, Pearson Publisher, 2007.
- [3] 김문권, 라현정, 김수동, "서비스 지향 컴퓨팅을 위한 GoF 디자인 패턴 적용 기법", 정보처리학회논문지 D, Vol. 19-D, No. 2, pp. 188-201, 2012.
- [4] 이홍로, 백정호, 문영채, "디자인 패턴을 활용한 LBS 기반 모바일 시스템 구현", 한국지리정보학회지, Vol. 12, No. 1, pp. 26-35, 2009.
- [5] 김희천, 박찬진, 김택수, 유찬우, 이형원, "레퍼런스 흐름에 기반한 디자인 패턴의 확장 지점 식별", 정보처리학회, Vol. 19-D, No. 4, pp. 293-298, 2012.
- [6] 김태호, 천현재, 이홍철, "관점지향 소프트웨어 개발 방법론과 디자인 패턴을 적용한 출입 보안 시스템 개발", 한국산학기술학회논문지, Vol. 11, No. 3, pp. 943-950, 2010.



(그림 1) Visitor 패턴의 Composite 패턴 적용 기법