

가변 속성 추출을 위한 객체구성 방법에 관한 연구

이은서*, 김종수*
 *안동대학교 컴퓨터공학과
 e-mail:
eslee@andong.ac.kr
kimjs@andong.ac.kr

A Study on Cofigation of Objects for Variable Attribute Extraction

Eun-Ser Lee*
 *Computer Engineering, Andong National University

요 약

재사용의 활용이 많아짐에 따라서 객체를 효율적으로 구성하고자 하는 방법이 요구되게 된다. 객체는 추상화를 통한 효율적인 재사용을 지원하여 사용자에게 편의를 제공하고자 하는 측면이 있다. 따라서 본 논문에서는 속성을 기반으로 영역별 객체 구성방법을 제시하고자 한다.

1. 서론

객체는 상태와 그 상태에 적용되는 오퍼레이션들의 정의된 집합을 갖는 개체이다. 상태는 객체의 속성들의 집합으로 표현된다. 객체에 연관된 오퍼레이션들은 어떤 연산이 요구될 때 서비스를 요청하는 다른 객체들에게 서비스를 제공한다[1][2][3].

객체들은 객체 클래스 정의에 따라 생성된다. 객체 클래스 정의는 객체를 생성하기 위한 타입 명세와 템플릿 모두이다 객체 클래스 정의는 클래스의 객체에 연관되어야 하는 모든 속성들과 오퍼레이션들에 대한 선언을 포함한다. 객체는 속성을 기반으로 사용되게 되고 속성은 다음과 같은 특성을 갖는다[4][5][6].

속성은 자료 객체의 내용을 정의한다. 또한 자료 객체 인스턴스에 이름을 붙이거나, 인스턴스를 설명하거나, 다른 표의 또다른 인스턴스를 참조하거나 할 때 사용된다. 속성들중 일부는 식별자로 사용되며, 원하는 인스턴스를 찾을 때 이용된다. 식별자의 값은 자신만의 고유한 값이지만 항상 그렇지는 않다. 자료 객체에 적합한 속성은 문제의 맥락에 의해 정해진다[7][8].

본 논문에서는 속성을 체계적으로 관리하여 구현의 완성도를 높이고자 속성의 가변성을 추출하고, 이를 기반으로 클래스의 형태를 제시하고자 한다.

2. 속성 추출

객체의 속성을 추출하기 위하여 영역의 특성을 고려한

다. 영역은 많은 가변성을 포함하고 있으며, 영역의 가변적 요인을 분석하여 객체 구성을 수행하도록 한다. 가변성 추출은 요구사항 분석에서부터 관리를 해야 한다. 따라서 요구사항 추출 및 분석자료를 충실히 수행한 후에 속성의 가변성을 분석할 수 있게 된다.

영역에서 가변성 부분의 속성을 분석하여 가변성과 공용성을 구분하기 위한 요구사항을 수행한다.

속성은 행위와 그에 따른 특성이 포함되어 있으므로 이를 기반으로 가변성을 추출하기 위해서는 다음과 같은 양식을 활용하고자 한다.

속성 추출을 위한 양식은 다음과 같다.

<표 1> 속성추출을 위한 항목

추출항목	기능적 속성	비기능적 속성	의존관계 속성	사용되는 data	추적성
항목1					
항목2					

① 기능적 속성

기능적 요구사항의 속성을 포함하고 있으며, 다른 기능 속성과 연관성이 높을 가능성이 있다. 이를 추출하기 위하여 기능적 속성의 구분이 필요하게 된다.

② 비 기능적 속성

기능적 속성 이외의 것으로 하드웨어 속성을 포함하고 있다.

③ 의존관계 속성

의존관계 속성은 속성간의 연관성을 분석하고자 하는 부분이다. 속성간의 의존관계는 클래스 설계시 완성도를 높일 수 있게 된다.

④ 사용되는 data

속성과 연동되는 data를 식별하여 속성의 가변성 확인 시 data도 동시에 관리하여 가변성 식별하고자 한다.

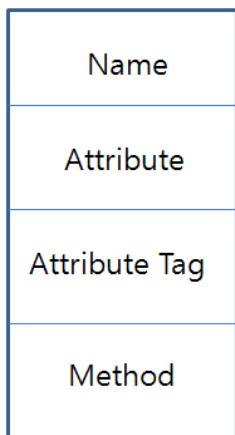
⑤ 추적성

항목의 가변성과 공용성의 이력을 관리하고자 추적성을 제공한다.

3. 객체구성

요구사항의 클래스 설계시 가변성과 공용성 속성을 적용하고자 한다.

가변성과 공용성은 아키텍처 구성시에도 활용이 되어서 가변성에 의한 계층별 오류 발생 기능을 구분할 수 있게 된다. 이와 같은 구조를 설계하기 위하여 새로운 클래스 형태를 제시하고자 한다.



(그림 1) 가변 속성 추출을 위한 클래스 다이어그램 형태

기존의 클래스와 다른 내용은 속성 태그를 두어서 추출된 속성의 가변성을 개별 속성별 구분을 지을 수 있다. 속성 태그는 가변성을 단순히 표현하는 것이 아니라, 요구사항 분석에서부터 설계 및 코딩까지 완성도를 높일 수 있도록 추적성까지 제공을 하고자 한다.

5. 결론

본 연구에서는 분석, 설계 구현의 완성도를 높이기 위하여 객체 속성의 가변성과 공용성을 추출하고자 한다. 이를 위하여 추출 양식과 형태를 제시하였다.

또한, 향후 연구에서 디스패처 사용의 효율성을 수치적으로 입증하고 효율적인 아키텍처를 개발한다. 또한 의존성의 문제를 해결한 방안을 찾고 향후 개선된 디스패처를 연구한다.

참고문헌

[1] 우치수 공역 “소프트웨어 공학 실무적 접근” Mc Groaw Hill
 [2] Roser S.Pressman, “Software Engineering :A practitioner’s approach”, Mc Groaw Hill
 [3] 최은만 “객체지향 소프트웨어 공학” 사이텍미디어
 [4] Amold, Gosling, Holmes “The java programming language” 4th Ed. addison-Wesley
 [5] 윤 청 “공적인 소프트웨어 개발 방법론” 생능출판사
 [6] 권기태 역 “소프트웨어공학” 홍릉과학 출판사
 [7] 최은만 “소프트웨어공학” 정익사
 [8] 최은만 “소프트웨어 공학론” 사이텍미디어