

효율적인 설계패턴의 검색 및 관리에 관한 연구

최영건*, 김귀정, 송영재

경희대학교 전자계산공학과

e-mail:ykchoi@case.kyunghee.ac.kr

A Study on Retrieval and Management of Efficient Design Patterns

Young-Keon Choi*, Gui-Joung Kim, Young-jae Song

Dept of Computer Engineering, Kyunghee University

요 약

본 연구는 점차로 늘어가고 있는 설계 패턴의 효율적인 관리와 재사용을 위하여 패킷 방식을 이용한 패턴 분류와 검색 방법을 제안하고 이를 UML 다이어그램으로 나타낼 수 있도록 설계·구현하였다. 재사용 가능한 설계 패턴은 기본적인 특성을 기준으로 영역별로 라이브러리에 저장하고 각 패턴의 특성을 표현하기 위하여 패킷과 항목을 설정하였다. 또한 모든 패턴의 패킷 항목에 대해 유사성을 측정하여 관련 패턴을 찾을 수 있도록 하였으며, 패턴 분류 체계에 따라 패턴을 추가·갱신함으로써 적절한 경험을 자동화된 방식으로 제공할 수 있도록 하였다. 제안한 분류 방식은 검색 결과 Gamma의 분류방법을 사용했을 때 보다 질의 작성이 간단하고 관련 패턴을 쉽게 찾을 수 있어 재사용에 용이하며 이를 UML 다이어그램으로 표현할 수 있다.

1. 서 론

객체지향 설계 개념을 이론적으로 제시하는 것으로 그치지 않고 프로그래밍에 적용 가능하도록 구체적으로 규격화시킨 것이 설계 패턴이다[1]. 수많은 설계 패턴의 재사용성을 향상시키기 위해서는 부품의 효율적인 관리와 설계 패턴을 보다 쉽게 이해하고 적용하기 위한 UML의 사용이 필수적이다.

설계 패턴을 사용함에 있어 설계자는 정확한 패턴 명이나 기능을 알아야 하는 어려움을 가지고 있다. 이러한 이유로 Gamma에 의한 분류만으로는 원하는 패턴을 검색하여 이용하기에 어려움이 있다. 이러한 단점을 보완하기 위해 패턴에 대한 여러 가지 특성을 추출하여 그것을 패킷항목으로 설정하는 패킷 분류방법을 사용하였다. 패킷 분류방법은 부품들이 갖는 공통적인 측면의 특성을 합성하여 하나의 패킷으로 표현한 후 하나의 부품을 여러개의 패킷으로 나타낸 것이다. 이러한 방법을 사용함으로써 패턴의 정확한 기능이나 패턴명을 알지 못하여도 원하는 패턴이 가지고 있는 특성을 패킷 항목에서 선택하므로써 사용자의 요구사항에 맞는 패턴을 효율적으로 검색할 수 있다. 그러나 이와 같은 패킷분류에 의한 패턴검색은 각

패턴에 대한 객체들 간의 관련성을 쉽게 찾을 수 없기 때문에 이를 보완하기 위해 패턴들 사이의 유사도를 측정하였다. 유사도 측정은 요구사항과 일치하는 패턴이 없을 경우에도 다수의 관련 패턴 검색을 지원하여 패턴 사이의 관련성을 표현할 수 있다. 또한 패턴 분류 체계에 따라 패턴을 추가·갱신할 수 있고 각 패턴에 대한 정보를 제공하여 패턴 검색 후 여러 후보 패턴 중 가장 적합한 패턴을 선택하여 재사용할 수 있도록 도와주며 저장된 패턴을 표현하기 위해 UML 다이어그램으로 나타낼 수 있도록 하였다.

본 연구는 패킷 분류방법을 사용하여 Gamma의 분류방법에서 제공하지 못한 질의에 대한 정확성과 유사도를 이용해 관련 패턴으로 효율적으로 검색함으로써 Gamma의 방법만으로 사용했을 때보다 좋은 결과를 얻을 수 있었으며, 검색된 패턴을 수정하고 새로운 패턴을 등록시킬 수 있으며 검색된 패턴을 UML 다이어그램으로 나타낼 수 있도록 하였다.

2. 관련 연구

본 절에서는 기존에 연구된 부품 분류 방법과 유사도 측정 방법에 대하여 살펴본다

2.1 부품 분류방법

분류방법으로는 Gamma에 의한 패턴 분류가 있고 GTE사 시스템의 패킷분류가 있다. Gamma의 패턴 분류는 설계 패턴을 크게 두 종류로 분류하였는데, 첫 번째는 패턴이 하는 역할에 따라 생성패턴, 구조패턴, 행위패턴으로 구분하였다. 두 번째는 각 패턴의 적용에 대한 범위로써 클래스와 객체로 구분하였다.[2] 또한 Diaz에 의해 제안된 GTE사의 시스템은 패킷 분류 방법을 이용한 대표적인 재사용 시스템이다[3]. 패킷 분류 방법은 열거형 분류 방법의 단점인 부품의 확장성을 개선시킨 방법으로 부품들이 갖는 공통적인 측면의 특성을 함성하여 하나의 패킷으로 표현한 후 하나의 부품을 여러 개의 패킷으로 나타낸다.

2.2 Dice의 유사성 측정방법

문서들의 유사성을 결정하는 대표적인 유사성 측정 방법은 Dice 방법이다[4]. 이 방법은 먼저 두 문서 (i, j)간에 가중치를 결정하는데 가중치는 두 문서의 용어들에 의해 결정되며 결정된 가중치를 이용하여 두 문서의 유사성을 식 (1)의 정의에 따라 측정한다.

$$S_{ij} = \frac{2 \times C_{ij}}{A_i + B_j} \quad \text{식(1)}$$

여기에서 S_{ij} 는 문서 i와 j사이의 유사도를 나타내며 A_i 는 문서 i에 존재하는 의미 있는 단어의 수이고, B_j 는 문서 j에 존재하는 의미 있는 단어의 수이고, C_{ij} 는 문서 i와 j에 공통적으로 존재하는 의미 있는 단어의 수이다. 이 방법은 패킷 항목 사이의 관련성 유지에 대한 문제를 보완할 수 있다.

3. 설계 패턴 관리 시스템

설계 패턴 관리 시스템은 패턴 라이브러리, 패턴 관리 시스템, 패턴 뷰어 3부분으로 구성되어 있으며 패턴의 효율적인 관리를 위하여 패턴을 패킷으로 분류하고 각 패킷 항목을 선택함으로써 사용자 요구사항에 따른 패턴을 검색할 수 있도록 하였으며 유사성을 이용한 관련 패턴을 검색할 수 있도록 하였다. 또한 패턴 등록 및 삭제, 수정이 가능한 인터페이스를 지원하며 UML 다이어그램으로 표현하는 패턴 뷰어를 지원한다.

3.1 설계 패턴 재사용 시스템 구조

설계 패턴 재사용 시스템은 (그림 1)에서와 같이 크게 패턴 라이브러리, 패턴 관리 시스템, 패턴 뷰어 3부분으로 구성된다. 기존의 시스템과는 달리 패턴을 부품화시켜 도메인 종속 여부에 따라 라이브러리에서

분리하여 관리하며, 재사용성을 극대화하기 위하여 실제 부품이 적용될 수 있는 상황을 제시할 수 있도록 패턴을 패킷 분류하여 효율적인 검색이 가능하도록 하였고 검색된 부품은 UML의 클래스 다이어그램으로 표현되어 객체지향 방법론을 효과적으로 수행할 수 있도록 설계하였다.

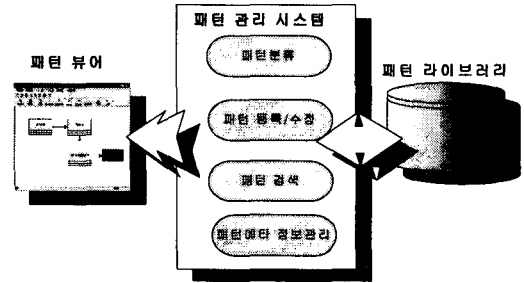


그림 1 패턴 재사용 시스템

(그림 2)는 본 연구에서 제시한 패턴 검색 및 관리 과정을 나타낸 것이다. 패킷 분류된 패턴은 유사도가 측정되어 적용 도메인 영역에 따라 라이브러리에 저장된다. 이때 각 패턴의 패턴 이름, 다른 패턴과의 관련성, 분류를 위한 패킷과 각 항목에 대한 정보 등이 메타정보로 관리된다. 사용자의 요구에 따라 각 패킷 항목이 선택되는데 다양한 재사용 상황과 적용 영역, 역할 등의 항목을 제공하여 적합한 부품을 검색할 수 있도록 해주고, 유사도에 의해 다수의 재사용 가능한 후보 패턴까지 검색해준다. 선택된 패턴은 UML 다이어그램으로 표현되어져 재사용 부품으로 사용된다.

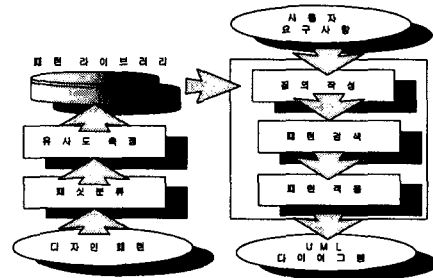


그림 2 패턴 검색 및 관리 과정

3.2 제안한 패턴 분류방법

제안한 패턴을 분류하는 과정에서의 패킷 구성 방법과 각 패킷 항목에 대해서 알아본다. <표 1>에서 패킷은 패턴의 기본적인 기능과 목적 그리고 패턴을 구현하고 실행할 수 있는 실제 상황을 기술할 수 있도록 도메인 종속 여부, 패턴이 속한 영역, 적용 목적, 적용 범위의 4개 패킷을 포함하도록 정의하였다.

<표 1> 각 패시의 항목

패시	특성
도메인 종속 여부	도메인 독립 도메인 종속
영역	클래스(class) 객체(object)
목적	생성(creation)패턴 구조(structural)패턴 행위(behavioral)패턴
적용범위	요구 사항을 객체로 분할 시스템 기능을 객체로 분할 객체의 인터페이스 지정 객체 지향 프로그래밍 적용 델리게이션 메카니즘 적용 런타임,컴파일타임 구조의 관계규정

도메인 종속 여부는 각 패턴의 재사용 시 특정 응용 도메인에 국한된 패턴인지를 선택하는 기준이 되며 또한 사용자가 특정 도메인에만 적용 가능한 패턴을 따로 관리하거나 검색할 수 있도록 해준다. 영역은 각 패턴의 적용에 대한 범위로서 클래스와 객체로 구분되며 패턴이 하는 역할에 따라 생성 패턴, 구조 패턴, 행위 패턴으로 구분된다. 적용 범위에 대한 패시 항목 중 첫 번째는 요구 분석서에 제시된 명사와 동사를 바탕으로 그에 상응하는 클래스와 오퍼레이션을 추출하는 방법을 사용하지만 객체지향 설계의 내용이 실제와 연관이 적은 클래스로 구성되는 경우나 추상화하기 힘든 개념을 프로그래밍에서 다룰 수 있는 객체로 변환시키는 경우에 이러한 문제를 해결하는데 사용할 수 있도록 하였다. 두 번째는 큰 기능의 시스템 모듈을 객체 단위의 서브 모듈로 표현하고자 할 때 많은 모듈을 효과적으로 관리하고자 할 때 사용할 수 있도록 항목으로 제시하였다. 세 번째는 객체의 인터페이스를 지정하는 솔루션으로 인터페이스의 종류와 인터페이스간의 연관 관계를 지정하는 경우에 사용할 수 있도록 하였으며 네 번째 항목은 실제 프로그램을 작성 할 때 클래스 상속과 인터페이스 상속의 차이에서 오는 문제를 해결하거나 객체 생성 시 객체 연동에 필요한 인터페이스의 구현에 해결책을 줄 수 있도록 하였다. 또한 실제 구현 시 개발자가 코드를 재사용 할 때 상속 기법의 재사용성이 포함된 복합(Composition) 기법인 델리게이션(Delegation) 메카니즘을 사용함으로써 재사용성이 증가할 수 있도록 하였으며, 마지막항목은 복잡한 런타임 구조를 구축하거나 런타임에 대한 구조와 컴파일 타임 구조와의 구

분을 명확히 하기 위하여 사용할 수 있도록 하였다.

3.3 유사도 측정

패시의 항목이 많아지면 그들 사이에 관련성의 명세가 어렵고, 검색 시간이 길어진다는 단점이 있다. 이를 보완하기 위하여 본 연구에서는 사용자가 요구하는 패턴에 대해서 쉽고 효율적인 검색이 가능하도록 패턴 사이의 유사도를 측정하였다.

유사도 S_{ij} 를 식(2)와 같이 정의하였다.

$$S_{ij} = \frac{2 \times C_{ij}}{(I_i + 1) + (I_j + 1)} \quad \text{식(2)}$$

I_i 와 I_j 는 패턴 i 와 패턴 j 에서 각각 선택된 항목의 개수이고, C_{ij} 는 패턴 i 와 j 에서 공통적으로 선택된 항목의 개수이며 I_i 와 I_j 에 1을 더한 이유는 패턴의 구성 클래스 수에 대한 항목을 비교하기 위해서이다.

이와 같이 유사도를 측정함으로써 요구 사항과 일치하는 패턴이 없을 경우에도 다수의 관련 패턴 검색을 지원할 수 있도록 유사도로써 패턴 사이의 관련성을 표현하였다.

3.4 패턴 검색

라이브러리에 저장되어 있는 패턴은 특성별로 패시로 분류되고 검색시 각 패시 항목을 선택함으로써 패시 항목과 일치하는 속성을 가진 패턴이 검색되도록 하였다.

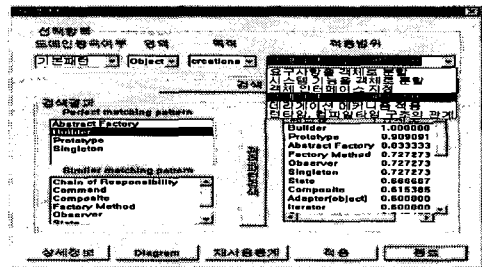


그림 3 패턴 검색

(그림 3)에서는 검색된 패턴에 대해서는 일치하는 항목과 구성 클래스의 수에 따라 계산된 유사도를 바탕으로 관련 패턴 리스트가 제공되며 또한 4개의 패시에 대해서 선택한 항목과 모두 일치되는 패턴이 없을 경우의 검색을 위하여 4개의 패시 항목 중 3개의 항목이 일치하는 패턴 리스트의 검색이 가능하도록 하였다. 다수의 후보 패턴들에 대해서는 패시 항목에 대한 속성과 구성 클래스 수, 패턴 파일명, 재사용 횟수 등의 상세 정보가 제공되어 진다. 검색 결과 나타난 각각의 패턴에 대해서는 「관련패턴검색」 버튼을

클릭하면 검색된 패턴과 일치하는 항목과 구성 클래스 수에 따라 계산된 유사도가 오른쪽 창에 리스트로 제공되며 재사용 될 패턴을 선택하기 전에 각 패턴의 자세한 정보와 다이어그램의 형태를 보기 위해서 「상세정보」와 「Diagram」 버튼을 사용할 수 있다.

검색된 후보 부품 중 재사용 될 패턴이 선택되어지면 패턴명을 더블 클릭하거나 「적용」 버튼을 클릭하면 패턴이 UML의 클래스 다이어그램으로 표현되어져 윈도우에 나타난다.

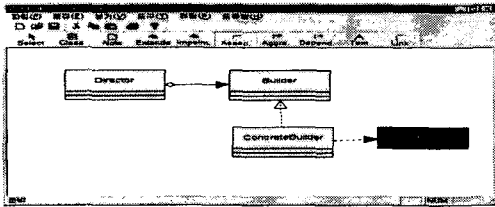


그림 4 빌더 패턴 적용 예

(그림 4)은 검색된 후보 패턴 리스트 중 Builder 패턴을 선택했을 때의 클래스 다이어그램을 보여준다.

3.5 패턴 관리

소프트웨어 재사용 시스템에서 부품의 효율적인 검색과 함께 부품의 추가 및 삭제·수정과 같은 부품 관리란 매우 중요한 부분을 차지한다. 따라서 검색할 패턴의 확장성을 고려하여 현재 라이브러리에 패턴을 추가로 삽입하여 검색이 가능하도록 하였으며 추가된 패턴과 다른 패턴들과의 유사도가 자동적으로 계산되어 메타정보로 저장되도록 하였다. 또한 삭제나 수정시 라이브러리에 있는 패턴의 데이터베이스 정보를 제공하여 삭제·수정하고자 하는 패턴에 대한 정보를 미리 알 수 있게 하였다.

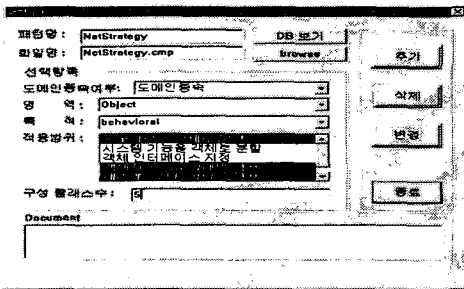


그림 5 패턴 추가

(그림 5)는 “NetStrategy”라는 새로운 패턴을 추가하는 과정을 보여준다. 추가할 패턴을 클래스와 연관 관계 그리고 텍스트 등의 클래스 다이어그램으로 표

현한 다음(그림 5)와 같은 인터페이스를 이용하여 새로운 패턴을 추가할 수 있다.

4. 결 론

본 연구는 설계 패턴의 효율적인 관리와 재사용을 위하여 패킷 방식을 이용한 패턴 분류와 검색 방법을 제안하였고 이를 UML 다이어그램으로 나타낼 수 있는 설계 패턴 관리 시스템을 구축하였다. 패킷은 패턴을 구현하고 실행할 수 있는 실제 상황을 기술할 수 있도록 특정 응용 도메인 종속 여부와 패턴이 속한 영역, 적용 목적 그리고 적용 범위 등을 포함하도록 정의하였으며 질의에 부합하는 패턴이 검색되지 않았거나 요구 사항과 일치하는 부품이 없을 경우에 다수의 유사한 부품의 검색을 지원할 수 있도록 패턴들 사이에 유사도를 측정하여 검색 시 유사도가 높은 순서로 관련 패턴이 유사도에 따라 리스트 형식으로 제공될 수 있도록 구현하였다. 또한 설계 패턴을 모델링하기 위하여 UML의 다이어그램 표기 방법 중 클래스 다이어그램을 이용하여 라이브러리에 저장된 재사용 가능한 패턴을 표현함으로써 객체지향 분석과 설계과정을 보다 효과적으로 수행할 수 있도록 하였다. UML 모델링을 사용한 패턴 뷰어와 인터페이스를 이용하여 새로운 패턴의 추가와 삭제·수정 등의 패턴 관리가 가능하도록 하였고 전반적인 부품의 재사용성을 향상시킬 수 있도록 하였다.

앞으로의 연구방향은 유사도 측정 시 항목별로 차등있게 가중치를 설정하는 방법과 다양한 패턴에 대한 폭 넓은 적용이 요구된다.

참 고 문 헌

- [1] <http://www.omg.org>
- [2] E.Gamma, R.Helm, R.Johnson, and J.Vlissides, "Design Pattern : Elements of Reusable Object-Oriented Software," Addison-Wesley, 1995.
- [3] R.Prieto-Diaz and P.Freeman, "Classifying Software for Reusability," IEEE Software, Vol.4, No.1, pp.6-16, Jan. 1987.
- [4] William B.Frakes and Ricardo Baeza-Yates, "Information Retrieval," Prentice Hall, pp.419-442, 1992.