

패턴정보를 이용한 디자인패턴 검색 시스템 설계

A Design of Design Pattern Retrieval System using Pattern Information

김귀정
건양대학교

Kim Gui-Jung
KonYang Univ.

요약

본 연구는 디자인패턴의 효율적인 관리와 재사용을 위하여 패턴 정보를 이용한 디자인패턴 검색 시스템을 설계하였다. 패턴 정보는 패턴 속성정보와 패턴 메타정보로 구성하였고, 패턴 속성정보는 패턴 패킷 분류와 패턴 검색을 위한 유사도 측정에 이용되며, 패턴 구조를 UML로 모델링하기 위하여 패턴 메타 정보를 이용하였다. Gamma 분류 방법을 확장하여 각 디자인패턴이 사용될 수 있는 여러 경험적 상황을 패킷 항목으로 설정하고 패턴 정보로 관리하였다.

Abstract

In this paper, we implemented design pattern retrieval system for efficient management and reusability of design patterns. Pattern is consisted of property information and meta information. Property information is used for similarity measurement on classification and retrieval of patterns. Meta information is used for UML modeling of patterns. We classified design patterns with the empirical scope in addition to Gamma's basic classification.

I. 서론

수 백 가지에 이르는 패턴을 효율적으로 재사용하기 위해서는 패턴이 수행하는 역할에 따른 3가지 분류와 패턴명만으로 검색해야하는 Gamma의 분류 방법[1]으로는 사용자가 원하는 패턴을 찾아 적용하는데는 많은 어려움이 따른다. 그러므로 상황에 맞는 적절한 경험을 자동화된 방식으로 제공할 수 있는 패턴 분류 방법과 검색, 그리고 새로운 패턴의 등록과 UML을 이용한 모델링 도구 지원 등의 제공이 요구된다[2]. 따라서 본 연구는 Gamma 패턴 외에도 사용자에게 의해 신규 패턴의 등록과 관리가 가능하며, 증가하는 패턴에 대한 효율적인 분류와 검색을 지원하고, 패턴이 UML로 표현되고 모델링 될 수 있는 디자인패턴 검색 시스템을 설계하였다. 재사용성을 극대화하기 위해 패턴 구조를 재사용 컴포넌트로 사용될 수 있도록 구성하고, 패턴 구조를 가시적으로 모델링하기 위하여 UML기반의 패턴 정보를 이용하였다. 패턴 정보는 일반적인 패턴의 속성정보를 가지고 있는 패턴 속성정보와 패턴 구조 정보를 가지고 있는 패턴 메타정보로 구성하였다. 패턴 속성정보는 컴포넌트화된 패턴의 재사용성을 향상시키기 위하여 Gamma의 패턴 분류 방법을 확장한 패킷 분류와 요구사항에 맞는 패턴을 검색하기 위한 유사도 측정에 이용된다. 또한 디자인패턴의 구조를 UML로 모델링하고 컴포넌트화 시키기 위하여 디자인패턴 구조로부터 추출한 클래스 메타모델, 속성 메타모델, 오퍼레이션 메타모델, 관계 메타모델 등과 같은 패턴 메타정보를 이용한다.

패턴 등록을 비롯한 패턴 관리와 패턴구조 모델링, 검색의 사용자 인터페이스를 구현하고, 패턴 정보와 구조를 저장할 수 있는 데이터베이스를 구축하였다.

II. 관련연구

객체 지향 프로그램 개발을 위한 많은 도구들이 출현하였고, 설계를 위한 부품, 패턴 등을 지원함으로써 시스템의 효율적 구축을 목적으로 많은 연구가 진행되고 있다. OmniBuilder[3]는 설계 단계에서 패턴을 사용하는 도구로써 애플리케이션의 전체 생명주기를 거쳐 애플리케이션을 생성하는 CASE 도구이며, 복잡한 애플리케이션 개발에 필요한 시간과 비용을 감소시킨다. 디자인패턴은 OmniBuilder안에서 보통의 객체처럼 모델링되고, 애트리뷰트·서비스·이벤트·메소드와 행위를 가질 수 있다. 그러나 Gamma의 디자인패턴과 그 분류 방식은 다르며 특별한 검색 도구가 지원되지 않기 때문에, 계속적으로 추가되는 패턴들을 효율적으로 관리하고 검색하는데 어려움이 있다.

ModelMaker[4]는 볼랜드 델파이를 위한 클래스와 컴포넌트 패키지를 개발하기 위한 CASE 도구이다. 클래스·유스케이스와 시퀀스 다이어그램을 지원하며, 클래스와 멤버들 사이에 모든 관계를 저장하고 유지하는 동적인 모델링 엔진을 가지고 있어 클래스명을 바꾸거나 상위클래스의 내용을 바꾸는 행

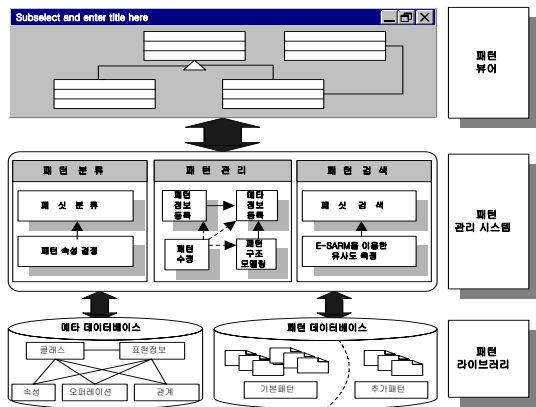
위는 생성된 소스 코드에 즉시 반영된다. ModelMaker의 특징은 패턴이 관련된 코드에 패턴을 삽입한 후에 동적으로 유지된다는 것이다. 그러나 UML과 같은 객체 지향 방법론을 이용한 모델링 도구가 독립적으로 존재하지 않고, 마치 매크로처럼 동작하는 템플릿만을 지원하기 때문에 패턴 구조를 다양하게 모델링하는데 어려움이 있다.

IBM의 디자인패턴 라이브러리[5]는 디자인패턴의 구현을 자동화하는 도구이다. 사용자가 주어진 패턴을 위한 특정 애플리케이션 정보를 제공하면, 자동적으로 규정된 패턴 코드를 생성한다. IBM의 DPL은 재사용 컴포넌트 구현에 적합한 분산 구조를 갖으며, 사용자 제공 정보로부터 자동적으로 디자인패턴 코드를 생성한다. 그러나 신규 패턴의 등록을 지원하지 못하여 자체적인 모델링 도구나 방법을 지원하지 못한다.

III. 디자인패턴 검색 시스템 설계

1. 시스템 구조

본 논문에서 제안한 디자인패턴 검색시스템은 크게 패턴 라이브러리, 패턴 분류·관리·검색을 담당하는 패턴 관리 시스템, 그리고 패턴 뷰어로 구성되며, (그림 1)은 구현한 디자인패턴 검색시스템의 구조를 보여준다. 패턴 라이브러리는 일반적인 패턴의 속성정보가 저장되는 패턴 데이터베이스와 UML로 표현된 패턴 구조를 메타모델로 저장하는 메타 데이터베이스로 구성된다. <표 1>은 2가지 패턴 정보를 나타낸 것이다.



▶▶ 그림 1. 디자인패턴 검색 시스템의 구조

[표 1] 패턴 정보

패턴 정보	데이터베이스	사용 범위
패턴 속성정보	패턴 데이터베이스	패킷 분류, 유사성을 이용한 검색
패턴 메타정보	메타 데이터베이스	UML 모델링

패턴 속성정보는 디자인패턴을 분류하고 검색하는데 필요한 정보를 포함하고 있으며, 이 속성에 의해 패턴 데이터베이스에 저장된 패턴은 도메인 종속여부에 따라 기본 패턴과 추가 패턴으로 분류되어 저장된다. 패턴 메타정보는 패턴을 UML로 모델링하는데 필요한 정보를 포함하고 있다. 디자인패턴 구조를 생성하기 위해서 패턴 등록 시 그 구조를 UML로 모델링하여 구성 정보를 데이터베이스화하였다.

패턴 관리시스템은 패턴의 분류와 등록, 검색, 패턴 정보 관리 등의 역할을 수행하여 라이브러리와 패턴 뷰어 사이의 데이터 흐름과 처리를 담당한다. 증가하는 패턴을 효율적으로 관리·검색하기 위해, 기존의 패턴명으로 검색하는 스트링 매칭 검색 외에 응용 도메인 종속 여부와 패턴이 속한 영역, 적용 목적 그리고 적용 범위에 의한 분류의 패킷 항목을 갖는 패킷 분류를 지원하며, 관련 패턴까지의 검색이 가능하도록 기존에 클래스 컴포넌트에 사용되었던 E-SARM을 패턴 검색에 적용하여 재사용성을 극대화시킬 수 있도록 하였다.

패턴 뷰어는 검색 결과 선택된 패턴을 클래스 다이어그램으로 보여주거나, 신규 패턴을 모델링하고 등록하기 위해서 사용된다. 또한 클래스의 속성을 편집하기 위한 편집기와, 검색 인터페이스, 패턴 추가·삭제의 패턴 관리기 등의 인터페이스를 제공한다. 편집기를 통해 모델링된 패턴 구조는 패턴 등록 시에 일반 패턴 속성정보와는 별도로 디자인패턴을 구성하는 메타모델들을 이용하여 메타 데이터베이스를 구축하게 되고, 검색 시 호출되는 패턴은 메타 데이터베이스에 저장되어 있는 메타모델을 이용해 UML 모델링 규칙에 따라 클래스 다이어그램으로 표현된다.

2. 디자인패턴 정보

2.1 패턴 속성정보

패턴의 속성정보는 기본적으로 Gamma의 디자인패턴 표현에 사용되는 형식을 사용하며, 그 정보를 저장하는 패턴 데이터베이스에는 그 외에도 4가지 방법에 의한 패턴의 패킷 분류 정보와 재사용 정보가 추가로 저장된다. (그림 2)는 패턴의 속성정보에 사용된 데이터를 나타낸 것이다. 패턴 속성정보에는 패턴 이름, 패턴 파일명, 분류를 위한 패킷과 항목, 패턴구조의 이미지, Gamma의 디자인패턴 표현 방법, 관련성, 패턴의 재사용 횟수, 구성 클래스의 수 등이 있다. 패킷과 항목들은 각 패턴이 패킷 항목 중 어느 부분에 속하는가를 나타내어 항목과 패턴 사이의 관계를 지정해 주는 역할을 함으로써 관련성을 표현할 수 있도록 하였다. 이 관련성은 검색 시 패턴간 유사성을 계산하는데 사용된다. 항목과 패턴의 연결 상태는 패턴이 분류된 패킷 항목에 따라 초기 값으로 설정되고, 이 항목과 패턴 사이의 연결을 통하여 활성값이 전달되어 항목과 관련된 패턴들

을 검색할 수 있다. 디자인패턴들은 특정 응용 도메인에 상관 없이 적용될 수 있는 도메인 독립적인 기본 패턴과 특정 응용 도메인에 따라 선택적으로 적용될 수 있는 도메인 종속적인 추가 패턴으로 구분된다. 도메인 독립적인 기본 패턴은 Gamma가 정의한 23개 패턴을 이용하였으며 도메인 종속적인 추가 패턴은 사용자가 특정 응용에 따라 추가로 저장한 패턴을 포함한다.

필드명	데이터 형식	의미
index	일련번호	일련번호
pattern	텍스트	패턴이름
pattern_name	텍스트	저장된 패턴 파일명
strDomain	텍스트	도메인 종속 여부
strScope	텍스트	응용에 따른 분류
strPurpose	텍스트	사용 목적에 따른 분류
strEmpirical	텍스트	실제 응용 상황
image	OLE 객체	패턴 구조 이미지
intent	텍스트	의도와 의도
Motivation	텍스트	패턴의 추상적 기술의 이해를 돕는 예제
Participant	텍스트	디자인패턴에 참여하는 클래스와 오브젝트들
Consequence	텍스트	패턴 사용의 결과
Application	텍스트	패턴이 적용될 수 있는 상황
reusecount	숫자	재사용횟수
reuserate	숫자	패턴의 재사용률
class	숫자	구분 클래스 개수

▶▶ 그림 2. 패턴속성 정보 테이블

2.2 패턴 메타정보

디자인패턴을 UML로 모델링하고 컴포넌트화 시키기 위하여 패턴 구조를 메타 데이터로 저장하여 패턴 메타 데이터베이스로 구축하였다.

<표 2>는 클래스와 인터페이스 메타모델의 정보를 갖는 테이블 구조이며, 인터페이스는 스테레오타입이 Interface인 클래스로 기술된다. Visibility 필드는 접근권을 말하며, UML에서는 특정 언어에 독립적인 방법으로 표기한다. Stereotype 필드는 클래스를 한 단계 높은 차원에서 분류한 것이고, Constraint 필드는 제약 사항을 말하며 정형화된 규칙은 존재하지 않는다[6].

<표 3>은 <표 2>의 클래스와 인터페이스에 속한 애트리뷰트와 오퍼레이션 메타모델을 정보로 갖는 테이블의 구조이다. 이중 Value 필드는 애트리뷰트에만 해당하는 메타 데이터이다.

[표 2] 클래스/인터페이스 메타모델

필드명	데이터 형식	의미
ID	varchar	클래스, 인터페이스의 ID
Link_Pattern	varchar	소속된 디자인패턴명
Name	varchar	클래스, 인터페이스명
Visibility	varchar	가시성
Stereotype	varchar	스테레오타입
Constraint	varchar	제약 조건

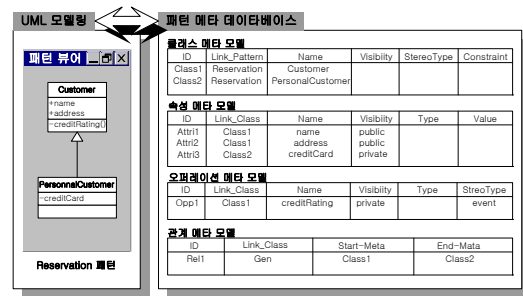
[표 3] 애트리뷰트/오퍼레이션 메타모델

필드명	데이터 형식	의미
ID	varchar	애트리뷰트/오퍼레이션 ID
Link_Class	varchar	소속된 클래스/인터페이스명
Name	varchar	애트리뷰트명/오퍼레이션명
Type	varchar	데이터 타입
Value	varchar	애트리뷰트값
Visibility	varchar	가시성
Stereotype	varchar	스테레오타입
Constraint	varchar	제약 조건

[표 4] 관계 메타모델

필드명	데이터 형식	의미
ID	varchar	관계 ID
Name	varchar	관계명
Start_Meta	varchar	관계가 시작되는 메타모델
End_Meta	varchar	관계가 끝나는 메타모델

<표 4>는 Dependency, Associations, Generalization, Refinement, Aggregation 같은 관계 메타모델들의 테이블 구조이며, 관계가 시작되고 끝나는 메타모델들의 ID를 기술한다.



▶▶ 그림 3. 패턴 메타정보의 데이터베이스 구축

패턴 뷰어에서 UML로 모델링 된 패턴 구조는 (그림 3)과 같이 각 메타 모델들을 추출해 메타모델 정보를 갖는 메타 데이터베이스에 저장된다. 패턴명이 Reservation인 패턴에서 Customer 클래스와 PersonalCustomer 클래스, 각각의 애트리뷰트와 오퍼레이션, 그리고 관계를 추출하여 메타모델 데이터베이스를 구축하였다. UML을 기반으로 한 패턴 구조 모델링을 지원하기 위하여 패턴 구조를 구성하는 메타모델을 이용하였다. 패턴 등록 시에 패턴 모델링 모듈이 수행되며, 사용자가 디자인패턴 구조를 모델링하면 일반 패턴의 속성정보는 패턴 속성정보 데이터베이스에, 패턴 구조는 메타 데이터베이스에 저장된다.

IV. 결 론

본 연구는 디자인패턴이 가지고 있는 속성과 특징, 메타모델 등의 패턴 정보를 이용하여 디자인패턴 검색 시스템을 설계하였다. 패턴 정보는 패턴의 분류와 검색을 위한 데이터를 가지고 있는 패턴 속성정보와 검색된 패턴을 뷰어에 UML로 표현하는데 필요한 패턴 메타정보로 구성된다. 본 시스템은 디자인 패턴의 효율적인 관리와 재사용을 위하여 패킷 방식을 이용한 패턴 분류와 패킷 항목과 패턴간의 관련성을 이용한 검색 방법을 사용하여 검색된 패턴을 UML 클래스 다이어그램으로 나타낼 수 있는 디자인패턴 검색 시스템이다. 앞으로의 연구방향은 유사성을 패턴 구조적인 면에서 비교하여 중복저장의 문제를 해결하고 패턴 클러스터링을 통하여 효율적인 라이브러리를 구축하는데 있다.

■ 참고 문헌 ■

- [1] <http://www.formsplayer.com/why-xforms.html>
- [2] <http://www.w3.corg/MarkUp/Forms/>
- [3] Richard Cardone, Danny Soroker, and Alpana Tiwari, "Using XForms to Simplify Web Programming," WWW 2005, May 2005, Chiba, Japan.
- [4] <http://www.modelmaker.demon.nl/>.
- [5] F.J. Budinsky, M.A. Finnie, J.M. Vlissides, P.S. Yu, "Automated code generation from design patterns," Object technology, IBM Systems Journal Vol.35, No.2, 1996.
- [6] <http://www.yy.ics.keio.ac.kr/~suzuki>, 1999.