패턴정보저장소를 이용한 인덱스 순서관계정보모델 설계 및 구현

선수균
*동원 대학 e-비즈니스과

Design and Implement Index Sequence Relation Information Model Using Pattern-In Repository

Su-Kyun Sun
Dept. of e-besiness, DongWon Collage, Korea

요 약

최근에는 웹 환경에 적합한 개방형과 컴포넌트들을 효율적으로 분류하고 추출하는 방법이 연구되고 있다. 본 논문은 개발환경에서 생성되는 산출물들 중 디자인패턴을 통합 관리하고, 추출, 검색하여 관리해 주는 인덱스순서관계정보모델을 설계 구현한다. 이 제안의 장점은 "인덱스 순서관계정보"로 클래스들 사이의 관련된 여러 관계정보를 UML 설계방법에 적응할 수 있는 구조로 변형할 수 있다. 두 번째 장점은 개발자가 인덱스 순서관계 정보에서 제공하는 정보를 가지고 관계정보를 쉽게 파악할 수 있으며, 디자인 패턴을 쉽게 추출함으로서 개발자는 설계정보에 쉽게 적용할 수 있다. 따라서 본 논문에서는 검색시간과 추출의 효율성을 입증하기 위해 시뮬레이션을 실시하여 향상된 기능을 입증하였다. 이 모델은 급변하는 소프트웨어 산업에 능동적으로 대체와 소프트웨어 개발에 시간을 단축함으로써 현존하는 다양한 디자인 패턴들을 최소한의 코드 수정을 통하여 재설계 함으로써 소프트웨어 개발 경제성을 높이는 데 있다.

1 서론

최근에는 웹 환경에 적합한 개방형과 컴포넌트들을 효율적으로 분류하고 추출하는 방법이 진행되고 있다[5.6]. 본 논문에서는 인터넷 환경에서 생성되는 산출물을 컴포넌트 객체 형태로 통합 관리하고, 패턴정보와 객체들을 효율적으로 관리하기 위한 패턴 정보저장소를 중심으로 인덱스순서관계정보모델을 설계 구현한다. 이 모델로 기존시스템을 재사용하고 급변하는 소프트웨어 산업에 능동적으로 대체와 소프트웨어 개발 시간을 단축하며 현존하는 다양한 데이터베이스들을 최소한의 코드 수정을 통하여 소프트웨어 개발 경제성을 높이는 데 있다. 본 논문에서는 기존 분류방법과 추출방법의 단점을 보완하고, 패턴을 인덱싱과정으로 패턴정보의 효율적인 추출 및 분류가 될 수 있도록 "인덱스 순서관계 정보 모델"을 제안한다. 또한 인덱스 순서관계정보모델을 제안함으로서 패턴을 효율적으로 추출, 검색하여 패싯항목을 코드화하고 데이터베이스화한 객체관리 모델을 제시함으로서 패턴정보 인덱싱과정으로 시스템 성능 향상과 분류의 효율성을 극대화시키는데 목적을 둔다.

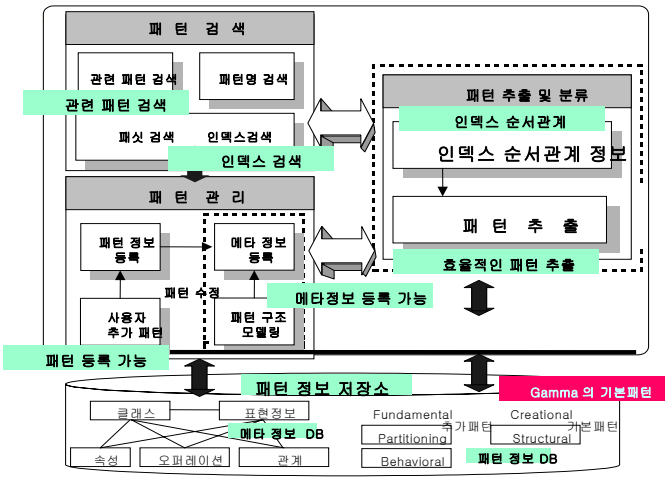
따라서 "인덱스 순서관계정보모델"을 설계로 효율적인 패턴정보를 추출 및 분류를 위해 패턴 인덱싱과정을 설계, 구현하였다. 이것은 패턴정보를 패턴 정보저장소에 저장함으로서 구조적인 추출과 패턴이 어느 상황에 적용될 수 있는가에 대한 개발자간의 경험적 상황을 고려한 기능적 인덱싱을 실행하여 디자인패턴에 인덱스를 자동부여 하여 디자인 패턴을 데이터베이스로 구축하였다. 이 과정은 정확하게 분류하는 방법으로 정확도를 향상시켜 소프트웨어 생산성을 향상시키는데 그 목적이 있다. 또한 웹환경에서 디자인패턴을 통합 관리하고, 효율적으로 추출, 검색하여 재설계까지 함으로서 인덱싱 패턴 정보 저장소설계와 디자인 패턴의 추출을 효율적으로 할 수 있다. 검증하기 위해서 시뮬레이션을 실시한 결과 향상된 기능을 입증하였다.

본 논문의 목적은 다음과 같다. 첫째 "인덱스 순서관계정보"로 클래스(class)들 사이의 관련된 여러 관계정보를 UML 설계방법에 적응할 수 있는 구조로 변형함으로서 개발자가 관계정보를 쉽게 파악하여 모델링 설계언어인 UML 설계방법에 쉽게 적응할 수

있게 한다. 둘째 패턴정보를 구성하고 있는 UML 관계 관련성에 대한 구조적 추출과 패턴정보 인덱싱과정으로 패턴이 어느 상황에 적용될 수 있는가에 대한 기능적 인덱싱과 패턴을 규격화시키는 순서기준 인덱싱로 패턴을 패싯분류 항목으로 코드화시킨다. 셋째 객체관리모델을 이용하여 패턴을 데이터베이스화하고 패턴 정보저장소를 구축함으로서 메타정보를 등록하여 패턴 구조를 모델링할 수 있으며 디자인패턴을 용이하게 추출, 검색, 분류함으로써, 궁극적으로 객체 지향 소프트웨어 개발의 생산성을 대폭 개선시키는 데 있다.

2. 인덱스 순서 관계 정보 모델 설계

분산객체 환경에서 프레임워크 객체 모델로 현존하는 다양한 플랫폼 및 응용 시스템을 그대로 살리고 웹기반 인덱스순서관계정보 모델을 설계한다. 이 모델은 패턴검색, 패턴 추출 및 분류, 패턴관리, 패턴정보 저장소로 나뉜다.



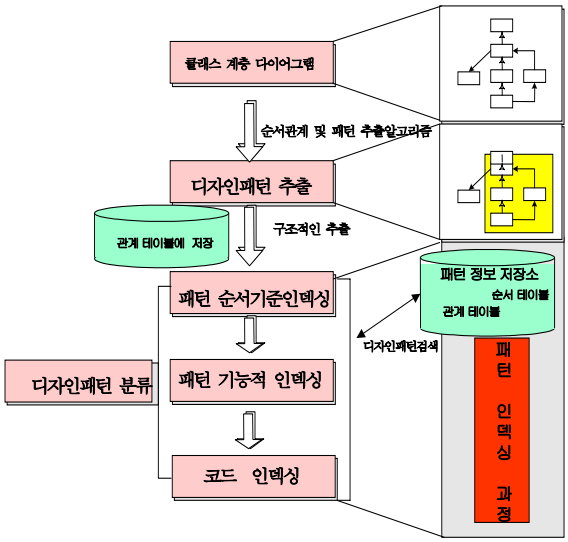
(그림1) 인덱스순서관계정보 모델 구성도

이 모델의 제안은 개발기간을 단축시켜 소프트웨어 산출물들을 효율적으로 유지보수 하고 저장 관리하여 생산성을 극대화시키는 것이 본 논문에서 제안한 목적이다. (그림1)은 인덱스순서관계정보 모델 전체 구성도를 나타낸 것이다.

2.1 인덱스순서관계정보 제안

패턴을 한마디로 정의한다면 “소프트웨어 엔지니어의 경험”이라고 말할 수 있다. 따라서 본 장에서는 “소프트웨어 엔지니어의 경험”과 “패턴의 코드화”로 패턴을 효율적으로 추출 및 분류하기 위한 인덱스 순서관계정보를 제안한다. 이것은 패턴 인덱싱 과정과 패턴을 추출하는데 구조적인 추출로 나눠 실시함으로서 개발자의 경험적인 면을 고려하여 패싯 분류방법으로 적용했고, 패턴 코드화 과정인 인덱싱함으로서 패턴을 효율적으로 분류하는 방법을 제시한다.

패턴을 보다 쉽게 이해하고 적용하기 위해서는 UML(Unified Modeling Language)의 관계(Realtion) 정보 사용이 필수적이다. 그러나 패턴의 체계적인 분류, 공유가 이루어지지 않아 적당한 패턴을 찾지 못하거나, 참조되지 못하는 경우가 많다. 기존 추출 방법 [9]은 패턴을 추출하는데 구조적인 추출만 하기 때문에 UML로 모델링하여 패턴 메타 정보를 나타낼 수 없어 효율적인 패턴 정보를 추출할 수 없었다. 이런 단점을 해결하기 위하여 패턴 메타 정보를 구축할 수 있는 패턴 정보저장소를 구축함으로서 패턴을 메타모델링하여 설계 재사용을 극대화할 수 있는 시스템을 구축하였다.



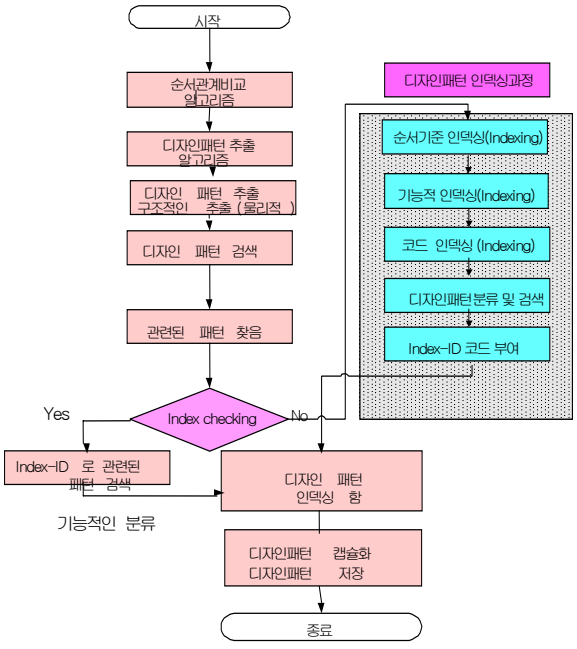
(그림2) 패턴 추출 및 분류과정

따라서 본 논문에서는 패턴 효율적으로 추출 및 분류하기 위해서 “인덱스 순서관계정보”를 제안하여 클래스들 사이의 관련된 여러 관계(Relation)정보를 UML 설계 방법에 쉽게 적응할 수 있는 구조로 변형함으로서 구조적인 추출과 패턴 인덱싱과정이 가능하게 하였으며 패턴의 역할정보와 패턴이 어느 상황에 적용될 수 있는가에 대한 개발자간의 경험적 상황을 고려한 기능적 정보와 순서기준 인덱싱, 기능적 인덱싱, 코드 인덱싱, 자동생성 분류방법을 설계함으로서 검색하는데 걸리는 응답시간을 단축하기 위함이다. 패턴을 코드화함으로서 효율적인 분류, 관리 및 검색이 용이해졌으며 재사용 극대화할 수 있었다.

3. 인덱스 순서 관계 정보 모델 구현

인덱스 순서관계 정보를 제안하고 구현하게된 목적은 다음과 같다. 첫째 인덱스 순서관계 정보로 클래스(class)들 사이의 관련된 여러 관계(Relation)정보를 UML 설계방법에 적응할 수 있는 구조로 변형함으로서 관계정보를 쉽게 파악하여 모델링 설계언어인 UML 설계방법에 쉽게 적응할 수 있게 한다. 둘째 인덱스 순서관계 정보는 원시 코드 중심의 재사용에서 탈피하고, 시스템 이해도를 높이기 위해 UML방법의 클래스다이어그램의 구성 요소 중 클래스 상호 간의 관련성을 파악한 후 구조적인 추출을 한다. 셋째 디자인패턴을 구성하고 있는 UML 관계(Relation) 관련성

에 대한 구조적 정보뿐만 아니라 패턴이 어느 상황에 적용될 수 있는가에 대한 경험적 정보를 추가하여 기존 단점을 보완했다. 넷째 패턴 인덱싱과정을 실시하여 복잡한 시스템의 복잡도를 감소시킬 수 있으며 패턴을 코드화할 때 패턴이 어떠한 역할을 수행하는가에 따른 기능적 분류와 패턴이 사용될 수 있는 여러 경험적 상황을 패턴 정보로 관리하여 가장 적합한 코드를 부여함으로서 증가하고 있는 수많은 패턴들을 효율적으로 분류하기 위한 것이다. 다섯째 패턴을 인덱싱함으로서 패턴의 효율적인 유지보수와 패턴 재사용을 극대화시키고, 검색분류 추출하는데 향상된 환경을 제공하기 위한 것이다. 논문의 접근은 클래스다이어그램으로부터 순환되는 패턴의 추출에서 시작되며, 각 패턴이 사용될 수 있는 여러 경험적 상황을 패턴 정보로 관리하고 이 정보를 이용하여 패싯 항목으로 설정하고 관련된 패턴 검색, 분류, 인덱싱으로 이어진다. 패턴은 클래스들의 관계에 대한 조직이고 시스템을 여러 가지 패턴으로 분류하여 활용할 수 있다. 패턴을 분류하기 위해 UML 클래스다이어그램으로 패턴을 추출하고, 추출된 패턴을 기준으로 검색하여, 분류와 인덱싱하는 과정을 다루는 것이 "인덱스 순서관계정보"이다.



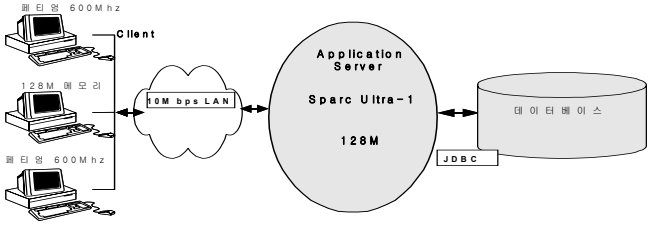
(그림3) 패턴 인덱싱과정 알고리즘 구성도

3.1 패턴인덱싱 과정

패턴 인덱싱 과정은 세 가지로 구분하는데 ,패턴이 어떠한 역할을 수행하는가에 따른 역할정보와 패턴이 어느 상황에 적용될 수 있는가에 대한 개발자간의 경험적 상황을 고려해서 분류하는 기능인 기능적 인덱싱, 패턴을 규격화시키는 순서기준 인덱싱이다. 이것은 패턴간의 관련성을 쉽게 찾을 수 있도록 패턴을 규격화하여 상속성(inheritance)을 가짐으로 시스템의 복잡성을 줄일 수 있었다. 패턴검색은 패턴정보저장소에 저장되어 있는 패턴을 효율적으로 검색하여 최상의 관련된 패턴을 찾아내 재사용 할 수 있도록 하고 관련 패턴검색 패싯 검색, 인덱스 검색등 여러 항목을 고려해서 검색함으로서 패턴을 효과적으로 찾아 구조적인 추출의 단점을 보완할 수 있다. 패턴관리는 패턴을 관리하고, 패턴정보를 등록하는 역할을 한다. 패턴 정보저장소는 패턴을 저장시키는 곳으로 추가패턴을 등록시킬 수 있으며 기본패턴은 따로 저장되어 있다. 이런 정보를 데이터베이스화한 것이 패턴 정보 저장소이고 데이터베이스와 JDBC과 연동된다. 또한 메타 정보 데이터베이스를 두어 각 클래스와 속성, 오퍼레이션, 관계 표현정보를 데이터베이스화한다. 디자인패턴을 UML로 모델링하고 패턴 구조를 메타 데이터로 저장하여 패턴 메타 데이터베이스로 구축하였다. (그림3)은 패턴의 인덱싱과정 알고리즘 전체 구성도를 나타내고 있다.

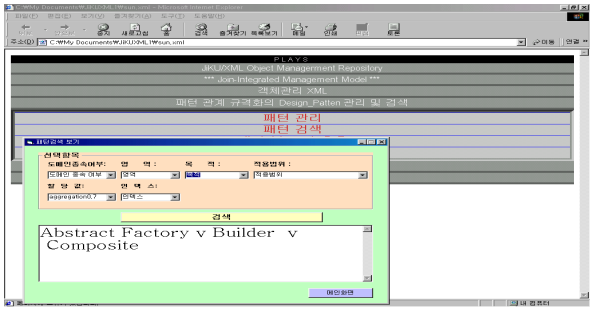
3.2 구현과 시뮬레이션 환경

디자인패턴에 코드화가 안된 패턴과 본 논문에서 제안한 인덱스 순서관계정보로 코드화가 된 패턴의 성능 평가를 위하여 시뮬레이션을 실시하는데 인덱싱된 패턴들을 검색하는데 걸리는 시간을 측정하고 인덱스로 인한 검색결과를 비교, 분석하여 인덱스 순서관계정보의 타당성과 장점을 검증한다. (그림4)는 디자인패턴 추출 및 검색과정에서 정보를 요청하는 클라이언트와 정보를 요청 받는 서버 그리고 패턴정보 저장소가 있는 데이터베이스계층으로 구성되는데 그 구성도를 나타낸 것이다. 성능평가와 검증을 위한 시뮬레이션 환경은 다음과 같다. 클라이언트 측과 성능평가에 사용한 클라이언트 시스템은 펜티엄 600MHz CPU에 128MB 메모리를 가지고 있고 운영체제는 Window 98이다. 구현언어는 자바와 Visual C++, XML를 사용하였고 구현화면은 (그림5)와 같다.

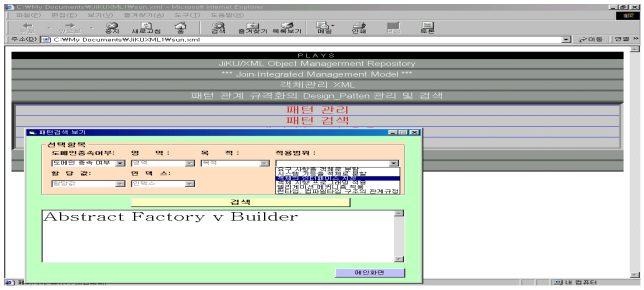


(그림4) 클라이언트와 서버 및 데이터베이스계층 구성도

패턴 정보 저장소인 데이터베이스 측과 데이터베이스는 데이터베이스가 설치된 계층으로 시스템은 펜티엄 600MHz CPU에 128MB 메모리를 가지고 있고 운영체제는 리눅스 6.0이다.

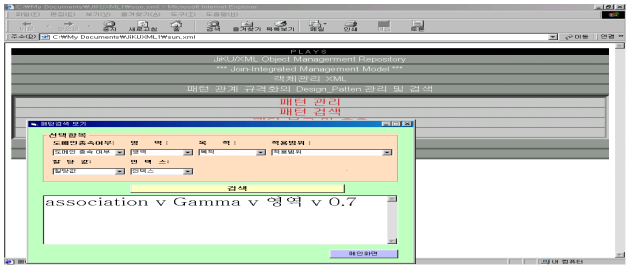


(그림5) 디자인패턴 검색 구현화면



(그림6) 디자인패턴 검색 질의어 구현화면

사용한 데이터베이스는 Mysql 2.0와 Access Database, Oracle Database를 사용했으며 애플리케이션과의 연결은 JDBC[4]를 사용하였다. 여기서 패턴 수와 질의어 수를 나타나는데 질의어 형태와 구현 화면은 (그림5),(그림6),(그림7)과 같다. 그리고 이것은 디자인패턴 검색 질의어 형태의 구현 화면을 나타낸 것이다.

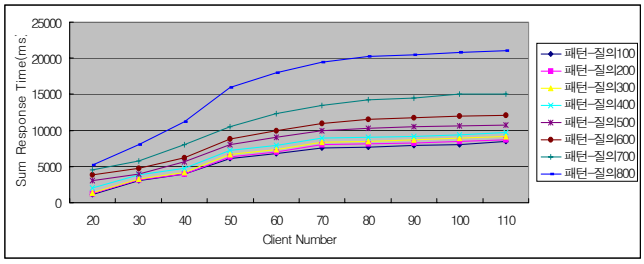


(그림7) 디자인패턴 검색 결과 구현화면

3.3 시뮬레이션 결과

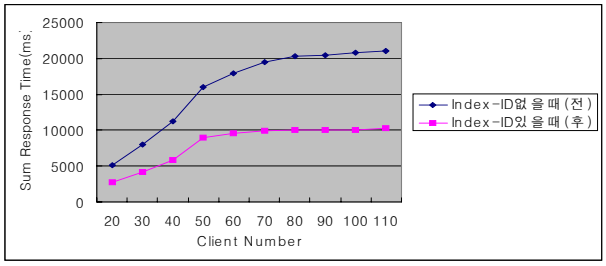
인덱싱된 디자인패턴 경우 클라이언트의 응답 시간와 측정함으로서 본 논문에서 제안한 모델을 구현한다.

이 모델 구현과 시뮬레이션은 사용한 시스템의 성능에 따라 동시에 처리되는 클라이언트의 수를 110개까지로 제한하였다. 클라이언트의 수를 20개부터 10개씩 증가시키면서 110개까지 증가 시켰다. 그리고, 클라이언트의 요구를 처리하는 서버 수를 1개로 하고 디자인패턴 수와 질의어를 변경하면서 클라이언트 당 검색 응답 시간을 측정하여 누적하였다. 측정한 누적 응답 시간의 결과는 (그림8)과 같으며 첫 번째는 100번에서부터 800까지의 그래프를 함께 나타낸 것이고 그 다음부터는 100개씩 증가시킨 그래프를 나타낸 것이다. (그림8)은 측정결과를 그래프로 나타낸 것이다. 측정 결과 클라이언트의 응답 시간은 클라이언트의 수에 비례하며 디자인패턴 수와 질의어 수에도 비례함을 알 수 있다.



(그림8) 검색 응답시간의 결과

마지막 성능평가로 패턴이 순수한 인덱스가 있는 것과 없는 디자인패턴을 서로 성능평가를 하였다. (그림9)는 클라이언트의 수를 20에서 10씩 증가하여 110개일 때까지 측정하였다. 인덱스가 있는 디자인패턴 수와 인덱스아이디가 없는 디자인패턴 수를 비교하여 그 수를 100개에서 200개로 증가할 때는 55% 검색 시간이 빨라졌으나 또한 200개에서 300개로 증가할 때는 55%, 300개에서 400개로 증가할 때는 56%의 검색시간 향상이 있음을 알 수 있었다.



(그림9) 인덱스로 비교한 누적 응답시간의 결과

따라서 700개에서 800개로 증가할 때는 56%의 성능 향상이 있음을 알 수 있었다. 따라서 클라이언트의 수가 110개일 때 인덱스가 있는 디자인패턴은 없는 것 보다 약 56%로 응답 속도가 향상되었음을 알 수 있다.

4. 결론 및 향후 연구과제

본 논문은 웹기반 인덱스순서관계정보 모델을 제안하고 설계하여 구현한 것을 입증하였다. 인덱스순서관계정보 모델을 제시하고 패턴 메타 정보를 구축할 수 있는 패턴 정보저장소를 구축하였다. 패턴을 메타 모델링한 데이터베이스에 저장하여 설계하는데 재사용이 가능한 시스템을 제시하였다. 이 과정의 장점은 패싯항목과 패턴간의 관련성을 쉽게 파악 할 수 있도록 패턴을 코드화함으로써 효율적인 분류, 추출, 관리를 할 수 있어 설계 경험을 공유하고, 설계정보 검색이 용이해졌으며, 정확하게 분류하여 정확도를 높여 설계정보를 재사용 할 수 있도록 하였다. 또한 이 모델을 이용하여 시스템을 구현함으로서 정확도와 응답시간을 측정하여 향상된 기능과 효율성을 입증하였다. 향후 연구과제는 제안의 모델을 다른 시스템과 비교 분석하여 검증을 받는 것이다.

참 고 문 헌

[1] Jon Meyer & Troy Downing "JAVA Virtual Machine", O'REILLY, 1999

[2] Orfali, Harkey and Edwards, "The Essential Distributed Objects Survival Guide", Wiley Press, 1996

[3] Jim Waldo, Geoff Wyant, Ann Wolrath and Sam Kenedall, "A Note on Distributed Computing", Sun Microsystems, 1994

[4] "JDBC specification", http://splash.javasoft.com/jdbc

[5] Emerson, Darnovsky, and Bowman, "The Practical SQL Handbook", Addison-Wesley, 1989

[6] Eric Gamma,Richard Helm, Ralph Johnson, John Vlissides,"Design Patterns : Element of Reusable Object-Oriented Software", Addison-Wesley, 1995

[7] Grady Booch, Ivar Jacobson, and James Rumbaugh."Unified Modeling Language". 2000. Version 1.3.

[8] Jagdish Bansiya, "Automating Design-Pattern Identification", Dr Dobb's Journal June, 1999