

# 객체기반 시소러스 구축과 부울리언을 이용한 설계패턴 검색

손은영\*, 한정수\*, 송영재\*\*

\*경희대학교 전자계산공학과

\*천안대학교 정보통신학부

\*\*경희대학교 전자계산공학과

e-mail:eyshon@dreamx.net, jshan@cheonan.ac.kr

yisong@khu.ac.kr

## Object-based Thesaurus Construction and Design Pattern Retrieval using Boolean

Eun-Young Shon\*, Jung-Soo Han\*, Young-Jae Song\*\*

\*Dept of Computer Engineering, Kung-hee University

\*\*Division of Information & Communication, Cheon-an Univ.

\*\*Dept of Computer Engineering, Kung-hee University

### 요 약

본 논문은 설계 패턴의 효율적인 관리와 재사용을 위하여 패킷 방식을 이용한 패턴 분류와 패킷 항목과 패턴간의 관련성을 이용한 패턴 검색 시소러스를 구축하였다. 저장소에 패턴들의 관계성을 정의하고 새로운 패턴을 삽입 시 임의의 패턴과의 관계성을 지정해주면 시스템은 이 패턴에 연관된 패턴들을 보여주고 사용자는 패턴과의 관계성을 지정해 준다. 패턴 검색은 입력한 질의를 추론하여 검색할 수 있도록 부울 질의 형식을 사용하였다. 그리고 기존의 시소러스의 중복성과 복잡한 비교시간을 최소화하기 위하여 "Gamma 확장 + 패킷" 분류방법을 이용하여 패턴들을 분류하였고, 본 연구의 검색 결과가 보다 더 정확한 패턴이 검색될 수 있도록 하였다.

### 1. 서론

설계 패턴은 Gamma에 의해서 패턴의 역할에 따라 생성 패턴, 구조 패턴 그리고 행위 패턴으로 분류된다. 그러나 수 백 가지의 패턴을 효율적으로 재사용하기 위해서는 3가지 분류와 패턴 명만으로 검색해야하는 Gamma의 분류 방법으로는 사용자가 원하는 패턴을 찾는 데 많은 어려움이 따른다. 따라서 정보 검색을 위한 저장소 구축을 효율적으로 하기 위해서 시소러스 개념을 도입하였다. 설계 패턴 검색을 위하여 패턴 사이의 관계성을 의미적으로 표현하고, 새로운 패턴을 추가, 삭제할 때 기존의 관계성을 이용하여 패턴을 검색할 수 있는 객체기반 시소러스로 구축에 그 목적을 두었다.[1][2]

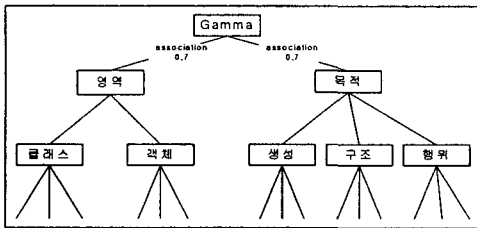
검색 방법은 Gamma의 패턴을 중심으로 패킷 기

능을 이용한 메뉴 방식의 질의어를 이용하였다. 객체는 클래스의 의미를 나타내는 개념 객체와 인스턴스 객체로 구분하고, 각 객체 사이의 관계는 일반화(generalization), 연관화(association), 집단화(agggregation), 클래스화(classification) 등을 중심으로 정의하였다. 패턴 검색은 질의가 갖는 의미를 제한된 범위 내에서 표현할 수 있고 확장된 부울 형식을 사용하였다. 따라서 본 연구는 패턴 검색을 위하여 질의와 객체사이의 관계를 개별적으로 해야하는 단점을 극복하고, 자동적으로 관계를 연결시켜 주는 객체기반 시소러스를 구축하였다.

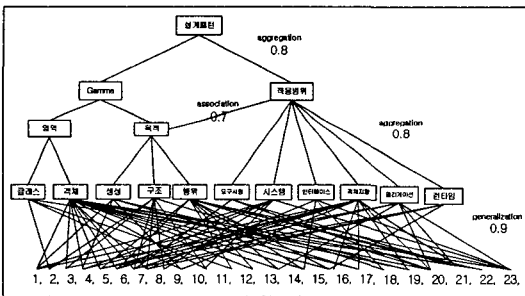
### 2. 설계패턴 시소러스 표현 레벨

#### 2.1 개념 표현 레벨

개념 표현 레벨에서, 하나의 최상위 객체는 다른 최상위 객체들과 집단화, 연관화 관계를 가질 수 있다. 최상위 객체들 사이의 관계성들을 구축하기 위해서 하나의 최상위 객체와 집단화 연관화 관계에 있는 다른 최상위 객체들을 입력하면, 시스템은 두 최상위 객체 사이의 관련 정도에 미리 정의한 값을 할당하게 된다[5]. (그림 1)에서처럼 최상위 객체 "Gamma"를 생성하였을 경우, "Gamma"와 연관화 관계 있는 최상위 객체 "영역"과 "목적"을 입력하면, 시스템은 두 개의 관계(association, Gamma, 영역, w), (association, Gamma, 목적, w)에 대한 관련정도 w를 미리 정의한 0.7의 값으로 할당하게 된다.



(그림 1) 최상위 개념객체 "Gamma"의 생성



(그림 2) 설계 패턴 계층도

(그림 2)는 Gamma 패턴들의 전체 계층 구조를 보여주고 있다. 여기서 "1, 2, 3, ..."은 각 계층도의 최하위 객체이며 인스턴스 객체를 의미한다. 이들은 Gamma의 영역과 목적 그리고 하위 개념 객체인 클래스, 객체, 생성, 구조, 행위로 분류하여 각각의 개념 객체에 속한다. 또한 경험적 방법 즉, 적용범위에 의하여 분류한 6가지 개념객체와 인스턴스 객체와의 관계를 나타내고 있다. <표 1>은 인스턴스 객체의 실제 패턴명들, <표 2>는 각 개념객체에 속하는 인스턴스 객체와의 관계를 보여주고 있다.

2.2 인스턴스 표현 레벨

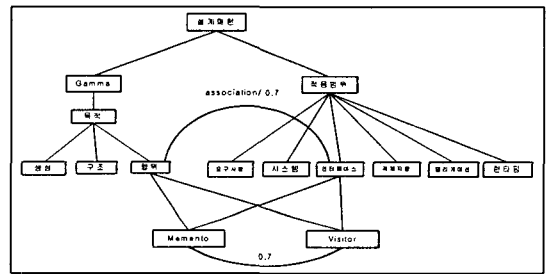
객체 기반 시소러스에 표현되는 객체들 중에서 많은 부분이 인스턴스 (instance)로 분류될 수 있

<표 1> Gamma의 설계 패턴 종류

index	Design Pattern	index	Design Pattern	index	Design Pattern
1	Abstract factory	9	Facade	17	Prototype
2	Adapter	10	Factory Method	18	Proxy
3	Bridge	11	Fly weight	19	Singleton
4	Builder	12	Interpreter	20	State
5	Chain Responsibility	13	Iterator	21	Strategy
6	Command	14	Mediator	22	Template Method
7	Composite	15	Memento	23	Visitor
8	Decorator	16	Observer		

<표 2> 개념 객체와 인스턴스 객체 관계

클래스(class)	2, 10, 12, 22
객체(object)	1,3,4,5,6,7,8,9,11,13,14,15,16,17,19,20,21,23
생성 패턴	1, 4, 10, 17, 19
구조 패턴	2, 3, 7, 8, 9, 11, 18
행위 패턴	5, 6, 12, 13, 14, 15, 16, 20, 21, 22, 23
요구사항을 객체로 분할	7, 20, 21
시스템 기능을 객체로 분할	1, 4, 6, 9, 11, 23
객체의 인터페이스 지정	8, 15, 18, 23
객체지향 프로그래밍 적용	1, 4, 5, 6, 7, 10, 16, 17, 19, 20, 21
엔리게이션 메카니즘 적용	3, 5, 14, 20, 21, 23
런타임, 컴파일타임 구조의 분계 규정	5, 7, 8, 16



(그림 3) 추가된 관계의 상속

며, 인스턴스는 질의 처리 시에 매우 높은 사용 빈도를 보이고 있다. 그러나, 시소러스에 삽입되는 한 인스턴스에 대해, 모든 집단화·연관화 객체들을 알아내기는 매우 어렵다. 객체 기반 시소러스는 집단화·연관화 관계가 상속되는 성질을 이용함으로써 이를 해결할 수 있다. 즉, 인스턴스와 집단화·연관화 관계를 맺을 객체들은 자신이 속하는 상위 개념 객체들과 다른 상위 개념 객체들 내의 인스턴스들 중에서 선택되기 때문에 인스턴스들은 시스템에 의하여 자동으로 제시될 수 있다. 따라서 필요한 객체만을 선택함으로써 새로운 관계를 형성할 수 있게 된다[3]. (그림 3)에서와 같이 상속의 성질에 의하여 상위 객체들의 관계의 값들을 그대로 상속받게 되는데, "Visitor"와"Memento"사이의 관련정도는 그 상위객체의 관계 (association, 행위, 인터페이스, 0.7)에서 관련정도 "association/0.7"을 상속받아 (association, Visitor, Memento, 0.7)로 정의할 수 있다.

### 3. 추론에 적합한 질의의 형태

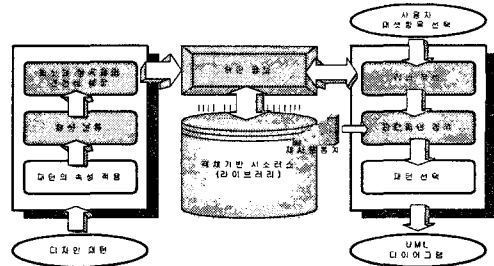
#### 3.1 부울리언 연산

본 논문에서는 자연어 질의가 갖는 의미를 제한된 범위 내에서 표현할 수 있는 부울 질의 형식을 사용한다. 질의 형식에서 사용되는 연산자는 존재 연산자와 개념 표현 연산자로 세분될 수 있다. 존재 연산자는 질의에 표현된 개념들이 의미적으로 완전히 독립적일 때 사용되며, 개념 표현 연산자는 질의에 표현된 여러 개념들이 종속적으로 수식 관계를 가지고 한 개념을 표현할 때 사용된다. 예를 들어, “객체 AND 생성 AND 객체지향”이라는 개념이 완전히 독립적으로 세 개념이 동시에 나오는 패턴을 사용자가 원하는 경우이고, 두 번째 해석은 “객체”와 “생성, 그리고” 객체지향”이라는 세 개념이 종속적 관계를 의미하여 <표 2>에서처럼 세 개념을 동시에 포함하는 “Abstract Factory”, “Builder”, “Prototype”, “Singleton”이라는 개념을 의미하고 있는 경우이다.

#### 3.2 객체 기반 추론

객체기반 추론 관계에는 일반화 관계, 구체화 관계, 클래스화 관계로 분류할 수 있다. 사용자가 구체적인 개념들은 알고 있지만 일반적인 개념을 모르고 경우, 질의는 구체적인 개념과 일반화 연산자 “∨”를 통하여 구성될 수 있고, 추론기는 질의에 표현된 구체적인 개념들을 모두 포함하는 상위 개념 객체를 추론하게 된다. 예를 들어, 자연어 질의 “Gamma 패턴 중에 Abstract factory, Builder, Chain Responsibility를 포함하는 패턴”에 대하여 분석기는 “Abstract factory ∨ Builder ∨ Chain Responsibility”로 변형할 수 있고, 추론기는 “Abstract factory”, “Builder”와 “Chain Responsibility”의 인스턴스들이며 연산자 “∨”이 일반화 연산자임을 인식하여, 여기에 해당하는 인스턴스들을 동시에 소유하고 있는 개념 객체들 중에서 가장 구체적인 개념 객체 “객체”와 “객체지향”을 추론한다. 구체화되는 개념은 IS-A 계층에 존재하는 일반적인 의미를 지니는 개념들은 알고 있지만, 개념들의 의미를 동시에 만족하는 구체적인 개념을 모르고 있을 경우, 사용자 질의는 일반적인 개념들과 구체화 연산자 “∧”을 통하여 표현될 수 있다. 예를 들어, 자연어 질의 “생성이면서 시스템”에 대하여 분석기는 “생성 ∧ 시스템”과 같이 변형될 수 있고, 추론기는 “∧”이 구체화 연산자이며 “생성”과 “시스템의 의미를 동시에 상속받는 하위 객체

“Abstract factory, Builder”를 추론한다. 또한 자연어 질의 “도메인 독립이며 인터페이스와 런타임에 모두 사용되는 설계패턴”에 대하여, 분석기는 “설계패턴[독립, 인터페이스, 런타임]”으로 변형할 수 있고, 추론기는 “설계패턴”이 주개념이며, “독립, 인터페이스, 런타임”이 주변 정보임을 인식한다. 따라서, 주개념인 “설계패턴”의 하위 객체 중에서 “도메인, 독립, 인터페이스”와 “런타임”을 집단화 또는 연산화 관계로 가지는 객체 “Decorator”를 추론한다.



(그림 4) 설계 패턴 검색 과정

### 4. 패턴 검색

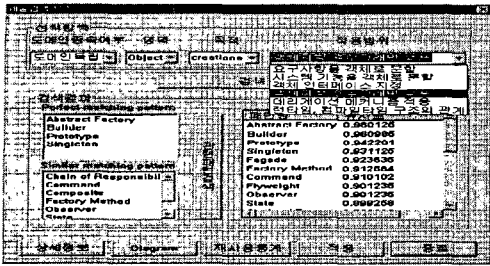
#### 4.1 설계 패턴 검색 과정

패턴 검색 과정은 (그림 4)에서처럼 각 패턴에 대한 정보를 패킷 분류한 후 객체 사이의 관계를 지정해 주면 시소리스에 의하여 관계 깊은 객체들이 제공되고, 이 객체들 중 가장 관련있는 객체들을 연결시켜 저장함으로써 저장소가 구축된다. 검색을 위해서는 적용 영역, 역할 등의 패킷 항목으로 적합한 패턴을 부울 연산자를 이용하여 검색되도록 해주고, 관련 패턴까지 검색해줌으로서 패턴 선택의 폭을 넓힐 수 있도록 하였다. 검색된 다수의 관련 패턴들에 대해서는 패턴 적용의 도메인 종속 여부, 영역, 목적, 적용 범위 등의 패킷 항목에 대한 속성과 구성 클래스 수, 패턴 파일명 등의 상세 정보가 제공되며 각 패턴의 재사용 횟수를 보여줌으로서 전반적인 패턴의 재사용성을 평가할 수 있어 효율적인 재사용이 가능하도록 하였다[4].

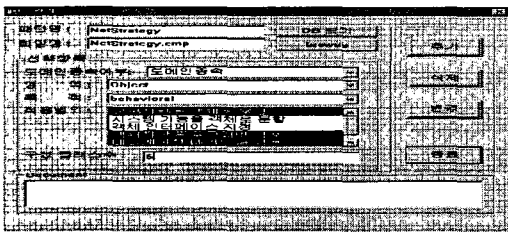
#### 4.2 패턴 검색

패턴 검색은 패킷에 대한 각 항목을 선택하여 부울 연산자에 의하여 검색된다. 검색 결과에서 「Perfect matching pattern」은 선택한 4개의 항목을 모두 만족하는 패턴 리스트를 나타내며 「Similar matching pattern」은 4개의 항목 중 3개의 항목만을 만족하는 패턴 리스트를 나타낸다. 이

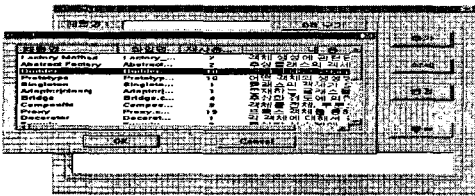
는 질의에 대한 “^” 연산자를 이용하여 검색한 결과이다. 「관련패턴검색」 기능은 선택한 항목에 따라 계산된 인스턴스 객체가 제공된다.



(그림 5) 패턴 검색



(그림 6) 패턴 추가



(그림 7) 패턴 데이터베이스

(그림 5)는 “도메인 독립”, “object”, “creational”, “객체지향프로그래밍에 적용”의 패킷 항목을 선택하여 검색한 결과로서 Abstract Factory 패턴을 중심으로 11개의 패턴이 검색되었는데 선택한 질의와 “^” 연산자와의 결합에 의한 검색은 상위 4개인 Singleton 패턴까지이다. 본 논문에서는 검색할 패턴의 확장성을 고려하여 라이브러리에 있는 패턴들에 추가로 삽입하여 검색이 가능하도록 하였으며, 추가된 패턴과 항목들과의 연결관계가 자동적으로 패턴 정보로 저장되도록 하였다. (그림 6)은 패턴의 추가 인터페이스로서 패킷 항목 중 적용 범위에 대해서는 패턴의 실제 적용 상황에 따라 여러 개 항목의 선택이 가능해야 하므로 각 항목들이 다중 선택이 가능하도록 하였다. 추가된 패턴은 라이브러리에 저장되고 선택된 항목에 따라 항목과 패턴과의 관련성이 설정되며, 패턴 속성정보로 관리되기 위해 (그림 7)과 같이 패턴 데이터베이스에 저장된다. 패턴 삭제

와 변경 시에는 현재 라이브러리에 있는 패턴의 정보를 알고 있어야 처리가 가능하므로 라이브러리의 패턴 정보를 데이터베이스의 테이블 형식으로 제공하였다.

### 5. 결론

본 논문은 설계 패턴의 효율적인 관리와 재사용을 위하여 패킷 방식을 이용한 패턴 분류와 패킷 항목과 패턴간의 관련성을 이용한 패턴 검색 시소리스를 구축하였다. 패킷은 도메인 종속 여부와 패턴이 속한 영역, 적용 목적, 적용 범위 등을 포함하도록 정의하였다. 패턴을 검색하기 위하여 패턴의 관계성을 의미적으로 표현하고, 새로운 패턴의 추가, 삭제 시 기존의 관계성을 이용하여 객체 관계가 형성되도록 하였다. 객체 기반 시소리스에서 객체는 개념객체와 인스턴스 객체로 구분하였고, 각 객체 사이의 관계는 일반화, 연관화, 집단화, 클래스화 등을 중심으로 정의하였다. 패턴 검색은 질의를 추론을 위해 부울 질의 형식을 사용하였고 중복성과 복잡한 비교 시간을 최소화하기 위하여 “Gamma 확장 + 패킷” 분류방법을 이용하여 패턴들을 분류 검색하였다.

앞으로의 연구방향은 많은 설계 패턴이 확장되었을 경우 자연어 처리를 위한 추론 기법에 대한 연구가 필요하며, 패턴 삽입시 유사 또는 중복된 패턴에 대한 정보 제공이 필요하다.

### 참 고 문 헌

- [1] E.Gamma, R.Helm, R.Johnson, and J.Vlissides, “Design Pattern : Elements of Reusable Object -Oriented Software,” Addison-Wesley, 1995.
- [2] William B. Frakes and Ricardo Baeza-Yates, “Information Retrieval,” Prentice Hall, pp. 419-442, 1992.
- [3] R. Rada, and B. K. Martin, “Augmenting Thesauri for Information System,” ACM Transaction on office Information System, Vol. 5, NO. 4, pp. 378-392, 1987.
- [4] 김귀정, 송영재, “패턴정보를 이용한 설계패턴 검색 시스템 구축”, 정보처리학회논문지 제8-D권 제 1호, 2001, 2.
- [5] 최재훈, 한종진, 박종진, 양재동, “구조적인 시소리스 구축을 지원하는 객체 기반 정보 검색 모델”, 정보과학회논문지, 제24권 제11호, 1997.