

경험적 분류 클래스를 도입한 객체 지향 데이터베이스 모델링

(The Modeling of Object oriented Database introducing Heurilistic Classfication Class)

김 준 모(Jun-Mo Kim)¹⁾

요 약

기존의 객체지향 데이터베이스에 경험적 분류 모델에 기반을 둔 새로운 클래스를 도입한 확장된 객체 지향 데이터베이스의 모델을 설계한다.

이를 구현하기 위해 기존의 객체 데이터 베이스에 경험적 분류 클래스를 도입하였으며, 이 클래스들을 연산하기 위한 경험적 분류 연산 클래스를 설계하였다.

그리고 확장된 객체 지향의 데이터 모델 상에서 데이터베이스에 저장된 데이터의 경험적 분류 모델에 기반을 둔 검색이 가능한 질의어를 설계하였다.

ABSTRACT

This paper has been designed extend object-oriented database model that introduced new class basing the Heurilistic Classfication model. In order to implement this model, we have introduced heurilistic class to traditional object-oriented database. And we designed query for search data that basis on the heurilistic classfication model using stored data in extened object-oriented data model.

1. 서론

경험적 분류 모델은 전문가 시스템 개발 및 객체 지향 데이터 베이스 시스템 개발에서 분류 단계에서 적합하고 융통성이 있기 때문에 분류 과정을 단축하는 장점이 있다.[1][2][3] 객체 지향 개념은 응용 분야와 사용 용도에 따라 이용되는 특성이 다양하지만, 데이터 추상화 특성 계승, 객체

참조 등을 그 주요한 특성으로 한다. 데이터 추상화는 데이터 형식을 정의 할 때, 허용되는 연산을 정의하며 그 실제 수행을 염두에 둘 필요성을 제거해 준다. 특성 계층은 객체 지향 개념에서 가장 주요한 특성 중에 하나로서 두 클래스가 계승 계층을 이루고 있을 때 상위 클래스의 특성 즉 인스턴스 변수와 메소드(method)등이 하위 클래스(subclass)에 계승되는 개념이다. 따라서 특성 계

1) 정회원 : 전주기전여자대학 인문사회학부 조교수

승에 의하여 하위 클래스들은 상위 클래스의 정보를 계승하게 된다. 객체 참조는 각 객체마다 고유한 객체 식별자(object id;oid)를 갖고 있어서 이를 이용하여 각 객체들이 서로 참조하게 된다. 객체 지향 시스템에서는 이와 같은 특성을 객체, 클래스, 인스턴스, 메소드, 메시지 등으로 구성되어 지원한다.[1][2][7]

2. 경험적 분류 모델

간단한 문제의 경우 분류가 간단할 수 있지만 복잡한 문제의 경우 입력된 데이터만으로 정확한 분류를 하기 어렵다. 따라서 이러한 문제점을 해결하기 위한 것이 클래스시가 제시한 경험적 분류 방법이다.[3]

2.1 경험적 분류모델의 활용

이 경험적 분류 모델은 전문가 시스템 개발 및 객체 지향 데이터베이스 시스템 개발에서 분류 단계에서 적합하고 융통성이 있기 때문에 분류 과정을 단축하는 장점이 있다. 전문가의 문제 해결과정은 자료추상화 과정, 경험 연상과정, 정제 과정 등 3단계가 있다.[4][5]

자료추상화 과정은 사용자가 입력한 구체적인 자료들을 분류하여 추상화시키는 과정이다. 경험 연상과정은 추상화된 자료로부터 추상화된 결과를 연상하는 단계이다. 이단계는 전문가의 지식이 가장 많이 포함되어 있다. 정제과정은 경험 연상에서 얻어진 결과를 검색하고 관련 자료를 수집하여 해를 줄여 나가서 결론을 구하는 단계이다. 이런 과정을 이용해 효율적인 데이터의 검색이 가능하다. 따라서 이과정을 지원해 주는 클래스를 도입, 설계하여 기존의 객체 지향 데이터 모델을 확장하고자 한다.

2.2 기존의 객체 지향 데이터베이스 문제점

객체 지향 데이터 모델에서는 현실 세계의 모든 엔티티(entity)를 객체로 모델링 한다. 객체는 객체 지향 데이터 모델은 데이터베이스에서 객체 지향 개념을 기본 단위로, 개별적인 메모리를 갖고 있어서 그 객체의 상태를 기억하게 된다. 이 개별적인 메모리를 인스턴스 변수라 하며 데이터의 내용은 일련의 인스턴스 변수에 저장된다. 객체 지향 데이터 모델에서의 연산은 메시지와 메소드에 의하여 처리하게 된다. 객체, 즉 클래스나 인스턴스를 연산하고자 할 때, 해당 클래스나 인스턴스에 원하는 연산을 메시지로 보내면 허용되는 메소드 중에서 메시지에 해당되는 메소드를 실행시키게 된다.

객체 지향 데이터 모델에서는 상위 클래스의 특성을 계승하며 데이터 형식에 제한이 없으므로 소프트웨어의 재사용성이 가능하여 데이터베이스 관리시스템 프로그램의 코드를 절약할 수 있다. 그리고 도형이나 음성과 같은 데이터를 관리할 수 있다는 장점을 가지고 있어 모델링 능력이 우수하다. 그러나 기존의 객체 지향 데이터베이스 시스템에서는 객체간의 경험적 분류된 데이터의 처리가 명확하지 않아서 대량의 객체 관리에 어려움이 있다는 문제점을 안고 있다.

3. 경험적 분류방식을 이용한 데이터 모델의 확장

기존의 객체지향 데이터 모델들은 엔티티 클래스들이 Refer-To관계로 다른 클래스나 그 클래스의 인스턴스를 참조하므로 관련된 클래스간의 효율적인 검색 및 인스턴스값의 변화 등이 어렵다는 단점으로 객체 지향의 특성을 잘 살리지 못하고 있다.

또한 기존의 데이터베이스 시스템 경험적 분류 방식으로 처리되지 않아서 어떤 교수가 여러 과목

을 지도할 때, 특정 과목을 지도관계로 하여 학생의 명단과 교수의 명단의 검색등이 불가능하였다. 그러므로 경험적 분류방식을 도입하여 데이터 검색을 명백하게 지원해 줄 수 있도록 객체 지향 데이터 모델을 확장하여야 한다.[1]

따라서 엔티티 클래스간의 경험적 분류방식을 적용하면 엔티티 클래스간의 경험적 분류를 이용하여 인스턴스를 검색할 수 있고 경험적 분류 방식을 이용하여 데이터의 분류가 가능하여 엔티티 클래스를 효율적으로 액세스 할 수 있다. 또한 수많은 클래스들이 존재하여 서로 복잡한 관계를 갖고 있는 기존의 객체 데이터베이스에서 경험적 분류 클래스를 도입한다.

4. 객체 지향 데이터 모델에서의 경험적 분류

기존의 객체 지향 데이터 모델은 객체 지향 개념이 제고해 주는 장점에도 불구하고 데이터베이스에서 요구되는 특성을 제대로 지원해 주지 못하고 있는데, 특히 엔티티 간의 경험적 분류를 별도로 명시해 주지 않으므로써, 현실 세계에 보다 근접한 데이터 베이스를 모델링하는데 문제점을 내포하고 있다. [1][2]

따라서 객체 지향 데이터 모델 상에 실제적인 엔티티 클래스간의 경험적 분류를 인스턴스로 하는 경험적 분류 클래스를 사용자 정의 클래스로 정의하며 이들 클래스를 삽입, 삭제, 검색하는 기능을 가진 시스템 정의 클래스인 경험적 분류 연산 클래스를 도입하여 경험적 분류를 효율적으로 지원해 줄 수 있게 한다. 경험적 분류 클래스는 사용자가 데이터베이스를 구성할 때 즉 엔티티 클래스를 정의할 때 클래스간의 경험적 분류를 추출해야 한다.[2][3]

4.1 경험적 분류 연산 클래스의 정의

경험적 분류 연산 클래스는 연산 클래스를 상위 클래스로 갖고 있어서 기본적인 인스턴스의 삭제, 삽입, 검색 등의 행동 양식을 계승 받고 실제 경험적 분류를 인스턴스로 하는 경험적 분류 클래스를 하위 클래스로 갖는다.

4.2 경험적 분류 클래스의 구성

경험적 분류 클래스는 시스템에서 제공되는 경험적 분류 연산 클래스를 상위 클래스로 갖고 있어서 그 특성을 계승받고 엔티티 클래스간의 내재된 관계에서 추출된 경험적 분류 이름과 관련된 엔티티 클래스의 인스턴스들을 하나의 경험적 분류 인스턴스로 갖는 클래스이다.

이 클래스는 실제적인 경험적 분류를 갖게 되어 관계 데이터베이스의 관계 엔티티 테이블과 그 개념이 유사하지만 관계 엔티티 테이블이 경험적 분류된 엔티티들을 테이블 형식으로 구성하는데 비해 경험적 분류 클래스는 관련된 인스턴스들의 객체 식별자를 갖는 복합 객체 즉 경험적 분류 인스턴스들로 구성된다.

상위 클래스 선언부에서는 계승 관계를 나타내기 위해 경험적 분류 연산 클래스를 상위 클래스로 선언해 준다. 경험적 분류된 클래스의 선언부에서는 경험적 분류가 추출되는 엔티티 클래스들의 이름이 명시된다.

변수 선언부에서는 그 클래스 특유의 변수를 선언하고, 메소드 선언부에서는 경험적 분류 인스턴스를 첨가, 삭제하는 연산을 수행하기 위해 허용되는 메소드를 명시한다.

```

Class          Direction
super          Classify
Classify       Teacher, Student
instance       variable
Direction_name=char 10;
D_Classify=professor, SetofSubject
instance_method
D_delete: /*delete instance */

Class          Lecture
super          Classify
Classify       professor, Subject
instance       variable
lecture_name=char 10;
lecture_sex=char 5
L_Classify=professor, SetofSubject
instance_method
L_Delete: /* delete instant */
L_Add: /* addition instant */

Class          Attend
super          Classify
Classify       Student, Subject
instance       bar
Attend_name=char 20;
A_Classify=Student, SetofSubject
instance_method
A_Delete: /*deletion of instance*/
A_Add: /*addtion of instance */
    
```

[그림 1] 경험적 분류 클래스 정의된 예
 [Fig. 1] Exmpl of definition of heurilistic classifc class

경험적 분류 연산 클래스는 사용자가 별도로 정의한 경험적 분류 클래스에 대하여 데이터베이스 연산을 함으로써, 기존의 객체 지향 데이터 모델 내에서 경험적 분류를 직접적으로 지원해 주게 된다.

관계 연산 클래스는 임의의 엔티티와 관계되는 어떤 엔티티를 검색하는 것은 물론 사용자가 경험적 분류 이름과 어떤 조건만을 가지고 엔티티

들을 검색할 경우에, 사용자가 작성한 검색 질의어에서 경험적 분류 이름을 분류해서 메시지로 하여 경험적 분류 연산 클래스에 보낸다. 그러면 그 경험적 분류 이름을 키로 하여 해당되는 경험적 분류 인스턴스를 검색하여 반환하는 기능을 가지고 있다. 또한 경험적 분류된 엔티티들이 있을 때 이 엔티티 사이의 경험적 분류 이름을 검색하여 반환한다.

이와 같이 기능을 실행하기 위하여 관계 연산 클래스는 경험적 분류 인스턴스를 삽입, 삭제하는 메소드와 경험적 분류 이름을 검색하는 메소드, 경험적 분류 이름으로 관련된 인스턴스를 검색하는 메소드 등을 클래스 메소드로 가져야 한다.

경험적 분류 확장 객체 지향 데이터 모델에서는 기존의 객체 지향 데이터 모델에서의 질의어에 경험적 분류 메소드(Method related) 이 도입되어 경험적 분류 이름을 이용하여 정보를 변경하게 된다. 경험적 분류가 확장된 질의어의 구성 형식은 [그림 2]와 같다.

```

Class          Insert: {( method:=argument)_list }
                [With: { information_selected_message!}
                { Classify: { relationship_predecate!}
                /*insert of classify instant*/

Class          Delete
                [Where: { instance_predicate!}
                { Classify: { relationship_predicate!}
                /*delete of classify instant*/

Class          Insert: {value of instance}
                [With: { informationselectd_message!}
                [Where: {relationship_predicate!}
                { Classify: { relationship_predicate!}
                /* insert instace variable using classify*/

Class          Delete: {method_predicate!}
                [where: {instance_predicate!}
                { Classify: {relationship_predicate!}
                /*deletet instace variable using classify*/
    
```

[그림 2] 경험적 분류 클래스가 도입된 질의어 형식
 [Fig. 2] Query formation introducing heurilistic classifc class

경험적 분류 확장 데이터 모델에서의 정보 변경 및 갱신 메시지의 실제 예가 [그림 3]에 나타나 있다.

```
Student Insert: {name:= "Kim W K" ,
                age:=21, sex:= "Male" }
Classify: {Relationship:= "Direction" ,D_name: Computer,
          T_name:= "Lee K L" }
```

(a) Lee K L" 라는 교수에게서 수강하는 학생 중 조건의 학생을 삽입

```
Teacher Delete: {name:= "Kim H D" , age:=40}
Classify:
{Relationship:= "Lecture" ,L_name:= "Math" }
```

(b) 수학을 강의하는 교사의 이름이 "Kim H D" 이고, 나이가 40세인 사람을 Teacher 클래스에서 삭제

```
Teacher, career Update: {career+5}
Where: {name:= "Hong S K" }
Related: {Relationship:= "Iecure" ,
         L_name:= "Korean" }
```

(c) 국어를 강의하는 "Hong S K" 라는 교사의 경력에 5년을 가산하여 갱신

[그림 3] 경험적 분류를 이용한 삽입, 삭제, 갱신
[Fig. 3] Insertion and Deletion and Update using heuristic classify

6. 결론

기존의 객체 지향 데이터베이스에서 경험적 분류모델을 지원해 주기 위하여 경험적 분류 클래스를 도입하였으며 이 클래스들을 연산할 수 있는 경험적 분류 연산 클래스를 설계하여 경험적 분류 모델을 이용한 검색과 삭제, 삽입이 가능한 확장된 객체 지향 데이터베이스를 모델링하였다. 또한, 경험적 분류모델을 이용하여 객체 지향 데이터 베이스에 액세스하기 위한 질의어를 설계하였

고 확장 질의어를 처리하기 위한 경험적 분류 모델하에서 메소드를 설계하였다.

따라서 지향 데이터 베이스에서 경험적 분류를 이용한 데이터 검색, 변경이 가능하도록 구현하였으며 이로인해 보다 복잡한 현실 세계를 모델링하고 내제된 경험적 분류방법으로 정보의 효율적인 검색 및 변경이 가능하게 되었다.

앞으로는 다양한 모델을 개발하여 현실세계에 보다 가까운 객체지향 데이터베이스를 모델링하는 연구가 필요하다.

※ 참고문헌

- [1] Banerjee, J. et al. " An Object Model Issues for Object-oriented Application, ACM TOOIS 1987.
- [2] Banerjee, J. ,Kim W., Kim,k., "Queries in Object-oriented Database",Proc. 4th Intl Conf. Data Engineering feb, 1990.
- [3] R.R.Burton, "Diagnosing bugs in a Simple Procedural Skill," Intelligent Tutoring System(Eds. D,Sleeman and J,S.Brown), Academic Press, pp.157-183,1982.
- [4] P.G.Kearsley, Artificial Intelligence & Instruction: Application and Methods, A0ddition- Wesley,1987.
- [5] W.H. Inmon, Building the Data Warehouse 2nd edition, Jhon Wiley & Sons inc.,1996
- [6] Kurt Vanlehn,"Student Modelling," Foundations of Intelligent Tutoring System (Eds .M.C.Polson & J.J,Richardson),Lawrence Erlbaum Associates Publishers, pp.55-78,1988.
- [7] R.C.Lippert. "Expert Systems: Tutors, Tools, and Tutees," Journal of Computer-Based Instruction, vol. 16, no. 1, pp. 11-19, 1989.

- [8] G.F.Luger and W.A.Stubblefield Artificial Intelligence and the Desing of Expert Systems, The Benjamin/Cummings Publishing Company,Inc., 1989.

김 준 모



1988년 이학사
(광운대학교 전산학과)
1990년 이학석사(광운대학교
대학원 전산학과)
1998년 이학박사 수료
(전북대학교 대학원
전산통계학과)
1991년 유한공업전문대 강사
1991년 배화여자전문대학 강사
1992년 강원대학교 강사
1992년- 현재
전주기전여자대학 조교수
관심분야 : 멀티미디어 교육,
컴퓨터교육