

메타 데이터 클래스를 이용한 객체  
지향 데이터 조작 모듈의 설계  
**The Design of The Object object Data Operation  
Module using META Data Class**

김 준 모(Kim, Jun-mo)<sup>1)</sup>

**Abstract**

This paper designed Object-oriented Meta\_data operation module that introduced new Meta\_class basis the Meta\_Classification model. In order to implement this Data model, we have introduced heuristic class to traditional object-oriented database. And we designed Meta\_data operation module for Implemented Meta\_data that basis on the heuristic classification model using stored Meta\_data in extended object-oriented data model.

논문 접수 : 2009. 5. 6.  
심사 완료 : 2009. 5. 18.

---

1) 정회원 : 전주기전대학 보건복지학부 교수

## 1. 서 론

객체 지향 개념은 데이터 추상화 특성 계승, 객체 참조 등을 그 주요한 특성으로 한다. 경험적 메타 모델은 전문가 시스템 개발 및 객체 지향 데이터베이스 시스템 개발에서 메타 단계에서 적합하고 속도가 줄어들고, 융통성이 있기 때문에 메타 과정을 단축 축소하는 특성이 있다. 데이터 추상화는 데이터의 형식을 정의 할 때, 허용되는 연산을 정의하며 그 실제 세부적인 수행을 고려할 필요성을 없애 준다. 특성 계승은 객체 지향 개념에서 가장 주요한 특성 중이며 두 클래스가 계승 계층을 이루고 있을 때 상위 클래스의 특성 즉 인스턴스 변수와 메소드(method)등이 하위 클래스(subclass)에 계승되는 개념이다. 따라서 특성 계승에 의하여 하위 클래스들은 상위 클래스의 정보의 특성을 물려받고 계승하게 된다. 객체 참조는 각 객체마다 각각 고유한 객체 식별자를 갖고 있어서 이를 참조하여 각 객체들이 서로 이용하게 된다.[5]

## 2. 메타 데이터 모델

기존의 간단한 인스턴트 메타데이터화의 경우 메타화가 간단할 수 있지만 복잡한 문제의 경우 입력된 검색어로는 정확하고 적합한 메타화하기가 어렵다. 따라서 이러한 문제점을 해결하기 위해 본 논문은 클랜시가 제시한 경험적 메타화 방법을 도입하였다.

이 메타데이터 모델은 전문가 시스템 개발 및 객체 지향 데이터베이스 시스템 개발에서 메타 단계에 적합하고 융통성이 있기 때문에 메타 과정을 획기적으로 시간을 단축하는 장점이 있다. 메타데이터모델의 문제 해결과정은 자료 추상화 과정, 경험 연상과정, 정제 과정등 3단계가 있으며 자료추상화 과정은 사용자가 입력한 구체적인 자료들을 메타하여 추상화시키는 과정이다. 경험 연상과정은 추상화된 자료로부터 추상화된 결과를 연상하는 단계이며 이 단

계는 전문가의 지식이 가장 많이 포함되어 있다. 정제과정은 경험 연상에서 얻어진 결과를 메타화하고 관련 자료를 수집하여 해를 줄여 나가서 결론을 구하는 단계이다.[6]

### 2.1 객체 지향 개념

객체는 객체 지향 데이터 데이터베이스에서 데이터를 표현하는 기본 단위로서, 개별적인 워킹 메모리를 갖고 있어서 그 객체의 상태 타입을 기억하게 된다. 이 개별적인 메모리는 인스턴스 변수이며 데이터의 내용은 연속적인 인스턴스 변수에 저장된다. 객체 지향 데이터 모델에서의 연산은 메시지와 메소드에 의하여 처리하게 된다. 객체는 해당 클래스나 인스턴스에 원하는 연산을 클래스가 메시지로 보내면 허용되는 메소드 중에서 해당메시지에 해당되는 메소드를 동작하게 된다.[5]

객체 지향 데이터 모델에서는 상위 클래스의 특성을 계승하며 데이터 형식에 문자, 그래픽, 숫자등에 제한이 없으므로 소프트웨어의 재사용성(reuseability)이 가능하여 데이터베이스 관리시스템 프로그램의 사용코드를 절약할 수 있다. 그리고 도형이나 음성과 같은 메타데이터를 관리할 수 있다는 장점을 가지고 있어 기존의 데이터베이스에 비해 데이터모델링의 능력이 우수하다. 그리고 기존의 객체 지향 데이터베이스 시스템에서는 객체간의 메타된 데이터의 처리가 명확하지 않다.

## 3. 메타검색 방식

기존의 데이터베이스 시스템에서는 어떤 교수가 여러 과목을 지도할 때, 특정 과목을 지도관계로 하여 학생의 명단과 교수의 명단의 메타검색등이 어렵고 그래픽이나 음성등의 메타데이터의 처리가 불가능하였다. 그러므로 경험적 메타방식을 도입하여 데이터의 메타화를 명백하게 지원해 줄 수 있도록 객체 지향 데이터 모델을 확장할 필요성이 있다.[5]

따라서 엔티티클래스 간의 메타메소드를 적

용하면 엔티티클래스 간의 메타인스턴스를 검색 할 수 있고 메타인스턴트는 그래프, 음성 데이터를 의미한다. 또한 메타 방식을 이용하여 데이터의 메타데이터화가 가능하여 엔티티클래스를 효율적으로 액세스 할 수 있고 많은 클래스들이 존재하여 서로 복잡한 관계를 갖고 있는 기존의 객체 데이터베이스에서 메타데이터의 조작을 위해 본 논문에서는 메타데이터 클래스를 도입한다.[1]

#### 4. 메타데이터 모듈의 설계

객체 지향 데이터 모델은 메타데이터상의 관계성을 명백히 제시해 주지 않음으로써, 현실 세계에 보다 근접한 데이터베이스를 모델링하는데 문제점을 내포하고 있으며 이를 해결하기 위한 많은 연구가 진행되었다.[5]

따라서 객체 지향 데이터 모델에서 실제 데이터를 내포하고 있는 엔티티 클래스간의 검색이 가능하고 메타 인스턴스를 베이스로 하는 메타 클래스를 사용자 정의 클래스로 정의할 수 있으며 이들 메타클래스를 삽입, 삭제, 메타데이터화하는 기능을 가진 시스템 클래스를 도입하여 메타데이터를 효율적으로 지원해 줄 수 있게 한다. 메타 클래스는 사용자가 데이터베이스를 구성할 때 즉 엔티티클래스를 정의할 때 클래스간의 메타데이터를 추출해야 한다.

또한 이를 구현하기위해 메타데이터조작 모듈을 설계 구현해야 한다.

메타 연산 클래스는 연산 클래스를 상위 클래스로 갖고 있어서 기본적인 인스턴스의 삭제, 삽입, 검색 등의 행동 양식을 계승 받고 실제 메타데이터를 인스턴스로 하는 메타 클래스를 하위 클래스로 갖는다.[2]

##### 4.1 메타데이터 클래스의 정의

메타 클래스는 실제적인 메타데이터를 갖게 되어 관계 데이터베이스의 관계 엔티티데이터

과 그 개념이 유사하지만 관계 엔티티데이터 엔티티들을 테이블 형식으로 구성하는데 비해 메타 클래스는 관련된 인스턴스들의 객체 식별자를 갖는 복합 객체 즉 인스턴스들로 구성된다. 시스템에서 제공되는 메타 데이터 연산 클래스를 상위 클래스로 갖고 있어서 그 특성을 계승받고 엔티티 클래스간의 내재된 관계에서 추출된 메타 이름과 관련된 엔티티 클래스의 인스턴스들을 하나의 메타 인스턴스로 갖는 클래스라고 할 수 있다.

상위 클래스 선언부에서는 계승 관계를 나타내기 위해 메타 연산 클래스를 상위 클래스로 선언해 주며, 메타 클래스의 선언부에서는 메타데이터가 추출되는 메타엔티티 클래스들의 이름을 명시된다.

변수 선언부에서는 그 클래스 특유의 변수를 선언하고, 메소드 선언부에서는 메타인스턴스를 첨가, 삭제하는 연산을 수행하기 위해 허용되는 메소드를 명시한다.[6] 객체지향 프로그래밍 언어인 SmallTalk를 이용한 메타데이터 클래스의 알고리즘은 [그림 1]과 같다.

```

Class          Direction
super         Meta_Search
Search        Teacher,Student
instance      variable

Direction_name=char 10;
D_           Search=professor,
SetofSubject instance_method
              D_delete: /*delete instance
*/
    
```

```

Class          Lecture
super         Meta_Search
Search        professor,Subject
instance      variable
              lecture_name=char 2 0 ;
              lecture_sex=char 15
    
```

```

L
Search=professor,SetofSubject
instance_metod
L_Delete: /* delete instant */
L_Add: /* addition instant */

Class Attend
super Meta_Search
Search Subject
instance bar
Attend_name=char 20;
A_Search=Student, SetofSubject
instance_method
    
```

[그림 1] 메타 클래스 정의

메타 데이터 연산 클래스는 사용자가 별도로 정의한 메타 클래스에 대하여 메타데이터의 조작이 가능한 데이터베이스 연산을 함으로써, 기존의 객체 지향 데이터 모델 내에서 메타관계를 이용한 메타데이터의 검색이 가능하게 한다.

관계 연산 클래스는 임의의 엔티티와 관계되는 어떤 엔티티를 검색하는 것은 물론, 사용자가 음성, 도형, 소리, 그림 같은 메타 데이터를 어떤 조건만을 가지고 엔티티들을 검색할 수 있고, 사용자가 작성한 메타데이터 질의어에서 해당 아이디를 메타화해서 메시지로 하여 경험적 연산 클래스에 보낸다. 그러면 그 이름을 키로 하여 해당되는 인스턴스를 검색하여 반환하는 기능을 가진다. 이는 기존의 객체지향 데이터베이스가 관계성을 이용하지 못하는 단점을 개선할 수 있는 또한 메타데이터를 효율적으로 검색하므로써 검색시간이 획기적으로 개선될 것이라는 점이 기대된다. 그리고 엔티티들이 있을때 이 엔티티 사이의 관계성이 이용된 메타데이터의 이름을 검색하여 반환된다.

이와 같이 기능을 실행하기 위하여 관계 연산 클래스는 메타 인스턴스를 [그림 2]과 같이 삽입, 삭제하는 메타메소드와 인스턴트이름을

검색하는 검색메소드, 메타 이름으로 관련된 인스턴스를 검색하는 메타 메소드 등을 포함한 메타 시스템을 설계하였다.

메타 확장 객체 지향 데이터 모델에서는 기존의 객체 지향 데이터 모델에서 검색, 삭제, 삽입질의어에 메타검색 메소드를 설계 도입하여 메타데이터 이름을 이용하고 데이터베이스를 변경하게 된다. 메타 데이터가 확장된 질의어의 구성 형식은 [그림 3] 와 같다.

```

Class Meta_Insert:
{ ( method:=argument)_list }
[With:
{information_selected_message}
[ Search:relationship_predecate}]
/*insert of Search instant*/

Class Meta_Delete:
[Where: { instance_predicate}]
[Search: relationship_predicate}]
/*delete of Search instant*/
    
```

[그림 2] 메타 클래스가 도입된 모듈 형식

메타 확장 데이터 모듈에서의 정보 변경 및 갱신 메시지의 실제 예가 [그림 3]에 나타나 있다.

```

Student Meta_Insert:
{name:="송 미 권",age:=20, sex:="Female"}
Search: {Relationship:="Direction",D_name: }

a) "송 미 권"라는 교수에게서 수강하는 학생을 삽입
    
```

```

Teacher Meta_Delete:
{name;" 박 현 회", age:=46}
Meta_Classfy:
{Relationship:="Lecture",L_name:="Education"}
    
```



김준모



1988년 이학사(광운대학교)  
1990년 이학석사(광운대학교 대학원)  
1998년 이학박사 수료(전북대학교 대학원)  
2007년 교육학석사(전북대학교 대학원)  
2009년 교육학박사과정(전북대학교 대학원)

1991년 유한공업전문대 강사  
1991년 배화여자전문대학 강사  
1992년 강원대학교 강사  
1992년-현 전주기전대학 보건복지학부 교수

관심분야: 멀티미디어 교육, 유아교육,  
교육공학, E-러닝