

# 그래픽 객체 질의어 OGQL에서 순환적 질의의 시각적 표현 기법<sup>†</sup>

## (A Visualization Technique for Cyclic Queries in the Graphical Object Query Language OGQL)

정태성<sup>\*</sup>, 안명상<sup>\*\*</sup>, 조완섭<sup>\*\*</sup>  
(Tae-Sung Jung, Myung-Sang Ahn, Wan-Sup Cho)

**요약** 본 논문에서는 웹 환경에서 객체 데이터베이스를 위한 그래픽 질의어인 OGQL(Odysseus Graphical Query Language)의 설계 및 구현에 대하여 논하였다. OGQL은 객체 질의어에 포함된 객체 개념을 간단한 시각적 도구로 표현함으로써 사용자에게 편리한 질의 환경을 제공한다. 특히, OGQL에서는 클래스 계층구조(class composition hierarchy) 상에서 질의 조건을 표시하는데 널리 사용되는 순환(cycle)을 가진 경로식(path expression)을 그래프 형태의 직관적인 표현 방식으로 시각화하고 있다. 순환을 가진 질의의 시각적 표현은 복잡한 이유로 대부분의 그래픽 질의어에서 다루지 않고 있다. 또한, 클래스 간의 상속(inheritance)과 같은 객체 질의어의 고유 특성도 시각적으로 표현하고 있다. OGQL은 JDBC를 이용한 자바 응용 프로그램이므로 웹 환경에서 원격지 데이터베이스에 접근이 용이하고, 컴퓨터 시스템 플랫폼에 독립적이라는 장점도 가진다.

**Abstract** We propose a graphical object query language called OGQL (Odysseus Graphical Query Language) for Odysseus object databases. By employing simple graphical notations, OGQL provides a user friendly database interface for object databases. OGQL provides a simple graphical notation for advanced features of the object queries such as path expressions including cycles (cyclic queries). Cyclic queries have not been handled in the previous graphical query languages because of their complexity and ambiguity in devising appropriate graphical notations. OGQL also provides a graphical notation for inheritance hierarchy. Since OGQL has been implemented in Java and JDBC standard interface, users can access the remote databases in a web environment and the system has platform-independent architecture.

### 1. 서론

그래픽 질의어는 데이터베이스에 저장된 대용량 자료에 대하여 그래픽 사용자 인터페이스를 사용하여 질의를 수행할 수 있도록 지원하는 데이터베이스 도구이다. 대부분의 상용 DBMS에서 고급 질의어와 함

께 그래픽 질의어를 제공하여 사용자가 편리하게 데이터베이스를 조작하고 관리할 수 있도록 지원하고 있다. 기존의 그래픽 질의어에 관한 연구는 주로 관계 데이터베이스를 대상으로 하고 있으며[10,15], 상용 관계 DBMS에서도 대부분 그래픽 질의어를 제공하고 있다. 최근 들어 복잡한 데이터 구조를 모델링 할 수 있는 객체 데이터베이스의 사용이 증가함에 따라 이를 대상으로 하는 그래픽 질의어에 관한 연구가 이루어지고 있다[1,2,12,14,17,19,20]. 그러나, 기존의 그래픽 객체 질의어는 간단한 형태로 그 표현력을 한정하고

<sup>†</sup> 본 논문은 한국과학재단의 해외 POST-Doc. 연수지원비에 의하여 연구 되었음.

<sup>\*</sup> 충북대학교 정보산업공학과

<sup>\*\*</sup> 충북대학교 경영정보학과, 컴퓨터 정보통신 연구소

있으므로 객체 질의어의 중요한 특성들을 시각화하는데 충분하지 못하다. 구체적으로 살펴보면 기존의 그래픽 객체 질의어는 클래스 계층구조(class composition hierarch) 상에서 조건을 표시하는데 널리 사용되는 경로식(path expression)[6]을 직관적으로 표현하지 못하고 있으며, 집합속성, 메소드(method), 상속(inheritance) 등과 같은 객체 질의어의 고유 특성도 지원하지 않고 있다. 특히, 경로식 자체가 순환(cycle)을 포함하는 경우도 자주 발생하지만[8], 대부분의 그래픽 질의어에서 이러한 경우를 다루지 않고 있다. 객체 데이터베이스는 기존의 관계 데이터베이스보다 복잡한 구조의 자료를 관리하므로 질의어 자체도 복잡하다. 그리고 관계형 데이터베이스보다 객체 데이터베이스의 사용자를 위한 편리한 그래픽 질의어가 더욱 필요하지만 충분한 표현력과 편의성을 제공하는 도구가 미흡한 실정이다.

OGQL은 Odysseus [9] 객체 DBMS를 위한 그래픽 질의어로서 객체 질의어의 주요 특징들을 시각화하고 있다. 특히, 클래스 계층 구조상에서 조건을 표시하는데 널리 사용되는 경로식(path expression)을 직관적인 방식으로 시각화하고, 클래스간의 상속 (inheritance) 과 같은 객체 질의어의 고유 특성도 시각화하고 있다. OGQL에서는 그래픽 질의어에서 다루기 어려운 순환 (cycle)을 가진 질의어와 순환적 질의어(recursive query) 및 명시적 조인(explicit join)을 간단히 시각화하는 방안을 제시한다. 논문에서는 순환을 가지는 질의 그래프의 유형에 따라서 대응하는 텍스트 질의어로의 변환 방법도 제시한다. 또한, 하나의 GUI를 이용하여 그래픽 질의어와 텍스트 질의어 모두 사용할 수 있도록 지원하고 있으며, 질의어의 재사용이 가능하도록 질의어 저장 기능도 함께 제공하고 있다. 그리고 OGQL은 JDBC를 이용한 자바 응용 프로그램으로 웹 환경에서 데이터베이스 접근이 용이하고 플랫폼에 독립적이라는 장점도 있다.

본 논문의 구성은 다음과 같다. 2장에서는 OGQL과 관련된 그래픽 질의어에 대해서 살펴본다. 3장에서는 OGQL의 구조, 설계 및 구현에 대해서 소개한다. 특히, OGQL이 그래픽 질의어로서 가지는 특징을 중심으로 설계 및 상세 구현에 대해서 살펴본다. 4장에서는 OGQL의 특징 및 기대효과에 대해서 살펴보고, 5 장에서는 결론을 맺는다.

## 2. 관련연구

이 장에서는 그래픽 질의어에 대한 관련연구와

OGQL을 설계하고 구현하기 위해 필요한 관련 기술에 대해서 살펴본다. 먼저 그래픽 질의어와 Odysseus DBMS에 대해서 살펴본다.

### 2.1 그래픽 질의어

데이터베이스 질의 언어는 데이터베이스 모델 및 데이터베이스 시스템의 발전과 밀접한 관계를 맺고 있다. 파일 처리 시스템에서 데이터 조작용 제 3세대 언어에 의해 지원되었다. 이와 비슷하게 네트워크 및 계층 데이터베이스시스템에서의 데이터 조작용을 위한 데이터베이스 언어가 개발되었다. 관계형 데이터베이스 관리 시스템의 탄생으로 선언적(Declarative)이며, 사용자 중심적 질의어인 SQL과 Quel 등이 개발되었다. 이러한 질의어의 출현으로 사용자가 데이터베이스 시스템에게 "how"가 아니라 "what"을 명시하여 원하는 질의를 수행할 수 있게 되었다. 이러한 선언적 질의어의 특징에 의해서 개발자나 고급 사용자가 아닌 일반 사용자들이 사용하기에 적합한 질의어가 탄생되었다. 테이블 형태의 그래픽 질의어도 관계형 데이터베이스 시스템을 위해 제안되었다. 이러한 질의어로서는 QBE[16]가 대표적인 예이다. QBE와 같은 테이블 형태의 질의어에서 사용자는 스크린상의 테이블 구조를 보면서 질의를 표현하였다.

그 후 Gql[11]이나 Visualizer Query for Window [5] 같은 그래픽 사용자 인터페이스가 제안되었다. 이러한 그래픽 질의어들은 데이터베이스를 개체-관계(Entity-Relationship) 모델로 표현하고 있으며, 그래픽 질의어들은 학습이 쉽고, 사용자에게 직관적으로 질의를 작성할 수 있게 하였다. 그러나 부질의어나 조인 등과 같은 복잡한 질의를 구성하는 데는 한계가 있었다.

객체 데이터베이스를 위해서도 선언적 질의어인 OQL이 개발되었다[3]. OQL은 객체 데이터베이스 O2[7]의 객체 질의어이며, 관계 데이터베이스를 위한 SQL보다 복잡하다. 그 이유는 객체지향 특징들의 복잡한 구조를 지원하기 위해 설계되었기 때문이다. OQL은 관계형 데이터베이스에서 사용되었던 조인을 하지 않고도 향해 경로(Navigational Path)를 사용해서 질의를 실행할 수 있다.

현재까지 객체 데이터베이스 시스템을 위한 다양한 그래픽 질의어가 개발되었다[4, 13, 18]. 지금까지 객체 데이터베이스 관리 시스템을 위한 대부분의 그래픽 질의어들은 질의 생성 기능을 주로 다루고 있으며, 스

키마 관리 기능이 미흡한 상태이다. 또한 스키마 관리 기능이 가능한 그래픽 질의어들도 기존에 존재하는 클래스에 대한 정보를 사용자에게 미리 보여주지 않으므로 스키마 구축이나 질의 작성 시 불편함을 가지고 있었다. 객체 데이터베이스 관리 시스템 O2의 그래픽 질의어인 O2talk [13]의 경우, 이미 존재하는 클래스나 사용자 정의 타입에 대한 정보를 사용자는 미리 알 수 없으므로 질의 작성 시 불편함을 감수해야만 했다. 또한 객체지향 데이터베이스를 위한 그래픽 질의어인 GOQL[18]은 ad-hoc 질의 작성 기능을 주로 다루고 있으며, 클래스간의 계층 구조나, 순환(cycle)을 가지는 질의의 표현에 있어서는 미흡하다.

## 2.2 Odysseus DBMS

Odysseus는 정보검색용 객체-관계형 멀티미디어 데이터베이스 관리 시스템(DBMS)으로, 대용량 멀티미디어, 텍스트 정보검색, GIS, OLAP, 데이터마이닝 등의 최신 응용에 적합한 DBMS이다. Odysseus는 관계형 DBMS 특징뿐만 아니라 다음과 같은 객체지향 DBMS의 특징을 지원하고 있다. 첫째는 클래스 상속 개념을 지원한다. 클래스 상속은 슈퍼 클래스 C로부터 상속받아 새로운 서브 클래스 D를 생성할 수 있는데 D는 C의 모든 속성을 가지게 된다. 게다가 D클래스만의 고유의 속성도 가질 수 있다. 두 번째로 클래스의 한 컬럼에 복수개의 데이터를 포함시킬 수 있는 Set(), Bag(), List() 등의 다중 값 데이터 타입을 지원한다. 이러한 데이터 타입을 통해 숫자형, 문자형 또는 객체 타입의 값을 여러 개 저장할 수 있다. 세 번째로 Odysseus는 OID(Object-Identifier)를 통해 객체를 조작할 수 있다. 모든 객체들은 서로 다른 OID를 가진다. 관계형 DBMS에서 서로 다른 두 개의 객체를 구분하기 위해서는 속성 값으로 키를 가지고 있어야 했다. 그러나 객체 DBMS에서는 OID를 통해 식별할 수 있으며, OID를 통해 객체 탐색도 가능하다.

Odysseus는 네트워크를 통한 클라이언트/서버 구조를 가진다. 서버 역할을 담당하는 Odysseus/OOSQL 서버는 멀티미디어 객체 저장 시스템인 Odysseus/COSMOS를 하부 저장 시스템으로 장착하여 객체지향 SQL 질의어(Odysseus Object-Oriented SQL: OOSQL) 기반 다사용자 동시 질의 처리, 파손 회복 기능, 트랜잭션 관리 등의 기능을 수행한다. 클라이언트 역할을 담당하는 Odysseus 클라이언트는 사용자 인터페이스,

Odysseus/OOSQL 서버와의 통신, 그리고 객체 버퍼 관리 등의 기능을 수행한다.

## 3. OGQL(Odysseus Graphical Query Language)

OGQL(Odysseus Graphical Query Language)은 Odysseus 객체 데이터베이스 시스템을 위한 그래픽 객체 질의어이다. 이 장에서는 OGQL의 설계 및 구현에 대해서 논한다.

### 3.1 OGQL의 구조

OGQL의 시스템 구조는 크게 Connection Manager, Query Manager, Schema Manager로 이루어진다. 그림 3.1은 OGQL 시스템의 전체적인 구조를 보여주고 있다. Connection Manager는 원격지 객체 데이터베이스 접속 기능을 담당한다. Query Manager는 사용자에게 그래픽 질의를 작성하고 실행하도록 지원한다. Schema Manager는 데이터베이스 스키마를 관리하는 기능을 한다. 그림 3.1과 같이 OGQL 사용자는 인터넷과 PC를 통해 원격 데이터베이스에 쉽게 접속하여 그래픽 환경에서 질의를 할 수 있다. 사용자가 작성한 OGQL은 텍스트 질의로 변환되어 Odysseus DBMS에서 실행된다.

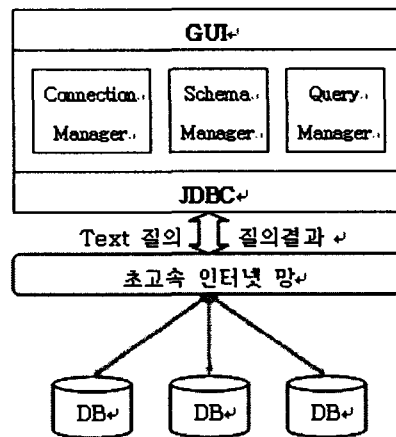


그림 3.1 OGQL 시스템 구조

그림 3.2는 OGQL의 초기화면이다. 그림에서 보는 바와 같이 OGQL의 시작 창은 크게 5가지 영역으로 나누어 볼 수 있다.

- 데이터베이스 브라우징 창: OGQL을 통해 연결 가

능한 데이터베이스 목록을 제공한다. 데이터베이스를 선택하여 연결하게 되면, 그 아래 영역에서 해당 데이터베이스 내에 존재하는 모든 클래스를 트리 형태로 표현하여 각 클래스들 사이의 상속 관계를 보여준다.

- 클래스 정보 창: 데이터베이스 내의 선택된 클래스의 스키마 정보를 제공한다. 필드 이름, 필드 타입 정보를 제공함으로써 사용자가 질의 작성 시 편의성을 제공한다.

- GQL 작성 창: 클래스를 선택하고 조건을 부과하여 그래픽 질의어를 작성하는 화면이다. 그래픽 질의어는 이 창에서 그래프 형태로 생성되며, 질의 작성이 끝나면 실행할 수 있다.

- OQL 작성 창: 고급 사용자를 위하여 텍스트 질의를 작성할 수 있도록 지원한다. 이 때 사용자는 클래스 정보창에 나타나는 정보를 활용하여 질의를 작성할 수 있다. 또한, 작성된 질의어는 다음에 반복사용하기 위하여 저장될 수 있으며, 저장된 질의를 검색하여 원하는 대로 수정한 다음에 질의를 실행할 수도 있다.

- 질의 결과 창: GQL 또는 OQL에서 작성된 질의의 실행 결과를 보여준다. 각 창은 스크롤 기능을 사용하여 확장될 수 있다.

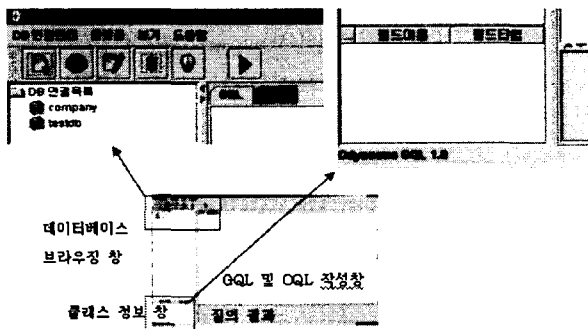


그림 3.2 OGQL의 화면 구성

OGQL에서 제공되는 기능은 JDBC를 통한 객체 데이터베이스에 연결, 스키마 정보의 조회 및 질의 작성, 경로식의 시각적 표현, 클래스간의 상속 관계를 직관적으로 표현하고 있다. 또한 고급 사용자를 위한 객체 질의어를 작성할 수 있는 기능도 제공하고 있다. 각 기능들에 대해서 좀 더 자세히 살펴보자.

### 3.2 Connection Manager

OGQL이 JDBC를 통하여 Odysseus에 접속하는 단위는 데이터베이스이다. OGQL의 Connection Manager는 간단한 그래픽 창을 이용하여 서버 측에 존재하는 데이터베이스를 연결등록, 삭제, 변경하고 접속 할 수 있다. 원격지 데이터베이스를 연결할 때 OGQL에 사용자가 입력할 정보는 아래와 같다.

- DB 명: JDBC 드라이버 이름으로 클라이언트에 설치되어 있어야 한다.

- DB 서버: 원격 데이터베이스에 접속하기 위한 위치정보(URL)를 입력한다.

사용자는 작업하고자 하는 여러 개의 데이터베이스를 등록시킬 수 있으며, 일단 등록된 후에는 IP 주소 대신에 데이터베이스 이름을 사용하여 연결할 수 있으므로 편리하다. 데이터베이스가 선택되면 OGQL은 JDBC를 통해 해당 서버에 접속하여 데이터베이스 스키마 정보를 가져오게 되며, 이 역할은 Schema Manager가 담당하게 된다.

### 3.3 Schema Manager

Schema Manager는 데이터베이스 내의 클래스를 보여주고, 이들의 상속 관계를 보여주는 기능을 제공한다. 데이터베이스에 접속 후 선택된 데이터베이스의 스키마 정보를 간단한 트리 구조로 표현하기 때문에 사용자는 해당 데이터베이스에 대하여 마우스 액션을 통해 데이터베이스 정보를 직관적으로 알 수 있다. 클래스간의 상속관계 또한 쉽게 알 수 있으므로 사용자로 하여금 해당 데이터베이스 내의 클래스 구조에 대한 이해를 높일 수 있다.

그림 3.3(a)은 예제 데이터베이스의 구조를 보여주고 있다. 그림과 같이 4개의 클래스(PERSON, ADDRESS, CAR, COMPANY)는 서로 객체 참조(OID Reference)를 하고 있으며, 2개의 클래스(TRUCK, BUS)는 CAR 클래스와 상속 관계를 가지고 있다. 또 SMALL BUS 클래스는 CAR로부터 상속받은 BUS로부터 다시 상속받아 생성된 클래스이다.

이러한 구조를 가지고 있는 데이터베이스를 OGQL에서는 그림 3.3(b)과 같이 표현하고 있다. 데이터베이스

스 내의 모든 클래스를 시각적으로 볼 수 있으며, 상속 관계 또한 직관적으로 알 수 있다.

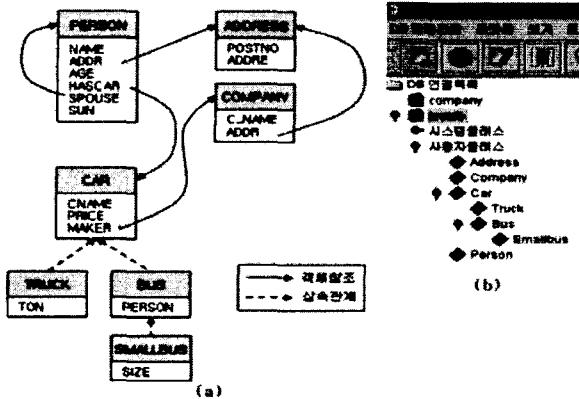


그림 3.3 예제 데이터베이스 구조와 OGQL화면

### 3.4 클래스 정보 창

클래스 정보 창은 선택한 클래스의 속성정보를 보여주는 기능을 한다. 클래스의 속성정보를 보고자 할 경우는 해당 클래스를 마우스 왼쪽 버튼으로 한번만 클릭 해주면 아래의 클래스 정보창에 나타나게 된다.

화면에 필드이름과 필드타입 정보를 보여주고 있다. 이처럼 클래스 정보창에서 클래스의 속성 정보를 제공해 줌으로써 사용자가 해당 클래스에 질의를 작성하기 전에 속성 정보를 미리 알 수 있기 때문에 질의 작성에 도움이 된다.

### 3.5 GQL(그래픽 질의어)의 설계 및 구현

그래픽 질의어를 이용하여 질의를 작성하려면, 우선 질의의 대상이 되는 클래스를 선택해야 한다. 질의 대상 클래스를 선택하는 방법은 데이터베이스 브라우징 창에 나타난 클래스 리스트에서 하나를 선택하면 GQL 창에 해당 클래스가 그래픽 노드로 표현된다. 그림 3.4의 왼쪽 노드는 클래스 PERSON이 선택된 모습을 보여준다. 여기서 사용자는 OID 참조 관계를 선택하여 속성의 도메인이 되는 클래스를 추가로 질의 대상으로 선정할 수 있다. 그림에서는 hascar 속성을 선택하여 Car 클래스를 추가한 모습이다. 이와 같이 OGQL에서의 질의 작성은 질의의 대상이 되는 클래스를 선택한 후, 속성도메인 관계를 추가로 선택함으로써 그래프 형태의 시각적 질의어를 만들어 나가게 된다. 클래스 선정과 함께 사용자는 속성에 대하여 조건을 부과할 수 있다.

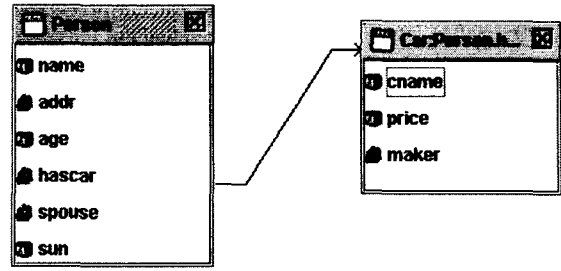
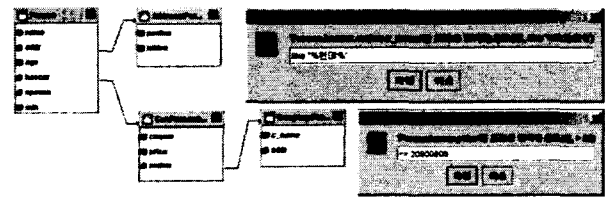


그림3.4 Path Expression 그래픽 표현 화면

좀 더 복잡한 질의를 작성해 보자. '제조사 현대 자동차에서 만든 차 중에 가격이 2000만원 이상인 차를 소유한 사람의 이름, 나이, 우편번호, 주소'에 대한 정보를 얻고자 한다. 질의 대상 클래스는 PERSON 클래스이며, 이름과 나이는 PERSON 클래스에 존재하고, 우편번호와 주소는 ADDRESS 클래스에 존재한다. 또한 제조사에 대한 정보는 COMPANY 클래스에 차량 가격에 대한 정보는 CAR 클래스에 존재한다. 그림 35 (a)는 PERSON 클래스로부터 속성 도메인 관계의 속성을 선택함으로써 이와 같은 클래스를 질의 그래프에 포함시킨 모습을 보여준다. 사용자는 그림 35(a)의 그래프에 대하여 조건을 부과할 수 있으며, 조건은 그림 35(b)와 같이 별도의 창을 통하여 표시된다. 조건의 작성이 완성되면 사용자는 실행 버튼을 클릭하여 그림 3.6과 같은 결과 테이블을 얻게 된다.

Person.no.	Person.age	Person.addr.postno	Person.addr.addr
장해경	28	333-333	충청북도 청원군

그림 3.5 예제 질의 그래픽 질의어 작성



(a) 그래픽 질의어 표현 (b) 조건 부여

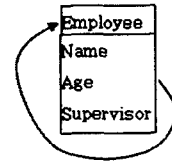
그림 3.6 예제 질의 수행 결과

### 3.6 순환(Cycle)을 갖는 질의어의 그래픽 표현

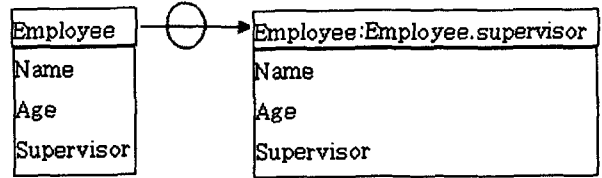
여기서는 순환(cycle)을 포함한 그래프 형태의 시각적 질의를 다룬다. 일반적으로 순환(cycle)이 포함된 질의는 복잡한 경우가 많으므로 대부분의 시각적 질의에서는 다루지 않는다.

(1) Recursive Query

“주어진 Employee의 Supervisor를 모두 찾아라”와 같은 재귀적(recursive) 질의의 시각적 표현은 대개 그림 3.7(a)와 같이 순환(cycle)을 포함한다. 그러나, 그림 3.7(a)와 같이 표현하면 질의 조건을 작성하는데 모호함이 발생한다. 예를 들어, 그림에서 Employee 노드에 조건 age=30을 부과하면 이 조건이 Employee.age = 30과 Employee.supervisor.age = 30 중 어느 것을 의미하는지 모호하기 때문이다. 이러한 문제를 해결하기 위하여 OGQL에서는 그림 3.7(b)와 같이 표현한다. 그림 3.7 (b)에서는 사용자가 속성 supervisor를 선택하면 Employee의 복사본인 ‘Employee: Employee.supervisor’가 새로운 그래픽 노드로 생성되고, supervisor에 해당하는 링크가 생성되어 두 그래픽 객체를 연결한다. 또한, 사용자는 supervisor에 해당하는 링크를 더블 클릭해서 그의 성질을 ‘recursive’로 설정하여 recursive 질의임을 표시한다. 이렇게 하면 두 노드가 분리되므로 조건 부여에서 모호함이 발생하지 않는다. 즉, Employee 노드에 조건 age=30이 부과되면 Employee.age=30로 해석하고, 복사본 노드인 Employee: Employee.supervisor’에 조건이 부과되면 Employee: e.supervisor.age=30로 해석한다.

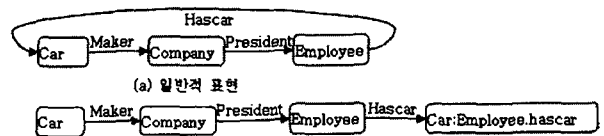


(a) 일반적 표현

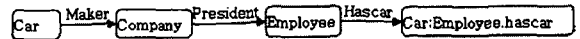


(b) OGQL에서의 표현

그림 3.7 순환질의 표현



(a) 일반적 표현



(b) OGQL에서의 표현

그림 3.8 Cycle 질의의 표현

(2) Cycle Query

재귀적 질의는 아니지만 순환(cycle)을 형성하는 질의도 있다. 예를 들어, 경로식 Car.maker.president.hascar에서는 Car 클래스가 2회 선택되어 그림 3.8(a)와 같이 순환을 형성한다.

OGQL에서는 이 경우에도 재귀적 질의와 마찬가지로 Car 클래스의 복사본을 만들어 그림 3.8(b)와 같이 표시한다. 여기서도 복사본 노드를 생성하여 그래픽 객체로 표현하면 질의 작성 시 발생할 수 있는 모호함을 해결할 수 있다. 예를 들어, 그림 3.8(a)에서는 Car 클래스에 있는 price 속성에 ‘price > 20000’ 과 같이 조건을 부여한다고 가정 할 때 조건이 Car.price에 부여한 것인지 Car.maker.president.hascar.price에 부여한 것인지 모호함이 발생하게 된다. 그러나, OGQL에서는 그림 3.8(b)처럼 복사본 노드를 사용하므로 조건의 대상이 명확해진다.

(3) 명시적 조인 질의 (Explicit join query)

경로식을 이용한 조인 연산의 표시에서도 복사본 노드를 사용한다. 그림 3.9는 ‘Car.maker.address.city = Car.maker.president.address.city’의 조인 질의 표현을 나타낸 것이다. 이러한 경우에도 cycle query 같은

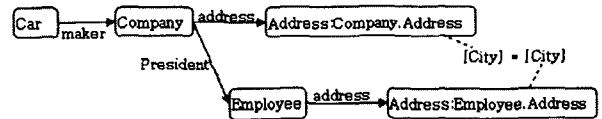


그림 3.9 OGQL에서의 조인 클래스의 표현

모호함이 발생할 수 있지만, OGQL에서는 Address의 복사본 노드를 사용함으로써 이러한 모호함을 해결할 수 있다.

3.6 OQL 작성 창의 설계 및 구현

OQL 작성 창은 객체 질의어의 문법을 잘 아는 고급 사용자를 위한 기능이라고 할 수 있다. 객체 질의어는 관계 데이터베이스에서 표준 SQL보다 복잡한 구조를 가지고 있기 때문에 일반 사용자들은 사용하

기 어려운 면이 있다. 그러나, 고급 사용자들은 그래픽 질의어를 이용한 질의도 필요하지만, 보다 복잡한 질의를 하고자 할 때 텍스트 질의어를 직접 작성하고 실행할 수 있는 OQL 작성 창을 제공하는 것도 필요하다.

OQL의 기능은 GQL 형태로 생성되어 저장된 질의를 불러와서 실행하는 기능이 있다. 그 외에 직접 질의를 작성하거나 작성된 질의의 실행은 물론 저장할 수 있는 기능을 지원한다. 또한, 히스토리 기능을 이용하여 사용자가 실행했던 질의를 탐색하여 다시 실행할 수 있도록 지원하고 있다.

#### 4. OQL의 특징 및 기대효과

본 논문에서는 객체-관계 DBMS인 Odysseus를 기반으로 개발된 데이터베이스 그래픽 객체 질의어 OGQL의 설계 및 구현에 대하여 논하였다. OGQL의 주요 특징과 장점은 다음과 같다.

##### ▷ 편리한 데이터베이스 연결 관리

OGQL에서 사용자는 처음에 한번 데이터베이스에 대한 IP나 URL 등록하면 그 이후로는 간단히 DB 이름만 선택하여 원격지 DB에 접속할 수 있다. 또한, 접속과 연결해제, 데이터베이스 등록 등 연결과 관련된 작업을 그래픽 환경에서 편리하게 수행할 수 있다.

##### ▷ 경로식의 시각화

경로식은 관계 데이터베이스에서의 조인에 해당하며, 다수의 클래스에 대한 질의를 표현하는데 사용되는 객체 질의어의 핵심 요소이다. OGQL에서는 경로식을 간단하게 표현하는 시각화 기법을 제공한다. 복잡한 의미를 가지는 경로식을 시각화함으로써 사용자 편의성이 향상된다.

##### ▷ 클래스 노드의 복사본 사용을 통한 순환적 질의의 그래픽 표현

경로식을 이용한 Recursive Query, Cycle Query, Explicit Join Query를 작성할 때 발생할 수 있는 순환으로 인한 모호함을 해결하기 위하여 OGQL에서는 클래스 노드의 복사본을 이용한다. 이렇게 함으로써 질의 조건의 의미가 명확해져 모호함이 해결된다.

##### ▷ 상속 개념의 시각화

객체 데이터베이스에서는 상속 개념을 제공하며, 객체 질의어에서도 이에 따라서 서브 클래스 혹은 전체 클래스 계층 구조에 관한 질의를 수행할 수 있다. OGQL에서는 슈퍼 클래스와 서브 클래스로 구성되는 상속 개념을 트리 형태로 시각적으로 제공하여 사용자가 이러한 상속의 개념을 사용할 수 있도록 하고 있다.

##### ▷ JDBC 기반 자바 응용 프로그램

OGQL은 JDBC를 이용한 자바 응용 프로그램으로 원격지에서의 데이터베이스 접근이 용이하여 사용의 편의성을 증대시키고, 자바의 플랫폼 독립적인 특성에 따라 플랫폼에 구애받지 않고 사용할 수 있어 실용성을 높였다.

OGQL은 객체 데이터베이스의 사용자 편의성을 증대시키는데 기여할 것이다. 기대 효과는 크게 세 가지로 나누어 생각할 수 있다. 그 중 하나는 그래픽 객체 질의어의 표현력 향상이다. OGQL은 기존의 그래픽 질의어에 비하여 객체 개념의 지원이라는 측면에서 향상된 질의 표현력을 가진다. 경로식, 상속, 순환적 질의 등의 주요한 객체 개념을 시각화하여 그래픽 질의어에 반영함으로써 질의 표현력을 향상시킨다. 향상된 질의 표현력은 그래픽 질의어의 유용성을 한층 높게 된다.

또 다른 하나는 편리한 질의 환경 제공이다. 사용자가 그래픽 질의를 작성하는데 필요한 스키마 정보를 그래픽 인터페이스를 통하여 사용자에게 제공하므로 사용자가 스키마 정보를 기억할 필요가 없어진다. 다만 데이터베이스 이름만 기억하면 되는 것이다.

마지막으로, 웹 환경에 적합한 그래픽 질의어라는 것이다. 인터넷을 통하여 원격(혹은 지역)지에 위치한 객체 데이터베이스를 연결하고, 질의하는 작업을 수행할 수 있도록 Java와 JDBC를 이용하여 그래픽 질의어를 구현했기 때문에 사용자는 플랫폼과 무관하게 원격지 데이터베이스를 편리하게 접근하여 사용할 수 있다.

#### 5. 결 론

본 논문에서는 그래픽 질의어에 대해서 살펴보고

웹 환경에서 Odysseus 객체 DBMS를 위한 그래픽 질의어를 설계 및 구현하였다. OGQL은 객체 질의어의 특징들을 직관적인 방식으로 시각화하여 사용자 편의성을 증대시켰으며, 사용자에게 편리한 질의환경을 제공한다. 또한, 웹 환경에서 사용이 적합하도록 설계되어 인터넷을 통하여 원격지 DB 접근이 용이하다. 특히, 그래픽 질의에서 표현이 어렵다고 알려진 순환을 가진 질의의 시각화 방안을 제시하였다.

현재 OGQL은 Odysseus DBMS를 기반으로 구현되었으며, 뷰와 중첩 질의 등의 문제는 향후 연구과제로 남겨 두었다. Odysseus DBMS에서 제공되는 JDBC Driver가 현재 완벽한 상태가 아니라는 점을 감안하여 구현되었기 때문에 현재 상태에서 다른 객체 데이터 베이스를 연결하여 사용하기는 어렵다. 향후 연구 과제로 Odysseus DBMS 뿐만 아니라 다른 객체 DBMS에도 자유롭게 연결하여 사용할 수 있도록 확장하고자 한다.

## 참 고 문 헌

- [1] N. H. Balkir et al., "VISUAL: A Graphical Icon-Based Query Language," In Proc. Int'l Conf. on Data Engineering, pages 524-533, 1996.
- [2] I. Cruz, "Doodle: A Visual Language for Object-Oriented Databases," In Proc. ACM SIGMOD Int'l Conf. on Management of Data, pages 71-80, 1992.
- [3] Delobel, C. et al., From Relational to Object-Oriented Systems, International Thomson Computer Press, 1995.
- [4] Dennebouy, Y. et al., "SUPER: Visual Interfaces for Object+Relationship Data Models," Journal of Visual Languages and Computing, Vol. 6:73-99, 1995.
- [5] IBM-ASTRAC Ltd, "Visualizer Query for Windows (VQ/W)," User Manual, 1995.
- [6] M. Kifer, W. Kim, and Y. Sagiv, "Querying Object-Oriented Databases," In Proc. ACM SIGMOD Int'l Conf. on Management of Data, pages 393-402, 1990.
- [7] Lecluse, C. et al., "O2 an object-oriented data model," In Proc. Int'l. Conf. EDBT, pages 556-562, 1988.
- [8] Won Kim, Introduction to object-oriented databases, MIT Press.
- [9] Odysseus User Manual, KAIST, 2000.
- [10] G. Ozsoyoglu and H. Wang, "A Relational Valculus with set Operators, its Safety, and Equivalent Graphical Languages," IEEE Trans. on Software Engineering, Vol. 15, No. 9, Sept. 1989.
- [11] Papantonakis, P. and King, H., "Gql: a Deductive Graphical Query Language Based on the Functional Data Model," In Proc. AVI, pages 113-122, 1994.
- [12] F. Staes, L. Tarantino, and A. Tiems, "A Graphical Query Language for Object-Oriented Databases," In Proc. IEEE Workshop on Visual Languages, pages 205-210, 1991.
- [13] Sentissi, T. and Pichat, E., "A Graphical User Interface for Object-Oriented Database", In Proc. 17th Int'l. Conf. Chilean Computer Society, 1997.
- [14] UniSQL, UniSQL/X Users Manual, Rel. 1.0, 1991.
- [15] K. Y. Whang et al., "Two-Dimensional Specification of Universal Quantification in a Graphical Database Query Language," IEEE Trans. on Knowledge and Data Engineering, Vol. 18, No. 3, Mar. 1992.
- [16] Zloof, M., "Query By Example: a database language," IBM Systems Journal, Vol.16:324-343, 1977.
- [17] 김정희, 조완섭, 이석균, 황규영, "VOQL : 시각적 객체 질의어," 한국전자공학회 논문지, 2001. 9
- [18] Euclid Keramopoulos, Philippos Pouyioutas and Chris Sadler, "GOQL, a Graphical Query Language for Object-Oriented Databases", In Proceedings of the Third Basque International Workshop on Information Technology (BIWIT'97): Data Management systems, Biarritz (France), July2-4, pp.35-45
- [19] Jan Paredaens, Peter Peelman, and Letizia Tanca, "G-Log: A Graph-Based Query Language", IEEE Transactions on Knowledge and Data



Engineering, Vol 7, No. 3, JUNE 1995

- [20] Lei Sheng, Z. M. Ozsoyoglu and G. Ozsoyoglu, "A Graph Query Language and Its Query Processing", In *Proceedings of the 15th International Conference on Data Engineering*, 23-26 Mar 1999. pp 572-581



정 태 성 (Tae-Sung Jung)

충북대학교 경영정보학과  
경영학사  
충북대학교 정보산업공학과  
석사과정  
관심분야 : DB, DM, OLAP, GUI



안 명 상 (Myung-Sang Ahn)

충북대학교 경영정보학과  
경영학사  
충북대학교 전자계산학과  
이학석사  
충북대학교 경영정보학과  
박사과정

관심분야 : DW, DM, OLAP, MIS



조 완 섭 (Wan-Sup Cho)

한국전자통신연구원,  
행망주전산기본부, 연구원  
한국과학기술원 전산학과  
공학박사  
현 충북대학교 경영정보학과  
부교수

관심분야 : 전자 상거래, 데이터 웨어하우스 & OLAP, CRM, DB, Web DB, Multimedia Information Systems, IR, GIS, MIS, Query Optimization and DBMS Performance, GUI, CG