

# 시각 요소와 시각 변수를 통한 시각 객체 질의어(VOQL)의 개선

이 석 균 †

요 약

최근 제안된 시각 객체 질의어(VOQL)는 시각 질의어로 스키마 정보가 시각적으로 질의 표현에 포함되므로 복잡한 구조의 데이터에 대한 질의를 효과적으로 표현할 수 있는 객체 지향 데이터베이스 질의어이다. VOQL은 귀납적으로 정의된 시맨틱을 갖는 그래프 기반 언어로 다양한 텍스트 경로식들을 그래프로 간결하게 표현할 뿐 아니라 복잡한 경로식의 시맨틱을 명확하게 전달한다. 그러나 기존의 VOQL은 모든 속성을 다중 값으로 가정하고 있고, 객체변수의 바인딩 개념을 시각화하고 있지 못하고 있다. 이로 인해 VOQL 질의문의 표현이 직관적이지 못할 뿐 아니라 이론적 확장이 쉽지 않다. 본 논문에서는 이러한 문제를 해결하도록 VOQL을 개선하고자 한다. 단일 값과 다중 값을 갖는 속성의 결과를 각각 시각요소와 서브블랍을 통해 시각화하고, 시각변수를 도입하여 객체 변수의 바인딩을 명시화하여 질의문의 시맨틱을 직관적이고, 명확하게 하고 있다.

## Improving Visual Object Query Language(VOQL) by Introducing Visual Elements and Visual Variables

Suk-Kyoon Lee †

ABSTRACT

Visual Object Query Language(VOQL) proposed recently is a visual object-oriented database query language which can effectively represent queries on complex structured data, since schema information is visually included in query expressions. VOQL, which is a graph-based query language with inductively defined semantics, can concisely represent various text-based path expressions by graph, and clearly convey the semantics of complex path expressions. However, the existing VOQL assumes that all the attributes are multi-valued, and cannot visualize the concept of binding of object variables. Therefore, VOQL query expressions are not intuitive, so that it is difficult to extend the existing VOQL theoretically. In this paper, we propose VOQL that improved on these problems. The improved VOQL visualizes the result of a single-valued attribute and that of a multi-valued attribute as a visual element and a subblob, respectively, and specifies the binding of object variables by introducing visual variables, so that the improved VOQL intuitively and clearly represents the semantics of queries.

### 1. 서 론

CAD/CAM 멀티미디어 시스템이나 지리정보 시스템

※ 이 논문은 1997년 한국학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

† 정 회 원 : 단국대학교 전산통계학과 교수

논문접수 : 1998년 10월 19일, 심사완료 : 1999년 3월 4일

템과 같은 응용분야는 복잡한 구조의 데이터 표현이 필수적으로 이러한 분야를 위해 객체 지향 데이터 모델이 제안되었다. 객체 지향 데이터 모델을 위한 질의어에 대한 연구는 주로 텍스트 기반의 질의어에 집중되어 있다[9]. 시각 질의어는 텍스트 기반의 질의어에 비해, 스키마 정보의 자연스러운 전달이 이루어지고

크 사용이 직관적이어서 오류의 빈도가 훨씬 감소한다는 장점이 여러 번 지적되어 왔다[10]. 일반화 계층(generalization hierarchy)와 집계화 계층(aggregation hierarchy)을 통해 스키마가 표현되는 객체 지향 데이터베이스에서는 시각 질의어의 역할이 관계형 모델에서 보다 더욱 중요해진다.

관계형 데이터베이스를 위한 시각 질의어에 비해, 객체 지향 데이터베이스를 위한 시각 질의어는 그다지 많지 않다. 대부분이 관계형 질의어 QBE[11]를 객체 지향 개념을 포함하도록 확장하거나, 그래프에 기초하고 있다. QBE에 기초한 객체 지향 질의어는 VQL[16], PESTO[3]를 들 수 있다. VQL[16]은 QBE를 확장한 형태로 템플레이트 기반의 언어이다. 경로식의 표현은 복수 개의 템플레이트와 이들 사이를 연결하는 레이블과 텍스트 변수를 통해 이루어진다. 이를 좀더 발전시킨 것이 PESTO인데, 이는 복합 객체 관계를 복수의 품과 연결에지를 사용하여 표현하고 있다. 그러나 품 사이의 관계는 단지 클래스간의 관계만을 반영하고 있어 복합 객체들 간의 접근 경로를 의미하는 경로식[7, 8]의 시맨틱을 제대로 전달하지 못하는 단점이 있으며 오히려 브라우징 시스템에 더 가깝다[9].

그래프에 기초한 시각 질의어들은 QBD\*[1], G+[5], ERC[6], GRAQULA[10] 등이 있다. 이 질의어들은 그래프 기반이나 관계형 또는 Entity Relationship (E-R) 모델에 기반하고 있어 객체 지향 데이터베이스 질의어로는 적합하지 않다. 그래프에 기초한 객체지향 질의어로는 VQL[15], QUIVER[4], VOQL[9]이 있는데, VQL은 다양한 그래프 기반의 구성 요소를 통하여 부정, 전체정량자, 재귀적 질의문, 스키마 질의 등을 처리하는 등 뛰어난 표현 능력을 지니고 있으나, 지나치게 텍스트 기반의 변수 사용 및 prolog 기반의 시맨틱과 기본적으로 관계형 모델을 기초하는 등 객체 지향 질의어로는 부족함이 많다. QUIVER는 풍부한 메뉴와 아이콘들로 사용하기 쉬운 사용자 인터페이스(user interface)를 제공한다. 그러나 QUIVER는 정형화된 이론적 기반에 정의되고 있지 못하며, 특히 경로식을 제대로 시각화하지 못하고 있다.

VOQL은 Harel의 Higraph[14]에 기초한 시각 질의어로 경로식을 객체 레벨에서 시각화함으로써 경로식의 시맨틱을 직관적으로 표현하고 있으며, 집합 관련 조건 연산을 벤다이어그램(ven Diagram)에 기초하여 표현함으로써 조건 연산의 표현이 직관적이고 간결하다. 뿐만 아니라 VOQL의 시맨틱(semantic)을 Object Ori-

ented Predicate Calculus(OOPC)[2]에 기초하여 그래프 기반의 시각 질의어로서는 귀납적(ductively)으로 정의하고 있어 이론적 기반이 확실하다는 장점이 있다. 그러나 VOQL에는 시맨틱의 정의와 단순화를 위해 모든 속성을 다중 값(multi-valued attribute)으로 가정하여 항상 집합으로 표현하고 있다(즉 단일 값인 경우는 싱글톤(singleton)으로 처리한다). 이러한 단순화는 경로식의 중간 및 최종 결과가 집합으로 표현되어 질의의 표현이 어색하며, 이는 동일한 경로식에 복수의 시각적 표현이 존재할 수 있는 문제점이 있다. 또한 VOQL은 귀납적 정의가 가능했었지만, 경로식의 표현에 있어 언어의 일부 구성 요소의 시맨틱의 정의가 복잡하여 질의의 표현이 직관적이지 못한 점이 있다.

본 논문에서는 이러한 문제를 해결하기 위해 속성 값의 표현에 다중 값과 단일 값을 허용하고, 이를 서로 다른 문법적 구성 요소로 시각화하여 위에서 언급한 문제들을 해결하고, VOQL 질의어의 구성 요소가 OOPC에서의 단일의 구성요소로 표현되도록 정의하여 시맨틱이 직관적이며 명확한 개선된 VOQL을 제시한다. 본 논문의 구성은 다음과 같다. 2장은 기존의 VOQL에 대한 설명으로, 2.1장에서는 VOQL의 구성요소의 정의를 위해 OOPC를 소개하며 2.2장에서는 기존의 VOQL의 구성 요소와 시맨틱을 설명한다. 2.3과 2.4장은 기존의 VOQL의 문제점을 제시한다. 3장에서는 2장에서 언급한 문제점을 해결하기 위해 개선된 VOQL을 소개하는 것으로 3.1장에서는 새로 도입된 시각 요소와 시각 변수의 개념을 정의한다. 3.2장에서는 개선된 VOQL에서는 앞에서 언급한 문제점들을 어떻게 해결되었는지를 설명하고, 끝으로 4장에서는 결론을 내리고 향후 연구방향을 제시한다.

## 2. VOQL 소개와 문제점

본 장에서는 VOQL과 개선된 VOQL의 시맨틱의 정의를 위해 OOPC와 [9]에서 제안된 VOQL를 소개하고 VOQL의 문제점을 제시한다. VOQL의 소개시, 정형화된 정의는 논리 전체에 필요한 경우에만 사용을 하고자 한다. 자세한 내용은 [9]를 참고하시오.

### 2.1 OOPC 소개

OOPC는 관계해석(relational calculus)을 객체 지향 데이터 모델을 위해 확장한 것으로 다음과 같은 구조

를 갖는다. 자세한 내용을 위해서는 [2]를 참조하시오.

**{목적절 | 범위절 ; 조건절}**

목적절(Target clause)는 추출할 정보를 나타낸다. 범위절(Range clause)는 v/c 형태로 작성되며, 여기서 v는 객체변수(object variable)이며 c는 객체변수 v가 바인딩(binding)될 클래스이다. 조건절(Qualification clause)은 원자술어(atomic predicate)나 정량술어(quantified predicate)로 구성된다. 원자술어인 경우 참, 거짓 또는  $t_1 \theta t_2$ 의 형태를 하게 된다. 여기서  $\theta$ 는 비교연산자로  $\in, \supset$  등 집합 연산자들도 포함되며,  $t_1$ 는 상수, 객체변수 v, 또는 객체 변수에 속성을 적용한 값  $v.att_i$ 이다. 정량술어인 경우 **v범위절(정량술어)**나 **범위절(정량술어)**로 형태를 취한다.

예로 "사장의 연봉이 \$300,000이상인 회사의 이름을 추출하라."라는 질의를 OOPC로 나타내면 아래와 같다.

```
OOPC 2.1 {x.name | x/company; ∃y/Employee((y = x.president) ∧ (y.Salary > $300,000))}
```

OOPC는 조건절에서 경로식을 사용할 수 있다. 다음 질의를 보자. "자동차(Vehicle)의 제조회사(Manufacturer)의 본사(Headquarter)가 있는 도시(City)가 'New York'인 자동차(Vehicle)의 이름(name)을 추출하라" 위 질의에 대한 OOPC는 아래와 같다.

```
OOPC 2.2 {x.name | x/Vehicle; x.manufacturer.headquarter.city = "New York"}
```

x.manufacturer.headquarter.city는 경로식으로 관계형 데이터베이스 시스템에서처럼 복수의 명시 조인으로 처리하는 것 보다는 훨씬 직관적이고 사용이 편하다. OOPC 2.3은 회사의 모든 부처들에 대하여 소속 직원들 중 60세인 직원이 있을 때 그 회사의 이름을 추출하는 질의이다.

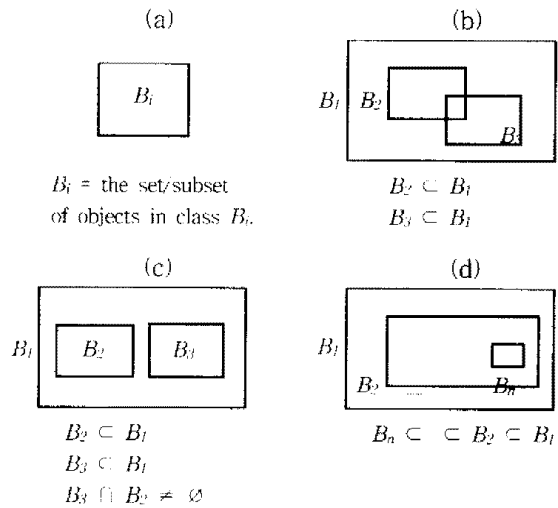
```
OOPC 2.3 {x.Name | x/Company; ∃y/x.Divisions. Employees(y.Age = 60)}
```

OOPC는 SQL의 질의문의 형태를 사용하여 목적절, 범위절, 그리고 조건절로 구분하는 데, 본 논문에서는 관계해석에서처럼 범위절과 조건절을 논리곱(AND)으로

표현한 것을 OOPC로 불러라고 부른다.

**2.2 VOQL 소개**

VOQL은 벤다이어그램에 기초하여 클래스 또는 클래스의 부분 집합을 표현하며, 또한 다양한 집합 관련 조건을 표현하고 있다. 블랍(blob)은 사각형으로 표기되며, 클래스 즉 클래스에 속한 모든 객체들의 집합을 명시한다. 서브블랍(subblob)은 블랍 안에 내포되어 있는 사각형으로 표기되며, 둘러싸고 있는 클래스에 속한 객체들의 집합에 대한 부분집합을 의미한다.



(그림 1) 블랍의 정의와 블랍과 서브블랍들간의 다양한 조건들의 표현

(그림 1)의 (a)에서  $B_1$ 는 블랍 또는 서브블랍의 시맨틱을 보여 주며, (b)-(d)에서는 VOQL 질의문에 사용될 수 있는 다양한 집합 관련 조건을 보여주고 있다. 이외에도 텍스트 기반 연산자의 사용이 허용된다. VOQL에서는 경로식을 시각화함으로서 질의문에 관련된 객체들의 구조를 그래프 형태로 보여주는 데 이를 위해 다음의 경로식의 정의를 사용하고 있다.

정의 1 (경로식): 임의의 클래스  $C_i$ 의 속성  $a_k$ 에 대하여 속성  $a_k$ 의 도메인을 클래스  $C_j$  라고 할 때, 속성  $a_k$ 는  $C_i$ 로부터  $2^{C_j}$ 로 매핑하는 함수로 정의한다. 이때, 경로식  $x.a_1.a_2...a_n$ 은 다음과 같이 정의된다. 단  $a_1$ 는 클래스  $C$ 의 속성이고  $x$ 는 클래스  $C$ 에 바인딩하는 객체 변수이다.

$$x.a_1 \dots a_n = \text{if } n = 1 \text{ then } a_1(x) \text{ else } \bigcup_{y \in x.a_1 \dots a_{n-1}} (y)$$

위의 경로식 정의에서 알 수 있듯이 모든 속성은 다중값 즉 집합을 리턴하는 함수로 정의되어 있다. 예를 들어  $x.Divisions.Employees$ 의 경로식에서  $x$ 가  $c1$ 이라는 Company의 객체로 바인딩될 때,  $Divisions(c1) = \{d1, d2\}$ ,  $Employees(d1) = \{e1, e2\}$ ,  $Employees(d2) = \{e3, e4\}$ 라고 하자. 이럴 경우, 경로식  $c1.Divisions = \{d1, d2\}$ 으로 표현되며, 경로식  $c1.Divisions.Employees = \{d1, d2\}.Employees = Employees(d1) \cup Employees(d2) = \{e1, e2, e3, e4\}$ 로 계산된다.

VOQL의 질의문의 예로 (그림 2)에는 OOPC 2.2에 대응하는 VOQL 질의 예가 주어져 있다. 모든 속성들은 다중 값, 즉 집합으로 가정하고 있기 때문에 조건 비교에 {"New York"}이라는 singleton이 사용되고 있다.

(그림 2)에서 각 블랍들은 Vehicle, Company, Address 클래스의 속한 객체들의 집합들을 각각 나타내며, 블랍 Company와 블랍 Address 안에 두 개의 서브블랍이 있다.

타원 모양의 투명한 스템프 블랍(stump blob)은 서브블랍의 특별한 형태로 원자적 속성(atomic attribute)를 나타낼 때 사용된다. (그림 2)에서 스템프 블랍은 {"New York"}이라는 값과 동등 비교를 나타내는 조건절에 참여한다. 불투명한 스템프 블랍과 레이블, 그리고 무방향 에지(undirected edge)는 프로젝트된 속성을 표시하는데 즉 목적절로서 사용된다. (그림 2)에서는 클래스 vehicle의 name 속성을 목적절로 지정하며 이는 OOPC 2.2의 목적절인  $x.name$ 을 의미한다.

정의 1의 경로식 정의에 기초하여 VOQL의 경로식의 구성요소들의 시맨틱이 정의된다. VOQL에서 경로식을 표현하기 위해서는 두 가지 형태의 방향 에지 즉 b-edge(binding edge)와 f-edge(flattening edge)가 사용되고 있다. b-edge와 f-edge는 각각 가는 선의 방향 에지와 굵은 선의 방향 에지로 표현되며, 정의 1의 경로식의 then 부분과 else 부분의 시맨틱에 각각 대응하는 구성 요소이다. 따라서 VOQL의 경로식은 1개의 b-edge와 0개 이상의 일련의 f-edge로 표현된다. 예를

들어, OOPC 2.2의 경로식  $x.manufacturer.headquarter.city$ 는 (그림 2)에서와 같이 하나의 b edge와 두 개의 f-edge로 구성된다. VOQL의 구성 요소 b-edge와 f-edge의 시맨틱은 b-edge와 f-edge라는 같은 이름의 함수들에 의해 정의되는 데, 이들 함수들의 정의는 다음과 같다.

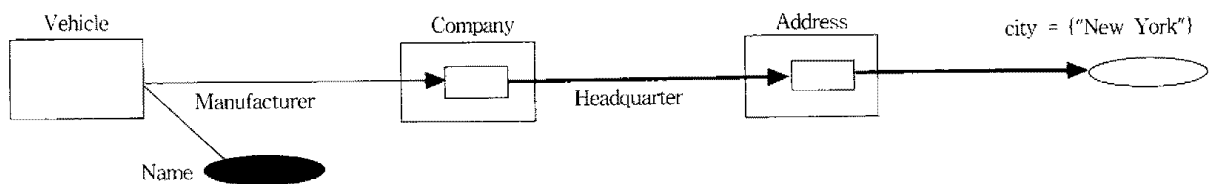
정의 2 (b-edge와 f-edge의 시맨틱): 블랍 (또는 서브블랍)  $B_i$ 에서 시작하여 속성  $attr_{ij}$ 로 레이블된 b-edge(또는 f-edge)는  $attr_{ij}$ 와  $B_i$ 를 입력받아 서브블랍  $B_j$ 을 출력하는 함수 b-edge (또는 f-edge)에 의해 시맨틱이 정의된다. 함수 b-edge (또는 f-edge)는 다음과 같이 정의된다:

$$b-edge(attr_{ij}, B_i) = attr_{ij}(x) \text{ where } x \in B_i$$

$$f-edge(attr_{ij}, B_i) = \bigcup_{y \in B_i} attr_{ij}(y)$$

b-edge는 방향 에지의 출발점이 접한 블랍(또는 서브블랍)에 대한 "객체변수의 바인딩" 개념과 방향에지에 레이블되어 있는 "속성의 적용"이라는 두 가지 개념을 의미한다. (그림 2)에서는 Manufacturer로 레이블되어 있는 b-edge는 OOPC 2.2의 "x/Vehicle"의 객체 변수 바인딩과 "x.Manufacturer"의 속성의 적용에 해당하는 부분 경로식을 나타내고 있다. 이 b-edge가 가리키는 서브블랍은 x.Manufacturer가 리턴하는 객체들의 집합 (이 경우에는 싱글톤)을 의미하게 된다. 연결된 한 개의 b-edge와 0개 이상의 일련의 f-edge들은 VOQL의 경로식을 나타내며, 이러한 VOQL 경로식은 b-edge와 f-edge의 시맨틱을 사용한 귀납적 증명방법을 통해 정의 1의 경로식과 동등함을 보일 수 있다[9].

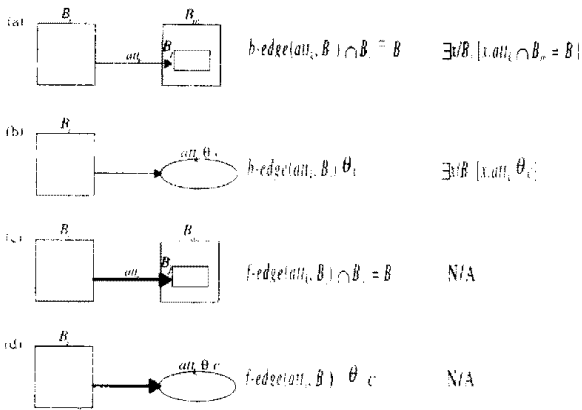
VOQL 질의문은 VOQL 포물라(formula)와 목적절로 구분된다. 목적절은 불투명한 스템프블랍과 레이블, 그리고 무방향 에지로 구성되며, VOQL 포물라는 VOQL 질의문에서 목적절을 제외한 나머지 부분이다. (그림 3)에는 VOQL 질의문을 구성하는 기본 포물라의 정의가 주어져 있다. 이들이 서로 결합하여 사용되면



(그림 2) OOPC 2.2에 대한 VOQL 질의문

서 VOQL 포블라를 구성하게 되면, VOQL 포블라의 시맨틱은 각 기본 포블라의 시맨틱이 귀납적으로 결합됨으로 정의되게 된다. (그림 3)의 (c)와 (d)는 단독으로 사용되지는 못하며 반드시 (a)와 더불어 사용되어야 하기 때문에 대응하는 OOPC 포블라의 정의가 주어지지 못하다. (그림 3)의 (a)와 (c)에서 속성  $att_k$ 의 도메인은 블랍  $B_m$ 인 경우에는 그 시맨틱이 각각  $b-edge(att_k, B_i) = B_j$ ,  $f-edge(att_k, B_i) = B_j$ 로 단순화되며 이를 단순화된 정의라 한다.

VOQL formula    Semantics of VOQL formula    OOPC formula

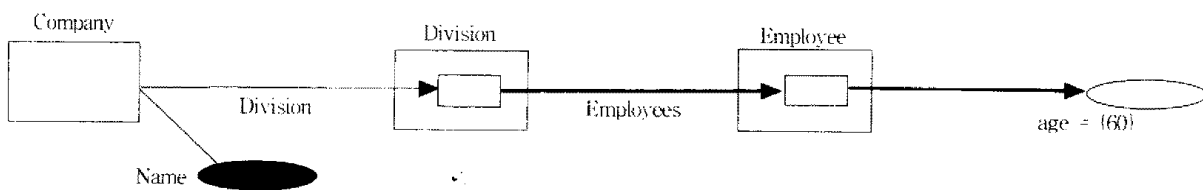


where  $\theta \in \{*, \times, \cap, \cup, \dots\}$  and N/A represents that the corresponding translation is not available.

(그림 3) VOQL 기본 포블라와 시맨틱의 정의

(그림 3)의 OOPC 포블라의 정의에서 알 수 있듯이, b-edge의 사용시 도입되는 객체 변수에는 존재 정량자가 사용된다. 또한 VOQL 포블라들은 논리곱 관계, 즉 AND 관계로 해석된다. (그림 2)의 VOQL 질의문에서 목적절을 제외한 부분은 다음과 같이 OOPC로 변환된다.

step 1 : (그림 2)의 VOQL의 포블라의 시맨틱 정의  
 $(b-edge(Manufacturer, Vehicle) = temp1) \wedge$   
 /\*그림 3(a)의 단순화된 정의 사용\*/  
 $(f-edge(Headquarter, temp1) = temp2) \wedge$



(그림 5) OOPC 2.3에 대한 VOQL 질의문

/\*그림 3(c)의 단순화된 정의 사용\*/  
 $(f-edge(City, temp2) = \{ "New York" \})$

/\*그림 3(c)의 단순화된 정의 사용\*/

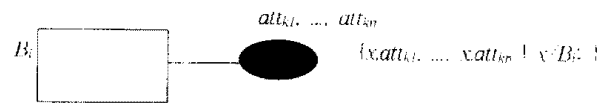
step 2 : OOPC 포블라의 변환

/\* OOPC 경로식의 정의로 변환 \*/

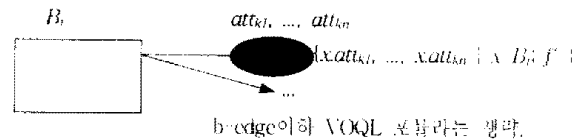
$\exists x/Vehicle [x.Manufacturer.Headquarter.City = \{ "New York" \}]$

VOQL의 목적절은 다양한 표현 형태가 있으나, (그림 4)에는 가장 대표적인 두가지 형태 만 보인다. 자세한 내용은 [9]를 참조하시오. VOQL 목적절은 블랍 또는 서브블랍에 접하여 표현하며, b-edge와 출발점을 공유할 때는 그 b-edge와 관련된 OOPC 포블라로부터 제약된다. (그림 4)의 (a)는 사실 상, VOQL 포블라가 존재하지 않는 경우로 해당 클래스에 속한 속성들만을 추출하는 경우이다. (그림 4)의 (b)는 블랍  $B_i$ 와 b-edge로 연결되는 VOQL 포블라가 가정되어 있는 경우로 목적절은 이 포블라에 의해 제약된다.

(a) 조건절이 없는 경우



(b) 조건절에 의해 제약받는 경우



$x$ 는 위의 b edge로부터 도입된 객체 변수이며,  $f$ 를 생략된 VOQL 포블라로부터 번역된 OOPC 포블라라고 할 때,  $f$ 는  $f$ 로부터 목적절에 관련된 객체 변수가 존재하는 범위절과 존재정량자를 제거한 OOPC 포블라이다.

(그림 4) VOQL 기본 포블라와 시맨틱의 정의

(그림 2)의 VOQL 질의문은 위에서 번역한 OOPC 포블라에 VOQL 목적절을 추가함으로써 OOPC 질의문으로 변환이 이루어진다. 이의 경우, VOQL 목적절에 언급된 객체 변수의 범위절과 존재 정량자를 생략

한 OOPC 포물라를 조건절에 기술하며 (그림 4)의 (b)에서 언급한 대로 목적절과 범위절을 추가함으로 표현이 이루어진다. 이때 (그림 2)의 VOQL 질의문은 다음과 같이 OOPC 질의문으로 번역이 이루어진다.

```
{x.name | x/Vehicle; x.manufacturer.headquarter.city = {"New York"}}
```

(그림 5)에서는 OOPC 2.3에 대응하는 VOQL 질의문이 주어져 있다. (그림 3, 4)의 정의대로 VOQL 질의문을 번역하여 보면, 다음과 같다. 자세한 절차는 [9]를 참조하시오.

```
step 1 : (b-edge(Divisions, Company) = temp1) ^
         (f-edge(Employees, temp1) = temp2) ^
         (b-edge(Age, temp2) = {60})
```

```
step 2 : ∃x/Company ∃y/x.Divisions.Employees(y.Age = 60)
```

```
step 3 : {x.Name | x/Company; ∃y/x.Divisions.Employees(y.Age = {60})}
```

2.3 속성 다중 값 가정으로 인한 질의 표현의 한계

VOQL에서는 모든 속성이 다중 값이라고 가정함으로써 속성의 결과는 실제로 단일 값이던, 다중 값이던 관계없이 하나의 표기인 서브블랍으로 표시되고 있다. 이는 (그림 2)에서의 VOQL의 질의에서와 같이 동등 비교가 경로식의 결과와 {"New York"}이라는 싱글톤 (singleton)과의 비교로 처리된 것에서 알 수 있듯이,

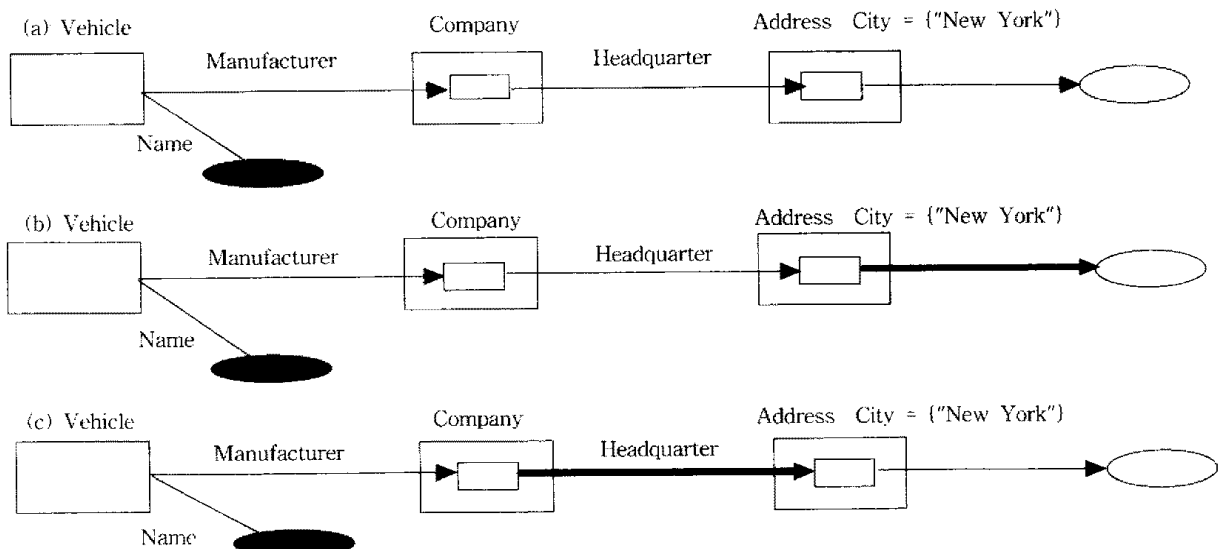
모든 비교는 집합간의 비교연산으로 밖에 표현할 수 없는 한계가 있다. 이는 이론적 단순화를 위하여 실용적인 측면이 희생된 것으로 실용적인 언어가 되기 위해서는 수정되어야 할 부분이다.

모든 속성을 다중 값으로 처리하는 데서 오는 또 하나의 문제는 서브블랍으로 표현되는 경로식의 중간 결과가 본래 단일 값 즉 싱글톤일 경우, 그 서브블랍으로부터 시작되는 b-edge나 f-edge가 같은 역할을 하게됨으로 같은 OOPC 경로식에 대해, 여러 VOQL 경로식이 존재하는 경우가 있다.

예를 들면, 2.1절의 OOPC 2.2 질의에서 만약 manufacturer, headquarter, city 모두가 단일 값 즉 싱글톤 속성들이라면 (그림 2)의 VOQL 표현과 (그림 6)의 3가지 VOQL 표현은 결국 같은 내용을 의미하게 된다.

(그림 2)와 6에서는 경로식상의 속성들이 모두 단일 값 즉 싱글톤이기 때문에 각 서브블랍에 적용된 b-edge나 f-edge의 결과는 같아진다. 즉, 싱글톤인 경로식 x.Manufacturer의 결과값에 b-edge "Headquarter"를 적용하는 것은 바인딩될 객체가 하나 밖에 없기 때문에 f-edge "Headquarter"를 적용한 결과와 같을 수밖에 없다. 마찬가지로 x.Manufacturer.Headquarter에 적용되는 b-edge나 f-edge도 같은 결과를 반환한다.

같은 질의에 대해 여러 개의 다른 표현은 SQL에서도 쉽게 발견된다. 그러나 SQL에서는 사용자의 편의를 위하여 다양한 표현 형태를 허용함에 비해 기존의 VOQL에서는 다중값의 가정으로 인한 즉 비현실적 가



(그림 6) OOPC 2.2에 대한 같은 의미의 다른 형태의 VOQL 질의들

정의로 발미암은 것으로 해결되어야 한다.

### 2.4 b-edge의 시맨틱의 문제

VOQL의 장점은 그 시맨틱이 로직에 기반을 둔 OOPC로 정의되어있고, VOQL의 질의문의 시맨틱은 그 언어의 구성 요소로부터 재귀적으로 시맨틱이 정의될 수 있는 점이다. 질의어의 시맨틱이 재귀적으로 명확하게 정의되기 위해서는 언어의 구성 요소가 번역 대상의 언어의 구성 요소와 1-1 관계에 있는 것이 바람직하다. 그러나, b-edge의 시맨틱은 다소 복잡하게 정의되어있다. 즉  $x/C$ 와 같이 표현되는 객체 변수 바인딩의 개념과  $x.Manufacturer$ 와 같이 속성의 함수적 적용(function application)의 개념이 b-edge라는 구성 요소로 시각화되어있다.

언어를 정의할 때, 각 구성 요소의 시맨틱은 분명하면서 불필요한 구성 요소는 도입하지 않는 것이 바람직하다. 그러나, b-edge의 경우는 두 가지 개념이 통합되어있어 그다지 직관적이지 못하며, 단일 값의 개념을 싱글톤을 나타내는 서브블랍이 아닌 새로운 구성 요소로 표현할 때, 또한 정량자의 개념을 시각화할 때, 지금의 b edge로서는 어려워 보인다.

## 3. 개선된 VOQL

이 절에서는 앞에서 언급한 문제점들을 개선된 VOQL을 제안한다.

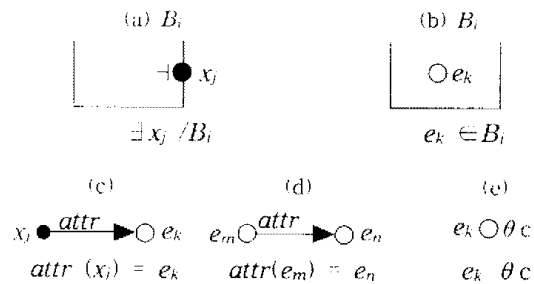
### 3.1 개선된 VOQL의 구성 요소

개선된 VOQL의 구성 요소는 블랍(blob), 서브블랍(subblob), 시각요소(visual element), 시각변수(visual variable), 방향에지(direct edge), 무방향에지(undirected edge), 불투명한 스템프 블랍(stump blob)으로 구성된다.

블랍과 서브블랍은 기존의 VOQL에서와 같다. 시각요소와 시각변수는 각각 투명한 작은 원과 불투명한 작은 원으로 표현된다. 서브블랍이 클래스에 속한 객체들의 부분 집합을 의미하는 반면, 시각요소는 하나의 객체를 표현한다. 블랍 또는 서브블랍 안에 존재하는 시각요소는 사용자가 정의한 클래스에 속하는 객체를 의미하며, 기본 영역(primitive domain)에 속한 객체는 블랍(서브블랍)에 내포되지 않은 시각요소로 표현한다. 시각변수는 언제나 블랍이나 서브블랍의 측면에 위치한다. 개선된 VOQL의 구성 요소의 시맨틱과

기본 포블라는 (그림 7)에 정의되어 있다.

기존의 VOQL에서는 b-edge가 객체 변수와 해당 클래스 내의 객체와의 바인딩과 그 객체에 대해 표기된 속성을 직용시키는 두 가지 역할을 하였다. 하지만 개선된 VOQL에서는 b-edge대신, 객체 변수의 객체와의 바인딩은 (그림 7)(a)와 같이 "블랍과 블랍의 측면에 위치한 시각변수의 관계"에 의해 표현되며, 해당 객체에 속성을 적용하는 것은 (그림 7)(c)에서처럼 방향 에지로 표현한다. 방향 에지는 적용되는 속성 이름으로 레이블링되며, 시각요소들 사이나, 시각변수로부터 시각요소로 연결된다.



시각변수와 시각요소는 레이블링되지 않으나 여기서는 설명의 편의를 위해 도입하였다. 다중 속성의 경우에는 시각요소 대신 서브블랍이 올 수 있으며, 이의 시맨틱은 유사하게 정의된다.

(그림 7) 개선된 VOQL의 구성요소의 시맨틱과 기본 포블라의 시맨틱 정의

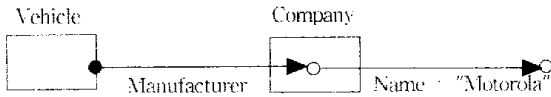
개선된 VOQL은 경로식 중간에 다중값이 오는 것을 허락하지 않는데, 이는 ODMG 2.0[12]의 표준과도 일치하는 것으로 다음의 정의에 기초한다.

정의 3 : (경로식) 임의의 클래스  $C_i$ 의 속성  $a_k$ 에 대하여 속성  $a_k$ 의 도메인을 클래스  $C_j$  라고 할 때, 속성  $a_k$ 는  $C_i$ 로부터  $C_j$ 로 매핑하는 함수로 정의한다. 이때, 경로식  $x.a_1.a_2 \dots .a_n$ ,은 다음과 같이 정의된다. 단  $a_1$ 는 클래스  $C$ 의 속성이고,  $x$ 는 클래스  $C$ 에 바인딩하는 객체변수이다.

$$x.a_1 \dots .a_n = a_n(a_{n-1}(\dots(a_1(x))))$$

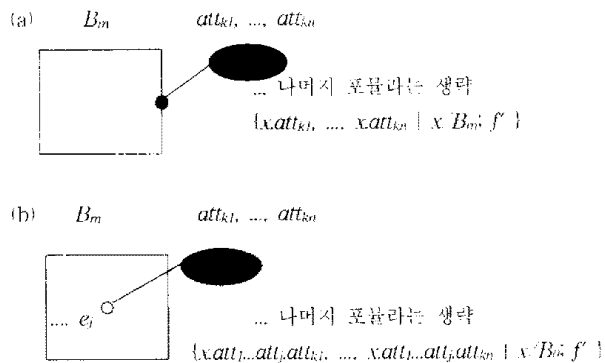
정의 3의 경로식의 정의는 (그림 7)의 정의를 통해 VOQL로 표현할 수 있는데, 이는 (그림 7)의 (c)와 일련의 (d)를 적용하여 표현이 이루어진다. (그림 7)의 정의를 통해 표현된 개선된 VOQL의 포블라의 예가 (그림 8)에 주어져 있다. (그림 8)의 VOQL 질의문은 다음과 같은 과정을 통해 OOPC 포블라로 번역된다.

- step 1. 그림 7)의 (c)와 (d)의 적용을 통해 Name (Manufacturer(x))의 생성.
- step 2. 정의 3을 통해, x.Manufacturer.Name의 생성.
- step 3. 그림 7)의 (a)와 (e)의 적용으로  $x/Vehicle$   $x.Manufacturer.Name="Motorola"$  생성.



(그림 8) 개선된 VOQL 포물라의 예

개선된 VOQL에서 목적절은 시각요소와 시각변수의 도입으로 (그림 9)와 같이 정의된다.



(a)  $B_m$   $att_{k1}, \dots, att_{kn}$   
 ... 나머지 포물라는 생략  
 $\{x.att_{k1}, \dots, x.att_{kn} \mid x/B_m f\}$

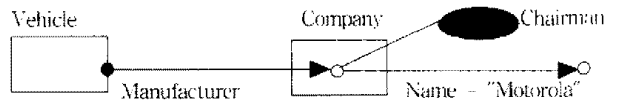
(b)  $B_m$   $att_{k1}, \dots, att_{kn}$   
 ... 나머지 포물라는 생략  
 $\{x.att_1, \dots, att_i, att_{k1}, \dots, x.att_1, \dots, att_i, att_{kn} \mid x/B_m f\}$

(f)을 생략된 VOQL 포물라로부터 번역된 OOPC 포물라라고 한 때,  $f$ 는  $f$ 로부터 목적절에 관련된 객체 변수가 존재하는 범위절과 존재정량자를 제거한 OOPC 포물라이다. (b)에서 시각요소  $e_i$ 는 OOPC 경로식  $x.att_1, \dots, att_i$ 으로 번역될 때, 목적절에 이의 내용이 반영되며, 범위절의  $B_m$ 는 시각요소  $e_i$ 가 연결되어 있는 시각 변수가 바인딩되는 블랍을 의미한다.

(그림 9) 개선된 VOQL에서 목적절의 시맨틱

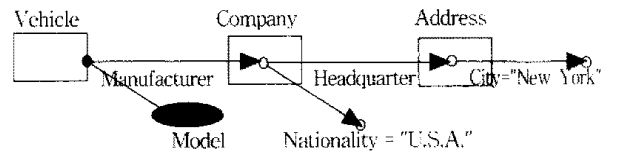
목적절의 정의에서 범위절과 존재정량자의 처리는 기존의 VOQL과 유사하다. 그림 9) (a)는 목적절이 시각변수에 연결되어 있는 경우로 (그림 4) (b)의 목적절의 시맨틱과 유사하다. 그림 9) (b)의 경우는 목적절이 시각요소에 연결되어 있는 경우이다. 그림 10의 VOQL 질의문은 (그림 9) (b)의 경우에 해당하는 것으로 (그림 8)의 VOQL 포물라에 목적절이 추가되었다. 그 의미는 "그 제조사의 이름이 Motorola인 자동차가 있을 때 그 회사가 위치한 회장이름을 추출하시오"으로 이를 OOPC로 번역하면 다음과 같다.

OOPC 3.1.  $\{x.Manufacturer.Chairman \mid x/Vehicle; x.Manufacturer.Name="Motorola"\}$ .



(그림 10) 개선된 VOQL 질의문의 예

Vehicle 블랍의 오른쪽 면에 위치한 시각변수는 Vehicle내의 한 객체에 바인딩된다. 방향에지는 속성의 다중 값/단일 값 여부에 관계없이 사용되며, 단일 값인 경우는 방향에지의 끝에 시각요소가 오게되며, 다중 값인 경우는 서브블랍이 오게 된다. 방향에지는 시각 변수나 시각요소에서 시작된다. VOQL 경로식은 시각 변수로부터 시작하여 일련의 방향에지와 시각요소로 연결되어 있는 트리로 정의된다.



(그림 11) 개선된 VOQL 질의문의 예.

(그림 11)에서 VOQL 경로식은 시각변수로부터 시작하여 동등 비교 연산에 참여하고 있는 두 개의 시각요소들이 리프(leaf)로 표현되는 트리로, 이는 OOPC의 두 경로식을 2차원 공간상에 표현한 것이다. 즉 트리 형태로 표현된 VOQL 경로식은  $x.Manufacturer.Nationality$ 과  $x.Manufacturer.Headquarter.City$ 라는 OOPC 경로식을 의미한다. 이 예에서 알 수 있듯이, 시각요소는 경로식의 중간 과정의 결과를 표현한 시각적 표현으로 시각요소는 새로운 경로의 출발이나 조건 연산에 참여할 수 있어 질의의 표현이 간결해지며 직관적으로 그 의미를 이해할 수 있게 한다. 목적절인 " $x.Model$ "은 시각변수의 도입이라는 차이를 제외하고는 기존의 VOQL과 흡사하다. 블랍 Vehicle에 바인딩되는 시각 변수와 "Model"이라고 표기된 스템프 블랍, 그리고 이들을 연결하는 무방향에지로 표현된다. 이를 OOPC로 번역하면 다음과 같다.

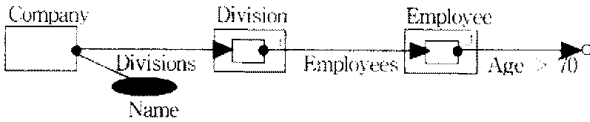
OOPC 3.2.  $\{x.Model \mid x/Vehicle ; x.Manufacturer.Nationality = "U.S.A." \wedge x.Manufacturer.Headquarter.City = "New York"\}$ .

다음과 같은 질의문을 생각해 보자 : "나이가 70 이



명칭 식원이 있는 부지의 회사의 이름을 나타사오" 등의 개선편 VOQL로 표현한 것이 (그림 12)에 주어져 있으며, 이를 OOPC로 번역하면 다음과 같다.

OOPC 3.3 {x.Name| x.Company: ±y.x.Divisions( ±z.y. Employees(z.Age>70))}



(그림 12) VOQL 질의문의 예

(그림 12)에서는 속성 Divisions과 Employees가 다중 값이기 때문에 서브블랍이, 속성 Age는 단일 값이기 때문에 시각요소가 사용되었다. 세 개의 시각변수가 사용되어 각각 Company 블랍, Division에 내포되어 있는 서브블랍, 그리고 Employee안에 내포되어 있는 서브블랍에 객체 변수의 바인딩이 이루어짐을 표현하고 있다. 이를 OOPC로 나타내면, 각각 x/Company, y/x.Divisions, z/y.Employees로 표현된다. 시각 변수는 목적설의 일부로 사용되는 경우를 제외하면 존재 정량자가 명시적으로 존재하여 존재 정량조건의 표현이 이루어진다. 이는 OOPC 3.3에서 ±y/x.Divisions ±z/y.Employees에 대응하는 VOQL 표현이다.

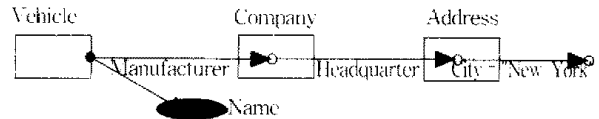
### 3.2 기존 VOQL의 문제들의 해결

2.3장에서 기존 VOQL에서 모든 속성을 다중 값으로 표기함으로써 발생하는 질의 표현의 한계를 언급하고, 이의 예로 (그림 6)에서 OOPC 2.2의 내용이 4가지 서로 다른 VOQL 질의문으로 표시될 수 있음을 보였다. 이러한 문제는 앞에서 언급했던 것처럼 단일 값과 다중 값을 서브블랍이라는 같은 기호를 사용함에 기인한다. 또한 2.4장에서는 b-edge가 객체 변수의 바인딩과 속성의 함수적 적용이라는 이질적인 시맨틱을 갖고 있으므로 오는 문제점을 지적했었다.

모든 속성을 다중 값으로 표기함으로써 발생하는 문제를 해결하기 위해 개선된 VOQL에서는 단일 값과 다중 값 각각에 대해 서로 다른 기호를 사용하였다. (그림 11)에서와 같이 각 방향에지의 결과 값(즉 속성)이 단일 값인 경우에는 시각요소로 표기하고, (그림 12)에서와 같이 방향에지의 결과 값(속성)이 다중 값인 경우 각 서브블랍으로 표기된다. 따라서 기존의 VOQL,

에서와 같은 질의 질의문의 아무 시각요소로 표현됨으로 (그림 6)에서와 같은 다양한 질의문이 생길 수 없다.

OOPC 2.2에서처럼 속성 manufacturer, headquarter, city 이 단일 값인 경우 개선된 VOQL에서는 (그림 13)과 같은 하나의 VOQL 질의문으로 표현된다.



(그림 13) OOPC 2.2에 대한 개선된 VOQL 질의문

위의 설명에서 그리고 (그림 13)에서 예시하였듯이 속성이 단일 값인 경우는 시각요소, 다중값인 경우는 서브블랍이 사용됨으로 기존의 VOQL에서처럼 단일 값을 싱글톤으로 표현할 필요가 없어짐으로 2.3장에서 언급한 문제는 개선된 VOQL에서는 발생하지 않는다.

b-edge의 시맨틱 문제는 b-edge의 시맨틱을 시각 변수와 방향에지로 구별하여 표현함으로써 해결되었다. 시각변수의 도입은 정량자의 표현을 명시적으로 가능하게 함으로 앞으로 전체 정량자의 도입 시에 발생할 수 있는 스코프(scope)의 문제의 처리의 기초를 마련하였다. 텍스트 기반 언어에서는 정량자들의 순서에 따라 시맨틱의 변화가 주어지는 데 그래프 기반의 시각 언어에서는 정량자의 순서를 표현할 수가 없다. 이는 스코프 연산자의 도입을 통해 처리가 가능한 데, 본 논문의 범위를 넘는 내용이다. 제각각 질의는 별도의 연산자의 도입을 통해 가능하리라고 보는 데, 이는 별도의 논문에서 언급하겠다.

개선된 VOQL에서는 f-edge의 개념에 대한 언급을 하지 않고 있는 데, 이는 ODMG 2.0[12]의 표준을 반영한 결과이다. 즉, ODMG 2.0에서는 다중값은 경로식의 중간에서는 쓸 수 없고, 경로식의 마지막에만 허용하고 있다. f edge을 도입하여도 이 논문의 결과에는 아무런 영향이 없으니, 설명을 간결하게 하기 위하여 제외하였다.

## 4. 결론 및 향후 연구 과제

객체 지향 데이터 모델은 복잡한 스키마의 구조를 갖는 응용 분야를 위한 데이터 모델로 그 구조적인 특성은 그래프 기반 질의어의 필요성을 제기하고 있다. 그러나 기존의 그래프 기반 질의 언어는 정형화된 이

분석 기반에 기초하고 있지 않은 뿐 아니라 경로식의 표현이 자연스럽지 못하고 이에 관련된 조건의 표현도 직관적이지 못한 측면이 있다.

본 논문에서는 이러한 문제를 해결하고자 제안되었던 그래프 기반 질의어인 VOQL을 소개하며 이의 문제점들을 해결하여 그래프 기반 질의어를 위한 이론적 기반을 다지고 있다. VOQL은 그래프 기반 시각질의어 중에서는 최초로 귀납적으로 정의된 시맨틱을 갖는 언어이다. 이는 텍스트 기반에서 발생될 수 있는 다양한 형태의 경로식을 그래프로 간결하게 표현함으로써 경로식의 시맨틱을 직관적으로 명확하게 전달하며, 집합 관련 연산을 벤다이어그램에 기초하여 자연스럽게 처리하는 장점이 있다.

VOQL은 이러한 장점들에도 불구하고 몇 가지 해결해야 할 문제들이 있다. 간결한 이론적 전개를 위해 모든 속성을 다중 값 즉 집합으로 가정함으로써 조건의 표현이 어색하여지는 등 질의 표현의 한계가 존재하였고, 또한 같은 내용의 질의문이 여러 형태의 VOQL 질의문으로 표현되는 문제가 있다. 뿐만아니라, b-edge의 개념이 객체변수의 바인딩 개념과 속성의 적용이라는 개념과 동시에 표현함으로써 직관성이 떨어지고 정량자의 개념을 시각화할 때 표현의 어려움을 주고 있다. 본 논문에서는 이러한 문제들을 시각요소와 시각변수의 개념의 도입을 통해 해결하고 있다.

개선된 VOQL이 시각요소와 시각변수의 도입을 통해 앞에서 정의한 속성의 다중 값 가정의 문제와 b-edge의 문제를 해결했지만 몇 가지 제약이 여전히 남아있다. 개선된 VOQL도 아직은 논리곱 질의 언어(conjunctive query language)의 범주를 넘지 못한다. 따라서 논리합(disjunction)의 표현, 부정(negation) 표현과 전체 정량 조건(Universal quantification) 표현이 필요하다. 현재 VOQL의 처리기의 프로토타입이 개발 중이며 이들 문제에 대한 연구도 진행 중이다.

### 참 고 문 헌

- [1] Angelaccio, M., Catarci, T., and Santucci, G., "QBD\*: A Graphical Query Language With Recursion," *IEEE Trans. on Software Engineering*, Vol.16, No.10, pp.1150-1163, October 1990.
- [2] Bertino, E. et al., "Object-Oriented Query Language : The Notion and the Issues." *IEEE Trans. on Knowledge and Data Engineering*, Vol.1, No. 3, pp.223-237, June 1992.
- [3] Carey, M., Haas, L., Maganty, V., and Williams, J., "PESTO : An Integrated Query/Browser for Object Databases," in *Proc. Intl. Conf. on Very Large Data Bases*, pp.203-214, 1996.
- [4] Chavda, M., and Wood, P., "Towards an ODMG-Compliant Visual Object Query Language," In *Proc. Intl. Conf. on Very Large Data Bases*, pp.456-465, Athens, Greece 1997.
- [5] Cruz, I., Mendelzon, A., and Wood, P., "Graphical Query Language supporting Recursion." In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pp.323-330, 1987.
- [6] Czejdo, B., Elmasri, R., and Rusinkiewicz, M., "A Graphical Data Manipulation Language for an Extended Entity-Relationship Model," *IEEE Computer*, Vol.23 pp.26-36, Mar. 1990.
- [7] Frohn, J., Lausen, G., and Uphoff, H., "Access to Objects by Path Expressions and Rules," In *Proc. the 20th VLDB Conference*, pp.273-294, 1994.
- [8] Kifer, M., Kim, W., and Sagiv, Y., "Querying Object-Oriented Databases," In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pp.393-402, San Diego, CA, 1992.
- [9] Jeonghee Kim, Tae-sook Han, Suk Kyoan Lee "VOQL : A Visual Object-Oriented Database Query Language For Visualizing Expressions," accepted for publication in *International Journal of Computer Systems Science and Engineering*, 1998.
- [10] Sockut, G. H., Burns, L. M., Malhotra, A., and Whang, K-Y., "GRAQULA : A Graphical Query Language for Entity-Relationship or Relational Databases," *Data and Knowledge Engineering*, Vol.11, pp.171-202, 1993.
- [11] Zloof, M., "Query By Example," *IBM Systems Journal*, Vol.16, pp.324-343, 1977.
- [12] Cattell, R.G., editor, *The Object Database Standard : ODMG 2.0* Morgan Kaufmann Pub-

lshers, San Francisco 1997.

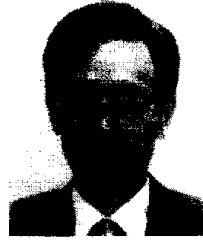
[13] Gyssens, M., Paredaens, J., Van den Bussche, J., and Van Gucht, D., "A Graph-Oriented Object Database Model," *IEEE Trans. on Knowledge and Data Engineering*, Vol.6, No.4, pp.572-586, August 1994.

[14] Harel, D., "On Visual Formalisms," In *Comm. of the ACM*, Vol.31, No.5, pp.514-530, 1988.

[15] Mohan, L. and Kashyap, R. L., "A Visual Query Language for Graphical Interaction With Schema-Intensive Databases," *IEEE Trans. on Knowledge and Data Engineering*, Vol.5, No.5, pp.843-858, 1993.

[16] Vadaparty, K., Aslandogan, Y. A., and Ozsoyoglu, G., "Towards a Unified Visual Database Access," In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pp.357-366, 1993.

### 이 석 균



e-mail : sklee@dankook.ac.kr

1982년 서울대학교 경제학 학사

1990년 University of Iowa 전산  
과학 석사

1992년 IEEE 8th International  
Conference on Data Engi-  
neering에서 최우수 논문  
상 수상

1993년 University of Iowa 전산과학 박사

1993년~1997년 세종대학교 정보처리학과 전임강사

1997년~현재 단국대학교 전산통계학과 조교수

관심분야 : 데이터 모델링, 데이터베이스에서 불완전  
정보관리, 객체지향 데이터베이스 시스템,  
데이터베이스 질의어, 다중처리기에서의 실  
시간 스케줄링, 데이터웨어하우스