論文2001-38CI-5-1

Stochastic VOQL : 시각적 객체 질의어

(VOQL : A Visual Object Query Language)

金 正 姬 *, 曺 完 燮 **, 이 석 균 ***, 황 규 영 ****

(Jeong-Hee Kim, Wan-Sup Cho, Suk-Kyoon Lee, and Kyung-Young Whang)

요 약

객체지향 데이터베이스를 위한 시각적 질의어(visual query language)의 설계에서 복잡한 질의 조건을 간단하고도 직관적으로 표현할 수 있도록 지원하는 것이 중요한 연구과제가 되고 있다. 본 논문에서는 시각적 객체지향 질의어인 VOQL(Visual Object Query Language)을 제안한다. VOQL은 그래프와 밴다이어그램(Venn Diagram)을 결합한 시각적 표현 기법을 사용하여 객체지향 데이터베이스의 스키마(schema)와 질의어를 하나의 통일된 시각적 표시법으로 표현하며, 객체지향 질의어에 포함된 한정된 경로식(quantified path expression), 집합 연산자, 상속 등의 객체지향 특성도 간단한 시각적 표시법을 이용하여 표현한다. VOQL의 가장 큰 특징은 기존의 시각적 객체지향 질의어들에 비하여 간단하고 직관적인 구문과 의미를 가지며, 뛰어난 질의 표현력을 가진다는 점이다.

Abstract

Expressing complex query conditions in a concise and intuitive way has been a challenge in the design of visual object-oriented query languages. We propose a visual query language called VOQL (Visual Object-oriented Query Language) for object-oriented databases. By employing the visual notation of graph and Venn diagram, the database schema and the advanced features of object-oriented queries such as multi-valued path expressions and quantifiers can be represented in a simple way. VOQL has such good features as simple and intuitive syntax, well-defined semantics, and excellent expressive power of object-oriented queries compared with previous visual object-oriented query languages.

* 正會員, BNC Intelligence
(BNC Intelligence)
** 正會員, 충북대학교 경영정보학과
(Department of MIS, Chungbuk National University)
*** 正會員, 단국대학교 전산통계학과
(Dept. of Computer Science and Statistics)
**** 正會員, KAIST 전산학과
(Department of Computer Science and Advanced Information Technology Research Center, KAIST)
※ 본 연구는 첨단정보기술연구센터를 통하여 한국과학재단의 지원을 받았음.
接受日字:2000年7月26日, 수정완료일:2001年6月11日

1. 서 론

SQL과 같은 텍스트 기반 질의어에 비하여 시각적 질의어(visual query language)는 사용자에게 편리한 데이터베이스 인터페이스를 제공하며, 질의 작성 과정에 오류 발생 빈도를 줄여준다[Soc93]. 텍스트 기반 질의어의 사용자는 질의를 할 때 질의어 구문 뿐 아니라 데이터베이스 스키마와 객체간의 관련성을 기억해야 하지만, 시각적 질의어의 사용자는 일반적으로 다이어그램 형태로 표시된 데이터베이스 스키마를 보면서 질의에 필요한 시각적 요소들을 선택한 후 간단한 형태

의 질의 조건만을 부과함으로써 질의를 작성할 수 있으므로 스키마와 질의어 구문을 기억할 필요가 없기 때문이다 [Soc93]. 텍스트 기반 질의를 사용하는 경우에 70 % 정도의 질의 작성 오류가 발생하였으나, 시각적 질의를 사용하는 경우에는 오류 발생 비율이 20 % 정도로 낮아진다는 보고도 있다[Wha92,Soc93].

시각적 데이터베이스 질의어에 관한 지금까지의 연구는 ER(Entity-Relationship) 데이터베이스와 관계(Relational) 데이터베이스에 집중되어 이루어졌지만[Ozs89,Wha92], 최근 들어 복잡한 데이터 구조를 모델링할 수 있는 객체지향 데이터베이스의 사용이 급증함에 따라 객체지향 데이터베이스에서의 시각적 객체지향 질의어에 관한 연구가 활발히 이루어지고 있다[Sta91,Uni91,Cru92, Bal96]. 그러나, 기존의 객체지향 데이터베이스를 위한 시각적 질의어에서는 그 표현력이 간단한 질의를 작성하는 수준으로 한정되고 있으므로 객체지향 질의어의 다양한 특성들 (features)을 모델링하는데 충분하지 못하다. 구체적으로 살펴보면 기존의 시각적 질의어는 클래스 구성 계층구조(class composition hierarchy) 상에서 조건을 표시하는데 널리 사용되는 경로식(path expression)[Kif92]을 직관적으로 표현하지 못하고 있다. 특히, 한정자가 첨가된 경로식(quantified path expression)[Ber92,Kif92]은 시각적 객체지향 질의어가 지원해야 하는 중요한 특성임에도 불구하고 대부분의 경우 지원하지 않고 있으며, 메소드(method)나 상속(inheritance) 등과 같은 객체지향 질의어의 고유 특성도 지원되지 못하고 있다.

UniSQL을 비롯한 대부분의 상용 객체지향 데이터베이스 시스템에서 제공하는 시각적 질의어들은 하나의 클래스에 대하여 간단한 조건을 제시하는 정도이다[Uni91]. 참고문헌[Gys90,Par95] 등에서 그래프를 사용한 객체지향 질의어를 제안하였지만 경로식이나 한정 경로식(quantified path expression)을 다루지 않았으며, 그래프 자체에 객체지향 데이터베이스에서의 중요한 특성인 데이터 추상화(data abstraction) 개념이 포함되지 않음으로써 스키마와 질의어의 복잡도가 증대된다. VQL[Moh93]은 자체 객체 모델을 기반으로 다양한 그래피칼 프리미티브(graphical primitives)를 사용하여 부정(negation), 전체 한정자(univeral quantification) 및 회귀성(recursive) 질의가 가능한 높은 표현력을 제공한다. 그러나 VQL로 작성된 질의는 관계형 모델을 기반으로 한 논리 언어로 번역되므로 조인 조건(join condition)을 위해 텍스트 변수(textual variable)를 사용하는 등 관계 모델의 제한 사항을 그대로 가진다. 또한, 경로식의 각 접근 링크(access link)를 표현하기 위해 각 템플리트 창(template window)에 텍스트 변수를 도입하여 변수들간의 조인 조건을 사용하므로 경로식을 시각적으로 적절히 표현하지 못한다. 참고문헌 [Bal96,Sta91]에서는 시각적 관계 질의어인 QBE(Query By Example)[Zlo77]를 확장한 아이콘 기반(icon-based)의 시각적 객체지향 질의어를 제안하였으며, 특히 전체 한정자(universal quantifier)를 시각적으로 표현할 수 있도록 하였다. 그러나 여기서는 그래프 형태의 객체지향 스키마를 보면서 그래프와는 다른 시각적 도구인 아이콘을 사용하여 질의를 표현하므로 질의 작성에서 두 가지 서로 다른 시각적 도구(그래프와 아이콘)가 사용되는 복잡성을 가진다.

본 논문에서는 객체지향 데이터베이스에서 VOQL (Visual Object Query Language)이라는 새로운 시각적 질의어를 제안한다. VOQL은 Higraph [Har88]라고 하는 통일된 시각적인 표시법 (notation)을 사용하여 객체지향 데이터베이스의 스키마와 질의어를 표현한다. Higraph는 기존의 그래프(graph)로부터 집합 이론에서 사용되는 시각적 표시법인 벤 다이어그램(Venn Diagram)을 결합하여 확장한 것으로 객체간의 집단화 관계(aggregation) 뿐 아니라 클래스간의 상속(inheritance) 관계를 자연스럽게 표시한다. VOQL에서 집단화 관계는 그래프로, 상속 관계는 벤 다이어그램으로 각각 표현한다. 특히, 벤 다이어그램에서 서브 노드는 상속 관계에서 서브 클래스를 표현하며, 서브 노드는 사용자의 명시적 요구가 있는 경우에만 표현함으로써 상속 관계로 인한 시각적 복잡도를 단순화한다. 사용자는 Higraph 형태의 객체지향 데이터베이스 스키마를 보면서 질의에 필요한 시각적 요소들(노드와 경로)을 선택한 후 간단한 질의 조건을 추가함으로서 질의 작성을 완성할 수 있다. VOQL에서는 시각적 질의어에서 다루기 어려운 문제로 알려진 한정자(quantifier)를 집합들 사이의 관계를 시각적으로 표시하는 벤 다이어그램 형태로 간단히 표현한다. 또한, 길이가 2 이상인 경로식(path expression)을 포함한 질의와 메소드를 포함한 질의에 대해서도 시각적인 표현 방법을 제시한다.

본 논문의 목적은 VOQL의 완전한 구문(syntax)과 의미(semantic)를 제안하는 것이 아니라 그래프와 벤다이어그램을 결합한 Higraph를 사용하여 객체지향 데이

터베이스의 스키마와 질의어를 모두 표현할 수 있다는 VOQL의 기본 아이디어를 제안하는데 있다. 특히, 대표적인 상용 객체지향 질의어인 XSQL[Kif92]로 질의를 작성할 때 가장 중요한 요소인 경로식(path expression)에 대한 조건을 VOQL에서의 시각적 경로로 쉽게 변환하여 표현할 수 있음을 입증한다. 현재 표준 객체 질의어로써 ODMG 2.0과 SQL3이 제안되고 있으며, 여기서도 관계 질의와 비교하여 대표적인 특성은 경로식의 사용과 집합값 속성에 따른 집합 연산자의 사용이다. 따라서 본 연구의 결과는 UniSQL의 XSQL 뿐 아니라 표준 객체 언어인 ODMG나 SQL3를 시각화하는데도 사용될 것으로 판단된다. 그리고, 본 논문에서는 SQL 명령의 일종인 삽입, 삭제, 갱신에 관한 내용은 다루지 않는다. 그러나, 데이터베이스에서 삽입, 삭제, 갱신 명령은 대부분 질의와 결합되어 사용되므로 본 연구의 결과는 이들 명령의 시각화에서 질의어 부분을 표현하는데 그대로 사용될 수 있을 것이다.

논문의 구조는 다음과 같다. 제 2 장에서는 객체지향 데이터베이스와 질의어에 관한 기존 결과를 요약한다. 제 3 장에서는 객체지향 데이터베이스 스키마를 Higraph로 모델링하고, 이를 기반으로 시각적 질의어인 VOQL을 제안한다. 제 4 장에서는 VOQL의 표현력을 XSQL과 비교분석한다. 제 5 장에서는 VOQL의 구현 결과를 소개하고, 제 6 장에서 결론을 맺는다.

2. 객체지향 질의어의 특성 및 시각적 모델링

본 장에서는 VOQL의 특징을 기술하는데 필요한 객체지향 데이터 모델을 설명하고, 이를 지원하는 텍스트 기반 객체지향 질의어의 특징들을 기술한다. 또한, 객체지향 질의어를 시각적으로 모델링하는데 있어 한계점을 기술한다.

실세계의 모든 정보는 객체로 모델링되며, 동일한 성질(속성과 메소드)을 갖는 객체들은 클래스로 분류된다. 임의의 두 클래스는 IS-A 관계나 어소시에이션(association) 관계1를 가질 수 있다. 그림 1은 본 논문에서 예제로 사용하는 객체지향 데이터베이스의 스키마를 OMT(Object Modeling Technique)[Rum90] 다이어그램으로 표시한 것이다. 그림에서 실선 사각형은 객체들의 집합인 클래스를 나타내며, 클래스에 정의된 속성들은 사각형 안에 표시하지만 필요한 경우가 아니면 복잡도를 줄이기 위하여 생략한다. 클래스 간의 어소시에이션 관계는 링크(link)로 표시하며 필요한 경우에는 롤(role)을 표시한다. 관계의 매핑수(mapping cardinality)는 경로의 끝에 불릿(bullet : 검은점)으로 1-N 관계의 N 측 노드를 표시한다. 한편, 클래스 간의 상속 관계는 중간에 삼각형이 있는 링크로 표시한다. 예를들어, 그림 1에서 Person과 Employee를 연결하는 링크는 상속 관계를 나타낸다.

대부분의 객체지향 질의어는 관계 질의어인 SQL (Structured Query Language)로부터 객체지향 개념을 지원하도록 확장한 형태이다. 객체지향 질의어에 대한 표준 규격이 국제 표준화 기구(ISO)에서 제정중에 있으므로 여기서는 기존의 객체지향 질의어 중에서 표현력이 비교적 뛰어난 XSQL [Kif92]을 가정한다. XSQL은 다음 식 (1)과 같은 확장된 경로식(extended path expression)2[Kif92]과 한정자(quantifier)를 사용하여 질의를 작성할 수 있다.

$$C_0\ P_1[\ C_1].\ \ P_2[\ C_2].\ \ P_3 \ldots P_n[\ C_n] \tag{1}$$

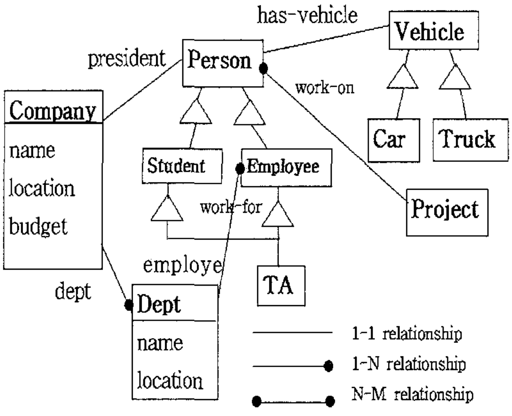


그림 1. OMT로 표시된 객체지향 데이터베이스 스키마.

식 (1)에서 Ci는 클래스 이름을, Pi는 도메인이 Ci인 속성 혹은 메소드를 나타낸다. 객체지향 데이터베이스에서는 속성을 인자(argument)가 없는 메소드로 간주

1. 참고문헌[Rum90]에서는 집단화(aggregation) 관계와 어소시에이션(association) 관계를 구분하였으나 여기서는 이들을 구분하지 않고 통틀어 어소시에이션 관계로 표시함.

2. 셀렉터(selector)를 사용하여 속성의 도메인을 특정 클래스로 한정하거나 메소드를 사용하는 경로식.

하여 메소드의 특별한 경우로 취급하므로 메소드와 속성은 서로 구분없이 사용된다. 각 Pi 다음의 괄호 '['와 ']'는 셀렉터(selector)로써 Pi의 도메인을 괄호안의 클래스로 한정하기 위함이다.

다음 질의어는 그림 1의 객체지향 데이터베이스에 대하여 XSQL로 작성한 질의어의 예를 보여준다.

(질의 1) Employee 중에서 35세 이상이며 빨간색 car를 가진 사람을 반환하시오.

```
SELECT *
FROM     Employee  E
WHERE    E.age>=35
 AND     E.has-vehicle[Car].color = Red ;
```

질의의 WHERE 절에서 경로식 E.has-vehicle [Car].color의 값은 임의의 직원 E가 가진 Car(Vehicle과 Plane은 제외)의 color이며, 이것이 Red이면 참이 된다. 질의에서는 속성 has-vehicle 을 단일값으로 가정하였다. 만일, 속성 has-vehicle이 집합값을 가지면 이 경로식의 계산 결과도 직원 E가 가진 Car들의 color로서 집합이 되고, 연산자도 집합 연산자를 사용해야 한다. 다음은 집합값 속성을 가지는 경우의 질의 예이다.

(질의 2) 소속 Employee 가 모두 $100,000 이상의 월급을 받는 Company를 반환하시오.

```
SELECT  *
FROM    Company  C
WHERE   C.dept.employee.salary
        ALL>=$100,000;
```

질의 2에서 경로식 C.dept.employee.salary를 계산한 결과는 임의의 Company C에 소속된 employee들의 월급으로서 집합값이다. 그리고 전체 한정자(universal quantifier) ALL이 연산자 >=에 수식되었으므로 좌측 연산항인 특정 회사 C의 직원들의 월급이 모두 우측 연산항인 100,000 보다 크거나 같으면 참이 된다. 즉, 임의의 Company C의 소속 직원 모두가 $100,000 이상의 월급을 받으면 회사 C는 해가 된다.

3. VOQL - 시각적 객체 질의어

본 장에서는 VOQL의 기반이 되는 Higraph[Har88]를 소개하고, 이를 이용하여 객체지향 데이터베이스의 스키마와 객체지향 질의어를 모델링한다.

Higraph[Har88]는 그래프(graph)와 벤다이어그램(Venn Diagram)을 통합한 형태로서 지식 표현(knowledge representation), ER(entity relationship) 데이터 모델링, 복잡한 병렬 시스템의 모델링 등을 위하여 제안되었다. Higraph는 기존의 노드와 경로로 구성되는 그래프에 서브 노드(subnode)라는 개념을 추가하여 확장한 것으로 다음과 같이 정의된다.

(정의 1) Higraph [Har88]

Higraph HG = (N, L, F) 이다. 여기서 N은 노드 집합이고, L은 링크(혹은 경로) 집합이며, F는 집합 N에서 N의 수퍼 집합 2N으로의 싸이클(cycle)을 갖지 않는 함수로서 노드 N의 서브 노드들의 집합이다.

그림 2는 간단한 Higraph를 보여준다. 그림 2의 Higraph는 노드집합 N = {A, B, C, D, E, G}, 링크 집합 L = {(A,E)}, 서브 노드 함수 F={(A,{B,C}), (C,D), (E,G)}로 구성된다. 서브 노드 함수 F는 노드 A와 노드 B, C가 서브노드 관계를 가지며, C와 D, E와 D가 각각 서브노드 관계를 가짐을 의미한다. 그림2의 Higraph에서 링크를 표현할 때 주의할 점은 링크 (A,E)가 실제로는 노드 A와 E의 서브 노드들에 대한 링크들인(A,G), (B,E), (B,G), (C,E), (C,G), (D,E), (D,G)를 포함하는 일반적인 의미(generalization)이라는 점이다. 즉, (A, E)가 L에 속하면 A의 임의의 서브 클래스 x와 E의 임의의 서브 클래스 y에 대해서도 링크 (x, y)가 존재함을 의미한다.

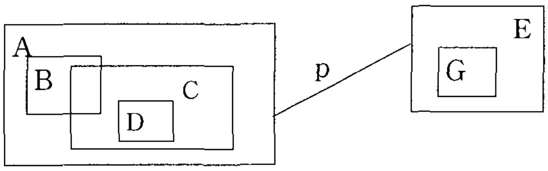


그림 2. Higraph의 예.

(정의 2) Higraph에서 경로와 길이

경로(path)는 Higraph에 속하는 노드 Vi와 링크 Ei의 순차적 나열(sequence)인 V0E1V1E2 V2...EnVn이다. 여기서, 노드 Vi-1와 Vi는 링크 Ei의 양쪽 끝 노드이다. 주의할 점은 노드 Vi-1와 Vi가 노드 혹은 서브 노드가 될 수 있다는 점이다. 경로의 길이는 경로에서 링크의 개수로 정의한다.

예를들어, 그림2의 Higraph에서 경로 A p E는 수퍼 노드에 인접한 링크 p를 사용하여 표시한 경우이며, 경

로 A p G, D p E, D p G 등은 서버 노드에 인접한 링크를 사용하여 표시한 것이다. 후자의 경로는 수퍼 노드를 사용한 전자의 경로보다 더욱 구체적인 관계를 나타내게 된다. 즉, A p E는 노드A(B, C, D 포함)와 노드 E(G 포함)와의 관계를 나타내지만, D p G는 노드 D와 G의 객체를 대상으로 한정하여 관계 p를 표현한 것이다.

3.1. Higraph [Har88]를 사용한 스키마 모델링

Higraph는 서브 노드 개념을 사용하여 객체지향 데이터베이스의 클래스 상속 계층 구조를 표현하고, 링크를 사용하여 클래스들 사이의 어소시에이션 관계를 표현하도록 함으로써 객체지향 데이터베이스의 스키마를 표시하는데 적합하다. 여기서는 객체지향 데이터베이스의 스키마를 Higraph로 모델링하는 방법을 제안한다.

객체지향 데이터베이스의 스키마는 클래스와 그들 사이의 관계로 구성된다. 클래스 사이의 관계는 IS-A 관계와 어소시에이션 관계로 구분된다. 이러한 스키마의 구성 요소들은 다음과 같이 Higraph로 모델링된다.

(1) 객체들의 집합을 나타내는 클래스는 Higraph의 노드로 표시한다. 이 노드를 클래스 노드라고 부른다. 단, 기본 타입에 해당하는 클래스(예를들어, 정수, 실수, 문자 자료형 등)는 별도로 표시하지 않고 사용자가 원하는 경우에만 텍스트로 보여준다.

(2) 클래스들 사이의 IS-A 관계는 서브 클래스의 객체 집합이 수퍼 클래스의 객체 집합에 포함된다는 의미를 가지므로 벤다이어그램의 포함관계를 사용하여 시각적으로 표현한다. 두 부모 클래스로부터 다중 상속을 하는 클래스의 경우도 서브 클래스를 벤다이어그램의 교집합으로 표시한다.

(3) 클래스들 사이의 어소시에이션 관계는 Higraph에서의 링크를 사용하여 표시한다. 특히, 서브 클래스 사이의 관계는 수퍼 클래스 사이의 관계로 추상화하여 한번만 표현함으로써 Higraph를 단순화한다.

그림 3은 그림 1을 Higraph 형태로 재작성한 것이다. 그림 1에 비하여 그림 3의 중요한 차이점은 클래스 Person의 서브 클래스들(Employee, Supplier, Student)이 IS-A 관계를 나타내는 점선으로 연결되는 것이 아니라 노드 Person의 서브 노드들로 표시된다는 점이다. 서브 노드 개념을 도입함으로써 Employee 나 Student 노드에 속한 객체 집합이 Person 노드의 객체 집합의 서브 집합(subset) 임을 시각적으로 알 수 있게 되며, 수퍼 클래스로의 데이터 추상화를 시각적으로 보여주게 된다. 다중 상속 관계에 있는 서브 클래스의 경우도 수퍼 클래스들의 교집합으로 표시함으로써 시각적인 표현이 가능해진다. 예를 들어, 그림에서 클래스 TA에 속하는 객체들은 노드 Employee에도 속하면서 노드 Student에도 속하는 공통 집합 (intersection)임을 시각적으로 알 수 있다.

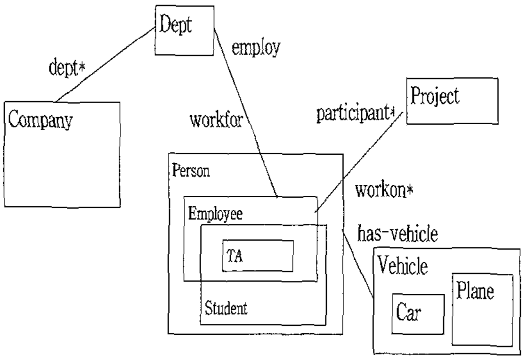


그림 3. Higraph로 모델링한 그림 1의 객체지향 데이터베이스 스키마

기존의 스키마 그래프[Kim90,Rum90]에서는 클래스들 사이의 IS-A 관계와 어소시에이션(association) 관계를 모두 링크로 표현함으로써 혼란이 초래될 수 있다. 또한, 링크로 표시된 IS-A 관계는 두 클래스에 속하는 객체 집합들 사이에 포함관계를 가진다는 사실을 시각적으로 표시하지 못하고 있다. Higraph에서 링크는 모두 어소시에이션 관계를 나타내며, ISA 관계의 표현은 집합노드와 서브 집합노드로 표시하여 더욱 직관적으로 의미를 표시하였다. 또한, 링크는 레이블을 가지며, 이 레이블은 도메인이 다른 클래스인 속성의 이름을 나타낸다. 즉, 그림3에서 레이블이 dept인 링크는 클래스 Company의 속성이고, 도메인이 Department 클래스임을 나타낸다. 만일 두 클래스에 포함된 속성이 서로 나머지 한 클래스를 도메인으로 하는 속성을 가진다면 (예를들어 그림 3에서 클래스 Employee와 Project에 포함된 속성 workon과 participant가 이 경우에 해당함), 시각화의 단순성을 위하여 하나의 링크로 표시하며, 속성의 이름(레이블)을 그 속성을 포함하는 클래스 노드 쪽에 위치시켜 서로를 구분한다. 즉, workon은 Employee 노드 쪽에, participant는 Project 노드 쪽에 위치시켜 서로를 구분하며, 시각적 질의를 작성할 때 workon(participant)을 선택하면 Employee(Project)에서

Project(Employee)로 향하는 링크를 의미한다.

3.2 경로식 표현을 위한 시각적 패턴 표시법

인스턴스 그래프는 데이터베이스에 저장된 객체와 그들간의 관련성을 보여주는 패턴으로써 질의에서 사용자가 원하는 객체의 패턴을 데이터베이스 관리 시스템에게 제시하는데 사용된다.

(정의 3) 인스턴스 그래프 (Instance Graph)

객체지향 데이터베이스에서 인스턴스 그래프 IG = (O, L) 는 객체를 나타내는 노드 (앞으로 객체 노드라고 부름)들의 집합 O={Oi}와 임의의 두 객체(노드) Oi와 Oj간의 관계를 나타내는 링크들의 집합 L= {(Oi, Oj)}로 구성된다. 앞으로 클래스내에 포함된 객체를 클래스(객체들의 집합)를 나타내는 사각형 노드(즉, 클래스 노드)와 구분하여 작은 회색 불릿(bullet)으로 표시한다.

그림 4는 Company Ci가 Dept Di를 가지며, Dept Di에는 Employee Ei 가 근무하고, Employee Ei 는 Project Pi에 근무하면서 Vehicle Vi를 소유한다는 사실을 나타내고 있다. 인스턴스 그래프에서 임의의 객체 노드와 링크로 연결된 객체가 두 개 이상인 경우도 있다. 이러한 경우는 객체의 속성이 집합값을 가지는 경우에 발생한다. 예를들어, 그림 4에서 객체노드 Ci 와 링크로 연결된 Dept 클래스의 객체 노드는 Ci 에서 속성 dept가 다중값을 가지므로 다수개이다. 이러한 경우에 객체 Ci 와 링크로 연결된 Dept 클래스의 객체 집합을 Dept 노드의 서브 노드로 표시하여 하나 이상의 Dept 객체가 Ci 와 연관됨을 시각적으로 표시한다.

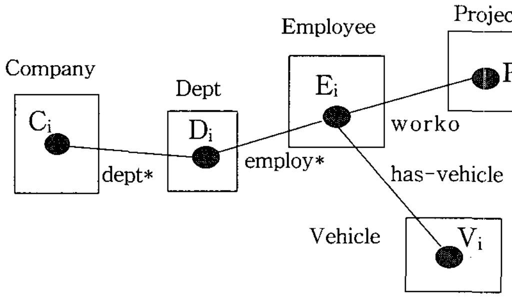


그림 4. 그림 3의 DB에 대한 인스턴스 그래프

이와 같이 집합 속성으로 인하여 하나의 객체와 연관된 객체집합을 서브 노드로 표시하여 만든 그래프를 추상화된 인스턴스 그래프(Abstracted Instance Graph)라고 부르며, 다음과 같이 정의한다.

(정의 4) 추상화된 인스턴스 그래프

추상화된 인스턴스 그래프란 인스턴스 그래프에서 집합값 속성에 대응하는 링크의 인접 객체노드들을 객체집합을 나타내는 서브노드로 표시한 인스턴스 그래프이다.

그림 5는 그림 4의 인스턴스 그래프에 대한 추상화된 인스턴스 그래프를 보여준다. 그림 5에서 노드 Dij는 Company Ci의 부서들의 집합을 나타낸다. 즉, Dij= {Dk | (Ci , Dk) ∈ L }이다. 유사하게, Eij는 Company Ci에 포함된 부서들에 근무하는 직원들의 집합을, Pij는 Ci의 직원들이 참여하는 프로젝트들의 집합을 각각 나타낸다.

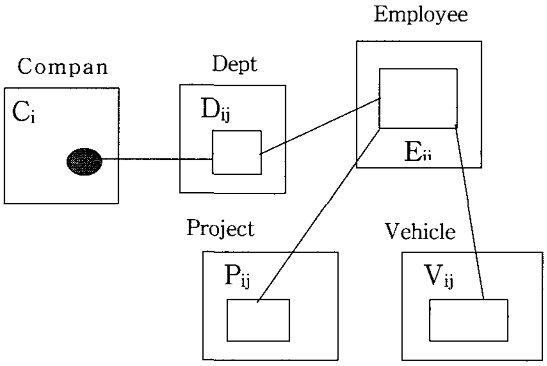
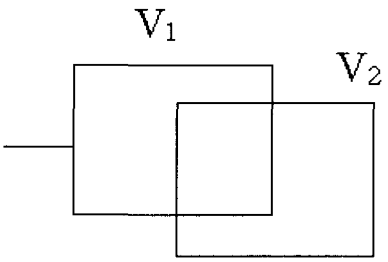


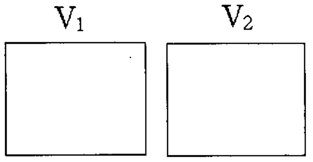
그림 5. 그림 4에 대한 추상화된 인스턴스 그래프

추상화된 인스턴스 그래프에서 링크의 인접 노드는 다른 노드와 연관되어 특정한 조건을 표시할 수 있다. 기존의 밴다이어그램 자체는 집합과 집합의 포함관계만을 시각적으로 표현하지만 여기서는 객체지향 질의어의 여러 특성들을 표현할 수 있도록 벤다이어 그램을 그림 6과 같이 확장한다. 그림에서 θ는 두 스칼라 값의 비교 연산자로써 =, <, ≤, > , ≥ 등을 나타낸다.

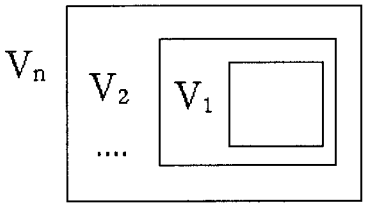


$V_1 \cap V_2 \neq \emptyset$

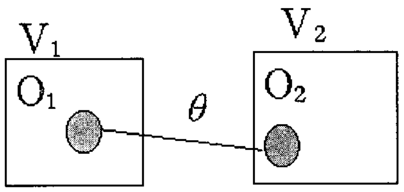
(a)교집합 있음(intersection)



$V_1 \cap V_2 = \emptyset$
(b) 교집합 없음(disjoint)



$V_1 \subset V_2 \subset \ldots \subset Vn$
(c) 포함됨(inclusion)



$O_1 \ \theta \ O_2$
(d) 두집합의 원소 비교: $O_1 \ \theta \ O_2$

그림 6. 벤다이어그램을 이용한 다양한 조건의 시각적 표현

3.3 VOQL 질의 작성

VOQL은 Higraph 형태의 시각적 질의어로서 객체지향 데이터베이스를 검색하는데 사용된다. VOQL로 작성된 객체지향 질의어는 Higraph로 표시된 스키마 그래프의 서브 그래프에 질의 조건과 프로젝션 리스트를 추가한 형태이다. 질의 조건은 하나의 노드(클래스)에 대한 조건인 노드-조건과 두 개 이상의 노드에 대한 조건인 전역-조건으로 구성되며, 노드-조건은 노드의 조건-레이블로, 전역-조건은 조건-박스에 각각 표시한다. 프로젝션 리스트는 P: [클래스-이름.]속성-리스트 혹은 전체 속성을 의미하는 * 로 표시하며, 질의 결과로 선택된 객체 중에서 프로젝션 리스트에 포함된 내용만 최종적으로 사용자에게 전달된다.

한가지 주의할 사항은 질의 조건들은 AND로 연결되었다고 가정한다. OR를 포함한 질의는 변환을 통하여 다수개의 AND 만으로 연결된 질의들의 합집합(Union)으로 표현될 수 있으므로 본 논문에서는 다루지 않는다.

VOQL의 사용자는 다음 세 단계를 거쳐서 시각적 질의를 작성한다.

(1) Higraph 형태의 스키마 그래프를 보면서 질의하고 싶은 클래스(들)와 경로(들)를 선정한다. 선정된 결과는 스키마 그래프의 서브 그래프로서 Higraph로 표시된다. 여기서 주의할 점은 노드 선택의 다양성을 이해하는 것이다. 먼저 수퍼 클래스(예를들어 P)에 해당하는 노드를 선택한 후, 노드 P에 대하여 마우스 오른쪽을 클릭하여 "show subclass" 라는 메시지를 보내면 그의 직속 subclasses(예를들어, 노드 S1, S2, S3)가 나타난다. 사용자는 이들 subclasses 들과 수퍼 클래스에 대하여 집합 연산자를 가할 수 있으며, 질의 대상은 이 집합 연산의 결과로 한정된다. 예를 들어, 노드 P와 S1을 선택한 다음에 집합 연산자 "차집합"을 가하면 질의 대상은 P - S1로 한정된다.

(2) 선정된 노드(들)에 대하여 질의 조건을 부과한다. 질의 조건은 단계 1에서 선정된 클래스 노드에 대하여 노드-조건의 경우 조건-레이블로 부과하고, 전역 조건인 경우 조건-박스 안에 부과한다. VOQL에서는 다양한 조건을 시각적으로 표시하기 위하여 조건식 외에도 다음 두 가지 형태의 질의 조건을 추가로 사용한다.

- 객체 집합들 간의 관계에 대한 조건 : 벤다이어그램으로 표시
- 사용자가 원하는 객체 패턴 : (추상화된) 인스턴스 그래프로 표시

(3) 사용자에게 반환될 질의 결과 항목들을 프로젝션 리스트에 명시한다.

단계 1에서 사용자는 스키마 그래프를 보면서 질의하고자 하는 클래스와 경로를 선정하고, 그 결과 서브 그래프에 대하여 단계 2에서 질의 조건을 부과하므로 스키마를 기억하면서 텍스트 질의를 작성하거나 혹은 아이콘과 같은 다른 종류의 시각적 도구를 사용하여 질의를 작성하는 경우보다 질의 작성시 발생하는 오류를 줄일 수 있다. 또한, 단계 2에서 사용자는 기존의 시각적 질의어보다 다양한 방식으로 질의 조건을 시각적으로 표현할 수 있게 된다. 즉, 사용자는 조건-레이블이나 조건-박스에 질의 조건을 명시하는 것과 더불어 벤다이어그램과 인스턴스 그래프 형태의 객체 패턴을 사용하여 원하는 질의 조건을 다양하게 표현할 수 있다.

다음에서 몇가지 예를 통하여 VOQL을 살펴본다.

먼저, 질의에 포함된 경로식이 하나의 값을 나타내는 스칼라 경로식을 다루고, 다음으로 집합값을 나타내는 집합 경로식의 경우를 다룬다. 특히, 집합 경로식에서는 한정자(quantifier)를 시각적으로 표현하는 방법을 제시한다. 마지막으로 메소드와 전역 조건식을 사용한 질의의 표현에 관하여 살펴본다.

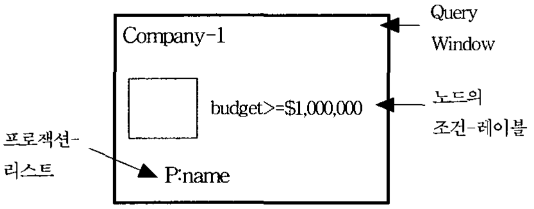
3.3.1 스칼라 경로식을 포함한 질의어의 VOQL 표현

여기서는 한정자(quantifier)와 전역-조건을 사용하지 않으면서 스칼라 경로식에 대한 조건만을 포함하는 간단한 질의어를 VOQL로 작성하는 방법을 살펴본다.

(질의 3) 예산이 $1,000,000 이상인 Company의 이름을 반환하시오.

```
XSQL : SELECT   C.name
       FROM     Company C
       WHERE    C.budget >=$1,000,000;
```

VOQL:



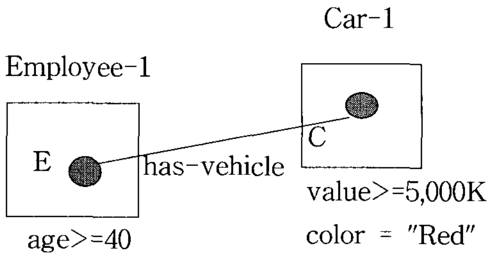
질의 3에 대한 VOQL의 작성 순서는 먼저, 스키마 그래프로부터 Company 노드를 선택한 후[3], 조건 레이블 >= $1,000,000을 Company-1 노드의 budget에 추가한다. 그리고, 조건을 만족하는 객체로부터 사용자에게 보여질 속성인 name을 프로젝션 리스트에 첨가한다.

(질의 4) 40세 이상이며, 가격이 5,000K 이상인 빨간색 자동차를 가진 Employee는 ?

```
XSQL :
    SELECT   *
    FROM     Employee
    WHERE    E.age >= 40   AND
             E.has-vehicle[Car].color= "Red"
    AND      E.has-vehicle[Car].value>=5,000K;
```

3. VOQL에서는 하나의 노드가 질의 작성에서 여러 번 선택되는 경우에도 이들을 서로 구분하기 위하여 스키마 그래프로부터 노드 N이 선택되면 그 노드의 이름을 N-i, i=1,2,...,n으로 표시해 나간다.

VOQL :



질의 4에 대해 VOQL에서는 노드의 조건-레이블과 함께 사용자가 원하는 객체의 패턴을 인스턴스 그래프 형태로 제시하고 있다. 사용자가 Employee 노드를 선택한 후 미리 정의된 객체-패턴 연산을 실행하면 Employee에 속하는 임의의 객체 노드 E가 Employee-1 노드 안에서 불릿으로 표시된다. 이 후 사용자가 스키마 그래프에서 노드 E에 대하여 has-vehicle 경로를 선택하면 has-vehicle의 도메인인 Vehicle 노드가 생성된다. 이 때 사용자는 Vehicle 중 Car에만 관심이 있으므로 Vehicle의 서브 노드인 Car를 재선정함으로써(그림에서 Car-1으로 표시함) has-vehicle의 도메인을 Car로 한정하는 XSQL의 셀렉터 기능을 시각적으로 구현한다. 또한, 속성 has-vehicle이 단일값을 가지므로 객체 E가 소유한 Car 객체(그림에서는 C로 표시됨)가 Car-1내에서 보여지며, 사용자는 E와 C로 구성된 객체 패턴에 대하여 추가로 조건을 제시할 수 있다. 여기서는 E.age에 대하여 조건 >= 40, C.color에 대하여 조건 = Red, C.value에 대한 조건 >= 5,000K를 표시하였다. 마지막으로, VOQL에서 프로젝션 박스를 별도로 지정하지 않으면 선택된 객체를 모두 반환하게 된다.

3.3.2 집합 경로식과 한정자의 VOQL 표현

객체지향 데이터베이스의 질의어에서 중요한 특징은 집합 속성을 포함하는 집합 경로식과 그에 대한 한정자(quantifier)를 사용한다는 점이다. 여기서는 집합 경로식에 대한 조건을 시각적으로 표현하는 방법을 제시한다. 특히, 제안된 방법은 기존에 시각적으로 표현하기 어렵다고 알려진 ALL/EXIST 한정자가 첨가된 조건의 경우에도 Higraph의 벤다이어그램 형태를 사용하여 시각적으로 표현할 수 있다. 예를들어, 그림 1의 데이터베이스에서 속성 has-vehicle이 집합값을 가진다고 가정하고, 질의 4를 약간 수정한 질의 5를 고려하자.

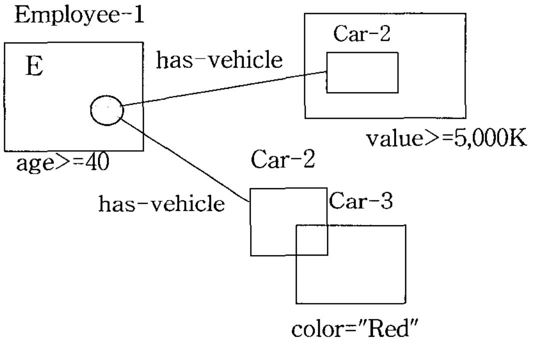
(질의 5) 40세 이상이며, 가격이 5,000K 이상인 자동

차만을 소유하고, 빨간색 자동차를 적어도 하나 가진 Employee를 찾으시오.

XSQL :

```
SELECT   *
FROM     Employee  E
WHERE    E.age >= 40  AND
         E.has-vehicle[Car].value ALL>=
         5,000K
AND      E.has-vehicle[Car].color  SOME
         = Red;
```

VOQL :



질의 5에서는 has-vehicle이 집합값을 가진다고 가정하였으므로 객체 E와 연관된 Car의 객체는 집합이며, Car-2로 표시하였다. Car-2에 포함된 모든 객체가 조건 value >= 5,000K를 만족해야 하므로 사용자는 Car의 객체중 value>=5,000K를 만족하는 객체들의 집합인 Car-1을 생성하고, Car-2를 이 집합의 서브 집합으로 설정하여 주어진 조건을 표시하도록 한다. 한편, 객체 E가 가진 Car 중에서 적어도 하나는 Red 이어야 하므로 그림의 하단에서 이 조건을 표시하였다.

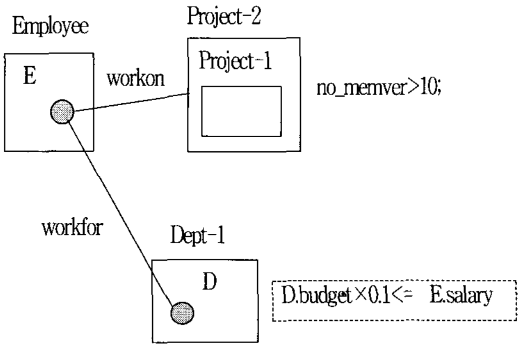
또 다른 예로써 질의 6은 전역-조건(그림에서 점선 박스로 표시됨)과 객체 집합간의 다양한 관계를 질의 조건으로 명시한 것이 특징이다.

(질의6) Employee 중 자신이 소속된 부서의 예산에 1/10을 곱한값보다 더 많은 월급을 받으면서 10 명 이상이 참여하는 과제에만 참여하는 사람의 이름은 ?

XSQL :

```
SELECT   E.name
FROM     Employee  E
WHERE    E.salary>=E.workfor.budget * 0.1
AND      E.workon.no_member  ALL>10;
```

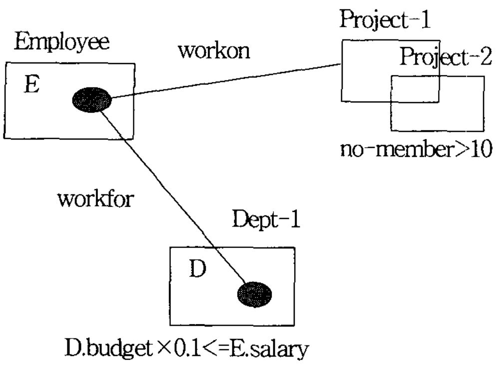
VOQL :



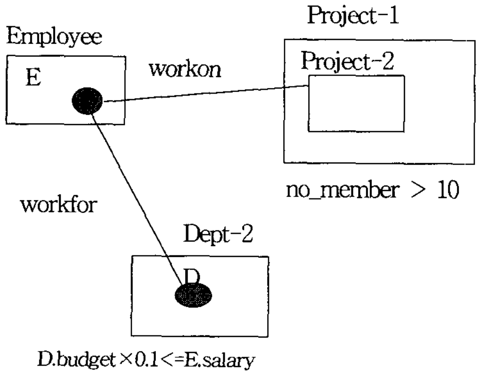
질의 6에서 객체집합 Project-1은 Employee E가 참여하는 Project들이며, 객체집합 Project-2는 member의 수가 10명 이상인 Project들의 집합을 의미한다. 질의 조건에서 Employeer가 참여하는 Projects가 모두 member의 수가 10명 이상인 Project이어야 하므로 객체집합 Project-1이 Project-2의 부분집합이 되도록 VOQL을 작성하였다. 사용자는 두 객체집합 Project-1과 Project-2의 포함관계를 조정함으로써 다양한 질의 조건을 쉽게 표현할 수 있다. 예를들어 그림 7은 객체집합 Project-1과 Project-2의 포함관계를 조정하여 얻은 질의 조건들이다. 그림 6 (a)는 질의 6에서 E.workon.no_member ALL > 10 대신에 E.workon.no_member SOME > 10을 표현한 것이며, no_member > 10 인 projects에 하나라도 참여하는 Employee를 반환한다. 한편, 그림 6(b)는 부 질의어를 포함하는 다음 질의어를 VOQL로 표현한 것으로 no_member > 10 인 projects에 모두 참여하는 Employee 를 반환한다. 마지막으로 점선 박스는 전역조건이며, 두 개 이상의 노드에 걸쳐서 제시되는 전역 조건을 표시한다.

(질의 7)

```
XSQL : SELECT  E.name
       FROM    Employee  E
       WHERE   E.salary>=E.workfor.budget *0.1
       AND     E.workon  CONTAINS
               (SELECT  *  FROM  Project
               WHERE  no_member ALL>10)
```



(a)



(b)

그림 7. 질의 6을 변형한 새로운 질의어 : (a) 질의 6에서 E.workon.no_member SOME >10으로 수정한 경우, (b) 질의 7에 대한 VOQL

3.3.3 메소드를 포함한 질의어의 VOQL작성

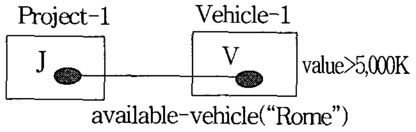
여기서는 메소드를 포함한 객체지향 질의어의 시각적 표현에 관하여 살펴본다. 객체지향 데이터베이스에서 메소드는 속성의 일반화된 형태로 간주할 수 있다[Kim90, Kif92]. 즉, 속성은 인자를 가지지 않으며 질의에서 참조되어 실행되면 기본 타입의 객체(들이)나 사용자 정의 타입의 객체(들)을 반환하는 간단한 형태의 메소드로 볼 수 있다. 메소드의 실행후 반환되는 객체의 타입을 그 메소드의 타입(혹은 도메인)이라 부른다. 본 논문에서의 경로식은 식 (1)의 형태로서 메소드와 속성을 구분하지 않고 있으므로 VOQL에서도 메소드를 속성과 동일하게 경로를 사용하여 시각적으로 표현한다.

그림 3의 데이터베이스 스키마에서 메소드 available_vehicle()은 클래스 Project에 정의되었으며, 도착지 도시를 인자로 받아서 그 곳까지 갈 수 있는 하나의 Vehicle을 반환한다고 가정할 때 다음의 질의 8을 VOQL로 표시하면 다음과 같다.

(질의 8) 로마 출장에 사용할 수 있으면서 가격이 5,000K 이상인 자동차를 가진 프로젝트의 이름은 ?.

XSQL : SELECT J.name
FROM Project
WHER J.available_vehicle("Rome").value>5,000K;

VOQL :



VOQL에서는 Project 노드를 선택한 다음에 (Project-1) 임의의 객체 J에 대하여 메소드 available_vehicle()을 선택하면 시스템은 인자를 요구한다. 사용자로부터 인자 Rome이 제공되면 available-vehicle()의 도메인인 Vehicle(Car, Plane 포함) 노드에서 실행 결과로 반환될 객체가 보여진다(그림에서 V로 표시하였으며, 메소드의 결과로 하나의 객체를 반환한다고 가정하였음). 사용자는 또 다른 조건으로 Vehicle 노드에 조건-레이블 value > 5,000K를 추가하여 질의를 완성한다.

3.3.4 VOQL 기호의 요약

표 1. VOQL 기호의 의미 요약

기호	의미
사각형 노드	객체들의 집합(클래스 등)
사각형 노드 레이블	객체 집합에 대한 조건(조건 레이블)
점선 노드	객체들에 대한 전역 조건
원형 노드	단일 객체를 나타냄
링크	도메인(리턴값의 타입)이 사용자 정의 클래스인 속성(메소드)
링크 레이블	도메인(리턴값의 타입)이 사용자 정의 클래스인 속성(메소드) 이름
노드간의 포함관계	객체 집합들의 연관성 조건(포함, 부분집합, 교집합 등)

여기서는 VOQL에서 사용된 기호들의 의미를 요약한다. VOQL은 제 3.1 절에서 소개한 바와 같이 Higraph를 기반으로 생성된다. 따라서 VOQL의 구성요소는 기

본적으로 노드와 레이블을 가진 링크이며, 노드에 대하여 조건을 레이블로 추가할 수 있다. 다음은 VOQL에 사용된 기호들의 의미를 요약한 표이다.

VOQL은 기본적으로 그래프 형태로써 사각형 노드는 객체들의 집합을 나타내며, 원형 노드는 객체를 나타내고, 링크는 속성 혹은 메소드를 나타낸다. 특이한 점은 노드들간의 포함관계를 이용하여 질의 조건을 표시한다는 점이다.

4. VOQL과 XSQL의 비교

본 절에서는 VOQL과 기존의 객체지향 질의어의 표현력을 비교한다. 관계 질의의 경우에는 관계 대수가 관계적으로 완전한 질의임이 증명되어 있으므로 새로운 질의어를 제안할 때 관계 대수와 비교하여 그 표현력의 정도를 측정할 수 있다. 그러나, 객체지향 질의어의 경우는 질의어 자체가 연구 단계에 있으며, 대다수가 공감하는 표준 질의어도 아직 제정되지 못한 상태이므로 여기서는 기존의 여러 객체지향 질의어 중에서 표현력이 비교적 뛰어난 XSQL[Kif92]과 VOQL의 표현력을 비교한다.

XSQL의 가장 중요한 특징은 경로식에 대한 조건을 사용한다는 점이다. 여기서는 XSQL의 경로식에 대한 조건을 VOQL의 노드와 링크를 사용하여 시각적으로 표시할 수 있음을 입증한다. 이를 위하여 먼저 경로식의 길이가 1인 경우에 대하여 VOQL로 표시할 수 있음을 보이고, 이를 기반으로 귀납법을 사용하여 임의의 길이를 가지는 경로식도 VOQL로 표시할 수 있음을 보인다.

4.1 길이가 1 인 경로식에 대한 조건의 VOQL 표현

길이가 1인 경로식이란 속성 하나를 포함한 경로식을 의미한다. 예를들어, Employee.age > 30은 길이가 1인 경로식에 대한 조건이다. 이 경우 경로식에 포함된 클래스는 VOQL에서 사각형 노드로 표시하며, 속성과 그에 대한 조건은 노드의 VOQL에서 조건-레이블로 표현된다. 그림 8은 경로식에 대한 조건 class.attribute θ value을 VOQL로 표현한 것이다.

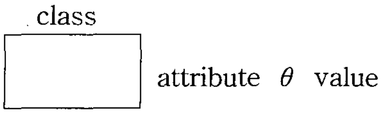
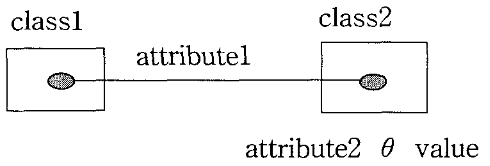


그림 8. 길이가 1인 경로식에 대한 조건의 VOQL 표현

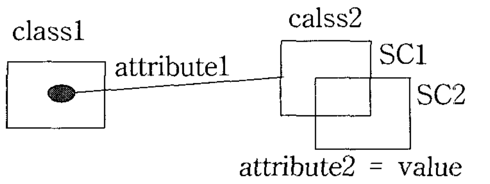
4.2 길이가 2인 경로식에 대한 조건

속성이 두 개 포함된 경로식에 대한 조건을 일반적으로 class1.attribute1[class2].attribute2 θ value로 표시하면 atttribute1이 단일값 속성인가 혹은 집합값 속성인가에 따라서 그림 9와 같이 두가지로 구분하여 VOQL로 작성할 수 있다. 먼저, attribute1이 단일값을 가지는 경우에는 그림 9의 (a)와 같이 class1의 임의의 객체와 연관된 class2의 객체를 객체 노드로 표시하고, 조건 attribute2 θ value를 class2의 노드 조건-레이블로 첨부한다. 다음으로, attribute1이 집합값을 가지는 경우 - θ가 ALL=이나 SOME= 등과 같이 ALL 이나 SOME으로 한정된 scalar 연산자인 경우 - 에는 세단계로 나누어 그림 9의 (b)와 같이 VOQL을 작성한다. 첫번째 단계로써 class1의 임의의 객체에서 속성 attribute1에서 참조되는 class2의 객체들의 집합을 class2의 서브노드 SC1로 표시한다. 두번째 단계로써 class2의 객체중 조건 attribute2 θ'value를 만족하는 서브집합을 SC2 노드로 표시한다. 여기서 θ'는 θ에서 한정자 (SOME이나 ALL)을 삭제한 나머지 부분이다. 마지막 단계로서 θ의 의미를 노드 SC1과 노드 SC2의 벤다이어그램 관계로 표시한다. 그림 9 (b)는 편의상 θ를 SOME=로 가정하여 SC1과 SC2가 교집합이 있도록 표시하였다. 이 경우, attribute1의 속성 값으로 SC2에 속하는 하나의 객체라도 참조하면 되기 때문이다. 만일, θ가 ALL= 이라면 그림 9 (c)와 같이 벤다이어그램 조건과 연산자를 표시한다. 이 경우에는 attribute1에서 참조하는 모든 객체가 SC2에 속하는 객체이어야 하기 때문이다. 만일, θ가 $\subseteq$ 이라면(즉, class1.attribute1[class2].attribute2 $\subseteq$ {v1, v2,...,vn}) 그림 9 (d)와 같이 표시하여 attribute1에서 참조하는 객체들이 {v1, v2,...,vn}의 일부임을 표시한다[4]. 나머지 집

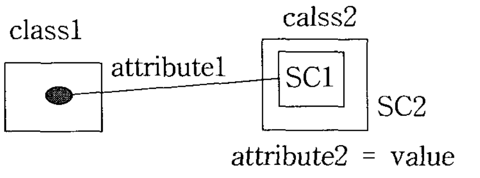


(a) attribute1이 단일값을 가지는 경우

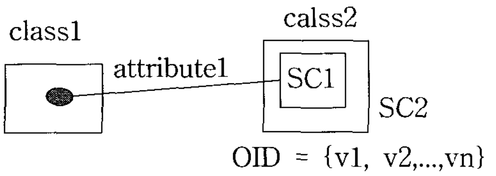
4. 여기서 vi는 단순 타입 (integer, character 등)의 객체이거나 사용자 정의 데이터 타입(즉, 클래스)이 될 수 있으며, 단순 타입의 객체인 경우 별도의 OID를 가정하지 않고 값 자체를 OID로 간주함.



(b) attribute1이 집합값, θ가 SOME=인 경우



(c) attribute1이 집합값, θ가 ALL=인 경우



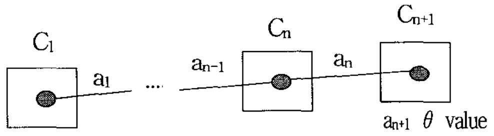
(d) attribute1이 집합값, θ가 ⊆ 인 경우

그림 9. 길이가 2인 경로식에 대한 조건의 VOQL표시.

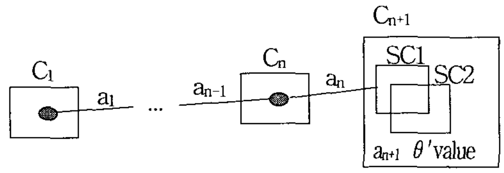
합 연산자에 대해서도 유사한 방식으로 벤다이어그램 관계로 연산자를 표시한다.

4.3 길이가 n+1 인 경로식의 VOQL 표현

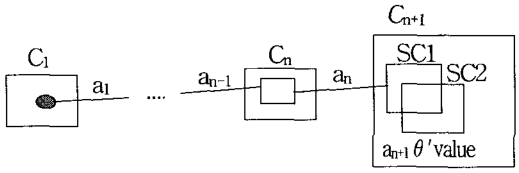
여기서는 VOQL을 사용하여 길이가 n+1인 경로식을 표현할 수 있음을 귀납적인 방법으로 증명한다. 이를 위하여 길이가 n (n>=2)인 경로식에 대한 조건 C1.a1[C2].a2 ··· an-1[Cn].an θ value가 VOQL로 표현 가능하다고 가정하자. 이로부터 길이가 n+1 (n>=2)인 경로식에 대한 조건 C1.a1[C2].a2 ··· an-1[Cn].an [Cn+1].an+1 θ value도 VOQL로 표현할 수 있음을 보인다. 길이가 n (n>=2)인 경로식에 대한 조건은 가정에서 VOQL로 표현할 수 있으므로, 그림 9와 같이 속성 ai (1<=i<=n-1)이 모두 단일값을 가지는 경우와 그렇지 않은 경우로 나누어 생각할 수 있다. 먼저, 속성 ai (1<=i<=n) 가 모두 단일값을 가지는 경우는 그림 9(a)로부터 그림 10(a)처럼 간단히 확장된다. 다음으로 속성 ai (1<=i<=n-1)는 단일값을 가지지만 an이 집합값 속성을 가지는 경우에는 그림 9 (b)를 확장하여 그림 10 (b)처럼 표현된다. 그림 10 (b)에서 SC1은 집합값 속성 an이 참조하는 객체들이며, SC2는 Cn+1의 객체중에서 조건 an+1 θ' value을 만족하는 객체들의 집합이다. 여기서 주의할 점은 원래 조건 C1.a1[C2].a2 ··· an-1[Cn].an[Cn+1].an+1 θ value에서의 θ가 두 집합 SC1과 SC2 사이의 집합 관계로 변환된 후에는 스칼라 연산자(여기서는 임의로 θ'으로 표시하였음)로 대치된다는 점이다. 이와 같이 집합 연산자가 벤다이어그램 관계와 다른 스칼라 연산자로 변환되는 경우는 4.2 절에서 설명되었다. 마지막으로, 속성 ai (1 <= i <= n-1) 중에서 어느 하나라도 집합값을 가지면 그림 9(b)를 확장하여 그림 10(c)와 같이 표현된다. 그림 10(c)에서는 C1의 임의의 객체(불릿으로 표시함)에서 참조하는 Cn의 객체가 집합값 속성에 의하여 이미 집합으로 표시되므로(Cn 내의 서브 노드) 속성 an에서 참조하는 Cn+1의 객체도 집합을 형성한다(SC1으로 표시). 이 서브 노드 SC1과 조건 an+1 θ' value을 만족하는 객체 집합 (SC2로 표시) 간의 벤다이어그램 관계로 전체 조건을 표시한다.



(a)



(b)



(c)

그림 10. 길이가 n+1인 경로식에 대한 조건의 VOQL 표시 : (a) 모든 속성이 단일값을 가지는 경우. (b) 속성 an만 집합값을 가지는 경우. (c) 속성 ai (1 <= i <= n) 중에서 어느 하나라도 집합값을 가지는 경우.

5. 구 현

VOQL은 현재 UniSQL 객체-관계 DBMS에서 구현중에 있다. 현재 경로식과 메소드에 대한 질의를 시각적으로 표시하였으며, 복잡한 집합 연산과 메소드 등에 관하여 구현중에 있다. 그림 11은 경로식을 포함한 VOQL 객체 질의를 시각적으로 표현한 화면을 보여주고 있다. 이 질의는 클래스 resort에 대하여 속성 리스트를 보면서 프로젝션 리스트와 셀력션 조건을 시각적으로 작성하는 화면이다. 여기서는 속성 beach에서 시작하는 경로식에 대한 조건을 보여주고 있으며, 지면 제약상 경로식 전체에 대한 윈도우들은 생략하였다.

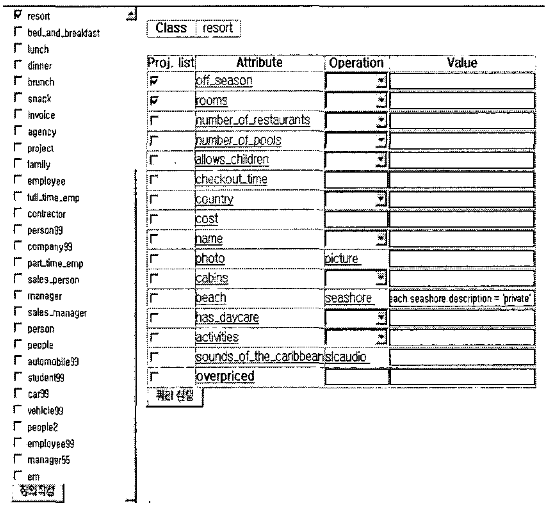


그림 11. VOQL 질의어 작성 예 : 경로식 resort.beach.description

Query문 : select resort.off_season , resort.rooms from resort where beach.description = 'private'

resort.off_season	resort.rooms
October 15 through March 15	
October 15 through March 15	@128|11|0 @149|1|0
October 15 through March 15	

그림 12. 그림 11에 대한 텍스트 질의와 질의 실행 결과

그림 12는 그림 11의 질의를 텍스트로 변환하여 UniSQL에서 처리한 후 결과를 화면에 보여주고 있다. VOQL은 인터넷 상의 클라이언트에서 원격지 데이터베이스 서버에 접근하는 것을 지원하기 위하여 UniSQL에서 제공하는 인터넷 프로그래밍 언어인 UniTCL [Uni91]을 사용하여 구현하였으며, 현재 다른 ORBBMS에 대하여 Java와 JDBC를 사용하는 패키지로 확장중에 있다.

6. 결 론

객체지향 데이터베이스에서 시각적 질의어인 VOQL을 제안하였다. VOQL은 그래프와 벤다이어그램을 통합한 형태의 새로운 그래프인 Higraph라고 하는 단일의 개념을 사용하여 객체지향 데이터베이스의 스키마와 질의를 시각적으로 표현한다. 기존의 시각적 질의어에서는 대체로 데이터베이스 스키마는 그래프 형태로, 질의어는 그래프가 아닌 아이콘 등의 방식으로 표현함으로써 서로 다른 표현 도구가 사용되지만 VOQL에서는 HiGraph라고 하는 특별한 그래프 형태로 스키마와 질의를 모두 표현한다. 특히, 제안된 VOQL은 객체지향 데이터베이스에서 널리 사용되는 길이가 2 이상인 경로식에 대한 조건과 한정자가 사용되는 질의어도 그래프에서의 경로와 벤다이어그램을 사용하여 간단히 표현할 수 있다는 장점을 가진다. VOQL은 대표적인 객체지향 질의어인 XSQL에서의 경로식에 대한 조건을 표현할 수 있으며, 집합 속성을 포함한 간단한 형태의 중첩 질의도 VOQL로 표현할 수 있다. 향후 VOQL과 XSQL의 상호 변환 알고리즘을 개발하여 두 언어의 표현력을 비교 분석하는 연구와 뷰를 포함한 질의어 및 일반적인 형태의 중첩 질의어를 VOQL로 표현하도록 확장하는 연구를 계속하고자 한다. 그리고, VOQL의 안전성(safety)에 관한 연구도 계속하고자 한다.

참 고 문 헌

[Bal96] N. H. Balkir et al., "VISUAL : A Graphical Icon-Based Query Language," In Proc. Int'l Conf. on Data Engineering, pages 524~533, 1996.

[Ber92] E. Bertino et al., "Object-Oriented Query Languages : The Notion and the Issues," IEEE Trans. on Knowledge and Data Engineering, Vol. 1, No. 3, pages 223~237, June 1992.

[Cru92] 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.
[Gys90] M. Gyssens, et al., "A Graph-Oriented Object Model for Database End-User Interfaces," In Proc. ACM SIGMOD Int'l Conf. on Management of Data, pages 24~33, 1990.
[Har88] D. Harel, "On Visual Formalisms," Comm. of the ACM, Vol. 31, No. 5, May 1988.
[Kif92] M. Kifer et al., "Querying Object-Oriented Databases," In Proc. ACM SIGMOD Int'l Conf. on Management of Data, pages 393~402, 1990.
[Kim90] W. Kim, Introduction to Object-Oriented Databases, MIT Press, 1990.
[Moh93] L. Mohan and R. L. Kashyap, "A Visual Query Language for Graphical Interaction With Schema-Intensive Databases," IEEE Trans. on Knowledge and Data Engineering, Vol.5, No.5, pages 843~858, 1993.
[Ozs89] G. Ozsoyoglu and H. Wang, "A Relational Calculus with Set Operators, its Safety, and Equivalent Graphical Languages," IEEE Trans. on Software Engineering, Vol. 15, No. 9, Sept. 1989.
[Par95] J. Paredaens et al., "A-Gog : A Graph-Based Query Language," IEEE Trans. on Knowledge and Data Engineering, Vol. 7, No. 3, June 1995.
[Rum90] J. Rumbaugh, et al., Object-Oriented Modeling and Design, Prentice Hall, 1991.
[Sat91] F. Staes et al., "A Graphical Query Language for Object-Oriented Databases," In Proc. IEEE Workshop on Visual Languages, pages 205~210, 1991.
[Soc93] G. H. Sockut et al., "GRAQULA : A Graphical Query Language for Entity-Relationship or Relational Databases," Data and Knowledge Engineering, Vol.11, No. 2, pages 171~202, 1993.
[Tha94] A. K. Thakore and S. Su, "Performance Analysis of Parallel Object-Oriented Query Processing Algorithms," Distributed and Parallel Databases, pages 59~100, 1994.
[Uni91] UniSQL, UniSQL/X Users Manual, Rel. 1.0, 1991.
[Vad93] K. Vadaparty et al., "Towards a Unified Visual Database Access," In Proc. ACM SIGMOD Int'l Conf. on Management of Data, pages 357~366, 1993.
[Wha92] 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.
[Zlo77] M. Zloof, "Query-By-Example : A Data Base Language," IBM Systems Journal, Vol. 16, No. 4, pages 324~343, 1977.

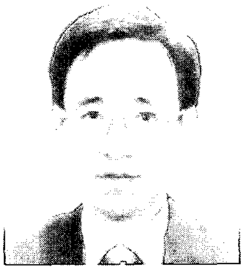
저 자 소 개

金 正 姬(正會員)

1979년 2월 : 서울 대학교 이학사. 1982년 8월 : 미국 Purdue 대학교 전산학 석사. 1998년 8월 : 한국과학기술원 전산학 박사. 1983년 7월~1985년 7월 : 한국 IBM National System Center, 삼성종합기술원 정보시스템연구소 선임연구원, Project Leader. 1989년 8월~1990년 5월 : 삼성 Software America, 삼성전자 Multimedia 연구소 수석 연구원. 1999년 3월~1999년 9월 : 삼성SDS Solution 사업부 SW 사업팀 수석. 2000년 2월~2000년10월 : 이화여자대학교 과학기술대학원 겸임교수. 1999년 12월~2001년 5월 : 비엔씨인텔리젼스 주식회사 사장. 2001년 6월~현재 : 3DNet-works 부사장

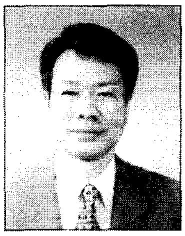
曺 完 燮(正會員)

1985년 : 경북대학교(학사). 1987년 : 한국과학기술원(석사). 1996년 : 한국과학기술원(공학박사, 전산학). 1987년~1990년 : 한국전자통신연구원(연구원). 1997년~현재 : 충북대학교 경영정보학과(조교수). <관심분야> 데이터베이스, 인터넷 및 전자상거래



이 석 균(正會員)

1982년 : 서울대학교 경제학 학사. 1990년 : University of Iowa 전산과학 석사. 1992년 : IEEE 8th International Conference on Data Engineering에서 최우수 논문상 수상. 1993년 : University of Iowa 전산과학 박사. 1993년~1997년 : 세종대학교 정보처리학과 전임강사. 1997년~현재 : 단국대학교 전산통계학과 부교수. <관심분야> 데이타 모델링, 데이타베이스에서 불완전 정보관리, 객체지향 데이타베이스 시스템, 데이타베이스 질의어, 다중처리기에서의 실시간 스케쥴링, 데이터웨어하우스, 데이터 마이닝, 데이터베이스 시각 질의어



황 규 영(正會員)

1973년 : 서울대학교 전자공학과 졸업(B.S.). 1975년 : 한국과학기술원 전기 및 전자학과 졸업(M.S.). 1982년 : Stanford University(M.S.). 1983년 : Stanford University (Ph.D.). 1975년~1978년 : 국방과학연구소(ADD), 선임연구원. 1983년~1990년 : IBM T.J. Watson Research Center, Research Staff Membber. 1992년~1994년 : 한국정보과학회 데이터베이스 연구회(SIGDB) 운영위원장. 1995년 : 한국정보과학회 이사 겸 논문지 편집위원장. 1999년~현재 : 한국정보과학회 부회장. Editior : The VLDB Journal. 1990~현재. Editor : Distributed and Parallel Databases : An International : Journal, 1991~1995. Editor : International Jouranl of Geographical Information Systems, 1994~현재. Associate Editor : IEEE Data Engineering Bulletin, 1990~1993. 1998년~2004년 : Trustee, The VLDB Endowment. 1999년~2004년 : Steering Committee Member, DASFAA. 1990년~현재 : 한국과학기술원 전자전산학과 전산학전공 교수. 1999년~현재 : 첨단정보기술연구센타(과학재단 우수연구센타) 소장 <관심분야> 데이터베이스 시스템, 멀티미디어, GIS