

지식 추상화 계층을 이용한 근사해 생성

허순영*, 문개현*

Providing Approximate Answers Using a Knowledge Abstraction Hierarchy

Huh, Soon-Young, Moon, Kae-Hyun

Cooperative query answering is a research effort to develop a fault-tolerant and intelligent database system using the semantic knowledge base constructed from the underlying database. Such knowledge base has two aspects of usage. One is supporting the cooperative query answering process for providing both an exact answer and neighborhood information relevant to a query. The other is supporting ongoing maintenance of the knowledge base for accommodating the changes in the knowledge content and database usage purpose. Existing studies have mostly focused on the cooperative query answering process but paid little attention to the dynamic knowledge base maintenance.

This paper proposes a multi-level knowledge representation framework called Knowledge Abstraction Hierarchy(KAH) that can not only support cooperative query answering but also permit dynamic knowledge maintenance. On the basis of the KAH, a knowledge abstraction database is constructed on the relational data model and accommodates diverse knowledge maintenance needs and flexibly facilitates cooperative query answering. In terms of the knowledge maintenance, database operations are discussed for the cases where either the internal contents for a given KAH change or the structures of the KAH itself change. In terms of cooperative query answering, four types of vague queries are discussed, including approximate selection, approximate join, conceptual selection, and conceptual join. A prototype system has been implemented at KAIST and is being tested with a personnel database system to demonstrate the usefulness and practicality of the knowledge abstraction database in ordinary database application systems.

* 한국과학기술원 테크노경영대학원

◆ 이 논문은 1998년 5월 1일 접수하여 1차 수정을 거쳐 1998년 6월 15일 게재확정되었습니다.

I. 서 론

기존의 데이터베이스 시스템은 스키마의 관점에서 정확하게 표현되지 못한 질의나 요구사항이 모호하게 표현된 질의는 처리할 수 없으며, 또한 질의 조건을 정확하게 만족하는 답만을 산출해 주기 때문에 정확한 답이 존재하지 않는 경우 아무런 정보도 제공해 주지 못한다. 따라서 사용자들은 자신이 원하는 답을 얻기 위해서는 데이터베이스 스키마와 문제 영역에 관한 지식을 갖고 있어야만 한다. 그러한 문제점을 해결하기 위한 노력으로 협력적 질의 응답(cooperative query answering)에 관한 연구가 수행 되어져 왔다[7, 8, 9, 10, [CCL90b][CHU ET AL 1994] 15, 17, 18, 24]. 협력적 질의응답은 정확히 작성되지 못한 질의문에 대응하여, 질의의 의도를 분석하고 데이터베이스에 내포된 여러가지 의미론적 지식을 활용하여, 정확한 답은 물론 근사적인 답이나 질의와 관련된 유용한 정보를 제공하는 지능적인 시스템을 개발하여 정보 검색시의 오류 처리와 의사결정 분석의 효과성 증대에 초점을 맞추고 있다.

협력적 질의응답을 통한 질의처리의 핵심 과정은 질의의 조건을 완화함으로써 이루어지는 질의 변환이라고 할 수 있다. 질의 완화란 지식베이스에 관리되는 하위 개념 수준의 원시 데이터와 상위 개념 수준의 추상적 개념들 사이의 의미론적 관계를 이용하여 질의의 탐색 영역을 확장하는 것을 말한다. 따라서, 협력적 질의응답에서는 데이터의 의미론을 이끌어낼 수 있게 해주는 지식 표현 프레임웍의 확립이 매우 중요하다고 할 수 있다. 지금까지 제시되어 온 다양한 지식 표현 프레임웍들은 의미론적 거리, 퍼지 집합, 규칙(rule), 개념적 분류에 기초한 기법들로 구분되어질 수 있다.

의미론적 거리 기법은 한 쌍의 데이터 개체 간의 유사성 정도를 수치화된 거리 개념으로 나타내어 질의완화 알고리즘을 쉽고 효율적으

로 개발할 수 있지만[12, 13, 15], 양적/질적 데이터 모두의 유사성을 수량화된 하나의 일률적인 척도로서 나타내기 힘들고, 다양한 데이터 개체간의 의미론적 유사성을 평가하는 객관적인 기준이 존재하지 않는다는 한계점을 가진다. 퍼지 데이터베이스 기법[1, 19, 25]은 각 도메인에 존재하는 데이터 개체들이 0과 1사이에 존재하는 유사성 정도값을 부여받을 수 있다고 가정하여 다양한 종류의 부정확성 정보, 예들 들어 혼합된 머리 색은 '갈색 0.6 검은색 0.4'로, 특정인의 거주지는 '서울 혹은 부산'으로 표현하여 저장하게 된다. 규칙 기반 기법[3, 9, 11]은 일차형 논리를 형식상의 프레임웍으로 사용하여 데이터의 의미론적 정보들과 무결성 제약을 표현한다. 규칙 기반 기법의 한계점은 질의 완화 과정을 유도해 나갈 체계적인 시스템이 부족하고 질의 응답 과정의 직관성이 떨어진다는 것이다.

개념적 분류 기법은 서로 유사한 원시 데이터 값들을 하나의 추상적인 클래스로 묶고, 각 클래스들이 다른 클래스들과 부모/자식 관계를 갖도록 하여 하나의 추상화 계층을 형성하게 된다[1, 2, 6, 8, 19, 22]. 따라서 동일한 클래스에 속한 데이터 개체들은 서로 유사한 개체들에 해당하고, 부모 클래스에 속한 개체들은 자식 클래스에 속한 개체들보다 더욱 일반화된 개체들로 여겨진다. 개념적 분류 기법은 특히 질적이고 분류 가능한 데이터 개체들에 적용할 경우에 장점을 갖는다. 또한 이 기법은 인간의 사고 체계와 유사하고 많은 조직에서 데이터의 의미론을 관리하기 위한 방법으로 이미 사용하고 있기 때문에, 데이터 개체들간의 유사성 관리를 위하여 쉽게 구현 되어질 수 있다.

이러한 기존의 연구들은 각각 데이터의 의미론적 지식을 표현하기 위한 프레임웍과 그를 바탕으로 한 협력적 질의처리 과정을 제시하고 있지만, 지속적인 지식의 변화를 효과적으로 처리해 줄 수 있는 지식 관리 측면을 고려하지

못하고 있다. 결과적으로 지식 표현 프레임워크에 의해 표현된 지식의 변화를 동적으로 처리해 주는데 많은 문제점을 갖게 되고, 지식을 실제로 관리하게 되는 데이터베이스를 확장 시키기도 힘들게 된다. 현재 가장 널리 이용되고 있는 개념적 분류 기법의 경우, 이러한 지식 관리상의 문제점은 지식 표현 프레임워크에 보다 다양한 정보를 포함시킴으로써 해결될 수 있다. 또한 더욱 효과적인 지식 관리는 물론 보다 다양한 유형의 질의를 처리할 수 있게 된다.

이 논문은 이러한 문제점을 해결하기 위하여, 값의 추상화에 관한 정보뿐 아니라 도메인 추상화에 관한 정보까지 포함하는 지식 추상화 계층 KAH(Knowledge Abstraction Hierarchy)을 제시함으로써 개념적 분류 프레임워크를 확장한다. KAH는 데이터베이스에 대한 의미론적 지식과 실제 데이터 값을 통합하여 협력적 질의응답을 수행할 수 있게 해줄 뿐 아니라, 동적인 지식 관리를 용이하게 해 주는 효과적이고 직관적인 지식 표현 프레임워크라고 할 수 있다. 또한 이 논문에서는, KAH에 포함된 의미론적 지식을 관리하기 위한 지식 추상화 정보 데이터베이스 KaDB(Knowledge Abstraction Database)를 구축하고, 값의 일반화와 세분화 과정을 바탕으로 다양한 질의처리를 수행해 나가는 협력적 질의응답 과정이 제시된다. 한편, 일반적인 데이터베이스 응용 시스템에서의 KAH의 유용성을 검증하기 위하여, 프로토타입 시스템이 구현되어 테스트 중에 있다

이 논문은 다음과 같이 구성된다. 2장에서는 지식 추상화 계층 KAH를 값의 추상화와 도메인의 추상화라는 두개의 추상화 관점에서 소개한다. 3장에서는 관계형 데이터 모델을 이용하여, KAH에 포함된 의미론적 지식을 관리하는 KaDB를 구축한다. 4장은 KaDB에 저장된 지식을 이용하여 협력적 질의응답을 수행할 수 있도록 확장된 질의어인 KaSQL(Knowledge

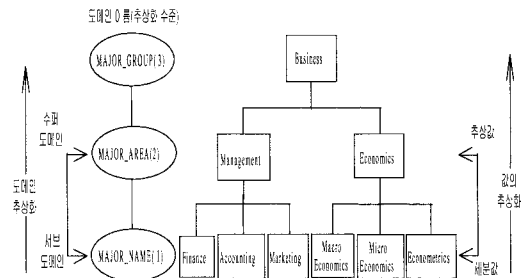
Abstraction SQL)을 설계하고, 그를 바탕으로 다양한 유형의 협력적 질의응답을 수행하는 질의 처리 메커니즘이 논의된다. 마지막으로 5장에 결론을 제시한다.

II. 지식 추상화 계층

데이터 추상화는 데이터 값들간의 의미론적 관계를 파악하는 효과적인 방법으로 간주되어 왔다[21]. 데이터 추상화는 보통 구체적인 세부 사항들은 은폐하고 추상화 된 개체와 그들 사이의 관계에 초점을 맞추게 된다. 본 논문에서는 데이터 추상화를 바탕으로 데이터 값들 사이의 의미론적 관계를 나타내는 프레임워크로서 지식 추상화 계층을 제시한다.

2.1 지식 추상화 계층의 구성

지식 추상화 계층은 하나의 지식 표현 프레임워크로서, 데이터 추상화를 통하여 데이터 값과 데이터베이스에 대한 의미론적 지식을 다단계 구조로 표현할 수 있게 해 준다. 데이터 추상화에서 하나의 데이터 값은 추상값으로 일반화될 수 있고, 일반화된 추상값은 세분값으로 세분화될 수 있다. KAH는 추상값과 세분값 사이의 이러한 일반화/세분화 관계를 바탕으로 구축되어 진다. <그림 1>에는 대학 전공에 관한 KAH가 나타나 있다.



<그림 1> 대학 전공에 관한 지식 추상화 계층

KAH는 값의 추상화 계층과 도메인 추상화 계층의 두 가지 추상화 계층으로 구성된다. <그림 1>은 대학 전공들을 전공 이름 (MAJOR_NAME), 전공 분야 (MAJOR_AREA), 전공 그룹 (MAJOR_GROUP) 등의 다단계 추상화 수준으로 분류한다. 값의 추상화 계층에서 Finance, Accounting, Marketing 등은 전공 이름 (MAJOR_NAME)이다. Management는 세가지 전공을 포괄하는 전공 영역 (MAJOR_AREA)의 한 값이며, 그들의 추상값으로서 더 높은 수준에 존재한다. 마찬가지로, Economics는 Macro Economics, Micro Economics, Econometrics를 포괄하는 또 다른 전공 영역의 값으로 그들의 추상값에 해당한다. 추상값 Management와 Economics는 다시 더욱 추상화 되어 2단계 높은 수준에 존재하는 값 Business와도 추상화 관계를 갖는다. 이처럼 한 계층내에서의 상위 수준은 하위 수준보다 좀 더 추상화 된 데이터 표현을 가능하게 함으로써, 상위 수준에 존재하는 값은 하위 수준에 존재하는 값의 추상값으로 이해된다. 따라서 최상위 수준의 값은 계층 내에서 가장 추상화된 값인 반면, 최하위 수준의 값들은 가장 세분화된 값들이다. 전체 추상화 수준의 깊이는 사용자가 문제를 적절하게 묘사하기 위해서 어느 정도의 추상화가 필요한지를 판단하여 자유롭게 결정할 수 있다.

추상값과 세분값 사이에 존재하는 일반화/세분화 관계는 IS-A 관계로 해석될 수 있다. 예를 들어 "Finance IS-A Management" 그리고, "Management IS-A Business"로 표현될 수 있다. 이때, 추상값과 그의 세분값 사이의 사상 원소수 (mapping cardinality)는 일대다 (one-to-many)로 가정한다. 따라서 Finance에 대한 일반화의 결과는 Management가 되고, 반면에 Management에 대한 세분화의 결과는 Finance, Accounting, Marketing 등의 세 개의 값이 된다.

도메인 추상화 계층은 값의 추상화 계층내의 모든 개별 값들이 속해있는 도메인들로 구성된

다. 도메인 추상화 계층에 존재하는 하나의 도메인은 유일한 이름과 계층 내에서의 절대 위치를 가지고 있다. 또한 도메인 이름은 값의 추상화 계층에 존재하는 값들의 추상화 수준을 결정할 수 있게 해준다.

한 도메인 D_1 내의 모든 값들이 도메인 D_2 에 자신의 추상값을 가지고 있을 경우 도메인 D_2 는 D_1 의 수퍼 도메인으로, D_1 은 D_2 의 서브 도메인으로 부르게 된다. 값의 추상화 계층과 마찬가지로, 도메인 추상화 계층에서 D_2 는 D_1 보다 더 높은 곳에 위치하게 된다. 수퍼 도메인 D_2 와 서브 도메인 D_1 사이의 이러한 관계는 값의 일반화/세분화에 대응하여 도메인의 일반화/세분화로 부르기로 한다. D_2 는 D_1 보다 더 일반화된 값들을 포함하고 있기 때문에 D_2 는 D_1 보다 더 일반화된 도메인이라고 할 수 있기 때문이다. <그림 1>의 MAJOR_AREA는 MAJOR_NAME의 수퍼 도메인이고 반면에 MAJOR_GROUP의 서브 도메인이다. 한편, 값과 도메인 사이에는 INSTANCE-OF 관계가 존재한다. 예를 들어 Finance, Accounting, Marketing, Macro Economics 등과 같이 1단계 수준에 존재하는 값들은 도메인 MAJOR_NAME의 인스턴스들이다.

인접한 수퍼 도메인과 서브 도메인 사이의 사상 원소수는 일대일 (one-to-one)로 가정한다. 따라서 한 서브 도메인은 인접한 하나의 수퍼 도메인을 가지며 반대로 하나의 수퍼 도메인은 인접한 하나의 서브 도메인을 갖는다. 하지만, 하나의 서브 도메인은 추상화 정도를 높여감에 따라 여러 개의 수퍼 도메인을 갖게 된다. 추상화 수준의 차이가 n 인 더 높은 수준의 수퍼 도메인내의 추상값들은 서브 도메인내의 세분값의 n 단계 추상값이라고 부른다.

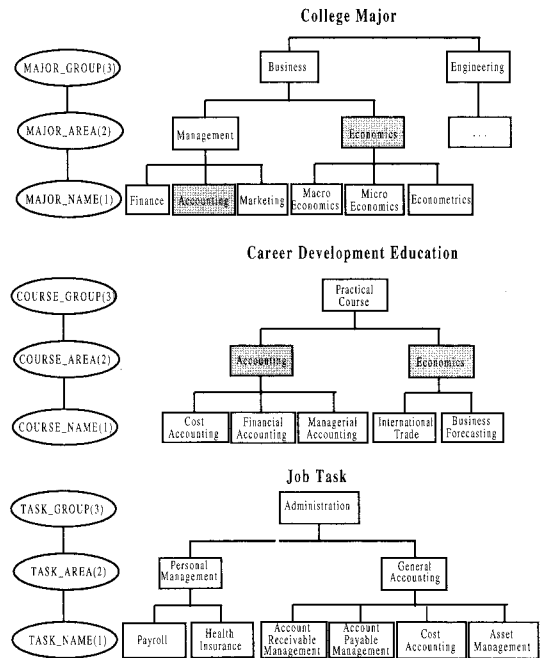
KAH는 개별 응용 문제와 이용 관점에 따라 그 구체적인 내용이 달라지게 되므로, 하나의 데이터베이스에 대하여 다수의 KAH 인스턴스들이 존재할 수 있다. <그림 2>는 동일한 데이터베이스로부터 유도된 세 개의 KAH를 보여준

다. 첫번째 KAH는 College Major 계층을 나타내고, 두번째 KAH는 Career Development Education 계층을 나타내며, 세번째 KAH는 Task 계층을 나타낸다. 필요할 경우, 동일한 데이터베이스로부터 또 다른 네 번째 계층이 추가될 수 있다.

다수의 KAH가 존재하는 경우, 한 도메인은 유일하게 하나의 계층에 존재하는 반면, 한 값은 다수의 계층에 존재할 수 있으며 따라서 다수의 도메인에 존재할 수 있게 된다. 예를 들어 <그림 2>의 한 값 Economics는 College Major 와 Career Development Education 2개의 계층에 존재하므로, 어느 계층에 존재하느냐에 따라서 다른 도메인에 속하게 된다. 다시 말해서, 하나는 College Major 계층에서 MAJOR_AREA 도메인에 존재하고 반면 다른 하나는 Career Development Education 계층의 COURSE_AREA 도메인에 존재하게 된다. 이처럼 Economics가 여러 계층에 존재할 수 있기 때문에 그의 추상값과 세분값은 Economics 값 이름만으로는 유일하게 결정되지 못한다. 주어진 값의 추상값과 세분값을 결정하기 위해서는 그 값의 도메인 정보가 추가적으로 요구된다. 예를 들어 Economics가 MAJOR_AREA 도메인에 존재할 경우 1단계 추상값은 Business가 되고, COURSE_AREA 도메인에 존재할 경우 Practical Course가 된다. 마찬가지로 주어진 값이 어느 도메인에 존재하느냐에 따라 1단계 세분값 집합이 결정된다. 본 논문에서는 이러한 특성을 추상값과 세분값의 도메인 종속성이라 부른다.

결국 KAH는 값의 추상화와 도메인의 추상화를 바탕으로, 기반 데이터베이스에 존재하는 데이터 값들과 그들의 도메인들 사이의 의미론적 관계를 다단계 계층의 형태로 표현할 수 있게 해주는 지식 표현 프레임워크라고 할 수 있다. KAH는 개별 응용 문제와 이용 관점에 따라 그 구체적인 내용이 달라지게 되므로, 하나의 데이터베이스에 대하여 다수의 KAH 인스턴스들이

존재할 수 있다.



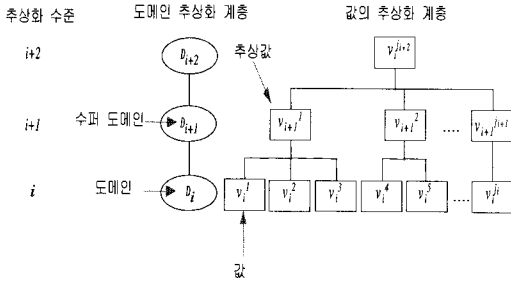
<그림 2> 다수의 지식 추상화 계층의 예

2.2 지식 추상화 계층의 형식화

값의 추상화 계층과 도메인 추상화 계층에 대한 형식화는 <그림 3>과 같다. 그림에서 도메인과 값은 각각 타원과 사각형으로 나타내어진다. 첨자 i 는 추상화 수준을 나타내고 j 는 추상화 수준 i 에 존재하는 값들의 무작위 순번을 나타낸다. 따라서 D_i 는 추상화 수준 i 에 존재하는 도메인을 나타내는 반면, v_i^j 는 도메인 D_i 에 존재하는 j 번째 값을 나타낸다.

도메인 D_i 와 그것의 수퍼 도메인 D_{i+1} 사이의 도메인 추상화 관계는 $D_i \Rightarrow D_{i+1}$ 으로 표현하고, 한 값 v_i^j 와 그것의 추상값 v_{i+1}^{j+1} 사이의 값의 추상화 관계는 $v_i^j \in * v_{i+1}^{j+1}$ 처럼 표현한다. 예를 들어 <그림 1>의 도메인 추상화 계층은 MAJOR_NAME \Rightarrow MAJOR_AREA \Rightarrow

MAJOR_GROUP과 같이 표현되고, Finance, Management, Business사이의 값의 추상화 관계는 $Finance \in^* Management \in^* Business$ 처럼 표현된다.



<그림 3> 지식 추상화 계층의 형식화

Finance \in^* Management \in^* Business 관계에서 나타난 것처럼, Finance와 같은 하나의 세분값은 서로 다른 추상화 수준에 존재하는 Management, Business 등의 다수의 추상값을 갖을 수 있다. 만약, v_{i+n}^{j+i} 이 v_i^j 의 한 추상값이라면 그들 사이의 추상화 수준의 차이는 n 이 된다. 이 때, 추상값 v_{i+n}^{j+i} 을 v_i^j 의 n 단계 추상값으로 부르고 그들의 추상화 수준의 차이를 추상화의 정도라 부르기로 한다. 한 값 v_i^j 와 그의 n 단계 추상값 v_{i+n}^{j+i} 사이의 값의 추상화 관계는 다음의 (식 1)과 같이 표현된다.

$$v_i^j \in^* v_{i+n}^{j+i} \text{ iff } \exists v_{i+1}^{j+1}, \dots, \exists v_{i+n-1}^{j+n-1} \text{ s.t. } v_i^j \in^* v_{i+1}^{j+1} \in^* v_{i+2}^{j+2} \in^* \dots v_{i+n-1}^{j+n-1} \in^* v_{i+n}^{j+n} \dots \text{ (식 1)}$$

(식 1)에서 \in^* 기호는 1단계 값의 추상화 관계 즉, \in^* 를 나타낸다. 따라서, <그림 1>에서의 Finance와 Business사이의 관계는 Finance \in^* Business처럼 나타낼 수 있고, 이는 Business가 Finance의 2단계 추상값이라는 사실

을 의미한다. 이때, 추상값들 사이에는 대수적 가법성이 존재하게 된다. v_{i+n}^{j+i} 이 v_i^j 의 n 단계 추상값이고 v_{i+n+m}^{j+i+m} 이 v_{i+n}^{j+i} 의 m 단계 추상값인 경우, v_{i+n+m}^{j+i+m} 은 v_i^j 의 $n+m$ 단계 추상값이 된

다. 다시 말해서, $v_i^j \in^* v_{i+n}^{j+i}$ 이고 $v_{i+n}^{j+i} \in^* v_{i+n+m}^{j+i+m}$ 이면,

$$v_i^j \in^* v_{i+n+m}^{j+i+m} \text{ 으로 표현할 수 있다.}$$

한편, 도메인 추상화 계층에서는, n 단계 수퍼도메인을 생각할 수 있다. D_{i+n} 이 D_i 의 n 단계 수퍼도메인인 경우, 그들의 추상화 수준 사이의 차이는 n 이 되고 다음의 (식 2)와 같이 정의할 수 있다.

$$D_i \xRightarrow{n} D_{i+n} \text{ s.t. } v_i^j \in^* v_{i+n}^{j+i}, v_i^j \in D_i, v_{i+1}^{j+1} \in D_{i+1}, \dots, v_{i+n}^{j+n} \in D_{i+n} \text{ for all } j_i \dots \text{ (식 2)}$$

(식 2)에서 \xRightarrow{n} 는 1단계 도메인 추상화 즉, $\xRightarrow{1}$ 의 특별한 경우로서 $n=1$ 일 경우를 나타낸다.

<그림 1>의 MAJOR_NAME과 MAJOR_GROUP사이의 관계는 MAJOR_NAME $\xRightarrow{2}$ MAJOR_GROUP

처럼 표현할 수 있다. 값의 추상화 계층과 마찬가지로, 도메인 추상화 계층의 도메인들 사이에도 추상화의 가법성이 존재한다. 다시 말

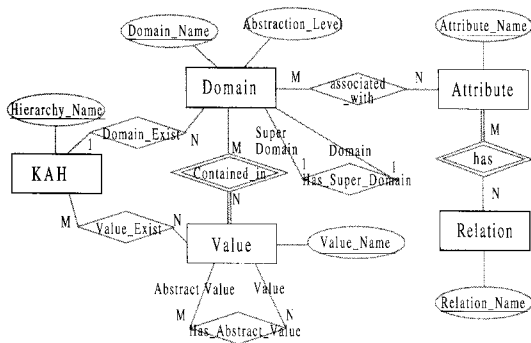
해서 $D_i \xRightarrow{n} D_{i+n}$ 이고 $D_{i+n} \xRightarrow{m} D_{i+n+m}$ 이면 $D_i \xRightarrow{n+m} D_{i+n+m}$ 과 같이 될 수 있다.

III. 지식 추상화 정보 데이터베이스 (KaDB)

KAH가 표현하는 값과 도메인에 관한 의미론적 지식은 협력적 질의처리를 수행하는데 활용될 수 있으며, 이를 위하여 이 논문에서는 관계형 데이터 모델을 바탕으로 추상화 정보 데이터베이스 KaDB를 구축한다. KaDB 설계시 지식의 효과적인 관리와 다양한 협력적 질의처리 연산을 고려해야 한다. 이러한 점들을 고려하여, KaDB에 대한 개체-관계도를 제시하고 관계형 스키마를 설계한다.

3.1 개념적 모델링

KaDB에 대한 개념적 모델링을 위한 개체-관계도는 <그림 4>에 나타난 것처럼 KAH, DOMAIN, VALUE, ATTRIBUTE, RELATION 개체와 그들 사이의 관계로 이루어진다.



<그림 4> 추상화 정보 데이터베이스에 대한 개체-관계도

KAH 개체는 기반 데이터베이스로부터 구축된 KAH를 나타내고 DOMAIN, VALUE 개체는 각각 도메인 추상화 계층과 값의 추상화 계층을 구성하는 도메인과 값을 나타낸다. DOMAIN 개체는 계층에서의 자신의 위치를 결정해 주는 유일한 이름과 추상화 수준을 가지고 있으며, 슈퍼 도메인과 서브 도메인 사이에 Has_Super_Domain 이라는 자기 순환적 일반화 관계를 가지고 있다. 이때, 하나의 슈퍼 도메인

은 하나의 서브 도메인을 가지기 때문에 Has_Super_Domain 관계의 사상 원소수는 일대일(one-to-one)이 된다.

VALUE 개체는 DOMAIN 개체와 contained_in 관계를 갖는다. 하나의 도메인은 보통 다수의 값을 포함하고 있고, 한 값은 다수의 도메인에 속할 수 있으므로 DOMAIN과 VALUE 개체의 사상 원소수는 다대다(many-to-many)가 된다. 한 값은 추상값의 도메인 종속성으로 인하여 하나의 도메인과 함께 자신의 추상값을 결정하게 되므로, VALUE 개체는 DOMAIN 개체와 함께 Has_Abstract_Value라는 자기 순환적 관계를 가지게 된다. DOMAIN 개체는 기반 데이터베이스에 존재하는 릴레이션들의 개별 애트리뷰트에 해당하는 ATTRIBUTE 개체와 관계를 갖는다. 한 애트리뷰트에 관련된 도메인은 자신이 어느 릴레이션에 존재하는가에 따라 달라지므로 RELATION 개체와 함께 Associated_With 관계를 갖는다.

협력적 질의 응답의 관점에서 볼 때, 그림 4는 KaDB가 질의를 어떻게 분석하게 되는지를 보여준다. 기존의 전통적인 질의 처리에서는, 질의 조건에 명시해서 찾고자 하는 목표값이 데이터베이스내의 애트리뷰트 값들과 직접적으로 비교 되어진다. 예를 들어, Finance를 전공한 직원을 찾고자 하는 경우, 질의문에는 목표값 Finance와 전공을 나타내는 애트리뷰트, 이를테면 Major가 명시된다. 이때, Major 애트리뷰트가 취하는 여러 값들과 Finance를 직접 비교하여, 정확히 일치하는 경우 질의의 결과로서 제시한다. 하지만, 협력적 질의 응답에서는 DOMAIN 개체가 목표값과 명시된 애트리뷰트 사이에 매개체로서 작용하여, 질의 조건을 정확히 만족하는 값들이 존재하지 않거나 유사한 값들이 요구되는 경우에, 그러한 유사한 값들이 애트리뷰트 값들과 비교되어 선택될 수 있게 해준다. 다시 말해서, DOMAIN 개체는 기존의 전통적인 질의 처리 과정에서는 가능하지 못했

던 일반화와 세분화 과정을 통하여 다양한 질의 완화 경로를 생성해 주는 것이다.

3.2 KaDB의 관계형 스키마

협력적 질의응답을 위해 구축될 KaDB는 기본 데이터베이스나 KAH에 발생하게 될 진화적 변화에 대하여 유연성을 가지고 적절히 대처할 수 있어야 하며, 협력적 질의 응답을 위해 일반화와 세분화 과정과 같은 질의처리 연산을 용이하게 수행할 수 있어야 한다. 그러한 KaDB의 관계형 스키마를 유도하기 위하여 <그림 4>의 개체-관계도로부터 애트리뷰트들 사이의 함수적 종속성 (functional dependency)을 토대로 Boyce-Codd Normal Form(BCNF) 분할 알고리즘을 적용한다.

함수적 종속관계의 최소 커버 (minimal cover)는, Hierarchy_Name(H), Domain_Name(D), Super_Domain(SD), Value_Name(V), Abstract_Value(AV), Abstraction_Level(L), Attribute_Name(A), Relation_Name(R)로 구성된 전체 애트리뷰트 집합 $R=(H, D, SD, V, AV, L, A, R)$ 에서 다음과 같이 생성된다.

- $D \rightarrow (H, SD, L)$
- $(V, D) \rightarrow (AV)$
- $(R, A) \rightarrow D$

등의 세가지 종속관계가 존재하게 된다.

위의 각 종속관계를 순차적으로 고려하여 전체 애트리뷰트 집합 R을 이원 무손실 조인 분할 (binary lossless-join decomposition)을 수행한 결과 KaDB의 관계형 스키마로 <그림 5>가 얻어진다. 또한 <그림 6>은 <그림 2>에 존재하는 KAH에 대하여 KaDB를 구축한 결과를 보여주고 있다. 설명을 위하여 기본 데이터베이스내의 릴레이션과 그들의 애트리뷰트 이름들이 ATTRIBUTE_MAPPING 릴레이션에 인위적으로 추가되었다.

```
DOMAIN_ABSTRACTION{
    (Domain, Super_Domain, Hierarchy,
     Abstraction_Level)}
VALUE_ABSTRACTION{
    (Value, Domain, Abstract_Value)}
ATTRIBUTE_MAPPING{
    (Relation, Attribute, Domain)}
```

<그림 5> KaDB의 관계형 스키마

KAH에 포함된 의미론적 지식을 관리하기 위하여 구축된 KaDB를 살펴보면 값의 추상화 정보, 도메인 추상화 정보, 그리고 추상화 계층과 기본 데이터베이스 사이의 대응 정보를 포함한 세가지 종류의 정보로 구분할 수 있다.

먼저 첫 번째 DOMAIN_ABSTRACTION 릴레이션은 도메인 추상화 계층에 포함된 의미론적 지식을 관리한다. 하나의 도메인 인스턴스는 자신의 유일한 이름과 슈퍼 도메인 이름, 계층 이름 그리고 계층 내에서의 위치를 나타내는 추상화 수준을 갖는다. 슈퍼 도메인은 서브 도메인과 일대일 (one-to-one) 대응관계를 갖는 일반화 관계를 갖기 때문에, 도메인 이름이 키 애트리뷰트가 되어 주어진 도메인의 이름만으로 슈퍼 도메인을 알아낼 수 있다.

두 번째, VALUE_ABSTRACTION 릴레이션은 값의 추상화 계층에 포함된 의미론적 지식을 관리한다. 하나의 인스턴스 값은 자신의 이름과 자신이 속한 도메인 이름을 갖는다. 세 번째 Abstract_Value 애트리뷰트는 이러한 인스턴스 값에 대한 1단계 추상값을 나타낸다. 하지만 다수의 계층이 존재하는 경우, 동일한 값의 이름이 서로 다르게 사용되어 서로 다른 계층, 서로 다른 도메인에 존재할 수 있으므로, 값의 이름과 자신의 도메인 이름이 VALUE_ABSTR- ACTION 릴레이션의 혼합키가 된다. 다시 말해서 추상값의 도메인 종속성으로 인하여, 값과 자신의 도메인 이름 모두가 추상값을 구하기 위해서 사용된다.

DOMAIN_ABSTRACTION

Domain	Super_Domain	Hierarchy	Abstraction_Level
MAJOR_NAME	MAJOR_AREA	College Major	1
MAJOR_AREA	MAJOR_GROUP	College Major	2
mAJOR_GROUP		College Major	3
COURSE_NAME	COURSE_AREA	Career Development Education	1
COURSE_AREA	COURSE_GROUP	Career Development Education	2
COURSE_GROUP		Career Development Education	3
TASK_NAME	TASK_AREA	Job Task	1
TASK_AREA	TASK_GROUP	Job Task	2
TASK_GROUP		Job Task	3
...

VALUE_ABSTRACTION

Value	Domain	Abstract_value
Finance	MAJOR_NAME	Management
Accounting	MAJOR_NAME	Management
Marketing	MAJOR_NAME	Management
Macro Economics	MAJOR_NAME	Economics
Micro Economics	MAJOR_NAME	Economics
Econometrics	MAJOR_NAME	Economics
Management	MAJOR_AREA	Business
<i>Economics</i>	<i>MAJOR_AREA</i>	<i>Business</i>
Business	MAJOR_GROUP	
Cost Accounting	COURSE_NAME	Accounting
Financial Accounting	COURSE_NAME	Accounting
Managerial Accounting	COURSE_NAME	Accounting
International Trade	COURSE_NAME	Economics
Business Forecasting	COURSE_NAME	Economics
Accounting	COURSE_AREA	Practice Course
<i>Economics</i>	<i>COURSE_AREA</i>	<i>Practice Course</i>
Practice Course	COURSE_GROUP	
Payroll	TASK_NAME	Personnel Management
Health Insurance	TASK_NAME	Personnel Management
Account Receivable Management	TASK_NAME	General Accounting
Account Payable Management	TASK_NAME	General Accounting
Cost Accounting	TASK_NAME	General Accounting
Asset Management	TASK_NAME	General Accounting
Personal Management	TASK_AREA	Administration
General accounting	TASK_AREA	Administration
Administration	TASK_GROUP	
...

ATTRIBUTE_MAPPING

Relation	Attribute	Domain
EMPLOYEE	EMP_ID	ID
EMPLOYEE	MAJOR	MAJOR_NAME
COLLEGE_MAJOR	MAJOR	MAJOR_NAME
TASK_HISTORY	EMP_ID	ID
TASK_HISTORY	TASK_PERFORMED	TASK_NAME
TASK_MAJOR	TASK_NAME	TASK_NAME
TASK_MAJOR	REQUIRED_MAJOR_AREA	MAJOR_AREA
...

<그림 6> 지식 추상화 정보 데이터베이스의 구성 릴레이션

예를 들어, College Major 계층에 존재하는 Accounting은 자신이 속한 도메인이 MAJOR_NAME이기 때문에, VALUE_ABSTRACTION에서 1단계 추상값이 Management가 된다. 동일한 방법으로, 한 추상값에 대해서 그의 세분값 집합을 구할 수 있게 해준다. 즉, College Major 계층에 존재하는 Management는 MAJOR_NAME 도메인에 존재하는 Finance, Accounting, 그리고 Marketing을 세분값으로 갖는다. 추상화 관계의 관점에서 볼 때, VALUE_ABSTRACTION 릴레이션이나 DOMAIN_ABSTRACTION 릴레이션 모두 하나의 튜플에 1단계 추상화 관계만을 표현한다. 임의의 n 단계 추상값은 추상화의 가법성에 기초하여, 이러한 1단계 추상화 관계를 바탕으로 이행적으로 유도될 수 있다.

세 번째, ATTRIBUTE_MAPPING 릴레이션은 기반 데이터베이스에 존재하는 릴레이션 이름과 애트리뷰트 이름, 그리고 그 애트리뷰트가 갖는 도메인을 관리함으로써, 질의문의 의도를 분석하고 VALUE_ABSTRACTION, DOMAIN_ABSTRACTION등과 결합하여 다양한 질의 완화 경로를 생성해 내기 위해 사용된다.

동일한 애트리뷰트 이름이 다수의 릴레이션에서 사용될 수 있기 때문에, 릴레이션과 애트리뷰트의 이름이 혼합기가 된다. 협력적 질의응답과 관련하여 ATTRIBUTE_MAPPING 릴레이션을 활용하는 방법은 4장에서 좀 더 자세하게 다루어질 것이다.

이러한 세가지 릴레이션들의 정규화 정도를 살펴보면, 모든 결정자가 후보키가 되기 때문에, 제 3정규형보다 더욱 강력한 보이스/코드(Boyce Codd) 정규화에 해당한다. 다음 절에서는, KaDB를 사용하여 협력적 질의응답의 기본 처리 과정에 해당하는 한 값이나 도메인에 대한 일반화와 세분화 과정을 어떻게 수행하는지 살펴보기로 한다.

3.3 값의 일반화 과정

KaDB를 이용하여, 질의 조건에 명세된 값에 대해 일반화 과정을 수행하는 것은, KAH의 상위 수준으로 이동해 가면서 추상값을 구해 그 추상값으로 대체해 나가는 것을 말한다. 다시 말해서 한 값에 대한 n 단계 일반화는 그 값의 n 단계 추상값을 구하는 것을 말한다.

한 값과 그 값이 속한 도메인이 주어졌을 때, 1단계 일반화는 단지 VALUE_ABSTRACTION 릴레이션만을 이용하여 수행할 수 있다. 왜냐하면, VALUE_ABSTRACTION 릴레이션의 Abstract_Value 애트리뷰트가 1단계 추상값을 나타내고 있기 때문이다. 이때 주목할 것은, 한 값의 추상값을 구하기 위해서는 추상값의 도메인 종속성으로 인하여, 그 값이 존재하는 도메인도 함께 주어져야 한다는 것이다. 예를 들어, <그림 6>의 VALUE_ABSTRACTION 릴레이션을 이용하여 College Major KAH에 존재하는 Accounting의 1단계 추상값을 구하면, Practice Course가 아니라 Management가 된다. 그 이유는 고려하는 Accounting의 도메인이 Career Development Education KAH에 존재하는 COURSE_AREA가 아니라, College Major KAH에 존재하는 MAJOR_NAME이기 때문이다.

임의의 n 단계 일반화 과정은 VALUE_ABSTRACTION 릴레이션과 DOMAIN_ABSTRACTION 릴레이션 모두를 사용해 이루어진다. 원칙적으로, n 단계 추상값은 1단계 일반화 과정을 n 번 반복함으로써 얻어질 수 있지만, 추상값의 도메인 종속성으로 인하여, 수퍼 도메인 정보를 필요로 하는데, 이것은 DOMAIN_ABSTRACTION 릴레이션을 이용해서 얻을 수 있기 때문이다. 예를 들어, <그림 6>의 VALUE_ABSTRACTION 릴레이션을 이용하여, Career Development Education 계층에 존재하고 COURSE_NAME을 도메인으로 갖는 Cost

Accounting의 2단계 추상값을 구해보자.

먼저 1단계 일반화 과정은 Cost Accounting의 도메인이 COURSE_NAME임을 이용하여, VALUE_ABSTRTCTION 릴레이션으로부터 1단계 추상값 Accounting을 찾아낸다. 하지만, 두 번째로 Accounting에 대한 1단계 일반화 과정을 수행하기 위해서는, Accounting의 도메인 이름을 알아야만 한다. 왜냐하면, VALUE_ABSTRTCTION에는 MAJOR_NAME에 속하는 Accounting과 COURSE_AREA에 속하는 Accounting등 두개의 Accounting이 존재하고, 그 둘 중에서 어느것이 Cost Accounting의 1단계 추상값에 해당하는지를 판단하기 위해서는, Cost Accounting이 속한 도메인 즉, COURSE_NAME의 수퍼 도메인이 MAJOR_NAME과 COURSE_AREA중에서 어느 것인지를 알아내야 한다. 다시 말해서 1단계 추상화를 통해 알아낸 추상값 Accounting의 도메인을 알아야 한다.

이러한 수퍼 도메인 정보는 DOMAIN_ABSTRTCTION 릴레이션을 이용하여 얻어질 수 있고, 그 결과 COURSE_AREA가 수퍼 도메인이 된다. 3단계 추상값 역시, 2단계 추상값과 2단계 추상값이 존재하는 도메인 정보를 가지고 세 번째 1단계 일반화 과정을 수행함으로써 찾아질 수 있다. 이러한 과정을 계속 반복함으로써, 임의의 n 단계 추상값이 구해질 수 있게 된다. KAH에서 상위의 추상화 수준에 존재하는 하나의 추상값은 하위 수준에 존재하는 다수의 세분값에 해당한다. 따라서, 협력적 질의 응답에서 한 추상값을 사용해 질의를 수행하는 것은 다수의 세분값들을 사용해 질의하는 것과 동일한 의미를 갖게 된다.

3.4 값의 세분화 과정

하나의 추상값은 그보다 하위 수준에 존재하는 다수의 세분값들에 대응한다. 주어진 값에

대한 세분화 과정은 KAH의 하위 수준으로 이동해 가면서 임의의 n 단계 세분값을 찾아내는 것을 말한다.

일반화와 유사하게, 세분화 과정 역시 임의의 n 단계 세분값을 찾아내기 위하여 VALUE_ABSTRTCTION 릴레이션과 DOMAIN_ABSTRTCTION 릴레이션 모두를 사용한다. 일반적으로 주어진 값에 대한 1단계 세분값들을 찾아내기 위해서는 추상값과 세분값의 도메인 종속성으로 인하여, 그 세분값들이 존재하게 될 도메인을 먼저 알아내야 한다. 이것은 DOMAIN_ABSTRTCTION 릴레이션을 이용하여 주어진 값이 존재하는 도메인의 서브 도메인을 찾아냄으로써 가능하게 된다. 일단 세분값들이 존재하게 될 도메인이 결정되면, VALUE_ABSTRTCTION 릴레이션을 이용하여 주어진 값의 세분값들을 찾아낼 수 있다.

예를 들어, <그림 6>의 KaDB를 이용하여 <그림 2>에서 College Major KAH에 존재하는 Economics의 세분값을 구해 보기로 하자. 설명한 것처럼, Economics의 1단계 세분값을 구하기 위해서는, 세분값의 도메인 종속성으로 인하여 Economics의 세분값들이 존재하게 될 도메인을 먼저 알아야 한다. 이는 Economics가 속하는 도메인 MAJOR_AREA의 서브 도메인에 해당하며, DOMAIN_ABSTRTCTION 릴레이션을 이용하여 MAJOR_NAME가 됨을 알 수 있다. 그 다음에, VALUE_ABSTRTCTION 릴레이션에서 Domain과 Abstract_Value 애트리뷰트 값으로 각각 MAJOR_NAME과 Economics를 갖는 튜플들을 선택함으로써 Economics의 세분값 Macro Economics, Micro Economics, Econometrics 등을 찾아낸다. n 단계 세분화 과정은 1단계 세분화 과정을 동일하게 n 번 반복함으로써 이루어질 수 있다. 다음 장에서는 일반화와 세분화 과정을 사용하여 전형적인 협력적 질의 응답을 수행하는 메커니즘을 설명한다.

IV. KaDB를 이용한 지식의 관리 및 협력적 질의응답 과정

구축된 KaDB는 KAH에 포함된 지식의 변화를 동적으로 처리하고 다양한 유형의 협력적 질의를 처리할 수 있어야 한다. 이 장에서는 KaDB의 지식 관리 연산들과 협력적 질의처리를 수행하기 위해 필요한 연산들을 논의한다. 마지막으로 기존의 다른 연구들과 비교할 때, 본 논문에서 제시하는 KaDB가 갖는 장점을 논의한다.

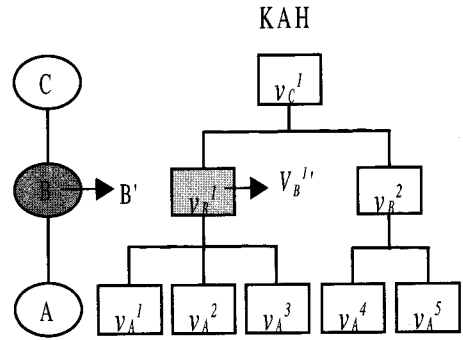
4.1 지식 관리 연산

기본 데이터베이스에 대한 추상화의 의미론적 지식뿐 아니라 KAH의 이용관점이 변해 나감에 따라, KAH에 포함된 지식이 변화해 나간다. KAH에 포함된 지식의 변화는 KAH의 내용상의 변화와 구조적 변화 두 가지로 구분할 수 있다. 내용상의 변화는 KAH 형태상의 구조는 변화하지 않은 채, 도메인이나 값 인스턴스들이 진화적으로 변경되어 나가는 것이다. 반면 구조적인 변화는 한 노드의 차수 즉, 세분값의 갯수가 변한다든지 KAH의 추상화 수준의 깊이가 변화하는 등, 트리 (tree) 구조가 변화하는 것을 의미한다. KaDB가 그러한 변화들을 처리하기 위하여 사용하는 연산들은 다음과 같다.

1) 내용값의 변화

KAH의 내용값의 변화는 값의 추상화 계층과 도메인 추상화 계층 모두에서 일어날 수 있다. 두 경우 모두, 한 내용값이 변화하게 되면 그 내용값 자체는 물론 변화하는 내용값이 관련된

추상화 관계 역시 변화하게 된다. 이러한 내용값의 변화를 <그림 7>에서 v_B^1 이 $v_B^{1'}$ 로, 도메인 B가 B'로 변하는 경우로 설명하기로 하자.

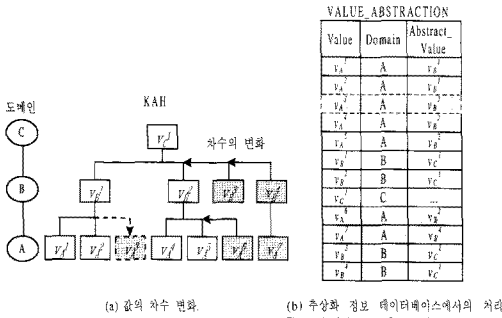


<그림 7> KAH의 내용값 변화.

<그림 7>에서 값 v_B^1 은 그의 자식값 v_A^1, v_A^2, v_A^3 등과 1단계 추상화 관계를 형성하고 있다. 따라서 v_B^1 이 $v_B^{1'}$ 로 변화하는 것은 VALUE_ABSTRACTION 릴레이션의 Value 애트리뷰트 값 중 v_B^1 을 갱신하는 것뿐 아니라, Abstract_Value 애트리뷰트 값 v_B^1 을 갱신함으로써 v_A^1, v_A^2, v_A^3 와의 추상화 관계도 갱신하는 것을 필요로 한다. 도메인 B의 변화에 대한 처리 역시, ATTRIBUTE_MAPPING, VALUE_ABSTRACTION 릴레이션의 Domain 애트리뷰트 값중 B를 갱신하고, DOMAIN_ABSTRACTION 릴레이션의 Domain, Super_Domain 애트리뷰트 값중 B를 갱신한다.

2) 값의 차수 변화

KAH의 두 가지 유형의 구조적 변화 중에서 먼저 한 노드값의 차수 변화를 고려해 보자. 한 도메인은 언제나 하나의 서브 도메인만을 갖게 되므로, 노드의 차수 변화는 값의 추상화 계층에서만 의미를 갖는다. <그림 8(a)>는 $v_A^6, v_A^7, v_B^3, v_B^4$ 가 추가되고, v_A^3 가 제거되는 경우를 나타내고 있다.



<그림 8> 노드값의 차수 변화

이처럼 노드를 추가하고 제거함에 따라, v_B^2 과 v_C^1 의 차수는 증가하고 반면에 v_B^1 의 차수는 감소하게 된다. 이러한 차수의 변화는 <그림 8(b)>와 같이 VALUE_ABSTRACTION 릴레이션에서 처리된다. 직선의 회색은 삽입되는 튜플을 나타내고 점선의 회색은 삭제되는 튜플을 나타낸다.

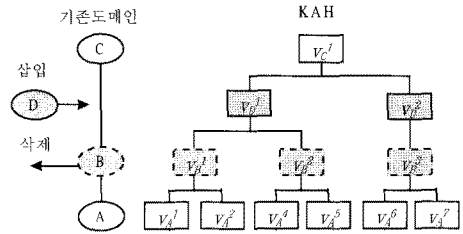
세분값들이 추가되어 한 값의 차수가 증가하는 경우는 추가되는 새로운 값과 그 값이 속한 도메인, 그리고 그 값의 추상값으로 구성된 새로운 튜플을 삽입해야 한다. 반면에 세분값들이 제거됨으로써 한 값의 차수가 감소하는 경우에는, 제거되는 값을 키로 갖는 튜플을 삭제하는 것으로 처리가 가능하다.

3) KAH 깊이의 변화

구조적 변화의 두 번째 경우는, 새로운 도메인이 추가되거나, 기존의 도메인이 삭제됨으로

써 KAH의 깊이가 변화하는 경우이다. <그림 9>는 새로운 도메인 D가 도메인 B와 C 사이에 추가되고, 그 다음에 B가 제거되는 예를 보여주고 있다.

이러한 도메인의 추가와 삭제는 항상 그 도메인에 속한 값들의 추가와 삭제를 동반한다. 따라서, 도메인 추상화 계층에서의 도메인의 추가나 삭제를 처리하기 위한 DOMAIN_ABSTRACTION 릴레이션의 갱신과, 도메인의 추가나 삭제에 의해 후속적으로 발생하는 값의 추가, 삭제를 처리하기 위한 VALUE_ABSTRACTION 릴레이션의 갱신 과정을 필요로 한다.



<그림 9> KAH의 깊이 변화

먼저, 도메인 D가 추가되고, 새로운 값 v_D^1 이 기존의 값들 v_B^1 과 v_C^1 사이에서 삽입되는 경우를 고려해 보자. 다음과 같이 정리될 수 있다. 추가된 도메인과 값들은 해당 릴레이션의 키 애트리뷰트에 해당하므로, 두 릴레이션 모두 키 애트리뷰트 값이 널 (NULL) 이거나, 가공의 값을 갖어야만 하는 삽입 이상이 발생하지 않게 된다.

대상 릴레이션	연산	설명
DOMAIN_ABSTRACTION	Insert	Domain, Super_Domain 애트리뷰트 값이 D, C인 새로운 튜플 삽입
	Update	C를 Super_Domain 애트리뷰트 값으로 갖는 기존의 튜플은 새로운 도메인 D를 그 애트리뷰트 값으로 갖도록 갱신
VALUE_ABSTRACTION	Insert	Value, Abstract_Value 애트리뷰트 값이 각각 v_D^1 , v_C^1 인 새로운 튜플 삽입
	Update	v_C^1 을 Abstract_Value 애트리뷰트 값으로 갖는 기존의 모든 튜플들은 새로운 도메인 v_D^1 을 그 애트리뷰트 값으로 갖도록 갱신된다.

다음으로, 도메인 B가 제거되는 경우도 다음과 같이 도메인의 추가되는 경우와 유사하게 처리된다.

대상 릴레이션	연 산	설 명
DOMAIN_ABSTRACTION	Update	B의 서브 도메인인 A를 Domain 애트리뷰트 값으로 갖는 모든 튜플은 자신의 Super_Domain 애트리뷰트 값으로, 제거되는 도메인 B의 수퍼 도메인 D를 갖도록 갱신
	Delete	B를 Domain 애트리뷰트 값으로 갖는 튜플을 삭제
VALUE_ABSTRACTION	Update	$V_A^1, v_A^2, \dots, v_A^6, v_A^7$ 의 추상값도 v_D^1, v_D^2 로 갱신
	Delete	도메인 B에 속하는 값들 v_B^1, v_B^2, v_B^3 를 삭제하게 된다

4.2 협력적 질의응답 과정

협력적 질의응답은 질의 조건의 유형과 질의 완화 대상의 추상화 수준에 따라 다양한 방법으로 이루어 질 수 있다. 다시 말해서 질의 완화가 선택조건, 조인조건에 적용 되어질 수 있고, 조건내의 애트리뷰트와 목표값들의 추상화 수준이 같을 수도 다를 수도 있다. 본 논문에서는 다음과 같은 네 가지 유형의 전형적인 협력적 질의응답 메커니즘을 제시하고자 한다.

적용조건 조건에서 비교되는 객체	선택조건	조인조건
동일한 추상화 수준의 객체	근사적 선택	근사적 조인
상이한 추상화 수준의 객체	개념적 선택	개념적 조인

세부사항을 설명하기 전에, 다음과 같은 단순화된 인사 정보 데이터베이스를 생각해 보자.

```

EMPLOYEE
  {(id, emp_name, dept, title)}
TASK_HISTORY
  {(id, beginning_date, ending_date,
    task_performed)}
COLLEGE_MAJOR
  {(id, major, entrance_data, graduation_date)}
    
```

```

CAREER_PATH
  {(task, prerequisite_task)}
TASK_MAJOR
  {(task, required_major_area)}
    
```

EMPLOYEE 릴레이션은 직원의 현재 직무상 지위에 관한 정보를 제공하는 반면 TASK_HISTORY 릴레이션은 직원들이 과거에 수행하였던 직무 기록을 제공한다. COLLEGE_MAJOR 릴레이션은 지권의 대학 전공 기록을 포함하고 있으며, CAREER_PATH 릴레이션은 여러 직무들간의 선행 관계를 정의하는 경력 개발 경로를 관리한다. 마지막으로 TASK_MAJOR 릴레이션은 개별 직무와 그 직무를 수행하는데 요구되는 대학 전공 영역 사이의 관계를 기술한다. 이러한 릴레이션들이 <그림 6>에 나타난 KaDB의 기반 데이터베이스를 구성한다고 가정하자.

1) 근사적 선택

근사적 선택이란 질의문의 선택조건에 명시된 목표값을 자신과 근사한 값들의 범위로 확장 시킴으로써 질의조건을 완화하고 탐색 영역을 확장하는 것을 말한다. 예를 들어, 인사 관리자가 특정 직무를 담당하기 위한 Finance 전공자를 찾겠다고 할 때, Finance를 전공한 직원이 존재하지 않거나 혹은 충분한 인원이 존재하지 않는 경우, 탐색 영역을 확장하여 관련된 전공을 갖는 다른 직원들을 찾을 수 있을 것이다. 하나의 질의가 근사적 선택 질의가 되기 위해서는, 질의에서 명시된 애트리뷰트와 목표값

의 도메인이 동일해야만 한다. 한편, KAH에서 한 값의 근사값들을 찾는 것은 그 값의 추상값을 찾아냄으로써 가능하게 된다. 왜냐하면 그 추상값의 세분값들은 주어진 값과 서로 유사한 값들이기 때문이다. 따라서, 근사적 선택 조건 C가 주어졌을 때, 질의완화의 핵심적인 과정은 다음과 같게 된다.

- 질의 일반화

KAH의 상위 수준으로 이동해 가면서 질의 조건에 명시된 목표값을 일반화시켜 추상화 된 질의 조건을 만든다

- 질의 세분화

KAH의 하위 수준으로 이동해 가면서 추상화 된 질의내의 추상값을 세분화시켜, 세분화된 값으로 표현된 정규 질의 형태로 변형시킨다

이러한 단계를 염두에 두고, Finance나 그와 유사한 전공을 갖는 직원을 찾는 근사적 선택 질의는 다음과 같이 작성될 수 있다.

- 근사적 선택 질의

```
select e.emp_name, e.dept
from employee e, college_major c
where c.major = "Finance" and e.id = c.id
```

질의문에서, 근사적 선택은 완화 연산자 =?에 의해서 표현된다. 따라서 질의내의 c.major=? "Finance"의 의미는 college_major 릴레이션 내에서 major 애트리뷰트 값이 "Finance" 이거나 그와 유사한 근사값인 토폴들을 찾는 것이다. 이러한 의미에서 위의 질의문 예에서는 두 도메인이 MAJOR_NAME으로 동일하기 때문에 근사적 선택 질의가 된다. 더욱이, KaDB를 이용한 협력적 질의응답 과정에서 질의문에 명시된 애트리뷰트의 도메인 정보는 질의처리 과정에서 수행되는 일반화, 세분화 과정의 축 역할을 한다. 왜냐하면, 그 도메인 정보가 주어진

목표값과 애트리뷰트 값의 추상값과 세분값 집합을 구하기 위한 질의 완화 경로를 생성해 주기 때문이다. 애트리뷰트의 도메인은 ATTRIBUTE_MAPPING 릴레이션을 통해서, 그리고 목표값의 도메인은 VALUE_ABSTRACTION 릴레이션을 사용하여 알아낼 수 있다. 그러한 도메인 정보를 이용하여 애트리뷰트, 목표값에 대한 일반화 세분화 과정을 수행하기 위한 연산들을 다음과 같이 정의한다.

- Get_Abstract_Value(value)

주어진 값에 대한 1단계 일반화 과정을 수행하여 1단계 추상값을 반환한다.

- Get_Abstract_Value(attribute)

주어진 애트리뷰트가 취할 수 있는 모든 값들의 추상값들을 제공한다. 1단계 추상값과의 선택연산이나 조인 연산을 수행할 수 있게 해준다.

- Get_Specific_Value(value)

주어진 값에 대한 1단계 세분화 과정을 수행하여 1단계 세분값 집합을 반환한다.

상위 수준에 존재하는 추상값은 하위 수준에 존재하는 다수의 세분값들에 대응하므로, 1단계 추상값을 찾는 것은 선택조건에 명시된 목표값의 근사값들을 생성할 수 있게 한다. 따라서, 다음과 같이 1단계 일반화된 질의를 생성해 낸다.

- 질의 일반화

```
select e.emp_name, e.dept
from employee e, college_major c
where c.major is-a Get_Abstract_Value("Finance") and e.id = c.id
```

is-a는 세분값과 추상값 사이의 일반화 관계를 나타낸다. 질의분석을 통하여 Finance의 도메인이 MAJOR_NAME임을 알고 있으므로, 일반화된 질의문의 Get_Abstract_Value("Finance")는 추상값 Management를 반환하며, 따라서 질

의 조건은 다음과 같이 완화된다.

```
select e.emp_name, e.dept
from employee e, college_major c
where c.major is-a "Management"
and e.id = c.id
```

Management를 세분화 하였을 때, c.major가 세분화된 값의 집합 내에 존재하는 값을 취하는지 알아내기 위하여, is-a 연산자는 추상값 Management를 다음과 같이 변환한다.

o 질의 세분화

```
select e.emp_name, e.dept
from employee e, college_major c
where c.major in Get_Specific_Value
("Management") and e.id = c.id
```

추상값 Management의 1단계 세분화는 Finance와 유사한 값들을 포함하는 세분값 집합 {Finance, Accounting, Marketing}을 반환한다. 따라서, 최종적으로 세분화된 질의문은 다음과 같이 작성되고 정규적인 SQL 질의문을 통하여 처리될 수 있다.

```
select e. emp_name, e.dept
from employee e, college_major c
where c.major in
("Finance", "Accounting",
"Marketing") and e.id = c.id
```

2) 개념적 선택

사용자들이 질의문을 작성하려고 할 때, 질의 작성시 필요한 값들을 명확하게 알고 있지 못할 경우가 종종 발생한다. 이것은 인사 관리 시스템이나 생산 데이터 관리 시스템과 같이 데이터의 분류와 범주화가 강조되는 데이터베이스 시스템일 경우에 더욱 그러하다. 예를 들어, 인사 관리자가 일반적인 Business 분야를 전공한 사람들을 찾고자 하지만 Business 분야의 대학 전공 이름에 익숙치 못하여 Accounting,

Finance, Marketing과 같은 값들을 잘 모를 경우, 질의문을 정확하게 작성할 수 없게 된다. 이러한 경우에 해당하는 개념적 질의는 협력적 질의 응답 메카니즘을 이용하여 처리할 수 있다. 따라서 사용자는 "일반적인 Business 분야에 속하는 전공을 갖는 직원을 찾아라"라는 보다 일반적이고 개념적인 질의를 작성할 수 있다. 질의문에 근사화 연산자가 존재하지 않고 선택 조건의 목표값이 애트리뷰트보다 추상화 수준이 높은 추상값이면 개념적 질의로 해석 되어지고, 목표값을 세분화하는 질의 세분화 과정을 거쳐 정규적인 질의로 변형시킨다. 따라서, 개념적 선택 조건 C가 주어졌을 때, 질의완화의 핵심 단계는 다음과 같다.

예를 들어, Business 영역을 전공한 직원을 찾는 개념적 질의는 다음과 같이 작성된다.

o 개념적 선택질의

```
select e. emp_name, e.dept
from employee e, college_major c
where c.major = "Business" and e.id = c.id
```

데이터베이스에서 major 애트리뷰트 값으로 Business가 존재하지 않기 때문에 질의조건을 만족하는 답이 존재하지 않을 뿐더러, major와 Business의 도메인이 다르기 때문에, 정규 질의 응답에서 이러한 질의는 잘못 작성된 질의로 여겨진다. 하지만 협력적 질의응답 메카니즘은 <그림 2>에 나타난 계층에서 Business의 도메인이 major 애트리뷰트의 도메인의 수퍼 도메인임을 발견하고, 그러한 조건을 개념적 질의 조건으로 해석한다. 이러한 개념적 선택 조건은 질의문의 애트리뷰트와 명시된 값 등 두 개의 비교 부분이, 동일한 계층의 서로 다른 도메인에 존재하는 선택 조건일 경우에 해당한다. 위의 질의에서 ATTRIBUTE_MAPPING 릴레이션을 이용하여 c.major 애트리뷰트의 도메인이 MAJOR_NAME임을 알 수 있고, VALUE_ABSTRACTION 릴레이션을 이용하여 Business

의 도메인이 MAJOR_GROUP임을 알 수 있다. 만약 개념적 조건으로 해석하는 것이 타당 하려면, Business의 도메인이 c.major 애트리뷰트의 도메인의 수퍼 도메인이 되어야 한다. 수퍼 도메인과 서브 도메인의 그러한 일반화 관계는 DOMAIN_ABSTRACTION 릴레이션에서 파악이 가능하다. 일단 질의 조건이 개념적 조건으로 판단이 되면, 다음과 같이 자동적으로 세분화 되어진다.

- 질의 세분화의 첫번째 단계


```
select e.emp_name, e.dept
from employee e, college_major c
where c.major in Get_Specialized
_Value("Business") and e.id = c.id
```

추상값 Business의 1단계 세분화는 세분값 집합 {Management, Economics}를 반환한다.

```
select e. emp_name, e.dept
from employee e, college_major c
where c.major in ("Management", "Economics")
and e.id = c.id
```

하지만 계속해서 목표값 Management와 Economics의 도메인이 c.major 애트리뷰트의 수퍼 도메인이므로, 비교되는 두 부분의 도메인이 동일한 도메인이 될 때까지 계층을 따라서 세분화가 계속 진행된다. 두 번째 질의 세분화의 결과로서 다음과 같은 정규 질의문이 생성되어 선택조건의 모든 값들이 동일한 도메인에 존재하게 된다.

- 질의 세분화의 두 번째 단계


```
select e. emp_name, e.dept
from employee e, college_major c
where c.major in ("Finance", "Accounting",
"Marketing", "Macro Economics",
"Micro Economics", "Econometrics") and
e.id = c.id
```

3) 근사적 조인

근사적 선택의 확장으로서, 조인 조건의 두 애트리뷰트가 유사한 값들의 집합으로 추상화 되는 근사적 조인을 생각할 수 있다. 비교되는 두개의 애트리뷰트 모두가 동일한 도메인을 갖는다는 면에서 근사적 선택 질의와 유사하다. 근사적 조인 조건에서는 두 조인 애트리뷰트의 애트리뷰트 값을 조인하는 대신, 조인 애트리뷰트의 값의 근사값을 통해 조인되어 진다. 즉, 두개의 애트리뷰트를 그들의 추상값을 이용하여 조인하여 일상적인 조인의 결과보다 더 넓은 범위의 조인 결과를 생성하게 된다. 예를 들어, Asset Management 직무의 선행 직무와 근사한 직무를 경험한 직원을 찾는 근사적 조인 질의는 다음과 같이 작성된다.

- 근사적 조인 질의


```
select e. emp_name, e.dept, e.title
from employee e, task_history t, career_path c
where e.id = t.id and
c.task = "Asset_Management" and
t.task =? c.prerequisite_task
```

근사적 조인의 질의 완화 연산자로서 =?가 사용된 위의 예제에서 두 조인 애트리뷰트 t.task와 c.prerequisite_task의 도메인 모두는 TASK_NAME이다. 다음에 두개의 애트리뷰트들이 자신이 원래 취하는 값 대신 추상값을 취하도록 다음과 같이 일반화 함으로써 질의 완화가 이루어진다.

- 질의 일반화


```
select e. emp_name, e.dept, e.title
from employee e, task_history t, career_path c
where e.id = t.id and
c.task = "Asset_Management" and
Get_Abstract_Value(t.task) =
Get_Abstract_Value(c.prerequisite_task)
```

조인 애트리뷰트의 추상값에 의거하여 두개

의 릴레이션을 조인하는 것은, KaDB를 이용할 경우 여러 가지 방법으로 이루어질 수 있다. 직관적인 한가지 방법은 VALUE_ABSTRACTION 릴레이션으로부터 Value와 Abstract_Value 애트리뷰트만을 뽑아낸 ABSTRACTION 릴레이션을 도입하는 것이다. 위 예의 경우, TASK_NAME 도메인을 갖는 레코드들이 <그림 10>과 같은 ABSTRACTION 릴레이션으로 추출된다. ABSTRACTION 릴레이션은 각 세분값의 추상값을 제공하므로, 두개의 조인 애트리뷰트 사이의 매개체로서 사용될 수 있고 다음과 같은 정규 질의가 생성되어 진다.

ABSTRACTION

Value	Abstract_Value
Payroll	Personnel Management
Health Insurance	Personnel Management
Account Receivable Management	General Accounting
Account Payable Management	General Accounting
Cost Accounting	General Accounting
Asset Management	General Accounting
.....	

<그림 10> Abstraction 릴레이션

o 질의 세분화

```
select e. emp_name, e.dept, e.title
from employee e, task_history t, career_path c
ABSTRACTION a1, ABSTRACTION a2
where
    e.id = t.id and
    c.task = "Asset Management" and
    c.prerequisite_task = a1.value and t.task
    = a2.value and
    a1.abstract_value = a2.abstract_value
```

4) 개념적 조인

개념적 선택의 확장으로서, 조인 조건의 두개

의 애트리뷰트가 서로 다른 도메인을 갖으며 따라서 서로 다른 추상화 수준에 존재하는 경우를 생각할 수 있다. 개념적 조인 조건에서는 한 조인 애트리뷰트의 도메인이 다른 조인 애트리뷰트의 슈퍼 도메인이 되는 경우로서, 서로 다른 도메인을 갖는 두 애트리뷰트임에도 불구하고 조인이 이루어질 수 있다. 두개의 조인 애트리뷰트의 도메인이 서로 다를 경우 질의는 개념적 조인으로 해석되고, 일반화 과정을 통해서 두개의 도메인을 동일한 도메인으로 변형시켜 정규 조인 질의를 생성한다. Health Insurance를 수행하는데 필요한 전공을 대학 전공으로 하는 직원을 찾는 개념적 조인 질의는 다음과 같다.

o 개념적 조인 질의

```
select e. emp_name, e.dept
from employee e, task t, college_major c
where t.task_name = "Health Insurance" and
t.required_major_area = c.major and
e.id = c.id
```

두 조인 애트리뷰트가 서로 다른 도메인이고 또한 하나의 도메인인 MAJOR_AREA가 다른 도메인 MAJOR_NAME의 슈퍼 도메인이 되기 때문에 질의는 개념적 조인 질의가 되는 것이 타당하다. 따라서 더 낮은 수준의 도메인을 갖는 애트리뷰트 c.major에 대해서 일반화가 다음과 같이 이루어진다.

o 질의 일반화

```
select e. emp_name, e.dept
from employee e, task t, college_major c
where t.task_name = "Health Insurance" and
t.required_major_area =
    Get_Abstract_Value(c.major) and
e.id = c.id
```

앞의 근사적 조인 메카니즘에서 사용되었던 것처럼, 두개의 조인 애트리뷰트를 연결하기 위하여 ABSTRACTION 릴레이션이 다음과 같이

사용된다.

o 질의 세분화

```
select e.emp_name, e.dept, e.title
from employee e, task_major t, college_major c
ABSTRACTION a1
where e.id = c.id and
c.task_name = "Health Insurance" and
c.major = a1.value and
t.required_major_area = a1.abstract_value
```

종합적으로 볼 때, 본 연구에서 제시한 네 가지 질의 유형의 각 메카니즘은 협력적 질의응답의 서로 다른 측면을 다루게 된다. 사용자의 요구와 KAH에 관한 지식에 따라, 하나의 질의에 네 가지 유형의 협력적 질의 응답 조건이 선택적으로 혼합될 수 있고 더 복잡하고 정교한 질의를 만들어 낼 수 있게 된다.

V. 결 론

데이터베이스에서의 정보검색과 의사결정의 효과성을 향상시키기 위하여, 협력적 질의응답에 관한 연구들은 오류처리가 가능하고 지능적인 특성을 갖춘 인터페이스를 제공하기 위한 노력이 오랫동안 이루어져 왔다. 본 논문에서는 보다 효과적인 지식 관리와 다양한 유형의 협력적 질의응답을 지원하기 위하여, 지식 추상화 계층 (KAH)이라는 지식 표현 프레임워크를 제시하고, 그를 바탕으로 한 KaDB를 설계하였다. 또한 KaDB를 이용하여 처리할 수 있는 질의의 유형을 분류하고 각각에 대한 구체적인 질의처리 과정을 제시하였다.

KAH는 값의 추상화와 도메인의 추상화라는 두 가지 종류의 데이터 추상화를 기본 개념으로 갖는 다단계 지식 추상화 계층이다. 값의 추상화는 기반 데이터베이스에 존재하는 데이터 값들 사이의 일반화/세분화 관계를 표현한다. 이러한 값의 추상화를 바탕으로, 최하위 수준에

존재하는 값이 가장 세분화된 값이 되고, 최상위 수준에 존재하는 값이 가장 일반화된 값이 되도록 값의 추상화 계층을 점진적으로 구축해 나간다. 도메인 추상화는 수퍼 도메인과 서브 도메인 사이의 추상화 관계를 표현하고, 값의 추상화 계층과 병렬적으로 도메인 추상화 계층을 구축해 나간다. 협력적 질의응답을 위한 다른 기법들과는 달리, KAH는 값과 도메인 모두에 대한 일반화/세분화 관계를 표현한다. 타입 추상화 계층 (type abstraction hierarchy)[7, 9]은 도메인 추상화를 명시적으로 포함하지 않으며, 객체지향 데이터베이스 기법은 도메인에 해당하는 클래스들 사이의 관계를 표현하긴 하지만, 한 클래스 내의 인스턴스들이 추상화된 여러 개의 인스턴스 그룹으로 분류되어지는 값의 추상화와 같은 개념이 존재하지 않는다.

KAH에 포함된 의미론적 지식을 관리하여 협력적 질의처리를 지원하기 위하여, 관계형 데이터 모델을 바탕으로 KaDB를 설계하였다. 하나의 투플에 다단계 추상화 관계를 표현하는 다른 기법들과는 달리 1단계 추상화 관계만을 표현함으로써 KAH에 포함된 지식의 변화를 동적으로 관리해 줄 수 있다. KAH의 구조와는 독립적으로 KaDB가 구축되고, 두가지 추상화 계층 모두가 정규화된 릴레이션으로 구현되기 때문에, KAH의 구조상의 변화는 KaDB 스키마를 변화시킬 필요 없이 단지 도메인의 삽입, 삭제와 같은 일반적인 데이터베이스 연산을 통해서 처리된다.

KaDB는 관계형 데이터베이스 시스템에 지능적인 인터페이스를 제공하기 위하여, 근사적 질의와 개념적 질의라는 두가지 유형의 모호한 질의를 처리하게 된다. 근사적 질의에서는 어떤 한 데이터 값이, 명시된 질의 조건이 명시된 값과 유사한 값들을 포함하도록 확장된다. 따라서, 명시된 값과 유사한 값들이 집합적으로 비교되고 선택될 수 있다. 개념적 질의에서는 사용자가 기반 데이터베이스에 존재하는 데이터 값들

의 세부사항을 모르더라도 개념적인 수준에서 질의 조건을 작성할 수 있다. 이러한 질의 조건들은 기반 데이터베이스 세부사항에 대한 사용자의 친숙성 정도에 따라 상호 보완적으로 이용된다. 한편, 이러한 두가지 유형의 질의 조건은 다시 선택조건뿐 아니라 조인 조건에 적용되어 근사적 선택, 개념적 선택, 근사적 조인, 개념적 조인등의 네가지 유형의 혼합 조건을 만들어 낸다. 사용자의 필요와 지식 추상화 계층에 대한 이해의 정도 따라, 네가지 유형의 질의 조건은 개별 질의로 작성되거나, 혹은 선택적으로 혼합되어 하나의 질의에 작성될 수 있다. 그 결과로 사용자 인터페이스는 스스로 오류 처리가 가능해지고, 좀 더 지능적인 특성을 갖게 된다.

KaDB는 협력적 질의응답을 위한 다른 기법

들과는 달리, 값의 도메인, 도메인 추상화 관계, 도메인의 추상화 수준과 같은 도메인 관련 지식을 추가적으로 관리함으로써, 좀더 인터랙티브하고 유연한 질의 변형 과정을 수행할 수 있게 해주고, 보다 다양한 형태의 질의에 대한 협력적 질의 응답을 수행할 수 있게 해준다. 타입 추상화 계층 (type abstraction hierarchy)[7, 9]을 바탕으로 한 기법에서는 근사적 선택과 개념적 선택 질의만을 해결하는 반면, KAH 기반 기법은 근사적 조인과 개념적 조인 질의를 포함한 좀더 다양하고 복잡한 질의를 처리할 수 있다. KaDB의 유용성을 검증하기 위한 프로토타입 시스템이 한국과학기술원에서 구현되었으며, 현재 인사 정보 데이터베이스 시스템에 적용하여 테스트중에 있다.

〈참 고 문 헌〉

- [1] B. P. Buckles and Petry, F. E., "A Fuzzy Representation of Data for Relational Databases," *Fuzzy Sets Systems*, Vol. 7, No. 3, May 1982, pp. 213-226
- [2] Y. Cai, N. Cercone, and J. Han, "Attribute-Oriented Induction in Relational Databases," in *Knowledge Discovery in Databases*, AAAI Press/ The MIT Press, 1993.
- [3] Q. Chen, W. Chu, and R. Lee, "Providing Cooperative Answers via Knowledge-Based Type Abstraction and Refinement," in *Proc. of the 5th International Symposium on Methodologies for Intelligent Systems*, Knoxville, TE, 1990.
- [4] L. Cholvy and R. Demolombe, "Querying a Rule Base," in *Proc. 1st Int. Conf. Expert Database Syst.*, 1986, pp. 365-371.
- [5] W. Chu, H. Yang, K. Chiang, M. Minock, G. Chow, and C. Larson, "CoBase: A Scalable and Extensible Cooperative Information System," *International Journal of Intelligence Information Systems*. Vol 6, 1996.
- [6] W. Chu, H. Yang, and G. Chow, "A Cooperative Database System (CoBase) for Query Relaxation," in *Proc. of the Third International Conference on Artificial Intelligence Planning Systems*, Edinburgh, May 1996.
- [7] W. Chu and Q. Chen, "A Structured Approach for Cooperative Query Answering," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 5, October 1994, pp. 738-749.
- [8] W. Chu, R. Lee, and Q. Chen, "Using Type Inference and Induced Rules to

- Provide Intensional Answers," in *Proc. of the 7th International Conference on Data Engineering*, Kobe, Japan, April 1991, pp. 396-403.
- [9] W. Chu, Q. Chen, and R. Lee, "Cooperative Query Answering via Type Abstraction Hierarchy," in S.M. Deen, editor, *Cooperative Knowledge Base System*, North-Holland, Elsevier Science Publishing Co., Inc., 1991, pp. 271-292.
- [10] F. Cuppens and R. Demolombe, "Cooperative Answering: A Methodologies to Provide Intelligent Access to Databases," in *Proc. of 2nd International Conference on Expert Database Systems*, October 1989, pp. 621-643.
- [11] P. Godfrey, J. Minker, and L. Novik, "An Architecture for a Cooperative Database System," in *Proc. of the 1994 International Conference on Applications of Databases*, June, 1994.
- [12] A. Hemerly, M. Casanova, and A. Furtado, "Exploiting User Models to Avoid Misconstruals," *Nonstandard Queries and Nonstandard Answers*, Oxford Science Publications, 1994, pp. 73-98.
- [13] Ichikawa, T., "ARES: A Relational Database with The Capability of Performing Flexible Interpretation of Queries," *IEEE Transactions on Software Engineering*, SE-12, 5, May 1986.
- [14] A. K. Jain and R. C. Dubes, *Algorithms for Cluster Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1988.
- [15] M. J. Minock and W. Chu, "Explanation for Cooperative Information Systems," in *Proc. of Ninth International Symposium on Methodologies for Intelligent Systems*. June 1996.
- [16] A. Motro, "VAGUE: A User Interface to Relational Databases that Permits Vague Queries," *ACM Transactions on Office Information Systems*, Vol. 6, No. 3, July 1988, pp. 187-214.
- [17] A. Motro, "FLEX: A Tolerent and Cooperative User Interface to Databases," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 2, No. 2, June 1990, pp. 231-246.
- [18] A. Motro, "Intensional Answers to Database Queries," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 6, No. 3, June 1994, pp. 444-454.
- [19] H. Prade and Testemale, C., "Generalizing Database Relational Algebra for the Treatment of Incomplete or Uncertain Information and Vague Queries," *Information Science*, Vol. 34, No. 2, Nov. 1984, pp. 115-143.
- [20] A. Scheer, *Enterprise-Wide Data Modeling: Information Systems in Industry*, Springer-Verlag, 1989.
- [21] C. D. Shum and Muntz, "An Information-

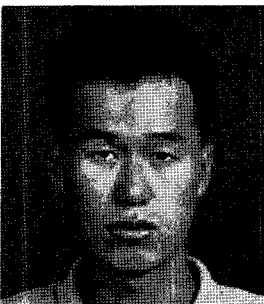
- Theoretic Study on Aggregate Responses," in *Proc. 14th International Conference on Very Large Databases*, Los Altos, CA: Morgan Kaughmann, 1988, pp. 479-490.
- [22] E. Turban, *Decision Support and Expert Systems: Management Support Systems*, Macmillan, 1988.
- [23] J. D. Ullman, *Database and Knowledge-Base Systems, Vol. 1*, Computer Science Press, 1987.
- [24] S. V. Vrbsky and W. S. Liu, "APPROXIMATE-A Query Processor that Produces Monotonically Improving Approximate Answers," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 6, December 1993.
- [25] M. Zemankova and Kandel, A., "Implementing Imprecision in Information Systems," *Information Science*, Vol 37., No. 1, Dec. 1985, pp. 107-141.

◆ 저자소개 ◆



허 순 영 (Huh, Soon-Young)

현재 한국과학기술원 테크노 경영대학원 조교수로 재직중이다. 서울대학교 전자공학과에서 공학사(1981), 한국과학기술원 경영과학과에서 공학석사(1983), 미국 UCLA대학교 경영대학원에서 경영정보시스템 분야로 경영학 박사학위(1992)를 취득하였다. 주요 관심분야는 객체지향 기술을 토대로 한 의사결정지원 시스템, 지능형 금융 트레이딩 시스템, 컴퓨터 기반 그룹웨어 등이다.



문 개 현 (Moon, Kae-Hyun)

현재 한국과학기술원 테크노 경영대학원 박사과정에 재학중이다. 한국과학기술원 경영과학과에서 이학사(1994) 및 공학석사(1996)를 취득하였다. 주요 관심분야는 지식 기반 데이터베이스 시스템, 근사해를 위한 데이터베이스 질의문 처리기법 등이다.