

論文95-32B-1-1

객체지향 데이터베이스의 검색을 위한 논리적 주밍기능을 가진 그래픽 사용자 인터페이스의 설계 및 구현

(Design and Implementation of a Graphic User Interface with Logical Zooming Functions for Browsing of Object-Oriented Databases)

崔 珍 晟 *, 朴 宗 熹 *

(Jin-Sung Choi and Jong-Hee Park)

요 약

복잡한 응용분야에 쓰이는 데이터베이스를 사용자가 효율적으로 검색할 수 있게하는 객체 지향 데이터베이스를 위한 그래픽 사용자 인터페이스를 설계하고 구현한다. 동일한 응용분야에 종사하는 다양한 사용자들의 수준과 필요성에 따른 정보를 객체지향 데이터 모델의 기본구조를 이루는 추상화 기능들에 맞추어 적절히 제공할 수 있게 하는데 설계의 초점을 맞춘다. 특히 논리적인 관점에서의 주밍기법을 도입하여 개별 사용자의 관심영역과 정도에 따라 집중적으로 탐색할 수 있는 기능을 제공한다. 이러한 설계에 관한 목표의 달성여부를 구현의 결과를 통하여 검증한다.

Abstract

A graphic user interface for effectively browsing object-oriented databases in complex applications is designed and implemented. The rationale behind our design lies in enabling the users of various levels and needs to investigate the database according to their respective interests and desired depths. A novel idea in our design is the introduction of zooming techniques from a logical view, which visualize the backbone abstraction concepts of the object-oriented data model. These objectives are verified by evaluating the results of its implementation.

I. 서 론

전통적인 사무처리와 다른 복잡한 문제영역 (예를들

* 正會員, 慶北大學校 電子工學科

(Dept. of Electronic Eng., Kyungbook Nat'l Univ.)

※ 이 논문은 1993년도 학술진흥재단의 공모과제 연구비에 의하여 연구되었음.

接受日字 : 1994年 5月 12日

어 제조업체의 여러 업무나 자동차 구조의 각 부분) 들은 서로 연관을 가지고 하나의 유기적인 전체를 형성한다. 이러한 유기적인 전체의 각 부분들에 관한 문제들은 개별적으로 해결하기 보다 전체적인 관점에서 종합적인 문제해결 방식이 요구된다. 한편 다양한 사용자들이 정보처리에 관여하게 되고 개별적인 사용자들은 서로 다른 관심영역과 정보 이용능력을 가지고 있게 된다.

사용자가 필요한 정보를 얻기 위한 질의를 생성할 때 문제영역이 방대하여 전체 데이터베이스에 대한 파악이 어려워 지체되고 결과적으로 원하는 정보의 효과적인 검색이 어렵다^[11]. 이와같이 비전문 사용자를 포함한 다양한 수준의 사용자들의 요구를 수용하기 위하여 사용편이성(usability)과 사용자의 수용도 (user acceptance)를 고려한 자연스러운 대화방식을 가지는 사용자에게 친숙한 환경으로서의 그래픽 사용자 인터페이스의 개발이 진행되어 왔다^[12,31]. 그래픽 사용자 인터페이스는 정보를 그래픽 도표 형태로 사용자에게 전달하고 메뉴와 그래픽 실체들을 조작하여 정보를 검색하게 한다^[4,5,61].

복잡한 문제영역의 정보표현 방법으로 객체지향 데이터 모델이 그 유용성을 인정받고 있다^[71]. 객체 지향 데이터 모델은 실세계의 모든 실체(entity)를 능동적인 객체(object)의 관점에서 파악함으로써 복잡한 문제영역에서 나타나는 객체들과 그들 사이의 다양한 관계를 효율적으로 표현할 수 있게 한다^[81]. 구체적으로 다양한 수준의 사용자의 요구에 따른 적합한 추상화 수준(abstraction level)을 제공할 수 있는 aggregation hierarchy, 수많은 종류의 객체들을 분류학적으로 조직할 수 있는 generalization hierarchy 등이 객체지향 데이터 모델의 기본구조를 이룬다.

본 논문에서는 복잡한 문제 영역을 표현하고 있는 데이터베이스를 사용자가 효율적으로 검색할 수 있게 하는 객체 지향 데이터베이스를 위한 그래픽 사용자 인터페이스를 개발하고자 한다. 구체적으로 사용자의 관심도에 따라 스키마정보를 다단계로 가변하여 보여 줄 수 있는 기능, 검색하는 과정에서 자신의 검색위치를 파악할 수 있는 기능, 검색의 시작단계에서 전체 스키마를 보여주는 기능등을 제공한다. 특히 논리적인 관점에서의 주밍(zooming)기법을 도입하여 그래픽 실체로 구성된 스키마를 사용자가 직접 조작하여 관심영역과 관심정도에 따라 집중적으로 탐색할 수 있는 기능을 제공한다.

논문의 구성을 보면, 먼저 본 시스템에 관련하여 기존의 시스템들을 데이터모델과 제공하는 검색기능에서 비교 검토한다. 비교검토에서 추출된 기능을 종합하고 미비점을 해결하기 위한 설계를 제시한다. 이러한 설계에 기초한 구현방법을 설명하고 구현 결과를 보여 준다. 본 시스템의 장단점을 검토하고 장래의 연구방향을 제시한다.

II. 그래픽 사용자 인터페이스의 기능 비교

본 논문에서 개발하고자 하는 그래픽 데이터베이스

사용자 인터페이스 시스템과 유사한 목적을 가지는 ISIS (Interface for a Semantic Information System)^[41], GUIDE (Graphical User Interface for Database Exploration)^[51], Ski (Semantics-Knowledgeable Interface)^[61]에서 구현된 그래픽 사용자 인터페이스를 기능적 측면 특히 데이터베이스 스키마를 검색하기 위해 제공되는 주요기능을 중심으로 살펴본다. 기존의 시스템들의 공통적인 미비점으로서 전체 데이터베이스의 조감, 문맥파악 및 검색기능등을 들 수 있다. 구체적으로 화면의 표현능력을 훨씬 초과하는 스키마에 대해 사용자가 검색의 연속성을 유지하면서 관심부분만을 집중하여 볼 수 있게하는 기능을 들 수 있다. 또한 다양한 사용자의 요구를 만족시키고 문맥파악을 효과적으로 하기 위한 여러 각도에서의 검색기능이 제공되지 않고 있다. 아래의 표 1에 그래픽 사용자 인터페이스로서 가져야할 유용한 기능들을 종합하고 기존의 각 시스템들의 해당 기능의 제공여부와 제공 기능의 수준을 비교하여 요약하였다.

표 1. 기존의 그래픽 데이터베이스 사용자 인터페이스가 제공하는 주요 기능 비교

Table 1. The comparison of facilities provided by existing graphic database user interfaces.

| 기 능 | GUIDE | ISIS | Ski |
|---------------------------------|--------------------|--------------------|--------------------|
| 데이터 모델 | E-R 모델 | semantic 모델 | semantic 모델 |
| 주밍 (zooming)기능 | 화면에 나타나는 객체의 수만 가변 | 제공없음 | 제공없음 |
| 전체 스키마 파악기능 | 주요 객체들의 계층구조로 파악 | 스키마 일부만 보여줌 | 스키마 일부만 보여줌 |
| 문맥파악(context-identification) 기능 | 주요 객체들의 계층구조 | 스키마 일부에 국한 | 스키마 일부에 국한 |
| 다중 상속 기능 | 제공없음 | 제공없음 | 제공없음 |
| 불필요한 화면요소 제거 (hide)기능 | 제공 | 제공없음 | 제공없음 |
| focus기능 | 제공 | 제공없음 | 제공 |
| 관련정보를 보여주는 기능 | 에트리뷰트 | 에트리뷰트와 도메인(domain) | 에트리뷰트와 도메인(domain) |
| 여러각도의 검색기능 | 제공없음 | 제공없음 | 제공없음 |
| 스키마수정기능 | 제공없음 | 제공없음 | 제공없음 |

그래픽 사용자 인터페이스는 그래픽으로 정보를 표현하고 사용자와 대화를 통해 사용자와 접속하는 시각적인 정보 교류 방법을 사용한다. 바람직한 그래픽 사

용자 인터페이스는 단순히 그래픽 실체로 보여 주는 것에 그치지 않고 주어진 문제영역의 의미를 충실히 표현해 줄 수 있어야 한다^[9,10]. 본 시스템에서는 위의 표에 열거된 모든 기능을 제공하도록 설계한다. 특히 문맥을 연속적으로 파악할 수 있게 하고 다양한 각도에서 검색할 수 있게 하는데 주안점을 두어 기존의 시스템의 미비점을 획기적으로 개선한다. 이를 위하여 다양한 주밍기능을 이용한다.

기존의 GUI 시스템들이 제공하는 주밍기능은 단순히 화면에 표시되는 그래픽 정보의 범위를 가변하는데 국한되어 있다^[5]. 이러한 주밍기능은 객체들의 물리적인 형태등을 검색하는데는 적당하지만 객체들의 논리적인 구조나 상호 관련성의 범위를 관심범위에 따라 효율적으로 검색하는데는 도움을 주지 못한다.

III. GUI 시스템의 설계

질의를 발생하는데 필요한 스키마의 구성요소와 제공되어야 할 검색기능을 차례로 기술한다. 사용자가 질의를 하는데 필요한 데이터베이스에 관한 주요 정보들은 스키마에 표현되어 있다. 스키마의 구성요소들을 그래픽으로 화면에 나타내어야 할 요소들과 그 이외의 요소들로 나눈다. 본 데이터베이스 사용자 인터페이스가 대상으로 하는 객체지향 데이터베이스의 스키마의 구성요소중 그래픽으로 표현해야 할 구성요소로는 객체 클래스와 그에 속하는 에트리뷰트들, 그리고 객체 클래스들 사이의 관계, 즉 aggregation 계층구조와 generalization 계층구조등을 들 수 있다. 그외에 각 클래스에 속하는 메소드들은 사용자의 요구에 따라 보여줄 수 있도록 배경정보로서 저장된다.

스키마정보들을 효과적으로 검색하기 위한 스키마의 브라우징에 필요한 기능들을 결정한다. 주요기능으로 추상화를 위한 기능, 문맥파악을 위한 기능, 그리고 generalization 계층구조에 따른 검색기능으로 나눌 수 있다. 부가적으로 hide 기능, help 기능, 사용자의 요구에 따라 관련정보를 보여주는 기능등이 제공된다. 이러한 기능들의 토대가 되는 기반 개념으로서 주밍이 다양하게 이용된다. 또한 다중 윈도우를 사용하여 여러 기능에 의해 제공되는 화면을 동시에 보여줄 수 있게 한다.

1) 주밍기능에 기초한 추상화기능(abstraction based on zooming): 복잡한 구조를 가진 정보에 관한 스키마를 검색하는데는 평면적인 화면표시 방법으로는 효과적인 검색이 불가능하다. 이를 해결하기 위해 필요에 따라 전체 스키마중 원하는 범위를 파악하게 하고

관심있는 일부 스키마 영역을 집중적으로 검색하기 위해 두가지 방식의 주밍(zooming)기법을 사용한다. 본 시스템에서의 주밍은 기존의 시스템에서와 달리 객체의 개수와 크기를 가변시키는데 주안이 있지 않고 주밍축적에 따라 aggregation의 레벨이 변화하게 되는 논리적인 관점에서의 주밍이라 할 수 있다.

Top-down방식의 주밍기법은 상위 개념의 객체 클래스로부터 상위개념을 구성하는 하위 개념의 객체 클래스(component object classes)로 이행하면서 검색할 수 있는 방법이다. 이 기법은 사람이 복잡한 객체를 파악하는 방식과 유사하게 객체의 구성요소중 상위개념부터 파악 후 상위개념을 구성하는 하위개념으로 점차 확장해 나가는 검색방법이다. 또한 이 검색방법은 객체 지향 데이터 모델의 aggregation에 의한 abstraction과도 합치하는 검색방법이다. 구체적인 검색방법을 살펴보면 먼저 주밍축척(zooming scale)을 증가하여 가장 상위 개념의 객체 클래스로부터 검색한다. 계속적으로 주밍 축척을 증가시키면 상위 개념의 객체 클래스가 가진 에트리뷰트(와 필요에 따라 메소드와 함께)를 보여준다. 모든 에트리뷰트가 보여진후 주밍 축척이 증가되면 상위개념의 구성 객체 클래스들을 화면에 보여준다. 이와같은 방식으로 주밍 축척에 따라 가시범위(viewing range)을 가변할 수 있으며 사용자의 관심정도에 따라 그 상세정도를 변화시키면서 검색할 수 있다.

Cross-sectional 주밍^[5]은 전체 스키마를 보여주면서 관심영역에 속하지 않은 부분을 화면에서 감추고 관심영역의 스키마를 확대 축소하면서 상세정도를 단계로 가변할 수 있는 기능이다. 구체적인 동작순서를 보면 먼저 전체 스키마를 객체 클래스와 링크들만으로 구성하여 화면에 보여준다. 사용자가 스키마의 관심영역으로 이동후 주밍 축척을 증가하여 관심영역의 에트리뷰트를 검색할 수 있다. 즉, 스키마중 관심영역의 스키마를 보면서 각 객체 클래스의 에트리뷰트와 메소드를 검색할 수 있는 방법이다. 이러한 주밍방식은 위에서 제시한 방식과는 달리 전체 스키마를 보면서 주밍축척에 따라서 각 객체 클래스가 가진 에트리뷰트와 메소드를 검색할 수 있다.

Top-down 주밍은 스키마를 어느 수준으로 알고 있는 사용자에게는 관심영역의 객체 클래스를 보기 위해 상위개념부터 검색하는 과정을 반드시 거쳐야 하는 단점이 있다. 이러한 단점은 cross-sectional 주밍기법으로 완화될 수 있다. 즉, 이 주밍기법은 전체 스키마중 일부단면을 검색할 수 있는 방법으로 top-down 주밍기능과 병행하여 사용할 수 있다. 이와같이 각기 독립된 윈도우에서 두가지 방식의 주밍기법을 사용하는

다중 윈도우를 활용하여 다중 문맥파악을 할 수 있다. 한편, 클래스 구조가 레벨별로 순차적으로 구성되어 있지 않고, 예를들어 상위 레벨에 속하는 클래스와 하위 레벨에 속하는 클래스가 직접 관련이 있는 경우에는 top-down 주밍기능만으로는 문맥의 파악이 불가능하다. 이 경우에 단면을 보는 주밍기능은 top-down 주밍기능의 전체적인 기능으로서 문맥파악을 가능하게 한다.

2) 문맥 파악 (context identification)기능과 focusing기능: 일부영역의 주밍과 다른 부분으로의 panning은 일부영역의 상세정보(local detail)를 제공해 주지만 전체적인 구조를 파악하기는 어렵다. 이런 단점을 극복하기 위해서는 주로 스크롤링을 하여 대규모의 스키마를 검색할 수 있지만 전체적 구조를 파악하기는 힘들다^[11]. 이런 단점을 보완하기 위한 전체 스키마를 보여주는 윈도우와 focus된 스키마 일부를 보여주는 두개의 윈도우를 사용하여 문맥 파악(context identification)과 focusing기능을 동시에 제공한다. 이때 전체 스키마를 객체 클래스자체와 링크로만 간략하게 구성하여 스키마의 골격만을 보여주며, 자신의 검색위치를 파악할 필요가 있을경우 언제든지 독립된 윈도우를 통해 참조할 수 있다.

일반적인 경우로, 최상위 레벨의 객체 클래스가 복수개 있을 경우의 문맥 파악 기능을 살펴보면 cross-sectional 주밍을 하는 윈도우에 최상위 레벨의 객체 클래스들을 보여주고 선택할 수 있도록 하는 것이다. 이렇게 객체 클래스가 선택되고 나면 단수의 문맥 파악 기능이 그대로 적용된다.

3) generalization 계층구조에 관한 검색기능: 특정 객체클래스의 에트리뷰트들의 통합 (merge) 기능은 generalization 계층구조를 따라 상위 객체 클래스로부터 상속되는 에트리뷰트를 보여주기 위한 기능이다. 즉 상위 객체 클래스들에서 상속되는 에트리뷰트들을 해당 객체 클래스에서 합쳐서 보여준다. 실제로 상속된 에트리뷰트의 통합은 스키마에서 임의의 객체 클래스가 선택되면 화면상에 팝업 윈도우가 생성되면서 상속되는 에트리뷰트의 집합을 보여준다. 특히 상속되는 에트리뷰트들이 어느 객체 클래스에서 상속되었는지를 나타내기 위해 객체 클래스 이름과 함께 보여주도록 한다.

4) 관련정보를 보여주는 기능: 객체 클래스에 관련된 정보들을 요약하여 보여주기 위한 헬프 시스템 윈도우는 두개의 서브 윈도우로 구성된다. 서브 윈도우중 하나는 객체 클래스 이름의 리스트를 보여주는 윈도우이고 다른 서브 윈도우는 리스트중 선택된 객체 클래스의 정보를 보여준다.

IV. GUI의 구현

1. 시스템의 구현 환경

본 사용자 인터페이스는 MS-WINDOWS를 운영체제로 한 IBM PC 시스템을 기반으로 한 개발환경에서 구현되었다. 화면상의 윈도우 생성 및 구성은 Borland C++ 3.1에서 제공하는 Object Window Library(OWL)를 사용하였으며 주요 기능들은 C로 구현하였다. OWL은 윈도우즈 애플리케이션이 공통적으로 수행하는 함수들을 제공하는 객체 지향 클래스 라이브러리이다. 본 시스템의 화면을 구성하는데에 OWL에서 제공하는 클래스들을 재정의하여 사용하였다.^[11]

2. 시스템의 구성

시스템의 전체적인 구성은 그림 1에서 보는바와 같다. 시스템을 구성하고 있는 구성요소와 각 구성요소들 사이의 처리순서를 나타내고 있다. 그림 1에 보여진

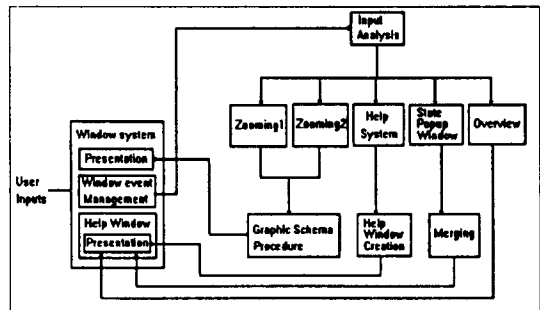


그림 1. 전체 시스템의 구성

Fig. 1. An overall System Structure.

각 구성요소의 기능을 간략히 설명하면 아래와 같다.

(1) 사용자 입력 (User Inputs) : 사용자는 마우스를 이용하여 메뉴, 주밍 박스, 그래픽 실체를 직접 조작하여 입력한다.

(2) 윈도우 시스템 (Window System) : 주어진 스키마 정보를 화면상에 그래픽 형태로 표현해주는 그래픽 서브 시스템으로 사용된다.

가) Presentation module : 사용자의 주밍명령에 의해 변화되는 스키마를 주 윈도우를 통해 보여준다. 스키마가 화면크기보다 크기 때문에 윈도우는 스크롤 기능을 갖는다.

나) Window Event Management module : 사용자의 입력에 의해 발생한 윈도우 이벤트는 각 윈도우 이벤트를 처리하기 위해 정의된 함수들을 수행시킨

다.

다) Help Window module : 임의의 객체 클래스를 선택할 경우 헬프 윈도우를 통해 정보를 보여준다.

(3) Zooming1 module : Top-Down 주밍기능을 수행하는 모듈이다.

(4) Zooming2 module : 전체 스키마의 일부 영역의 단면을 검색을 수행하는 모듈이다.

(5) Graphic Schema Procedure module : 스키마를 확대 축소하기 위해 필요한 각 객체 클래스의 공간적 위치를 계산하고 주밍 축척에 따라 화면상에 에트리뷰트, 메소드, 구성 객체 클래스(component classes)를 보여준다.

(6) Help System module : 두개의 서브 윈도우로 구성된 헬프 윈도우를 통해 에트리뷰트를 검색한다.

(7) Merging module 과 State Popup Window module : 스키마에서 임의의 객체 클래스가 선택될 경우 상속된 모든 에트리뷰트를 통합(merge)하는 기능이다. 통합된 결과는 상태 팝업 윈도우를 통해 보여준다.

(7) Overview module : 독립된 윈도우에서 전체 스키마를 조감할 수 해준다.

3. 사용자 데이터베이스에 관한 정보의 처리

사용자 데이터베이스의 스키마를 표현하기 위해 각 객체 클래스마다 아래와 같은 요소들이 필요하다. 즉,

- 1) 레코드 번호, 2) generalization 계층링크수와 generalization 링크로 연결된 객체 클래스의 레코드 번호, 3) aggregation 계층링크수와 aggregation 링크로 연결된 하위 객체 클래스들의 레코드 번호, 4) 계층 타입 플래그: 객체 클래스가 generalization, aggregation 계층링크중 어느 계층링크에 의해 파생되었는가를 표시, 5) 상위 객체 클래스의 레코드 번호 : 상위 객체 클래스가 복수개일 경우는 다중상속에 해당, 6) 스키마에서 나타내야할 객체 클래스의 이름, 에트리뷰트들의 이름, 메소드들의 이름등이다.

스키마에 관한 정보에서 그래픽 실체로 표현되는 요소들과 그래픽 실체와의 관계는 다음과 같다. 객체 클래스를 사각형으로 표현하고 에트리뷰트는 사각형과 라인으로 연결되는 라운드 사각형으로 표현한다. 이와 같이 표현되는 에트리뷰트의 수는 세개로 하고 나머지 에트리뷰트는 사각형내에 나타낸다. 그리고 각 객체 클래스의 이름과 에트리뷰트를 사각형과 라운드 사각형의 중심에 표현한다. 만약 (다음절에서 언급할)주밍에 의해 사각형이 확대 축소될 경우 사각형내에 객체 클래스의 이름, 에트리뷰트, 메소드의 수를 적절히 가변하여 나타낸다.

객체 클래스들 상호간의 관계표현은 공간적 위치와 객체 클래스들을 연결하는 선의 색을 달리하여 표현한다. 구체적인 방법으로 상위 개념의 객체 클래스는 하위 개념의 객체 클래스들보다 공간적으로 상위의 위치에 표시한다. 그리고 generalization과 aggregation의 계층관계에 속하는 하위 객체 클래스들을 같은 레벨의 위치에 배열한후 generalization과 aggregation의 계층관계를 상위 객체 클래스와 선으로 연결한다.

4. 스키마 검색 기능의 구현을 위한 알고리즘

주밍 알고리즘에서는 입력을 주밍 이벤트로 하였다. 사용자가 주밍 아이콘을 선택하거나 메뉴를 선택하면 이벤트 형태로 표현되고 주밍 축척이 가변되기 때문이다.

Algorithm1. TOP_DOWN_ZOOM(zoom event)

```

begin
  if(주밍 이벤트 발생) /* 주밍 이벤트 발생 */
  then
    주밍 축척의 증가:
    if( 주밍 축척이 증가 혹은 감소) then
      그래픽 실체의 파라미터 변경:
      객체 클래스들의 좌표계산:
      if( 다음 레벨의 구성 객체클래스를
        보여야 하는 축척에 도달) then
        구성 객체 클래스의 좌표계산:
      else
        화면에 보여줄 정보량결정 결정:
        화면에 변경된 스키마를 보여준다:
    end
  
```

Top-down 주밍이 수행되는 절차를 보면, 주밍 이벤트에 의해 주밍 축척이 증가하면 Algorithm1의 과정을 거쳐 그래픽 실체들의 크기, 보여지는 에트리뷰트, 메소드가 증가하며 상위 객체 클래스들과 generalization과 aggregation 계층링크로 연결되는 하위 객체 클래스들이 점점 많아진다. 아래의 Algorithm1은 top-down 주밍의 처리 단계를 보여준다. Cross-sectional 주밍은 전체 스키마를 보여주면서 화면상에서 보여지는 에트리뷰트와 메소드를 다섯단계로 가변하면서 볼 수 있다. 주밍 이벤트가 입력이 되면 Algorithm2의 과정을 거쳐 확대 또는 축소된 스키마가 보여진다.

문맥 파악(context identification) 기능과 focusing기능을 cross-sectional 주밍을 하는 윈도우와 전체 스키마를 보여주는 윈도우를 OWL의 TWindow로 계승하여 스크롤기능을 갖는 윈도우로 재정의하여 구성하였다.

스크롤기능은 수직, 수평방향으로 이동하여 전체 스키

```

Algorithm2. CROSS_ZOOM(zoom event)
begin
  if (주밍 이벤트 발생) then
    주밍 축척의 증가:
    if (주밍축척이 증가나 감소) then
      그래픽 실체의 파라미터 변경:
      객체 클래스들의 좌표계산:
      화면에 보여줄 정보량을 결정:
      화면에 변경된 스키마를 보여준다:
  end
  end
  
```

마중 일부만을 화면에 focus할 수 있는 기능으로 사용된다.

```

Algorithm3. SINGLE_INHERIT()/* 단일상속 */
begin
  if ( 스키마상에서 특정 객체 클래스가 선택 )
  then
    화면상에서 선택된 객체 클래스 결정:
    선택된 객체 클래스의 레코드번호 입력:
    do
    선택된 클래스의 스키마 정보 읽기:
    스키마 정보중 상위 클래스의 필드읽기:
    큐에 상위 클래스의 레코드번호 저장:
    큐의 포인터 변수를 증가:
  if ( 다중상속 ) then
    if (저장된 클래스의 상위 클래스가 동일)
    then
      상위 클래스의 레코드 번호로 변경:
    else 상위 클래스의 레코드 번호로 변경:
    while ( not 최상위 레벨의 객체 클래스 )
    do /* 애트리뷰트를 통합하는 과정 */
      큐에서 포인터변수가 가리키는 레코드 읽음:
      읽은 레코드번호의 스키마정보를 읽음:
      화면에 클래스의 이름과 애트리뷰트 보임:
      큐의 포인터 변수를 감소:
    while( 큐의 포인터 변수가 NULL이 아닐경우)
  end.
  end.
  
```

한편 전체 스키마를 보여주는 윈도우는 주윈도우에 속한 차일드 윈도우이며 언제든지 팝업하여 참조할 수 있다.

Generalization 계층구조로 표현된 객체 클래스에 관한 정보의 표현을 위하여 generalization계층링크를 따라 상위 객체 클래스들의 애트리뷰트를 상속하여 표현해 준다. 스키마상에서 임의의 객체 클래스에 대한 애트리뷰트들은 Algorithm3과 같은 과정을 통해 통합되어 보여진다. 다중상속의 경우도 고려된다.

V. 구현결과 및 검토

구현된 GUI 시스템의 동작과정을 보임으로써 구현 결과를 검토한다. 주 윈도우의 메뉴는 그림 2과 같다.

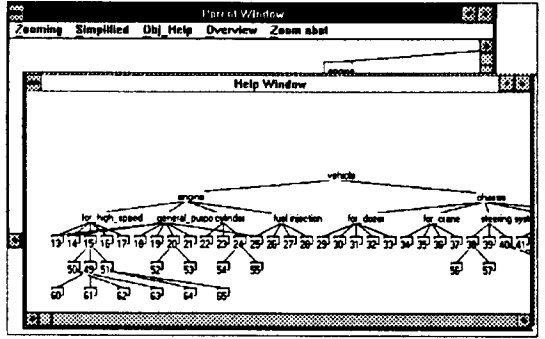


그림 3. 데이터베이스 스키마의 전체구조 화면
Fig. 3. Overview of Database Schema.

구현된 결과를 설명하기 위하여 복잡한 구조를 가진 대표적인 데이터인 자동차구조를 예로써 사용한다. 먼저 전체 스키마의 대략적인 구성을 파악하기 위해 메뉴중 Overview를 선택한다. 그 결과 화면에 그림 3와 같이 전체 스키마를 보여주는 윈도우가 팝업된다.

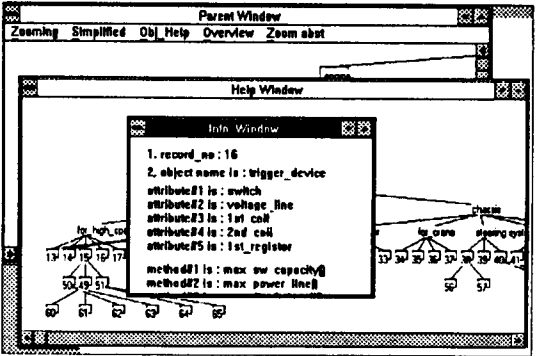


그림 4. 애트리뷰트를 보여주는 팝업 화면
Fig. 4. Popup Window with Attributes of the Selected Object Class.

전체 스키마의 전체 구조를 보여주는 윈도우와 병행

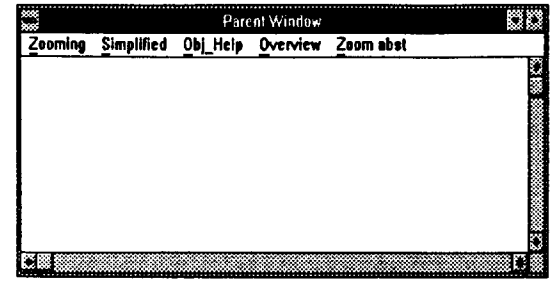


그림 2. 초기화면
Fig. 2. Main Window.

점적으로 검색하면서 자신의 검색위치를 파악할 수 있다.

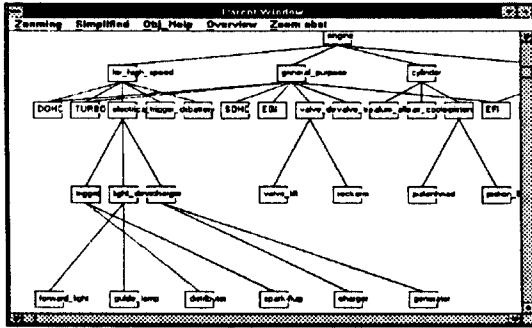


그림 5. 간략화된 데이터베이스 스키마
Fig. 5. Simplified Database Schema.

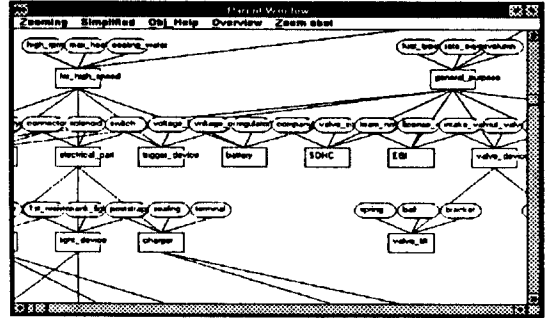


그림 7. Cross-Sectional 주밍의 결과화면(축적 = 1)
Fig. 7. Display of Schem As a Result of Cross-Sectional Zooming (scale=1)

Fig. 7. Display of Schem As a Result of Cross-Sectional Zooming (scale=1)

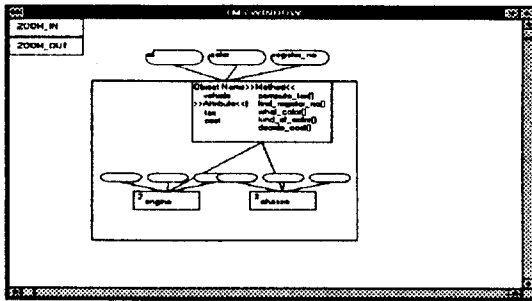


그림 6(a). 탑-다운 주밍의 결과화면(축적=6)
Fig. 6(a). Display of Schema As a Result of Top-Down zooming (scale=6)

즉 주윈도우와 전체 스키마의 전체구조를 보여주는 두 개의 윈도우를 사용하여 문맥 파악과 focus 기능을 제공한다. 스키마의 전체구조를 보여주는 헬프 윈도우는 주 윈도우의 스키마를 검색하는 과정에서 언제든지 생성 참고할 수 있다. 부가적인 기능으로 화면상 스키마 상에서 마우스를 이용하여 관심있는 객체 클래스를 선택하면 에트리뷰트와 메소드를 그림 4에서와 같이 검색할 수 있다.

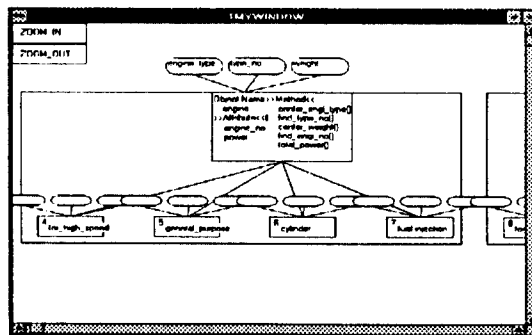


그림 6(b). 탑-다운 주밍의 결과화면(축적=11)
Fig. 6(b). Display of Schema As a Result of Top-Down Zooming (scale=11)

하여 주 윈도우에서 사용자가 관심영역의 스키마를 중

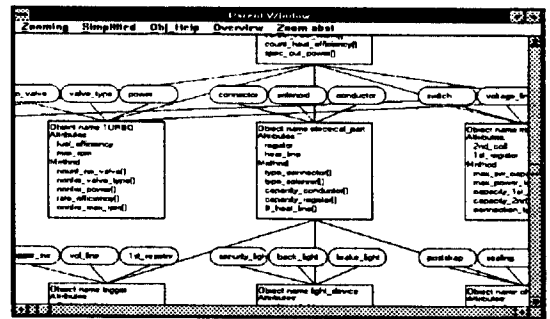


그림 8. Cross-Sectional 주밍의 결과 화면(축적=5)
Fig. 8. Display of Schem As a Result of Cross-Sectional Zooming (scale=5)

Fig. 8. Display of Schem As a Result of Cross-Sectional Zooming (scale=5)

Top-down 주밍기능에 관해 주 윈도우에서 제공하는 메뉴중 zoom_abs를 선택하면 서브 윈도우가 생성된다. 생성된 서브 윈도우의 주밍 박스내에서는 그림 5에서와 같이 zoom-in과 zoom-out 기능이 있다. 탑레벨의 객체 클래스만이 보여지는 초기 단계에서 계속

적인 zoom-in을 선택하면 탑 레벨의 객체 클래스의 에트리뷰트와 메소드가 보여진 후에 하위개념의 객체 클래스들을 보여준다. 이와같은 과정을 반복하면서 화면에 나타나는 객체 클래스의 수는 점점 많아지면서 더욱 세부적인 객체 클래스의 정보를 검색할 수 있다. 그림 6는 top-down 주밍과정의 전형적인 예를 보여주고 있다.

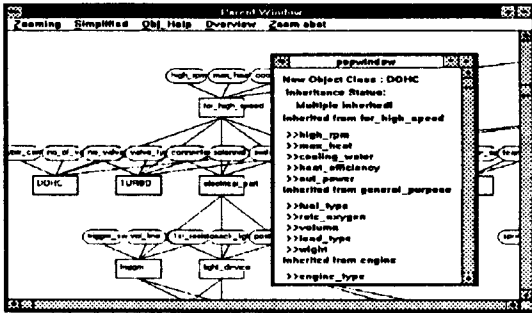


그림 9. 상태 팝업 윈도우
Fig. 9. State Popup Window.

Cross-sectional 주밍 기능을 선택한 후 주 윈도우의 Zooming 메뉴에서 scale1을 선택하면 그림 7에서와 같이 주 윈도우상에 객체 클래스의 이름, 에트리뷰트, 링크들로 구성된 스키마가 보여진다. 사용자가 스크롤 기능을 이용하여 관심 스키마영역으로 이동후 scale1에서 scale5 사이에서 상세정도를 가변할 수 있다. 그림 8은 scale5을 선택한 결과를 화면에 보여준다. 이와같은 방식으로 전체 스키마를 보여주면서 관심 영역의 각 객체 클래스의 에트리뷰트, 메소드를 검색할 수 있는 기능이다.

주 윈도우에 보여지는 스키마상에서 객체 클래스를 선택하면 상태 팝업 윈도우를 통하여 그림 9에서와 같이 사용자가 관심을 가진 객체 클래스로 상속되는 부모 객체 클래스의 이름과 에트리뷰트를 검색할 수 있다. 팝업 윈도우의 제한된 윈도우 크기로 통합된 에트리뷰트와 객체 클래스의 이름을 수직으로 배열하였으며 팝업 윈도우 영역을 수직으로 스크롤하여 모든 객체 클래스의 정보를 볼 수 있다.

헬프시스템은 사용자가 알고자 하는 객체 클래스의 이름을 미리 알고 있는 경우에 유용하다.

알파벳 순서로 배열된 객체 클래스 리스트를 보여주는 윈도우에서 임의의 객체 클래스를 마우스로 선택하여 그 객체의 에트리뷰트를 검색할 수 있다. 그림 10에서 보여지는 헬프 시스템의 윈도우에서 상위의 서브 윈도우는 객체 클래스의 리스트를 보여주고 아래의 윈

도우는 선택된 객체 클래스의 에트리뷰트를 보여준다.

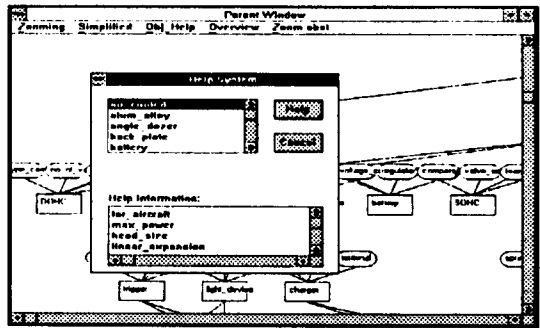


그림 10. 헬프 시스템 윈도우
Fig. 10. Help System Window.

구현의 결과에서 top-down 주밍은 상위 개념의 객체 클래스를 파악하고 사용자의 관심정도에 따라 하위 개념으로 스키마의 상세정도를 조정할 수 있는 기능을 확인하였다. 특히 cross-sectional 주밍을 병행하여 사용함으로써 문제영역의 전반적인 파악과 관심영역의 집중적으로 파악하기에 매우 효과적임이 실증되었다. 두가지의 주밍기법을 다양한 문제영역을 효과적으로 표현하고 조직화하기에 적합한 객체 지향 데이터 모델 기반의 데이터베이스에 적용함으로써 시스템의 유용성을 한층 증가시켰다. 결과적으로 자동차의 설계등 복잡한 데이터를 사용하는 응용분야들에서 수 많은 사용자의 다양한 검색요구를 개별적인 관심부분에 따라 효율적으로 충족할 수 있는 기능을 제공할 수 있음을 보여주었다.

주밍에 관련하여 보완이 필요한 기능으로는 화면에서 보여지는 스키마 정보를 선택적으로 간략화 할 수 있는 기능이다. 구체적으로 전체 스키마에서 하위 레벨의 객체 클래스로 주밍축척이 증가한 경우 화면에서 보여 주어야 할 상하위 객체 클래스가 제한된 화면의 범위를 벗어나는 경우가 많다. 이런 문제의 해결을 위해 generalization과 aggregation 계층 링크별로 화면에서 감추고 보여줄 수 있는 기능이 효과적일 것이다. 이러한 기능은 선택된 객체 클래스가 포함된 경로만을 따라 검색할 수 있는 방법과 병행해서 사용하면 효율적이 될 것이다.

VI. 결 론

지금까지 주밍기능을 가진 객체 지향 데이터베이스의 스키마 검색을 위한 그래픽 사용자 인터페이스를

설계하고 구현하였다. 구현과정으로 먼저 사용자가 데이터베이스 전반에 대한 이해를 쉽게 하도록 하기 위해 스키마를 그래픽 실체로 구성하여 화면에 보여주었다. 특히 대규모 스키마의 경우 평면적인 화면표시방법으로는 효과적인 검색이 어렵다. 이를 보완 하기위해 두가지 방식의 주밍기법을 구현하였다.

첫째, top-down 주밍은 상위 개념의 객체 클래스를 파악한 후 상위 개념을 구성하는 하위 개념으로 관심영역을 확장해 나가는 검색방식이다. 이러한 방식의 주밍은 객체 지향 데이터모델의 aggregation에 의한 abstraction과도 합치하는 검색방법으로 특히 사용자의 관심정도에 따라 상세정도를 달리하면서 가시범위(viewing range)을 가변할 수 있는 매우 효과적인 검색방법이다. 둘째, cross-sectional 주밍은 전체 스키마중 일부단면을 볼 수 있는 검색방법이다. 이와같이 두가지 방식의 주밍을 병행하여 사용하면 문제영역의 전반적인 파악과 관심영역의 집중적으로 파악하기에 매우 효과적이다. 두가지의 주밍기법을 다양한 문제영역을 효과적으로 표현하고 조직화하기에 적합한 객체 지향 데이터 모델기반의 데이터베이스에 적용함으로써 시스템의 유용성을 한층 증가시켰다.

한편 구현된 기능외에 몇가지 기능적 보완이 필요하다. 첫째, 그래픽 상태에서 스키마를 편집할 수 있는 스키마 편집기능의 추가이다. 둘째, 화면에서 보여지는 스키마 정보를 선택적으로 간략화 할 수 있는 기능의 추가가 필요하다. 구체적으로 generalization과 aggregation 계층 링크별로 화면에서 감추고 보여줄 수 있는 기능이며 전체 스키마에서 하위 레벨의 객체 클래스로 주밍축척이 증가한 경우 화면에서 보여 주어야 할 상하위 객체 클래스가 제한된 화면의 범위를 벗어나는 경우가 많다. 이러한 점의 해소를 위해 임의의 상위 객체 클래스와 상하위 관계를 갖는 객체 클래스에 적용할 수 있도록 top-down 주밍기법을 보완하는 것이 필요하다. 이 방식은 관심있는 객체 클래스가 포함된 경로만을 따라 검색할 수 있는 방법과 병행해서 사용하면 효율적이다.

참 고 문 헌

[1] A. Motro, "BAROQUE:A Browser for

Relational Databases," ACM Transactions on Office Information Systems, Vol. 4, No. 2, Apr. 1986, pp. 164-181.

- [2] 강일선, 엄기현, "데이터베이스를 브라우징하기 위한 사용자 인터페이스 설계," '91 동계 데이터베이스 학술대회 논문집 제 5권 제1호 pp. 239-257
- [3] D. Mandelkern, "GUIs: the Next Generation," Comm. of the ACM, Vol. 36, No. 4, Apr. 1993, pp. 37-39.
- [4] J. W. Davison and S. B. Zdonik, "A Visual Interface for a Database with Version Management," ACM Transactions on Office Information Systems, Vol. 4, No. 3, July. 1986, pp. 226-232.
- [5] Harry K. T. and I. Kuo, "GUIDE: Graphical User Interface for Database Exploration," Proc. of the 11th International Conf. on Very Large Databases, 1985, pp. 22-32.
- [6] R. King and S. Velville, "Ski: A Semantics-Knowledgeable Interface," Proc. of the 10th International Conf. on Very Large Databases, Aug. 1984, pp. 30-33.
- [7] W. Kim, "Introduction to Object-Oriented Databases," The MIT Press, 1990, pp. 1-20.
- [8] J. G. Hughes, "Object-Oriented Databases," Prentice-hall, 1991.
- [9] A. Morse and G. Reynolds, "Overcoming Current Growth Limits in UI Development," Comm. of the ACM, Vol. 36, No. 4, Apr. 1993, pp. 78-79.
- [10] 한민규, "인간과 컴퓨터와의 새로운 만남," Computer Magazine Sep. 1992 pp. 128-130.
- [11] User's guide, "ObjectWindows for C++ Resource Workshop," Borland, 1991.

저 자 소 개



崔 珍 晟(正會員)

1966년 10월 25일생. 1989년 2월 경북대학교 전자공학과 졸업 (공학사). 1994년 경북대학교 대학원 졸업 (공학 석사). 주관심 분야는 멀티미디어, 데이터베이스



朴 宗 熹(正會員)

1979년 서울대학교 졸업 (공학사). 1981년 한국과학원 졸업 (공학석사). 1990년 Univ. of Florida 졸업 (공학박사). 현재 경북대학교 전자공학과 조교수. 관심분야는 지능형 정보 시스템, 멀티미디어 응용, 분산 데이터 처리 시스템