데이터베이스 정보의 시각화 방법과 그 표현 언어들에 관한 연구

A Study of Visualization Methods and Languages
for Presenting Database Information

김 성 곤 (Sungkon Kim)
(주) 디지탈 드림 리서치 소프트 부설 연구소

 ● Received : Jun. 7. 2000 ; Accepted : Jun. 14. 2000

(要約)
최근 미디어 형태의 변화와 컴퓨터 그래픽 기술의 발전으로 데이터베이스 정보의 삼차원 동적 표현의 시도가 이루어지고 있다. 그러나 그 정보를 표현함에 있어서의 방법과 널리 디자이너 사이에서 통용되어지는 사용 언어에 대하여 연구는 부족한 실정이다. 따라서 본 논문에서는 삼차원 동적인 정보표현의 방법은 어떠한 것이 존재하며 그리고 그 방법을 사용, 설명하기 위한 언어는 무엇이 있는가에 대해 조사 연구하였다.
연구는 크게 네 부분으로 구성되어진다. 먼저 문헌 연구를 통하여 이차원 정적인 정보 표현에서 삼차원의 동적인 정보 표현이 가능하게 된 정보표현의 기초와 그 기능들을 정리 소개하였다. 즉, 컴퓨터 이차원 화면 속에 어떻게 삼차원적인 표현이 가능한가와 컴퓨터 그래픽 기술의 발달로 어떤 기능이 개선되어 졌는가를 소개하였다. 그리고 두번째로 데이터베이스와 그래픽 모델 사이에 이 논문에서 논하고자하는 정보표현 모델이 존재한다. 데이터베이스의 다이아그램 계층구조와 그래픽 모델을 설계할 때 만들어지는 씬그래프 계층구조 사이를 연결할 수 있는 정보 표현 구조 디자인 방법과 그 사용언어들에 대하여 제시하였다. 그리고 이 정보 표현 모델의 구조적 변화를 통해 어떻게 사용자가 필요한 정보를 관찰할 수 있는가를 논하였다. 세 번째로 그래픽 표현 기술을 분석하여 디자이너들이 그래픽 정보표현 모델 개발 시 참고 할 수 있는 정보 표현의 방법과 그 사용 언어들을 제시하였다. 마지막으로 구체적 사례 연구를 통하여 정보 표현 방법과 사용되어진 언어들의 예를 보이고 검증하였다.

(Abstract)
Recently, as the form of media has been various and computer graphic technology has developed, it has been possible and attempted to present database information 3-dimensionally and dynamically. However, there was not many studies done regarding how to present 3-dimensional and dynamic information. Therefore, this study was conducted to develop presentation techniques and languages of 3-dimensional and dynamic information.
This paper consists of four sections. First, it was introduced by secondary research that fundamental knowledge and functions for presenting the information by the computer graphics. In other words, it was stated what is possible to present 3-dimensional information on the 2-dimensional screen and which functions becomes better as computer graphic technology has developed. Second, computer graphic model that present database information should also have hierarchical structure that database information has. Thus, it was discussed how to present information regarding how to make a structure of graphic model that present database information. There are two ways to present information - by changing structure of information and by changing graphical presentation techniques. Third, all the graphical presentation language was collected and explained. Finally, the case study of presenting statistical data for baseball players demonstrated concrete example of using graphical presentation language.

1. 서론

1-1. 연구 배경

미디어 형태의 변화는 정보 표현 트랜드(Trend)의 변화에 많은 영향을 주었다. 다양한 미디어 매개체를 사용함으로써 정적으로만 표현되어지던 데이터들이 동적으로 표현 될 수 있으며, 컴퓨터 그래픽의 기술 발전과 더불어 실시간 3차원 모델을 보여줄 수 있게 되었다. 정보 표현은 이차원적 표현에서 삼차원적 표현으로 그리고 정적 표현에서 동적 표현으로 변화되었다.

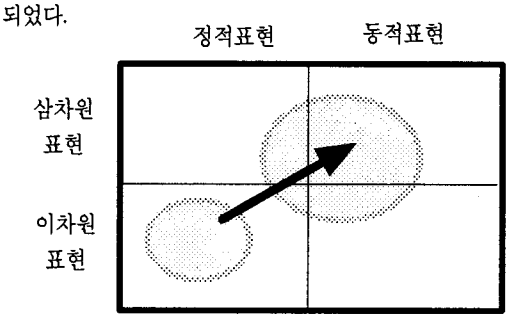

그림1. 정보 표현의 변화

동적인 표현 언어로서 정보를 표현하면, 보다 폭 넓게 데이터 구조(Structure)를 파악할 수 있게 되어지고, 그리고 또한 사용자가 인터페이스를 통하여 데이터를 조작하면서 생성되어지는 정보를 통해 새로운 사실을 파악하는 데이터 관찰(Observation)이 가능해진다. 삼차원 동적 정보 표현은 데이터베이스의 구조와 그래픽 모델의 구조를 이해하고 그리고 사용자들이 필요로 하는 정보가 무엇인가를 관찰 조사하여 그 표현하는 과정이 이루어진다. 그러나, 삼차원 동적 정보 표현을 위한 정보 표현의 방법에 대한 체계적인 연구는 아직 미비한 상태이다.

1-2. 연구 목적

본 연구를 수행함으로써 얻고자 했던 목적으로는 크게 두가지를 들 수가 있다.

첫째는 삼차원적 정보표현의 기본 원리와 현 컴퓨터 그래픽에서 가능한 기능을 조사하고, 그래픽 모델의 구조와, 데이터베이스 다이아그램 구조를 기초로 한 정보 표현구조를 정의하고 이러한 구조의 변화에 따른 정보 표현방법과 언어들을 밝혀낸다. 그리고 다양한 그래픽 표현 언어와 방법을 기존 사례들 속에서 찾아낸다.

둘째는 이러한 정보구조 변화에 따른 정보표현 방법과 언어 그리고 그래픽 표현방법에 따른 정보표현 방법과 언어들을 통하여 구현한 예를 제시한다.

1-3. 연구의 범위와 방법

2장, 3장, 4장은 문헌 연구와 실제 개발사례 조사를 통해 이루어진다. 2장은 이차원 그래픽과 비교하여 다른 삼차원적 그래픽 기초적 원리를 정의하고 현 삼차원 그래픽으로 구현되는 기능을 나열한다. 3장은 그래픽 모델 데이터의 구조와 데이터베이스 다이아그램 구조의 비교를 통해 표현 되어질 수 있는 정보표현 구조의 설계와 그 표현 방법에 대하여 설명한다. 4장은 현 다이나믹 시각 디자이너들이 개발한 개발 모델들의 분석과 현 컴퓨터 그래픽 언어들의 표현 가능성 분석을 통하여 그 가능방법과 예를 제시한다. 5장은 위의 제시한 방법을 기초로 야구통계자료 데이터를 어떻게 동적으로 표현 되어질 수 있는가를 보여주는 구체적인 예를 제시한다. 그리고 마지막 결론에서는 이 연구의 의의와 효용성에 대하여 논하고 앞으로의 향후 과제에 대하여 언급한다.

2. 정보 표현의 기초와 필요 기능들

2-1. 삼차원 그래픽 표현의 기초

삼차원 그래픽은 사실 평평한 컴퓨터 화면에 나타나는 2차원 이미지이다. 단지 깊이 정보를 제공함으로써 삼차원으로 보이게 하는 것이다. 깊이 정보를 사용자에게 전달하기 위하여 다음과 같은 요소들을 사용한다.

2-1-1 은선 제거(Hidden Line Removal)

선으로 만든 입면 체를 생각할 때 입면 체의 앞과 뒤를 구분할 수 있는 요소는 앞부분에 의하여 가려지는 뒷부분을 가리는 것이다. 이와 같이 앞쪽 선에 의하여 가려지게 될 선을 생략하는 것이 은선 제거이다.

2-1-2 색과 명암(Color and Shading)

실제로 은선이 제거된 입면체라도 컴퓨터에서 한가지 색으로 그린 이차원 그림과 유사하게 여겨진다. 한 사물이 같은색으로만 되어 있다하여도 삼차원에서의 그 사물은 명암이 다른 다른색으로 표현되어져야 한다.

2-1-3 빛과 그림자(Light and Shadows)

또 하나 간과하지 말아야 할 것은 빛의 효과이다. 삼차원 공간에서 빛은 두 가지 중요한 효과를 낸다. 첫번째로, 한쪽 방향에서만 조명이 비춰어진다고 가정할 때 같은 면은 하나의 색을 띄도록 한다. 두 번째로, 투명한 물체가 아닐 때 빛을 차단하게 되어 그림자를 만든다.

2-1-4 좌표 클리핑 (Coordinate Clipping)

사물을 컴퓨터 안에 그리기 전에 실세계 좌표(World Coordinate)를 컴퓨터 윈도우상의 좌표(Window Coordinate)로 변환하여야 한다. 이것은 실제로 화면에 나타내고자 하는 부분(World Coordinate 상의)을 지정해 줌으로써 이루어지는데, 이 영역을 클리핑 영역이라 한다.

2-1-5 뷰포트(viewport)

자신의 클리핑 영역의 폭과 높이가 픽셀로 되어있는 윈도우의 것과 일치하는 경우는 드물다. 그래서 좌표 시스템은 이론적인 데카르트 좌표계에서 실제적인 화면의 픽셀 좌표계로 매핑 과정이 필요하다. 이 매핑 방법은 뷰포트를 설정함으로써 정해진다. 뷰포트는 윈도우 클라이언트 영역내의 영역으로, 클리핑 영역의 내용이 여기에 그려지게 된다.

2-1-6 수직적인 투영(Orthographic Projection)

삼각법과 단순한 행렬 조작을 이용하여 삼차원에서의 정해진 한 위치를 컴퓨터 윈도우 이차원적 화면의 픽셀에 위치시킬 수 있다. 삼차원 좌표계는 이차원 면(윈도우 배경)으로 투영된다. 그것은 마치 어떤 물체의 앞에 유리판을 두고 유리판 위로 보이는 물체의 윤곽을 따라 그리는 것과 같다. 투영방법

중 수직적 투영방법은 모든 물체는 멀리 있건 가까이 있건 간에 원래 크기대로 나타난다.

2-1-7 원근 투영법 (Perspective Projection)

이 투영은 멀리 있는 객체를 가까이 있는 것보다 작게 보이도록 하는 효과를 낸다. 뷰잉 볼륨의 앞쪽에 위치한 가까운 물체는 거의 원래 크기대로 나타나고, 반면 볼륨의 뒷부분에 위치한 물체는 볼륨의 앞쪽으로 투영될 때 오그라든다. 이 투영방법을 사용함으로써 시뮬레이션이나 3차원 애니메이션에 사실감을 부여한다.

2-2. 삼차원 그래픽 표현의 필요 기능들

삼차원 이미지를 단순히 보여주는 과거의 기능에서 벗어나서, 삼차원 이미지 애니메이션과 네비게이션이 활용되어진 지도 이미 오래되었다. 또한 위의 기능을 제작할 수 있는 많은 전문 그래픽 언어들이 개발자를 위하여 개발되었다. 오픈지엘(OpenGL),다이렉트쓰리디(Direct3D), 코스모쓰리디(Cosmo3D), 옵티마이져(Optimizers) 그리고 오픈인벤터(Open Inventer)등이 국내에서 많이 쓰이고 있는 그래픽 언어들이다. 이 언어 각각은 안정성 문제 , 처리속도 문제 그리고 다양한 효과 문제 등에서 각각의 장단점을 보유하고 있다. 장단점들 중에서 아래의 기능은 그래픽 표현의 기본적 필요 기능들로 이를 통하여 다양한 그래픽 표현이 이루어진다.

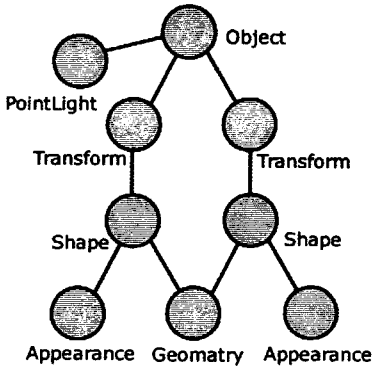

그림 2. 씬그래프 객체 계층구조

2-2-1 씬그래프 (Scene Graph)

삼차원 그래픽 모형이나 다이아그램을 효과적으로 활용하기 위해서는 객체 지향형의 구조로 모델이 설계되어야 한다.<그림2>와 같이 씬그래프는 객체 계층구조로 되어져 있다. 만약 객체지향형 형태로 모델이 설계되어지지 않으면 모델을 프로그램 언어로 다루는데 많은 어려움이 있다. 예로서 손 모델을 그리고 그곳에 손가락과 손톱을 그렸다고 가정하자. 손의 위치를 변화시킬 때 씬그래프 구조로 된 모델은 단순히 손의 위치 좌표만 바꾸어주면 된다. 그러나 만약 씬그래프 형식의 객체 구조로 설계되어 있지 않으면 손의 위치 변화를 위하여, 손, 손가락, 그리고 손톱의 모든 위치를 변화 시켜야 한다.

2-2-2 단순화 (Simplifiers)

비행기 모형을 만약 30,000개의 벡터 데이터로 그렸다고 가정하자. 그리고 10여개의 비행기가 동시에 뷰포인터에 등장한다면 그래픽 표현의 속도에 문제가 생긴다. 단순화 기능은 만약 비행기가 멀리 있을 때는 300개 정도의 벡터 데이터로 단순히 그려주고 가까이 있는 비행기는 30,000 벡터데이터로 그려주면 많은 양의 데이터 처리양을 줄일수 있게 하여준다.

2-2-3 쿨러 (Culler)

도시의 빌딩숲을 모델링 하여 네비게이션을 한다고 가정하자. 윈도우의 뷰포인터를 통하여 볼때, 높은 건물 뒤에 있는 작은 건물은 실제로 뷰포인터를 통하여 보이지 않는다. 이때 모델링 프로그램 엔진은 이러한 보이지 않는 사물에 대해서는 그리지 않아야 한다. 이러한 기능을 쿨러라 한다.

2-2-4 애니메이션 (Animation)

모델이 삼차원 공간에서 시간의 흐름에 따라 움직이는 것을 의미한다. 시간 템포, 시간 기간, 시간 방향, 시간 반복, 시간 속의 흐름길 지정 등 많은 요소들이 애니메이션에 존재한다. 애니메이션 기능과 함께 보다 다차원의 모델 표현이 이루어진다.

2-2-5 이벤드 처리자 (Event Handler)

사용자와의 인터액티브(Interactive)를 주기 위하여 필요한 기능들이다. 뷰포인터 어느 이상 가까이 접근하면 문이 열리다든지, 확대된다든지 아니면 영화 화면의 애니메이션이 시작된다든지 등의 이벤트를 가질 수 있다. 이 이벤트는 다르게 트리(Trigger)라고도 한다.

2-2-6 네비게이션 (Navigation)

보행자 관점(Walker View), 관찰자 관점(Examinator View), 비행자 관점(Fly View)등 다양한 관점을 가지고 사물을 관찰하는 것을 의미한다.

정보표현 방법은 크게 정보구조 변화를 통한 정보표현 방법과 다양한 그래픽 표현 방법을 통한 정보표현 방법으로 나누어진다.

3. 정보구조 변화를 통한 정보표현 방법과 언어들

정보구조 변화를 통한 정보표현 방법은 정보데이터 모델의 본질(Entities), 관계(Relations), 속성(Attributes)의 변화를 통하여 이루어진다.

정보데이터의 구조가 <그림 3>과 같은 계층구조(Hierarchical Structure)만으로 되어있지 않지만 본질과 속성의 차이를 보여주는 좋은 예이다. 본질은 속성보다 높은 레벨에 있다. 그리고 본질과 속성을 나누는 선은 언제나 유동적이며 개발자가 표현하고자 하는 의도(Intention)에 따라서 변화한다. 본질에서의 변형(Transformation)은 정보 데이터 모델의 거시적 변화라 할 수 있다. 정보데이터를 표현하기 위하여 새로운 본질을 생성할 수 있고, 제거할 수 있고, 결합할 수 있고, 분리할 수 있고 그리고 교환할 수 있다. 속성에서의 변형은 본질의 세부적 특성의 변화를 의미한다. 속성이 가지고 있는 형태, 값, 그리고 값의 범위 등을 새로 생성, 제거, 분해, 결합하는 과정을 통해 변화시킬 수 있다. 관계(Relation)는 본질과 본질 사이의 관계 변화를 통해 변형시킬 수 있다.

다시 말해서 속성은 데이터의 작은 단위 하나에 속한다. 이 위에 여러 층의 본질들이 존재한다. 본질들은 언제나 재정의 할 수 있다. 본질은 하부 속성들의 대표성을 의미하며, 상위본질은 하위 본질의 대표성을 의미한다. 본질의 변화는 그 하부 본질이 속성들을 어떻게 포함되는가에 달려있다. 여기서 사용

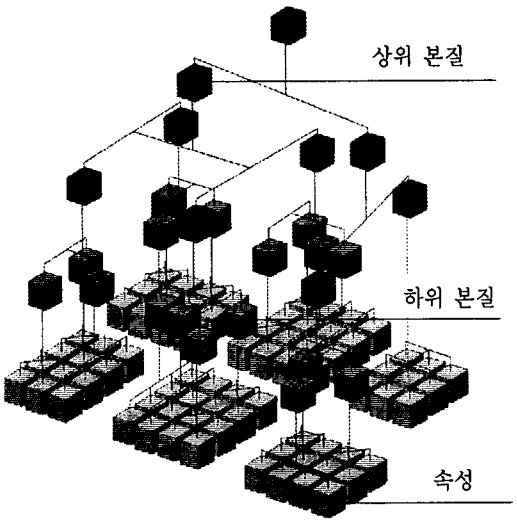

그림 3. 정보 계층 구조

자가 이 본질들을 나타내고 분류하여주는 관계를 바꾸면 다른 본질로 변하는 것이다. 또한 본질의 의미도 의미를 부여한 사용자에 따라서 다르게 나타내어진다. 그러나 속성은 달라질 수 없다. 다만 그 속성을 범위나 종류만 달라질 수 있을 뿐이다.

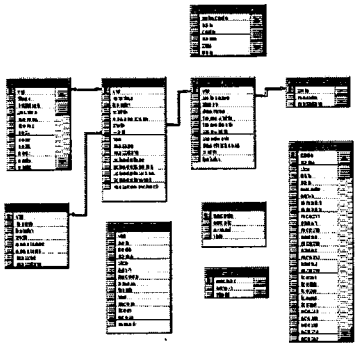

그림 4. 데이터베이스 다이아그램 구조

<그림 2.>, <그림 3.>와 <그림 4.>를 살펴보면 그 각각에서 본질(Entity)과 속성(Attribute) 그리고 관계(Relation)를 찾아 비교할 수 있다.

<그림 2.>는 그래픽 모델을 구성하는 객체-계층 구조이다. 여기서 Appearance와 Geometry는 모델을 나타내는 가장 하부 구조이다. 즉 속성에 해당된다. 그 위에 Shape은 속성들의 대표성을 의미하는 하위 본질이다. 그리고 그 위에 transform과 object는 상위 본질을 의미한다. 물론 object위에 또 상위의 Object가 존재할 수 있다. 그리고 Appearance, Geometry, Shape, transform, 그리고 object 을 연결하는 관계(relation)가 존재한다.

<그림 4.>은 데이터베이스 테이블 다이아그램을 나타내는 그림이다. 각각의 테이블(Table) 밑의 필드(Field)는 속성을 나타낸다. 그리고 테이블 명과 테이블들을 구별하는 상위 테이블 명은 본질에 속한다. 데이터 베이스 구조는 데이터의 효과적인 접근을 위하여 설계된다. 그러나 이 또한 계층구조를 가져야 한다. 그리고 각 테이블 사이를 연결하기 위하여 링크가 필요한데 이것이 관계(relation)이다. 그리고 데이터베이스의 구조에서는 키(Key)라는 개념이 있어서 관계를 설정할 수 있다.

위와 같이 그래픽 모델구조와 데이터베이스 다이아그램 구조에서 본질과 속성 그리고 관계를 찾을 수 있다. 정보표현의 구조를 설계할 때 위의 두 구조를 잘 반영할 수 있는 구조로 설계하고 그리고 각 이벤트(Event)나 조작 명령(command Control)으로 본질이나 속성의 요소를 효율적 변화하여 새로운 정보를 생성할 수 있도록 하여야 한다. 사용자는 이러한 본질, 속성 그리고 관계의 변형으로 새로운 데이터 정보를 찾을 수 있다.

4. 그래픽 표현방법을 통한 정보표현 방법과 언어들

그래픽 표현방법을 통한 정보표현 언어는 정보 데이터의 구조를 변형시키지 않고 사용자들에게 정보 데이터 모델을 바라보는 시점의 변화 그리고 세밀함의 변화를 통해 정보 데이터를 표현함을 말한다.

이 언어들은 컴퓨터 그래픽 기술의 발전으로 대용량의 데이터를 시뮬레이션 할 수 있음으로써 가능하게 되었고, 전환 언어들은 정보를 표현하기 위하여 독립적으로 사용할 때 보다 여러 개의 언어를 동시 조합을 하여 사용할 때 보다 효과적으로 표현할 수 있고, 개발자가 의도한 목적을 이룰 수 있다.

그래픽 표현방법은 삼차원 그래픽 모델뿐만 아니라 시간의 개념과 시간과 이벤트를 실행하는 개념을 포함한다. 다음의 표현 언어들은 다이나믹 시각 디자인 (Dynamic Communication Design) 관련 전문가들의 표현 사례 조사와 현 그래픽 언어들이 가지는 기능 조사를 통하여 개발되었다.

<그림 5.>는 뷰포인터(Viewpoint)의 회전(Rotation)을 통한 정보 표현이다. 회전 커맨드(Command)을 이용하여 계속적인 조작을 하지 않고 전체적인 모델 구조가 회전함으로써 사용자는 관찰을 할 수 있다. 이때 모델이 회전하는 것이 아니라 단지 관찰자의 창인 뷰포인터만 회전함으로 모델 데이터 구조에는 아무런 영향이 없다.

<그림 6.>은 뷰포인터의 팬(Pan)을 통한 정보 표현이다. 이 또한 회전과 동일하게 일정한 영역을 설정하여 사용자가 그 영역 안에서 반복적인 팬기능을 하는 모델을 관찰할수 있다. 이 또한 모델 데이터 구조는 변화가 없다.

<그림 7.>은 뷰포인터의 줌인(Zoom in)과 줌아웃(Zoom Out)이다. Z축을 통한 관찰자의 시점을 변화시키는 작업인데, 이 또한 모델 데이터의 구조에는 아무런 영향이 없다. 줌 기능과 회전 그리고 팬 기능을 함께 사용하면 사용자는 원하는 모델의 부위에 관찰 시점을 근접 접근하여 그 영역에서만 회전 혹은 팬 기능을 반복 수행할 수 있다.

<그림 8.>은 뷰포인터의 네비게이션(Navigation)이다. 정해진 길이 없이 사용자가 원하는 뷰를 자유로 변화하여 관찰한다. 이때 보행자 관점(Walker View), 관찰자 관점(Examinator View), 비행자 관점(Fly View)등 다양한 관점이 존재하는데,

주로 건축을 가상 공간 속에 놓고 관찰할 때 유용하다.

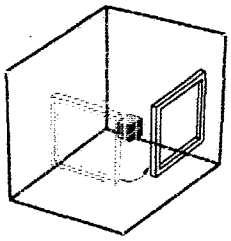

그림 5. Viewpoint Rotation

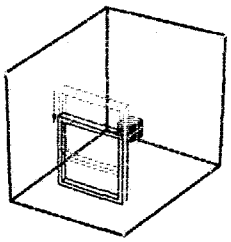

그림 6. Viewpoint Pan

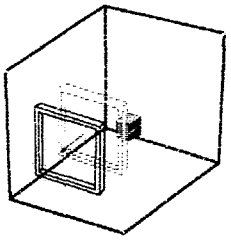

그림 7. Viewpoint Zoom

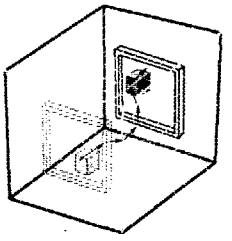

그림 8. Viewpoint Navigation

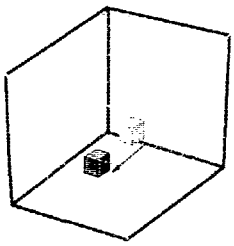

그림 9. Transposition

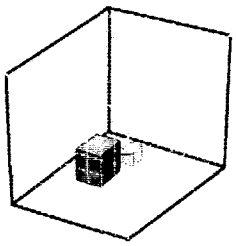

그림 10. Scale

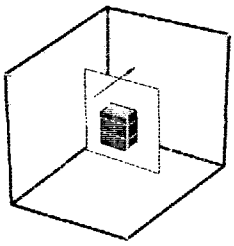

그림 11. Scan Section

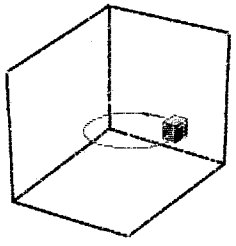

그림 12. Rotation

<그림 9.>는 이동(Transposition)이다. 삼차원의 좌표계에서 사용자가 일정 모델의 위치를 변화 시키는 것이다. 모델의 위치가 삼차원 좌표 속에서 변한다는 것은 그 모델 데이터가 의미하는 내용이 달라진다는 것을 말한다. 즉, 변화된 거리의 벡터 양은 모델 데이터의 수리적 크기에 영향을 주며, 사용자는 그 의미를 관찰하게 된다.

<그림 10.>는 스케일(Scale)이다. 삼차원의 좌표계에서 사용자가 일정모델의 크기를 변화시키는 것이다. 만약 그래픽 모델의 일정 부분의 크기가 변한다는 것은 그 모델이 의미하는 데이터의 수리적 치수가 커졌음을 의미한다. 예로서 증권시장의 데이터베이스를 표현하는 그래픽 모델이 있을 경우 일정 막대 모델 크기 변화는 증권가격이나 거래량이 많아졌음을 의미할 것이다.

<그림 11.>는 스캔 단면화(Scan Section)이다. 병원에서 뇌의 CT촬영을 통해 표현하는 그래픽과 유사하다. 일정 모델의 단면을 점진적인 이동과 함께 그래픽 결과를 보여주는 것이다.

<그림 12.>는 회전(Rotation)이다. 뷰포인터의 회전과 다른 점은 모델 구조의 일부분이 삼차원 좌표계의 일정한 점을 기준으로 회전함으로써 이루어진다. 그래픽 모델의 수리적 데이터는 Sin함수와 Cos함수의 범위값을 가지고 변화할 것이다.

<그림 13.>는 애니메이션 기간이다. 단순히 그래픽 모델을 일정 시간 동안 정해진 경로에 따라 움직이게 하는 것이다. 기본적인 애니메이션 방법으로 설계된 모든 애니메이션을 일정 동일한 시간으로 사용자의 조정 없이 실행됨을 의미한다.

<그림 14.>는 애니메이션 방향, 속도, 정지, 그리고 재시작이다. 사용자가 그래픽 모델의 애니메이션을 조정하는 것으로 DVD 플레이어를 조정하는 것과 같은 행위이다. 그래픽 모델이 변화되고 이동되는 방향과 속도를 사용자 스스로 조정 관찰 하는 것이다.

<그림 15.>는 제한적 반복 애니메이션이다. 그래픽 모델은 의도적으로 애니메이션의 일부분을 재반복하여 사용자에게 의미를 전달하고자 한다.

<그림 16.>는 초고속 애니메이션이다. 애니메이션이 초고속으로 움직이면 실제 보이는 그래픽 모델에서 다른 형태의 모델 패턴이 존재하게 된다. 여러 색으로 그려진 색깔 판을 초고속으로 회전하면 그 색깔들의 가감 색이 나타나는 원리와 같다. 혹은 그래픽 모델이 움직인 극단적인 패턴을 발견할 수 있다.

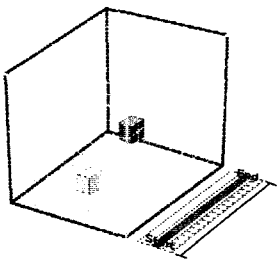

그림 13. Animation Duration

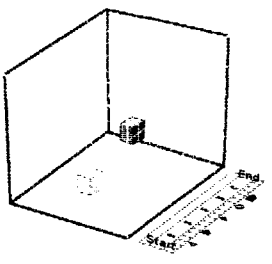

그림 14. Animation Speed, For/Reward, Stop

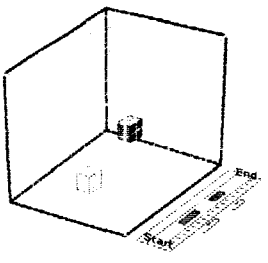

그림 15. Selected Repeat Animation

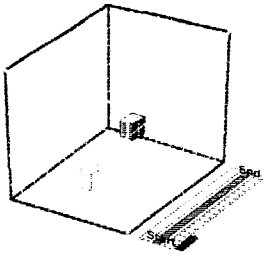

그림 16. Super Speed Animation

<그림 17.>는 애니메이션의 경로 설정이다. 만약 같은 그래픽 모델 애니메이션이라도 그 경로을 다르게 하여 관찰하면 기존의 애니메이션에서 발견되어지지 않았던 무언가의 데이터 패턴을 사용자는 발견할 수 있다.

<그림 18.>는 다중 뷰포인터(Multi-Viewpoint)이다. 동시의 여러 뷰포인터 관찰 창을 만들어서 사용자는 관찰하는 것이다. 여러 창을 동시에 관찰함으로써 보이지 않았던 이면의 새로운 데이터의 의미를 사용자는 발견할 수 있다.

<그림 19.>는 뷰포인터 오버레이(Viewpoint Overlay)이다. 하나의 뷰포인터위에 다른 하나의 뷰포인터를 겹쳐서 관찰하는 것이다. 사용자는 이 겹쳐진 뷰포인터를 통하여 단독 뷰포인터에서 발견하지 못했던 다른 의미를 찾을수 있고, 개발자는 의도적으로 뷰포인터를 겹쳐서 그래픽 모델 데이터의 의미를 전달할 수 있다.

<그림 20.>는 투명판 오버레이(Transparent Overlay)이다. 이는 여러 다이나믹 시각 디자이너들이 흔히 사용하는 방법 중 하나이다. 여러 그래픽 모델을 하나의 축을 기준으로 일렬로 정렬한 후 일정부분의 모델의 투명도를 바꾸면 뒤에 있는 모델의 형태는 보이게 된다. 이때 앞의 모델의 형태 또한 어렴풋이 사용자에게 전달할 수 있다.

<그림 21.>는 조명의 색깔이다. 여러 그래픽 모델 데이터는 이미 고유의 색깔을 가지고 존재할 것이다. 모델의 고유 색깔 이외에 조명이 주는 색깔로 분산되어져 있는 모델 데이터를 군집화 시킬 수 있을 것이다.

<그림 22.>는 조명의 위치변화와 크기이다. 연극 무대에서 관람객의 시선을 모으기 위하여 조명을 사용한다. 그래픽 모델들에게 일정한 조명을 비출 때, 그 조명의 크기와 위치에 따라 사용자에게 보여지는 모델은 다를 것이다.<그림 23.>는 장소 이벤트(Location Event)이다. 모델이 사용자의 조정으로 움직이다가 일정 좌표 영역으로 들어오면 어떤 이벤트를 사용자에 전달하는 것을 의미한다. 뷰포인터의 좌표나 그래픽 모델의 좌표 모두 사용될 수 있다. 평소 그래픽 모델에게는 보이지 않았던 정보가 가까이 근접하였을 경우 정보가 나타남으로써 사용자에게 필요한 정보만을 효과적으로 전달하기 위하여 사용된다.

<그림 24.>는 스케일 이벤트(Scale Event)이다. 모델이 사용자의 조정으로 정해진 크기 이상 혹은 이하로 변화가 되면 어떤 이벤트를 사용자에게 전달하는 것을 의미한다.

<그림 25.>는 절대 좌표축의 변화이다. 그래픽 모델 전체가 일정 이동하는 것과 좌표축 자체가 이동하는 경우에 차이점이 있다. 그것은 주변의 조명의 위치나 애니메이션의 경로 등은 그래픽 모델이 이동할 때 변화가 없지만 절대 좌표축의 이동시에는 그 변화가 있다.

<그림 26.>는 절대 위치설정(Super Position)이다. 아무리 뷰포인터의 위치가 변화더라도 뷰포인터를 통하여 항상 보이는 그래픽 모델을 설정할 수 있다. 일정 그래픽 모델의 위치를 뷰포인터의 좌표 앞에 설정해 놓은 경우이다. 뷰포인터의 이동을 따라서 항상 그래픽 모델은 움직이게 된다.

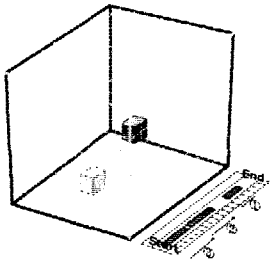

그림 17. Animation Path Design

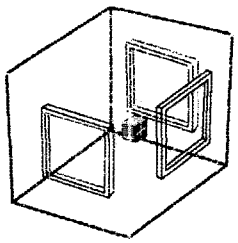

그림 18. Multi-Viewpoint

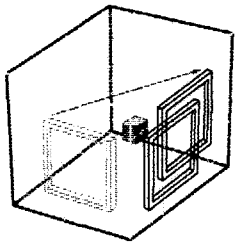

그림 19. Viewpoint Overlay

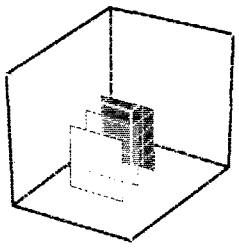

그림 20. Transparent Overlay

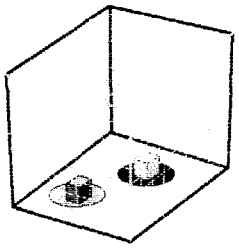

그림 21. Light Color

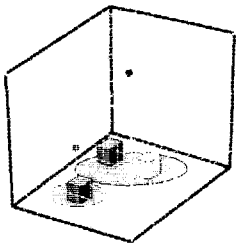

그림 22. Light Size and Location

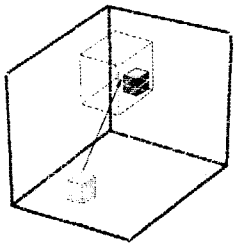

그림 23. Location Event

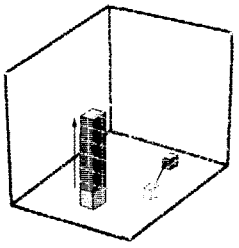

그림 24. Scale Event

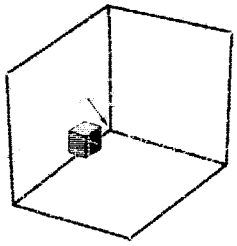

그림 25. Coordinate Axes

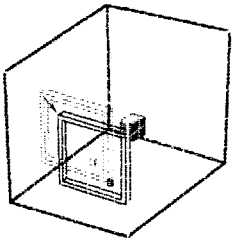

그림 26. Super Position

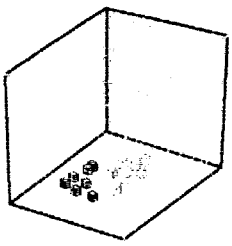

그림 27. Visible True/False

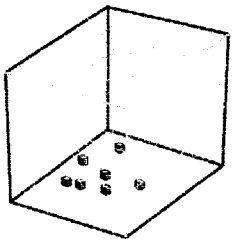

그림 28. Color Cluster

<그림 27.>는 사라짐/나타남(Visible True/False)이다. 여러 개 흩어진 그래픽 모델 중 필요한 모델을 사라지고 나타나게 함으로써 사용자가 官찰하는 범위를 조정할 수 있다. 복잡한 그래픽 모델일 경우 필요한 몇몇의 모델만 보이게 하고 나머지는 사라지게 한 후, 나중에 다시 등장하게 조정 비교 할 수 있다.

<그림 28.>는 색깔의 군집화이다. 색깔 또한 정보를 표현하는 방법 중에 하나이다. 색의 명도나 채도 그리고 그라데이션(Gradation) 등은 데이터 구조의 수리적 데이터를 그래픽 모델로 표현하기 좋은 요소이다.

5. 구현사례

- VRML을 이용한 야구 통계 자료 검색과 결과

Cosmo3D를 사용하여 그래픽 모델, 이벤트 조정, 그리고 애니메이션을 디자인하였으며, 일부분 자바(J++)언어를 구현을 하였다. 데이터베이스는 MS SQL을 사용하였으며 데이터 베이스와 모델의 연결은 OLEDB ADO방식과 자바(J++)로 구현하였다.

5-1. 구현 사례의 개발 요소

야구는 통계자료를 즐기는 게임이라 할 수 있다. 야구 팬들을 위하여 야구 데이터를 효과적으로 접근하여 관찰 그리고 필요한 정보를 전달하기 위해서는 그래픽 모델 개발 이전에 그 사용자, 배경, 내용, 그리고 의도 등을 설정하여야 한다.

5-1-1 사용자 (Audience)

인터넷을 통하여 스포츠 통계 자료를 검색하는 사이트의 사용자는 스포츠의 이야기 거리, 흥미 거리 그리고 때로는 도박을 위하여 방문한다. 그래서 단순한 통계자료만이 아닌 팀의 전체 전력을 파악할 수 있는 거시적인 시야를 제공하면서 개인의 자세한 정보 자료도 쉽게 연결하여 읽을 수 있는 그런 정보 표현 구조가 필요하다.

5-1-2 배경(Context)

야구 정보 검색에 관한 10여 개의 웹 사이트를 살펴보면, 원하는 통계자료를 얻기 위해서는 인터넷상에서 6번 혹은 많으면 10번까지의 하이퍼링크(Hyperlink)를 따라서 내려가야 한다. 그리고 그들은 2여장의 선수 사진과 선수의 기록을 나열한 수리적 데이터들을 얻을 수 있다. 짧은 하이퍼링크로 필요 정보를 찾을 수 있어야 한다.

5-1-3 내용(Content)

이러한 스포츠 통계자료 관련 웹사이트의 개선을 위하여 두 가지의 작업이 진행되어야 한다. 하나는 어느 정도의 자연어(Natural Language)를 가지고 검색을 할 수 있고 그리고 심볼 입력(Symbol-Selected Format)도 가능한 검색 엔진을 개발하는 것이고, 또 하나는 검색 엔진을 통하여 나온 결과를 다차원의 동적 다이아그램으로 결과를 보여 주는 것이다. 검색 엔진의 구조와 데이터베이스 그리고 결과를 보여줄 다이아그램은 동시에 개발되어야 한다..

5-1-4 의도(Intention)

사용자는 다음과 같은 질문을 할 수 있다. 1999년 시즌에서 4번 타자 중 누가 최고의 타자인가? 누가 최고라는 단어는 명확히 결론을 줄 수 없는 단어이다. 그 타자가 단지 타력이 높아서, 타점을 많이 얻어서, 주장으로 팀원을 잘 리더를 해서, 그리고 야외 수비를 잘해서 등 하나의 요소만으로 그들 중 누가 최고라는 결론을 내릴 수 없다. 결론을 내리는 자는 검색 엔진이 아니라 사용자이다. 사용자가 정보를 거시적으로 파악할 수 있고, 원하는 선수와 팀간의 데이터를 비교 평가 할 수 있으며, 그리고 시합의 연도나 시합 장소에 따른 선수의 기록 변화 등을 관찰할 수 있다면, 사용자 개개인은 자신의 주관에 따라 누가 최고라는 판단을 할 수 있다. 구현 사례의 정보 표현은 다른 사용자와 함께 대화의 주제를 만들 수 있는 복합적인 정보의 전달과 정보의 관찰이 이루어지도록 만들어져야 한다.

5-2. 구현 사례의 정보 표현

야구 통계 자료 속에서 속성(Attribute)은 개인의 신상 기록에 해당하는 타격(Batting Average, Hits, Doubles 등) 혹은 투구(Win, Run allowed, blown Saves등)에 관련된 자료, 시합(Fielding Percentage, Range Factor, Double Plays 등)에 대한 자료 그리고 시간(Season Total, In Day Games, Last 3 Games 등)에 대한 자료들이 이에 해당한다. 그리고 이런 자료들 상위 개념인 본질(Entity)은 소속된 팀이나 수비위치, 타격순서 그리고 시합과 선수의 역사 등이 이에 해당한다. 다시 말해서 속성은 개인 선수의 자세한 정보이고 본질은 선수를 분류, 소속시킬 수 있는 어떠한 그룹에 해당한다.

이런 본질과 본질을 연결하는 관계(Relation)는 사용자가 질의를 할 수 있는 예상 질문(Injury, Rookie, Playoff, Active Career Batting, Trade, 그리고White/Black/Foreigner 등에 관련된)들을 고려하여 설계하여야 한다.

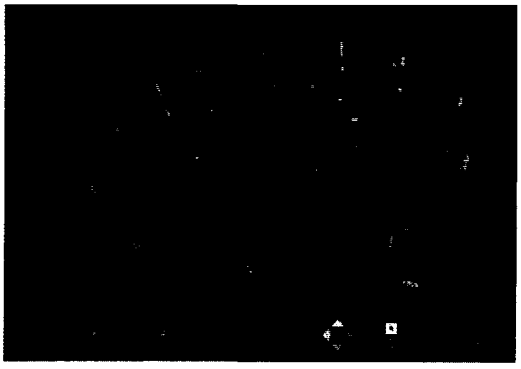

그림 29.

<그림 29.>는 각 선수들의 속성(Attribute)들을 하나의 조각상(Statue)으로 만든 후 그래픽 표현 언어중에서 뷰포인터 회전, 팬, 줌 그리고 네비게이션, 스케일, 회전, 장소이벤트, 절대 위치 설정등을 사용하여 표현하였다. VRML에서 제공되는 컨트롤 기능 외에 뷰포인터에 관계 없이 항상 같은 장소에 존재하는 컨트롤 기능 3가지를 더하였다. 하나는 마우스가 컨트롤러 위에 머무르면 전체의 선수상들이 회전을 가능하게 하는 기능인데 이는 사용자가 선수의 조각상으로 줌인(Zoom In)을 한 후 선수들의 조각상의 순차적인 회전으로 대략적인 한 팀의 선수들의
데이터를 보기 위함이다. 또 하나의 기능은 시합 시즌을 조정하기 위한 것인데 이 컨트롤 바를 움직이면 선수들의 조각상의 형태가 시합 시즌의 선수 시합 결과 데이터에 따라 변형한다.

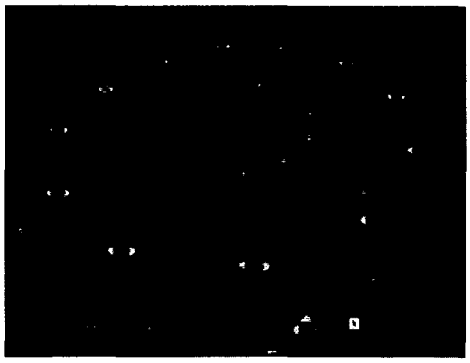

그림 30.

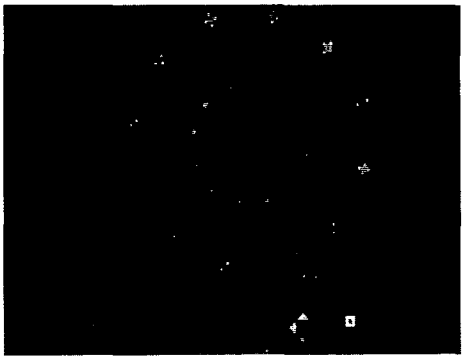

그림 31

나머지 기능은 조각상 각각들을 동시에 회전을 시켜서 선수들의 데이터를 비교하기 위한 기능인데 <그림 30.>와 <그림 31.>을 보면 각 선수들의 조각상이 일정 방향으로 향하고 있다. 각 조각상에 보여지는 막대들은 선수들의 일루타, 이루타, 삼루타, 홈런 등을 표현한다. 사용자는 검색 결과에 따라 나타난 조각상들 즉, 1999년 올스타 A팀 타자들의 검색 결과를 위와 같은 형태로 대략적인 타자들의 타격을 관찰할 수 있다.
만약 사용자가 한명의 선수 데이터를 의미하는 선수의 조각상으로 일정이상 Zoom In을 하면 <그림 32.>에서 보는 바와 같이 선수들의 자세한 수리적 데이터를 보여주는 윈도우가 자동적으로 등장한다. 이는 선수들의 정보 마지막 단계까지 보여주기 위함이다. 이때 위에서 조각상의 회전 컨트롤러 위에 커서를 머무르면 선수들이 순차적으로 회전하면서 선수들의 자세한 정보를 볼 수 있다.

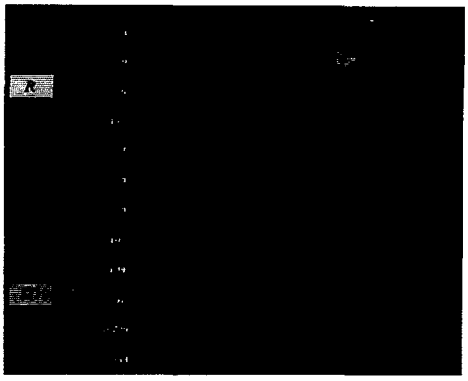

그림 32.

각 조각상에 표현된 색깔은 옆에 보여주는 수리적 데이터의 색깔과 동일하다. 선수들의 조각상의 길이, 넓이, 구의 지름, 그리고 사물과 사물과의 거리 등은 선수 데이터를 나타내기 위하여 사용되었다.

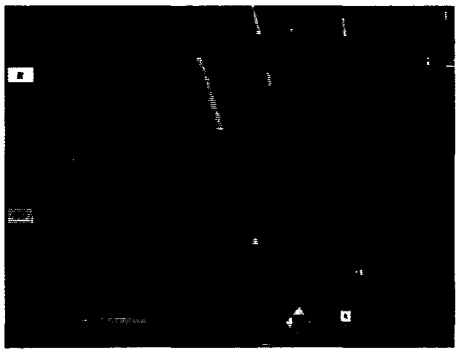

그림 33

<그림 33.>은 사용자가 사각형안의 게임에 대한 연도 데이터를 변화시키면 야구 선수들의 조각상들이 일제히 변화를 한다. 앞에서 설명한 봐와 같이 이 조각상들은 회전을 하는데, 해당 연도 변화와 함께 각각의 선수들의 데이터를 비교한다.
왼쪽 구는 포볼을 의미하며 옆의 막대그래프의 "BB"에 해당한다. 오른쪽 구는 스트라잌 아웃된 데이터를 의미하며 막대그래프의 "SO"에 해당한다. 십자모양의 긴바들은 각각 일루타수, 이루타수, 삼루타수, 그리고 홈런 수를 의미하며 옆의 막대그래프의 "H", "2B", "3B", 그리고 "H"에 해당한다. 이 십자모양의 색의 명도는 통상 타율을 의미하며 막대그래프의 "AVG"에 해당한다. 그리고 원기둥의 길이와 작은 십자모양의 타원형 기둥 길이와 색들은 일년동안의 게임 출전수, 타석수 그리고 1루 베이스의 진루율을 의미하며 막대그래프의 "G", "AB", 그리고 "SLG"에 해당한다. 이와 같이 각 조각상을 봄으로서 대략적인 선수의 기록을 파악할 수 있고 그리고 막대그래프를 봄으로써 자세한 기록을 살필 수 있다.

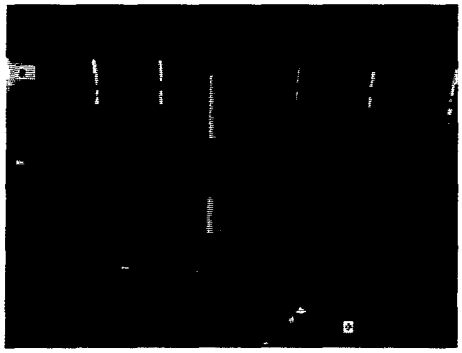

그림 34.

<그림 34.>은 조각상의 다른 선수의 기록 조각상을 보여준다. 위의 선수와 비교하면 대체적으로 1루타, 2루타 수가 적고 그리고 삼진아웃수와 포볼의 수가 적음을 알 수 있다. 그리고 그 근원적인 이유가 게임 출두률이 적어서 임을 알 수 있다. 그러나 선수의 타율이 높음을 감안 한다면 대타로서 일년동안 그 역할을 해옴을 짐작할 수 있다.

이와 같이 선수들의 기록상을 비교하면 그 팀의 선수들의 대략적 전력을 거시적 관점에서 파악할 수 있다. 이러한 거시적 파악은 사용자에게는 야구의 기록에 대한 자신만의 판단을 만드는데 도움을 준다.

6. 결론

이상에서 데이터베이스의 자료를 토대로 삼차원적이면서 동적인 그래픽 표현을 위한 방법과 그 언어들을 알아보았다. 데이터베이스의 구조와 그래픽 모델의 구조가 계층구조를 가지고 있음을 감안하여, 그 구조를 토대로 표현 모델 구조를 설계하고, 그 설계된 구조의 요소들인 본질, 속성 그리고 관계의 변형을 통하여 새로운 정보를 만들어 나가는 방법이 있었다. 그리고 보다 효율적으로 정보를 표현할 수 있도록 현존하는 그래픽 표현 방법을 정리하여 제시하였다. 실제로 삼차원의 동적인 표현을 위해서는 위에서 제시한 그래픽 표현방법들이 복합적으로 사용됨을 알 수 있다. 표현 방법을 설계하는 디자이너가 위에서 제시한 표현 언어들을 하나 하나 고려하면서 그 가능성을 생각하고 적용한다면, 방대한 자료의 데이터 베이스 정보라 할지라도 사용자들이 쉽게 이해할 수 있으면서 조작할 수 있는 정보표현을 설계할 수 있을 것이다.

그러나 한편, 구조 설계과정에서 어려움이 있을 수 있다. 그래픽 표현 모델 구조를 설계함에 있어서 디자이너는 많은 요소들을 동시에 고려하여야 한다는 점이다. 복잡한 구조를 단순히 설계자의 머리에서 나오기는 힘들다. 물론 표현 모델 구조에서 본질들을 사용자들이 관찰하고 해답을 얻고 싶어하는 정보로 만들면 가능하지만, 사용자의 의도보다는 데이터 구조의 분석을 통하여 사용자가 고려하지 않았던 요소들도 보여 주어야한다.

한편으로 5장에서 설계되어진 선수들의 조각상들이 전형적인 기하학들로 표현되지 않고 사용자가 미루어 짐작할 수 있는 형태로 설계되어진다면 보다 사용자가 데이터 자료를 쉽게 이해할 수 있을 것이다. 선수의 타율은 야구 방망이 형태가 될 수 있을 것이고, 선수의 총 게임 참석 수는 야구장의 필드형태가 될 수 있을 것이다. 하나의 데이터를 표현하는 그래픽 모델 부분의 의미를 사용자가 쉽게 이해할 수 있게 하며, 하나 하나가 모여 표현되어지는 한 개인의 선수 그래픽 모델 조각상이 또한 의미를 가질 수 있게 디자인되어져야 할 것이다. 따라서, 각 모델을 설계할 때 표현되어질 수 있는 메타포를 적용하여 연구가 필요하다고 생각한다.

그리고, 현 데이터 베이스의 데이터는 문자나 숫자뿐만 아니라 텍스트, 그래픽, 이미지, 오디오, 비디오와 같은 모노 미디어 데이터가 내용에 의한 논리적 관계와 시공적 관계, 사용자 상호 작용등에 따라 구성되어져 있다. 본 논문에서는 문자나 숫자에 국한되어 진행하였다. 그래픽, 이미지, 오디오, 비디오의 데이터가 효율적으로 삼차원 공간에서 표현되어질 수 그래픽 표현 방법과 언어들 그리고 표현 모델 구조 설계방법 및 언어 또한 연구가 필요하다.

참고문헌

- Charles L. Owen, **Diagrams as Tools for Worldmaking**, Visual Language, Volume 26, Numbers 3 and 4, 1992
- Edward R. Tufte, **Envisioning Information**, Graphic Press, 1990
- Edward R. Tufte, **The Visual Display of Quantitative Information**, Graphic Press, 1983
- Edward R. Tufte, **Visual Explanation**, Graphic Press, 1997
- Gerald L. Lohse, Kevin Biolsi, Neff Ealken, and Henry H. Rueler, **A Classification if Visual Representations**, Communication of The ACM, December 1994/Vol.37, No.12
- Hackos Redish, **User and Task Analysis for Interface Design**, 1998
- Jay Dobline, **A Structure for nontextual communications**, Processing of Visible Language, Vol. 2, 1979
- Vijay K. Sivasankaran, **Dynamic Diagramming**, Design Processes newsletter, Institute of Design, Volume3, Number1.
- Dennis Wixon, **Field Methods Casebook For Software Design**, 1996