

글. 성우제_ Sung, Woojae

OMA NYC office / Senior Architect

파라메트릭 디자인 II

Parametric Design II

지난 편에서는 parametric design의 발생과 흐름에 대해 간략하게 이야기 해 보았습니다. 이번 편에서는 parametric design을 하기 위해 고안된 여러 가지 parametric tool들이 바탕을 두고 있는 기본적인 개념 및 그 구성요소 등에 대해 간략하게 이야기를 해 보고자 합니다.

■ 이데아와 사물, Algorithm 과 Parameter

플라톤의 이데아는 정신에 의해 파악되는 사물의 본질이며 사물들은 이러한 이데아를 나누어 가지며 물리화 됩니다. 두 점을 연결하는 선을 모니터 상에 시각화 하는 데서 출발한 Autocad와 같은 설계 프로그램들이 공통적으로 가지고 있는 속성은 특정한 geometry 혹은 건축적인 요소에 대해 일반적으로 받아들여지는 암묵적 합의를 통해 그 객체를 정의 내린 후 이를 모니터라는 가상의 공간에 사물화 하는 여러 가지 방법들, 즉 algorithm을 제공한다는 점입니다. 여덟 개의 점으로 이루어지고 열두 개의 변을 가지고 있으며, 여섯 개의 면을 가지고 있다는 육면체의 정의 자체가 가장 이상적인 육면체에 대한 이데아가 되는 것이며, 이런 속성을 어떤 방식으로건 나누어 가지는 가상공간속의 육면체들 즉 사물을 구현하기 위한 여러 가지 방법을 별도의 algorithm으로 정의를 해 놓은 후 이중 사용자의 필요에 따라 택일 후 객체를 생성해 냅니다. 육면체를 만드는 수많은 algorithm 중 하나인 밀변과 높이를 정의하는 방법을 사용자가 선택하였다면, 컴퓨터 프로그램은 육면체의 아랫면을 정의하는 사각형을 먼저 그리기를 요구하고 그 후 육면체의 높이를 요구하게 됩니다. 이를 통해 생성되는 육면체는 구체적인 치수를 가지고 있으며 사용자의 정의에 따라 어떤 색상인지 어떤 재료인지까지 정의를 해주게 됩니다. Algorithm을 통해 객체를 생성하는 과정에 반드시 필요한 것이 바로 사용자의 입력 값인데요, 위의 육면체 예에서 볼 수 있듯이 밀변의 형태를 어떻게 생성할지 또 높이 값을 어떻게 주는지에 따라 각기 다른 육면체가 생성되게 됩니다. 이렇듯 변수, 즉 parameter는 geometry의 정의에서 출발하여 객체를 생성하는데 있어서 사용자의 의지가 반영될 수 있는 유일한 방법임에 그 중요성이 있다고 할 수 있습니다.

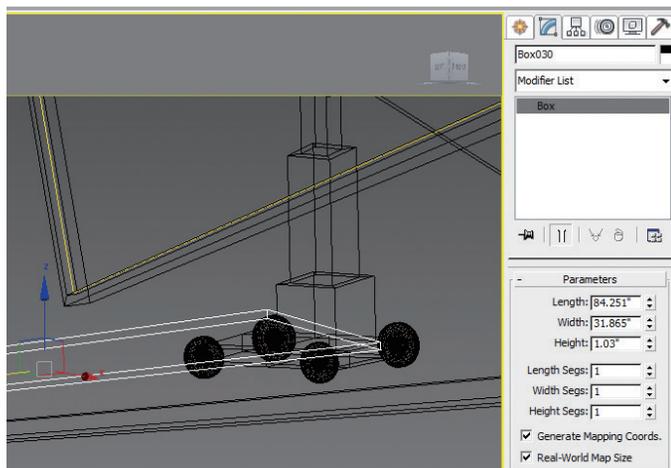
■ History

전통적인 삼차원 설계 프로그램에서는 이런 algorithm을 통해 객체를 생성하게 되는데 이렇게 생성된 객체는 하나의 결과물로서, 추가적인 변형을 요구하는 경우에는 전단계로 돌아가서 다시 객체를 생성하거나 아니면 추가적인 modifier를 통한 변형의 과정을 거치게 됩니다. 이제 이러한 객체 생성의 과정에 history¹⁾라는 개념이 추가됨으로써 객체를 생성시 이미 추후의 변형이 가능하도록 길을 열어주게 됩니다. 즉 이전의 예에서처럼 객체 생성시 육면체의 높이를 정의하여 주었다고 하더라도 생성 후 어떤 이유에서건 높이를 바꾸고 싶은 경우가 생긴다면 undo를 하고 새로운 객체를 생성하거나 이미 존재하는 육면체에 추가적인 변형을 가하는 것이 아니라 history editor에서 높이 값만 바꾸어 주면 되는 것입니다. history의 개념은 삼차원 설계프로그램에 시간이라는 하나의 차원을 추가 했다는 점에서 굉장히 기념비적인 발전입니다.

■ Block 과 Dynamic Block

이와 별도로 block이라는 개념이 발전해 왔는데요, 이는 필연적으로 같은 요소의 반복이라는 특성을 가지고 있는 건축의 속성상

1) 3d max - history



반복되는 건축적인 요소에 변화가 생겼을 때 발생하는 전체적인 변화에 좀 더 효율적이고 유연하게 대처를 하도록 도와주는 중요한 개념입니다. Block이 위에서 말한 parameter와 결합하면서 dynamic block²⁾이란 개념을 만들게 되는데 이는 동일한 형태를 가지되 각각 조금씩 다른 속성을 가지는 객체를 생성할 수 있도록 해줍니다. 예를 들어 대다수의 창호와 다르게 건물 특정부위의 몇몇 창호의 폭이 조금씩 달라야 한다면 block을 정의시 창호의 폭이라는 parameter를 추가함으로써 그러한 변화에 유연하게 대처할 수 있도록 해줍니다.

■ Parameter와 동적인 형태생성법³⁾

Algorithm이 history를 통해 특정한 변수를 추가하며 유연한 상태로 변화하는 과정과 block이 변수를 추가하여 dynamic block으로 변화하는 과정은 어찌 보면 동일한 개념이라 생각할 수 있습니다. 물론 전자는 후자보다 좀 더 추상적인 층위에서 작동하는 과정이고 후자는 좀 더 물리적인 층위에서 작동하는 과정이지만 공통적으로 변수라는 개념이 도입되면서 종래의 정적인 형태생성법이 동적으로 변화한다는 데에서 같은 맥락에서 이해 될 수 있습니다. 동적인 형태생성법이 가져오는 가장 근본적인 변화는 생성되

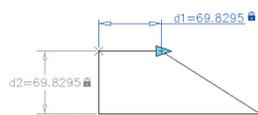
어지는 형태가 더 이상 자기 참조 적이지 않고 외부와의 소통을 획득하게 된다는 데에 있다고 할 수 있습니다.

■ Parametric Tool

위에서 살펴본 것처럼, 전통적인 삼차원 설계 프로그램들은 parameter의 도입으로 유연한 형태생성의 논리를 가지는 방향으로 변해 왔습니다. 우리가 흔히 말하는 parametric tool들은 전회에서 밝혔듯이 하루아침에 발명된 새로운 개념이라기보다는 사람들의 머릿속에 존재하여왔으나 기술적으로 구현할 수 없었던 것들이 기술의 발전으로 인해 가능해진 결과일 뿐입니다. 최근에는 많은 수의 삼차원 설계 프로그램들이 parametric의 개념을 적극적으로 도입하면서 그 경계를 굿기가 더욱더 힘들어 지고 있습니다. 또한 한 단계 더 나아가 VB, C#, Python등의 언어를 통해 API(Application Programming Interface)에 직접 접근하여 geometry 를 다루는 algorithm을 직접 생성하는 것 또한 가능해지고 있습니다. 다음 회에서는 최근에 조명을 받고 있는 몇 가지 parametric tool들에 대해 조금 더 구체적으로 살펴보도록 하겠습니다.

2) autocad - dynamic block

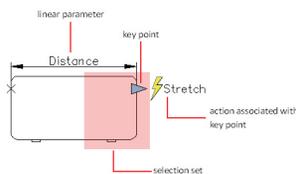
The following example shows a block reference with a constraint (in gray) and a constraint parameter (blue with grip).



Once the block is inserted into the drawing, the constraint parameters can be edited as properties by using the Properties palette.

Add Actions and Parameters

In a block definition, actions and parameters provide rules for the behavior of a block once it is inserted into the drawing.



Depending on the specified block geometry or parameter, you can associate an action to that parameter. The parameter is represented as a grip in the drawing. When you edit the grip, the associated action determines what will change in the block reference.

Like constraint parameters, action parameters can be changed using the Properties palette.

3) Catia - contextual tree

