

비정형 건축 구조 설계에서의 파라메트릭 디자인에 대한 이해

Understanding of Parametric Design in Structural Design of Complex-Shaped Buildings



이 상 수*

* (주)참소프트 아이앤아이 책임연구원

1. 서 론

오늘날의 건축계에는 IT를 통한 혁신의 바람이 거세다. 그 이면에는 같은 규격의 제품을 대량 생산하던 시대를 넘어서 고객마다의 다양하고 개성적인 요구에 맞는 제품을 대량으로 맞춤 생산해야 한다는 문화적이고 산업적인 요구가 있을 것이고, 전 지구적인 환경 문제에 대응하기 위한 친환경적인 요구도 있을 것이다. 건축계에는 이러한 요구에 대응하기 위한 기술적인 혁신의 방법론으로써 새로운

IT기술의 도입이 자연스럽게 요구되고 있다. 건설업계 전반적으로는 건축물의 전생명주기를 통괄해서 모든 정보를 통합적으로 전산 관리하고 자동 생산하게 해주는 BIM모델이 적극 도입되고 있고, 설계부문에 있어서는 다양한 설계 대안을 손쉽게 만들어내고 검토할 수 있는 파라메트릭 디자인 기법이 그 혁신의 중심에 있다.

파라메트릭 디자인이란 설계에서의 주요 속성들이나 치수들을 매개변수화하여 형태를 생성하는 디자인 방법론으로서 이 방식으로 설계를 하면, 어떤 변수가 변화할 때 자동으로 그와 연관된 형상이 변화하게 된다. 이를 통해 디자이너들은 다양한 알고리즘을 적용하여 형태를 생산해냄으로써 하나의 설계개념에 대한 다양한 대안을 실험해 볼 수 있고, 성가신 세부 디테일의 생성 문제에서 벗어날 수 있어 창의적이고 실험적인 방식으로 작업할 수 있다.

이러한 파라메트릭 디자인 기법은 1980년대에 기계산업 분야에서 처음 개발된 이후로 현재 기계, 조선, 항공 분야의 표준적인 설계방법론으로 자리잡고 있으며, 근래에는 게리 테크놀로지, 노만포스터, 자하 하디드 등의 비정형적 형상을 주로 다루는 건축가들과 아랍(Arup)등의 혁신적인 구조설계 회사들을 통해 건축분야에도 적극적으로 도입되고 있다.

본 기사에서는 이러한 파라메트릭 디자인 기법에 대해서 그것의 주요한 특징들과 한계에 대해서 일반적으로 기술해 보고자 한다.



그림 1 파라메트릭 디자인으로 설계된 바하이사원
(The Bahai Temple)

2. 파라메트릭 디자인의 개념과 구현원리

파라메트릭 디자인은 형상을 정의하는데 있어 일련의 상호 관련 파라미터(dependent parameter)와 구속조건(constraint)을 통하여 해당 형상을 정의하는 방법으로서 하나의 형상을 수정하면 상호 관련(또는 구속)된 다른 형상의 사이드나 위치가 변하게 된다. 이러한 방법론은 복잡한 형태를 구성함에 있어서 다층적인 복잡성의 수준을 가진 복합체를 만들게 되고, 그 각각의 수준들은 그 구성원들 간에 또 상하 위계적인 계층 간에 서로 연결되어 영향을 미친다는 점에서 '연관적인 설계(Associative Design)'라고도 불린다.

파라메트릭 방법론은 그 구현에 있어서 구성이력트리(through 단방향의 연관성을 가지고 구현되는 이력기반 시스템(history-based system)과 구성이력트리가 없이 상호연관이 가능하며, 구속조건들을 동시에 풀어내는 구속조건 솔버를 통해 구현되는 방식(constraint solver system)으로 나뉘지며, 하나의 시스템 내에서 그 두 가지가 결합되어 구현되기도 한다.

2.1 이력기반 시스템(history-based system)

이력기반 시스템은 모델의 구성요소들간의 의존성이 단방향이다. 즉 어떤 A라는 구성요소에 의존해서 B라는 구성요소가 생성된다면, B는 A의 자식이 되고, A는 B의 부모가 되며, 그 반대의 참조는 불가능하다. 이런 구조에서 상위 부모 요소의 변화는 즉각적으로 그 자식 요소들에게 전파(propagation)되며 전체 형상을 재구성하게 된다. 반대로 자식 요소의 변화는 상위에 있는 부모 요소에는 영향을 미치지 않는다. 이러한 부모-자식간의 참조관계는 부모가 자식

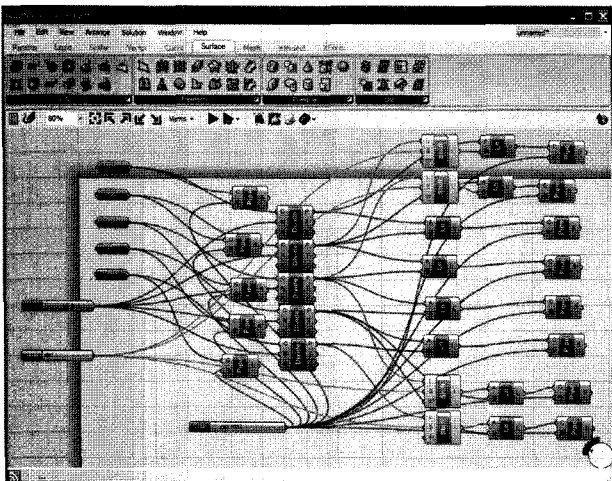


그림 2 이력기반 시스템의 예, Grasshopper의 모델 그래프

에 항상 선행하게 되므로 생성된 순서 또는 이력(history)을 기록한 것처럼 보이고, 또 입력값의 변화가 있을 때에는 그 순서대로 재실행하기 때문에 이력기반 시스템이라 불린다. 수학적으로는 방향성 비순환 그래프(directed acyclic graph)로 표현된다.

이런 시스템의 장점은 모델의 생성 과정이 단순하고 명확하다는 것이다. 또 대부분의 설계의도는 이러한 단방향 의존성에 의해 충분히 표현 가능하지만(예를 들면, 계단의 단높이는 층고에 의존적일 수 있지만, 계단의 단높이에 따라 층고를 변화시키고 싶은 설계자는 없을 것이다.), 복잡한 설계의도를 표현하기 위해 상호의존적인 관계를 설정해야 할 경우에는 한계가 있다.

2.2 구속조건 솔버 시스템(constraint-solver system)

모델에서 상호의존성을 허용할 경우의 가장 큰 문제는 인지적 복잡도가 기하급수적으로 증가한다는 것이다. 이러한 양방향 의존 모델의 복잡성을 해결하기 위해 선호되는 방식이 구속조건 솔버를 사용하는 것이다. 구속조건 솔버는 구성요소들 간에 정의된 모든 의존성을 동시에 만족시키는 해를 찾아내 최종적인 형상을 결정한다.

이러한 접근법은 앞에서 말한 이력기반 시스템과는 상당히 다른 접근법을 요구한다. 예를 들면, 직선 A, B를 평행하게 모델링한다고 할 때, 이력기반 시스템에서는 직선 A를 먼저 만들고, 직선 B를 만들 때, A로부터 d만큼 떨어져서 평행하게

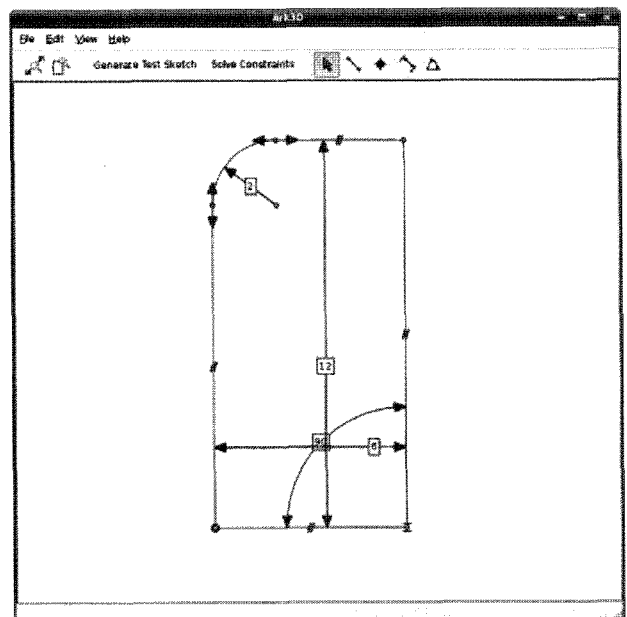


그림 3 구속조건 솔버 시스템에서 구현된 단면 스케치-평행 조건과 접선 조건, 사이각 조건 등이 형상을 규정하고 있다.

만든다라는 식으로 모델링하겠지만, 구속조건 솔버 시스템에서는 A, B를 동시에 그린후, A, B는 평행이라는 조건과 중심 간 거리는 d라는 조건을 걸어주어서 모델링하게 된다. 전자에서는 A를 움직이면 B가 조건에 따라서 갱신되지만, B는 A에 의존적이므로 독립적으로 이동할 수 없다. 후자에서는 A, B를 독립적으로 움직일 수 있으며, 그에 따라 각각 상대편 직선이 조건을 만족하는 형태로 갱신되게 된다.

구속조건 솔버를 사용하는 방식은 솔버에서 미리 정의된 구성요소와 구속조건만을 사용할 수 있으며, 늘 하나의 형상을 결정지을 만큼의 구속조건을 모자라지도 과하지도 않게 설정하기가 어렵고, 하나의 조건의 변화가 다른 형상에 미치는 영향을 예측하기도 어렵다는 단점이 있다. 이러한 한계로 인해 상호의존성을 허용하는 구속조건 솔버 시스템은 현재 파라메트릭 모델러의 일부(입체를 만들기 위한 초기 단면의 스케치 등)에서만 적용되고 있다.

3. 파라메트릭 디자인의 특징과 장점

기존의 디자인 방법론에 비해 파라메트릭 디자인이 가지는 주된 특징은 다음과 같다.

3.1 설계의도를 담아낸다.

파라메트릭 디자인은 하나의 설계요소가 다른 설계요소에 연관되고 종속되기 때문에 그 과정에서 자연스럽게 설계자의 설계의도가 명시적으로 기록된다. 설계자는 어떤 특정한 치수나 초기의 기하적인 구성에 의존하지 않고 설계요소들간의 배치나 상호 의존성을 규정할 수 있다. 예를 들면, 두 개의 직선이 있다고 했을 때, 직선 a는 직선 b와 연관하여 얼마의 거리로 어떤 조건(평행 등)으로 그려져야 하는지를 설계자가 자신의 의도대로 규정할 수 있다는 것이다. 또한, 형상 자체가 특정한 로직의 구현물이라면(오늘날 초고층 건물의 외피나 골조의 디자인에서 많이 적용되는 것처럼), 파라메트릭 모델링은 이러한 구조물을 설계로 구현하는 가장 자연스러운 방법이 된다. 로직의 단계별 산출물들간의 연관관계를 통해 최종 형상을 구현해 내는 과정으로 모델을 구성할 수 있기 때문이다. 파라메트릭 모델링의 이러한 로직 구현성과 관련된 특징을 특별히 강조하여 언급할 경우 알고리즘 기반 모델링(Algorithmic Modeling)이라고 부르기도 한다. 이러한 점은 각각의 선들이 아무런 관계없이 고정된 기하학적 좌표에 의해 나열되거나 단순한 위상적 연결 관계만을 가지던 기존의 설계모델과는 확연히 구분되는 특징이다.

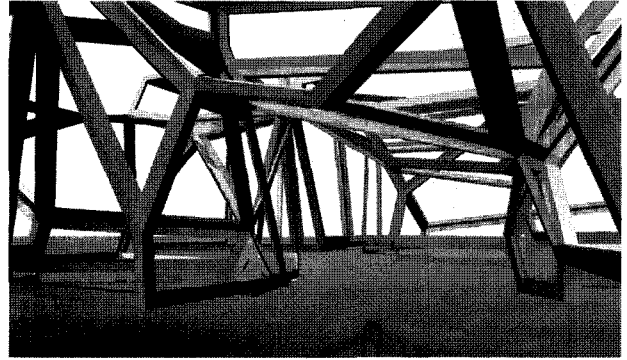


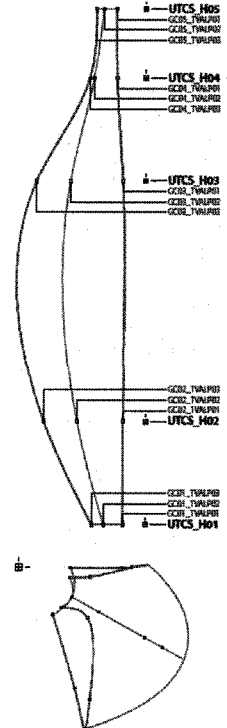
그림 4 보로누이 알고리즘에 의해 생성된 구조물 - 유기적인 자연스러움을 가지고 있다.

3.2 설계자가 일일이 설계 모델을 다시 만들지 않고도 대안적인 설계 시나리오를 검토할 수 있게 한다.

일단 모델과 관계가 정의되고 나면, 설계자는 매 번 완전한 모델을 재입력할 필요없이 그 모델을 통해 대안적인 설계 접근법을 가지고 다양한 실험을 해 볼 수 있다. 따라서, 파라메트릭 설계법을 통해 설계자는 자신의 설계전략에만 집중할 수 있게 된다. 예를 들면, 층간 높이, 기둥 간격, 하중의 작용점과 크기 등에 대해서 파라메트릭 디자인을 적용한 모델은 각각의 변수만 새로 적용시키면 그에 따라 전체

Name	Value	Property Value
GC01_EXPDIR01	28	
GC01_EXPDIR02	100	
GC01_RADC	5	
GC01_TVALP01	0	
GC01_TVALP02	0.920	
GC01_TVALP03	0.880	
GC02_EXPDIR01	28	
GC02_EXPDIR02	100	
GC02_RADC	10	
GC02_TVALP01	0	
GC02_TVALP02	0.920	
GC02_TVALP03	0.820	
GC03_EXPDIR01	28	
GC03_EXPDIR02	100	
GC03_RADC	11	
GC03_TVALP01	0	
GC03_TVALP02	0.920	
GC03_TVALP03	0.830	
GC04_EXPDIR01	28	
GC04_EXPDIR02	100	
GC04_RADC	7	
GC04_TVALP01	0.980	
GC04_TVALP02	0.920	
GC04_TVALP03	0.900	
GC05_EXPDIR01	28	
GC05_EXPDIR02	100	
GC05_RADC	5	
GC05_TVALP01	0.970	
GC05_TVALP02	0.920	
GC05_TVALP03	0.880	
UTCS_H01	0	
UTCS_H02	15	
UTCS_H03	50	
UTCS_H04	65	
UTCS_H05	75	

3.16 Visual sliders aid in the manipulation of Global Variables and create a customizable user friendly modeling environment



3.17 Variable Connection Diagrams

그림 5 파라메트릭 디자인 기법으로 만들어진 바하이사원의 모델 상세

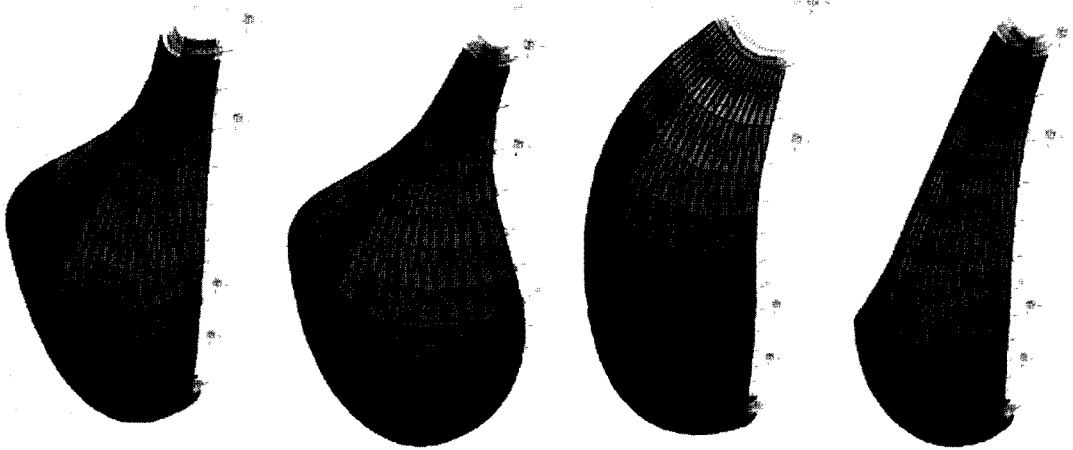


그림 6 자동생성된 설계대안들
그림 5에 나타난 슬라이더를 조절함으로써 그림 6과 같은 다양한 대안형상들을 자동으로 만들어낸다.

적인 모델이 새로 재구성되게 된다. 부수적으로 도면 추출 역시 3차원 모델에서 바로 매개변수적으로 이루어지게 되기 때문에 늘 최신의 변경사항이 도면에 자동 반영된다.

3.3 최적화가 가능하다.

3.2에서 바로 응용되어지는 분야가 바로 설계의 최적화이다. 특히 엔지니어링 분야에서 중요한 이 특징은 이미 기계계열의 CAE프로그램에서는 일반화된 사항이다. 주요한 설계 매개변수를 최적화의 대상으로 삼고 자동으로 그 값을 단계적으로 변화시키면서 생성되는 모델의 성능값이 어떻게 변화하는지를 추적할 수 있기 때문이다. 기존의 구조 해석 프로그램이 이미 고정된 구조형상을 두고 단면 형상만을 최적화의 대상으로 삼는 것과는 달리, 보다 근본적인 시스템의 최적화-전체적인 형상, 부재의 배치, 레이아웃 수준에서의 최적화가 가능하다. 예를 들면, 아트리움 지붕을 설계하는데 스페이스 프레임으로 지붕을 덮는다고 하자. 파라메트릭 모델 설정시, 스페이스 프레임을 아트리움 지붕의 곡면의 형상을 따라 특정 전략에 따라서 자동배치되도록 했다고 한다면, 설계자는 지붕의 곡면형상을 다양하게 변화시켜 보면서 자신의 스페이스 프레임 배치 전략이 잘 적용되는지 어떤 형상에서 최적화 되는지를 검토해 볼 수 있다.

3.4 설계의 재활용이 가능하다.

파라메트릭 디자인의 설계모델은 고정된 형상을 가지고 있지 않다. 그것은 설계개념의 구현물이며, 따라서 매개변수의 설정값을 변화시킴으로써 다양한 환경에 적합하게 최

종 형상을 재생성시킬 수 있다. 예를 들면, 파라메트릭 디자인 기법을 통해 만들어진 교량 설계물은 다양한 지형에 적합하게 변형되어질 수 있기에 같은 유형의 교량이 필요한 곳이면 얼마든지 손쉽게 재사용될 수 있다. 시공도면의 작성이나 부재의 공장제작과정까지 연결될 경우 그 활용성은 극대화된다.

3.1, 3.2, 3.3의 특징에서 살펴본 바와 같이 파라메트릭 디자인 방법론은 명확하게 설계의도를 반영한 구조물을 설계할 수 있도록 하며, 그 설계의도 내에서 다양한 대안들을 손쉽게 만들어 낼 수 있게 해주고, 또한 그 각각의 대안들을 평가하여 최적의 안을 도출해 내는 작업을 자동화할 수 있게 해준다. 이러한 특징들을 통해 설계의 여러 단계들 중에서도 특히 초기 단계에서 하나의 설계안이 도출되었을 때 그것을 평가하고 발전시키는 단계에서 파라메트릭 디자인은 매우 강력한 도구가 될 수 있다.

4. 파라메트릭 디자인의 한계와 단점

세상의 모든 것이 그렇듯이 파라메트릭 디자인은 만능이 아니며, 파라메트릭 디자인이 가지는 장점들은 저절로 얻어지는 것이 아니다. 파라메트릭 디자인이 가지는 한계 또는 단점은 다음과 같다.

4.1 구속조건 설정의 어려움

유용한 파라메트릭 모델을 만들기 위해서는 설계개념이 머릿속에 명확히 그려져 있어야 한다. 설계의도, 또는 모델이 어떤 기능을 할 것인가에 대한 개념을 정확히 가지고 있어야 구성 요소들간의 논리적 구속관계를 설정해 나갈

수 있기 때문이다. 그러나 이러한 명확한 설계의도가 아직 서지 않은 시점에서 무리하게 모델을 만들어 갈 때에는 창조적인 설계 프로세스에 제약을 받거나 최악의 경우는 나중에 완전히 모델을 재설계해야 하는 상황에 직면할 수도 있다. 디자인 초기의 창조적인 설계과정에서는 명확한 설계의도가 존재하지 않으므로, 모델의 어떤 부분을 매개변수화하고 변경가능한 부분으로 만들지, 또 모델을 어떤 논리적 과정으로 전개시켜 나갈지가 불분명하다. 이런 단계에서는 파라메트릭 디자인 방법에 의한 구속조건과 논리적 관계를 정의하며 모델을 전개시키는 것이 오히려 창조적 설계에 방해가 될 수 있다.

파라메트릭 디자인 기법이 가장 적절하게 사용될 수 있는 단계는 기본적인 설계개념은 명확해졌지만, 세부적인 변형에 대해 열려있는 상태이다. 즉 초기설계시 전혀 다른 시스템 A, B, C를 고안해 내는 데에는 적합하지 않지만, A가 결정되고 나서 A1, A2, A3 등의 다양한 변형을 만들어 내고 실험하기에 적합하다는 뜻이다.

4.2 거대화되는 모델관리의 어려움

파라메트릭 모델은 설계가 진행되어 갈수록, 구속조건간의 복잡성이 쌓여서 예기치 않은 문제를 일으킬 가능성이 있다. 모델 이력상의 상위부분의 변경은 그로부터 파생된 모든 형상에 영향을 미치게 되는데, 모델이 복잡해지면 그러한 파생된 형상들의 변화를 명확하게 알아채기 어렵고 의

도하지 않은 부작용을 낳을 수도 있다. 또, 구속조건간의 상호충돌, 또는 이력상 앞에 있는 요소의 변화가 후에 있는 요소의 요구조건을 만족시키지 못함 등의 이유로 인해서 모델 재생성이 실패하는 경우도 생길 수 있다. 나중에 만들어지는 형상들은 먼저 생긴 형상을 참조하며 의존적이므로 특정 형상을 수정하거나 제거하는 데에도 제약이 따른다.

이러한 수정의 어려움은 팀으로 작업하거나 다른 사람이 오래전에 작업한 결과를 가지고 수정해야할 때 더욱 커진다. 기존의 도면은 보이는 것이 전부이므로 관련된 부분들을 자유롭게 수정할 수 있지만, 파라메트릭 모델은 설계의도를 파악하고 신중하게 수정할 부분을 파악해 내야 하기 때문이다.

4.3 논리적 구성과 학습의 어려움

패트릭 슈마허가 파라메트릭 디자인에 대한 선언에서 언급한 것처럼 파라메트릭 디자인은 정교하고 세련된 파라메트릭 테크닉을 통해서만 존재를 유지할 수 있다. 여기서 세련된 파라메트릭 테크닉이란 형상을 만들어감에 있어서 매개변수를 설정하고 그들간의 논리적 의존관계를 설정해서 최종적인 형상을 이끌어 내는 감각과 기술을 의미한다. 알고리즘기반 모델링에서처럼 파라메트릭 디자인 방법론을 알고리즘과 결합해 보다 본격적으로 사용하기 위해서는 프로그래머가 하는 것과 같은 알고리즘을 작성하는 능력 또한 필요할 수 있다. 이 모든 개념과 기술들은 기존의 디자

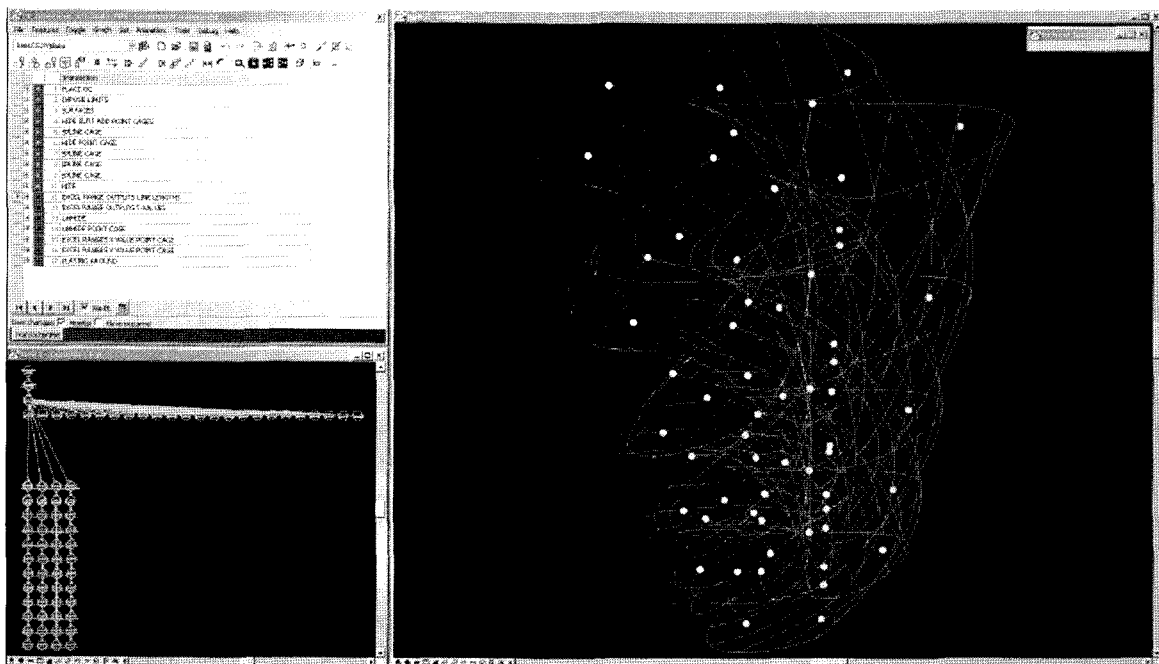


그림 7 GC의 GUI모델링 환경 - 우측에 3차원 그래픽 모델이 나타나고 좌측 하단에 이 모델의 그래프 구조가 나타나 있다.

이 너들에게는 새로운 것이며 쉽게 접근하기 어려운 진입장벽이 될 수 있다.

사실상 파라메트릭 디자인에 있어서 최초의 접근법은 성공적이지 못한 경우가 많으며, 그로부터 얻어진 경험을 통해 새로운 접근법을 고안해가며 설계가 진행된다. 이러한 접근법에서는 쉽게 재생성될 수 있는 형상 결과 보다는 주어진 설계 문제에 대한 이해를 반영하는 구성요소들간의 의존성과 관계들이 더 중요하다는 것을 이해할 필요가 있다.

5. 파라메트릭 디자인에서의 모델링 방법

5.1 모델기반 접근법

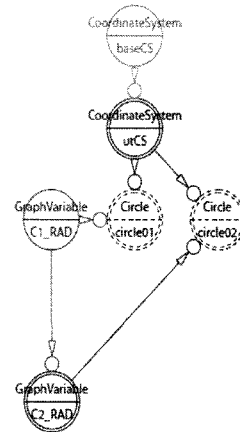
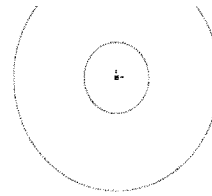
기존의 캐드 환경에서의 모델링은 주로 GUI환경 하에서 지오메트리 도구들을 중심으로 이루어진다. 파라메트릭 모델링 환경에서도 이와 유사한 수동 모델링 환경을 제공하는데, 요소들 간의 관계 및 요소가 생성되는 방식을 명시적으로 규정한다는 점에서 차이가 있다. 즉, 새로운 구성요소가 추가될 때 기존의 구성요소의 전체 또는 일부를 참조하며 그것을 기반으로 특정한 생성방식을 적용해 새로운 요소를 만들어낸다는 것이다. 예를 들면 하나의 실린더를 생성한다면, 미리 만들어 둔 원 형상을 단면으로 참조하고 그것으로부터 수직 방향으로 뽑아내는 방식으로 만들게 되며 이 모든 설정사항은 모델과 함께 저장되어 언제든지 수정할 수 있게 된다. 각각의 형상들은 고유의 이름이 있고, 그들 간의 관계는 그래프 형식으로 나타나게 된다.

5.2 스크립트 기반 접근법

5.1에서와 같이 지오메트리를 직접 만드는 방식 외의 또다른 접근법으로는 형상들을 정의하고 조작하기 위한 생성명령어들을 스크립트로 작성하는 방법이 있다. 스크립트는 수동 모델링 환경과 공존하면서도 프로그래밍에 한발짝 다가설 수 있는 방법이다. 스크립팅에 의한 모델과 수동모델링은 정확히 동일한 모델을 산출하지만 동시에 좀 더 효율적인 생성명령과 조건부 및 반복적 로직(반복문, 조건문 및 함수)을 적용할 수 있게 해 준다. 이러한 스크립트 기반 방법의 주된 장점은 다량의 입력형상들로부터 다량의 출력형상들을 반복적으로 만들어 낼 때의 편의성이다.

5.3 프로그래밍 기반 접근법

GC 등의 최근의 모델링 소프트웨어들은 매우 유연하고 강력한 프로그래밍환경(C# 등의 고급언어와 VisualStudio와



```

transaction modeBased "C2"
{
  feature GC.CoordinateSystem uCS
  {
    SymbolXY = {100, 101};
  }
  feature GC.GraphVariable C1_RAD
  {
    UsesNumericLimits = true;
    NumericLowLimit = 1.0;
    NumericHighLimit = 30.0;
    SymbolXY = {99, 103};
  }
  feature GC.Circle circle01
  {
    SymbolXY = {100, 103};
    Construction = ConstructionOption.Construction;
  }
  feature GC.GraphVariable C2_RAD
  {
    Value = C1_RAD*Pow(2, C1_RAD/3);
    SymbolXY = {99, 105};
  }
  feature GC.Circle circle02
  {
    CenterPoint = uCS;
    Radius = C2_RAD;
    Support = uCS.XYplane;
    Construction = ConstructionOption.Construction;
  }
}
    
```

그림 8 GC에서 2개의 원을 그렸을 때의 모델 그래프와 그에 대응하는 스크립트

같은 강력한 통합개발환경)을 제공한다. 이런 프로그래밍 기반 접근법은 많은 기하학적 구성요소를 가진 계산집약적인 형상구현 로직들(자연형상을 모사하거나 인공지능적인 최적화기법을 이용하는 로직 등)을 구현하는데 있어 매우 유용한 접근법이다. 이러한 로직은 수동모델링 기법으로는 거의 구현이 불가능하고, 스크립팅을 한다고 해도 지나치게 느려서 사용성이 매우 떨어지게 된다. 이러한 프로그래밍 기반 접근법은 전체적으로 적용되기 보다는 일부 계산집약적인 생성모듈에 적용되어 전체 모델의 일부로 사용된다.

이상의 세가지의 접근법들은 점진적으로 더 유연하고 강력한 설계를 구현할 수 있는 방법들로서 후자로 갈수록 복잡한 설계에 있어서 중요하며, 실제 설계에 있어서는 세가지 방법이 혼합되어 함께 사용된다.

6. 파라메트릭 디자인을 지원하는 소프트웨어 툴

PTC(Parametric Technology Corporation)가 1980년대에 동사의 CAD 시스템인 Pro/Engineer를 통하여 이러한 파라메트릭 모델링 방법론을 처음으로 소개한 이후 파라메트릭 디자인은 기계, 조선, 항공 분야에서 가장 널리 사용되는 설계방법론이 되었다. 기계분야에서는 AutoDesk사의 Inventor 나 CATIA가 이러한 방법론을 지원하는 대표적인 소프트웨어

어들이다. 이렇듯 파라메트릭 디자인을 지원하는 도구들은 이미 등장한 지 수 십년이 지났지만, 건축 실무에 본격적으로 사용되기 시작한 것은 극히 최근의 일이다. 오늘날에는 건축분야에서도 다양한 제품들이 개발되고 있는데, Autodesk사의 Revit, GraphiSoft사의 ArchiCad, Bentley의 Generative Component, Gehry Technologies의 Digital Project(CATIA의 건축분야의 파생제품), McNeal의 Grasshopper등이 대표적이다. 이들 중에서 Revit, ArchiCad 등은 BIM을 위한 제품들로서 자신의 3D BIM 모델을 구축하기 위한 방법으로서 다소 정형적인 형상의 구조물에 대한 파라메트릭 모델링을 지원하지만, GenerativeComponents(GC)나 Grasshopper는 보다 개념적 설계에 적합한 제품들로서 창의적이고 실험적인 설계를 위한 본격적인 파라메트릭 디자인 툴들이다. 특히 GC는 Bentley Structural과 연계하여 바로 구조해석과 최적화가 가능한 모델을 만들어 내고 그 결과를 피드백 받을 수 있다.


7. 결 론

이상과 같이 오늘날 건축계에 일어나고 있는 새로운 혁신의 중심에 있는 파라메트릭 디자인의 개념과 특징에 대해 설명해 보았다. 오늘날의 자유형상 건축 또는 신유기적 건축에 대한 새로운 관심은 강력한 컴퓨팅 성능의 출현으로 인해 가능해졌다. 복잡한 곡률을 가진 복잡한 형태들을 수학적으로 기술하고 변형시키며 다양한 실험을 해 볼 수 있게 된 것은 이러한 강력한 컴퓨팅성능에 의해 뒷받침되는 파라메트릭 디자인 방법론 덕분이다.

파라메트릭 디자인은 매우 유연하고 강력한 설계방법으로서 다양한 설계대안을 만들어 내고 검토하여 최적의 설계를 내는데 유용하며, 고도로 반복적이며 복잡한 계산을 요하는 로직의 적용을 통해 기존의 방식으로는 만들어낼 수 없었던 창의적인 형상을 만들어낼 수 있다. 그러나 새로운 방법론을 잘 활용하기 위해서는 그만큼의 새로운 지식에 대한 배움의 열의와 시행착오적인 탐구를 필요로 한다. 파라메트릭 디자인은 설계의 많은 부분을 자동화해주는 편리한 도구이지만, 그만큼 세심하고 명확한 사전설계가 전제되어야 하는 방법론이기도 하다.

이러한 어려움에도 불구하고 그 장점은 너무나 혁신적이고 명확하다. 새로운 시대적 요구는 이러한 새로운 접근법을 통해서만 해결될 수 있기에 오늘날 경쟁력을 가지고 시대를 이끌어 가려는 설계자들이라면, 열린 마음으로 파라메트릭 디자인이라는 새로운 방법론을 탐구하고 적용해 볼 필요가 있다고 하겠다.

참 고 문 헌

1. Parametric Design: An Implementation of Bentley Systems GenerativeComponents by Mark Andrew Cichy
2. Introduction to GenerativeComponents by Robert Aish, Ph.D
3. Algorithmic Modeling with Grasshopper by Mohamad Khabazi 

[담당 : 김현수, 편집위원]