글. 성우제_ Sung, Woojae

OMA NYC office / Senior Architect

www.woojsung.com, www.selective-omplification.net

파라메트릭 디자인 Ⅲ

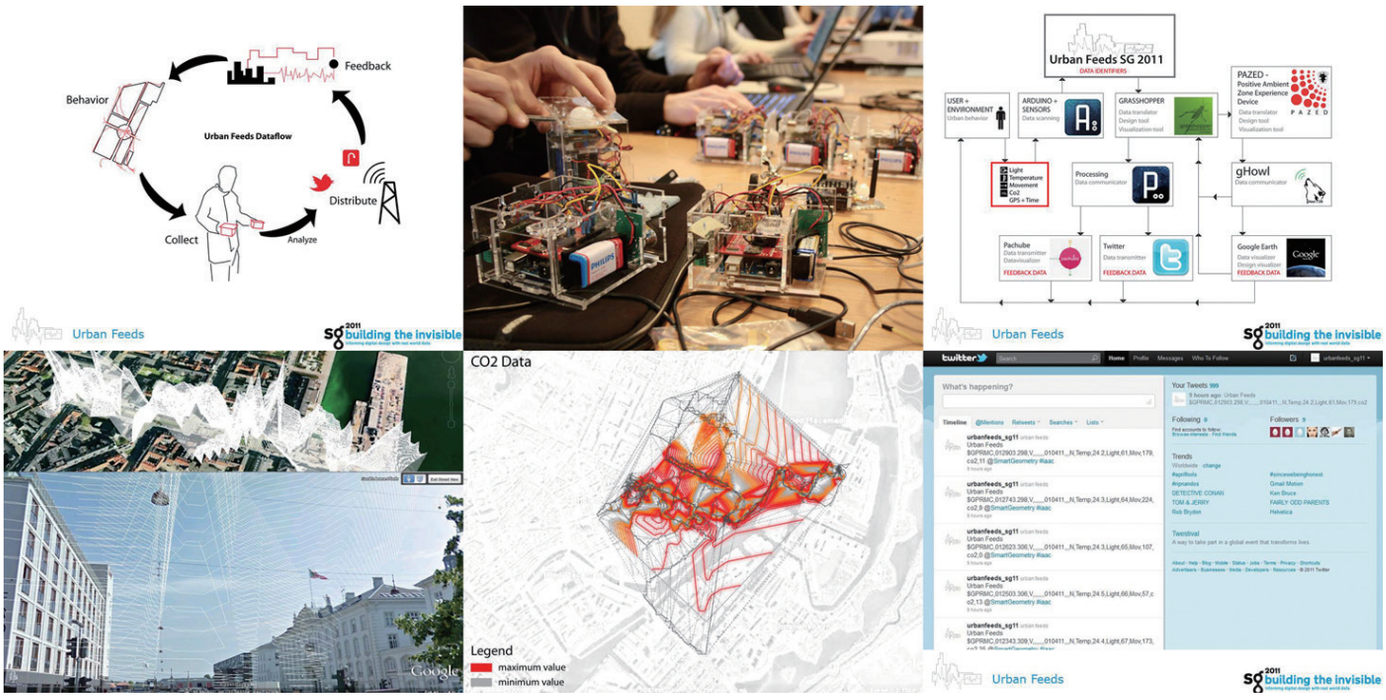
Parametric Design Ⅲ

지난 회에서는 parametric tool의 근간이 되는 기본 개념에 대해 간략하게 논의 하였고, 이를 통하여 삼차원 프로그램상의 형태 구현의 논리와 방법이 어떻게 발전해 왔는지에 대해 이야기 해 보았습니다. 전회에 말씀드렸듯이 이번회에서는 최근 실무에서 가장 많이 사용되고 있는 parametric tool에 대해 간략히 알아보도록 하겠습니다. 참고로 이번 회를 마지막으로 parametric tool의 개략적인 소개는 마치도록 하고 현장감 있는 실무에서의 적용에 대해 이야기 해 보도록 하겠습니다.

■ Rhino/Grasshopper

Rhino는 Autocad와 흡사한 command line interface및 아주 유사한 기본적인 명령어들로 인해 캐드를 다룰 줄 아는 많은 사용자들을 흡수했습니다. 그뿐 아니라 정확한 수치 작업 및 손쉬운 도면화에 큰 약점이 있는 Max나 Maya, 비정형 설계에 큰 약점을 가지고 있는 Sketchup, 혹은 매니아 층을 형성했으나 사용의 어려움과 이로 인한 건축분야의 저변확대에 실패한 FormZ나 Alias등의 사용자들을 흡수하며 가장 널리 사용되는 삼차원 설계 툴로 자리잡게 되었습니다. 그러나 geometry를 생성해 낼 수 있는 가장 방대한 library를 지닌 툴 중 하나임에도 불구하고 Max나 Maya가 가지고 있는 history 기능이 빈약하였고 Catia나 Generative-Component의 parametric 기능이 전무했었습니다. 이를 만회하기 위한 시도로 Rhino의 플러그인 형식으로 구현되는 초기 버전이 Explicit History라는 이름으로 처음 소개가 되었었습니다. 이전 회에서 언급한 바와 같이 parametric tool에서 history의 중요성을 단적으로 말해주는 이름이라고 생각됩니다. 하지만, 이후 동물이나 곤충의 이름을 따는 McNeel사의 관행에 따르기 위함인지 이름을 Grasshopper라고 변경하였고 2008년 이후 현재까지 비약적인 발전을 하게 됩니다.

1) Urban Feeds : Smart Geometry Workshop at Copenhagen, 2011. 코펜하겐 시내에 설치된 센서를 통해 온도, 습도, 바람의 세기와 같은 데이터를 엑셀 형식의 데이터로 얻은후 Grasshopper와 몇가지 플러그인을 통해 Rhino상에서 삼차원화 한후 이를 Google Earth및 Twitter에 포스팅하여 실시간 정보를 시각화한 프로젝트.



■ 입출력 및 Component, Graphical Editor

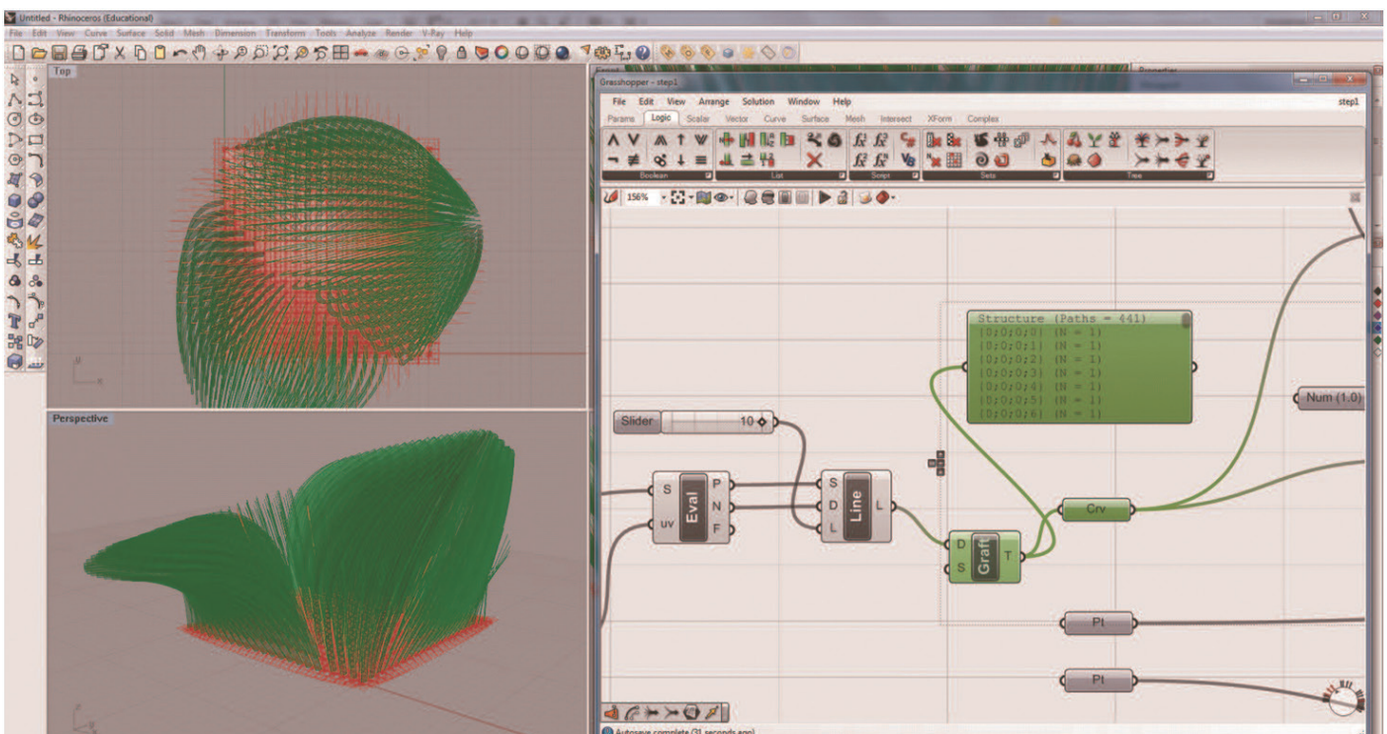
모든 parametric tool들이 그러하듯이 Grasshopper도 입력된 정보를 가공하고 이를 출력하는 세 가지의 큰 몸통으로 나누어 집니다. 입력되는 정보는 초기 버전에서는 Rhino와 Grasshopper 상에 존재하는 어떠한 형태의 정보, 즉 geometry및 문자열과 숫자 등으로 제한적이었으나 새로운 버전이 업데이트되고 Grasshopper 상에서 작동하는 또다른 플러그인들이 계속해서 개발됨에 따라 현재는 로컬 컴퓨터상의 혹은 인터넷을 포함한 네트워크상에서의 정보, 심지어는 센서를 통한 정보들까지 광범위한 데이터를 입력하는것이 가능해졌습니다[1]. 이렇게 입력되어진 정보는 단순한 하나의 작업을 수행하는 component에 입력이 됩니다. component는 Rhino나 윈도우 시스템에 이미 정의되어있는 API에 따라 입력된 정보에 기반하여 새로운 정보를 생성하거나 입력된 정보로 부터 필요한 속성 등을 추출하게 됩니다. 각각의 component는 어떠한 형태로든 출력을 하게 되는데요, component에 의해 가공되어지는 방식에 따라 입력가능한 정보의 형태만큼이나 다양한 형식을 취하게 됩니다. 이렇게 출력되어지는 정보는 Rhino상에 geometry로 표현이 되어 최종 결과물이 되거나 혹은 대다수의 경우 또다른 component의 입력값이 되기도 합니다. 이러한 일련의 과정은 Grasshopper상 canvas에 graphical editor[2] 의 형식으로 표현됩니다.

Graphical editor의 가장 큰 장점은 geometry에 대한 추상적인 층위에서의 이해에 큰 도움을 준다는 것입니다. 이는 특히 방대한 데이터를 정해진 algorithm에 따라 일괄적으로 다루어야 하는 대다수의 건축 프로젝트에 효율성과 일관성의 맥락에서 큰 도움이 된다고 할수 있습니다. Bentley Systems의 Generativecomponent나 Autodesk의 Revit에 기반을 둔 Dynamo와 같은 프로그램도 모두 이러한 graphical editor를 사용하고 있습니다. 반면 Catia와 같이 별도의 graphical editor를 가지고 있지 않은 프로그램 같은 경우 – 물론 tree라고 불리는 hierarchical navigation tool 이 있기는 하지만 – algorithm을 다루기 보다 삼차원 객체를 직접 다룸으로써 geometry에 대한 좀 더 직관적인 이해를 도와주지만 전체적인 큰 그림을 보는데는 어려움이 따르곤 한다고 생각됩니다.

■ Revit/Dynamo & Catia

전 회와 이번 회를 통해 잠시 언급된 바와 같이 Rhino/Grasshopper 이외에도 많은 수의 삼차원 설계 프로그램들이 parametric tool을 개발중에 있습니다. 하지만 범용성과 라이센스의 가격, 그리고 두터운 유저층에 힘입어 Rhino/Grasshopper가 가장 큰 성공을 이루고 있다고 생각됩니다. Revit에 기반을 둔 다이나모는 Grasshopper와 유사한 interface로 많은 관심을 모으고 있습니다. 특히나 많은 수의 미국내 프로젝트가 Revit으로 도면 작성을 요구함에 따라 parametric tool을 사용한 설계를 Revit에서 직접 할수 있다는 장점에 많은 주목을 받고 있는데요, 현재로서는 Revit이 가지고 있는 삼차원 라이브러리가 아주 제한적이라 Dynamo 자체의 가능성에 많은 걸림돌이 되고 있습니다. 단적인 예로, Revit에서는 현 버전까지도 기울어진 벽체를 벽으로 인식하지 못하고 generic mass로 인식하는데요, 이는 일반적인 도면뿐 아니라 디테일 그리고 적산에 이르기 까지 자동화를 추구하고자 하는 Revit의 사용이유에 부합하지 못할 뿐더러 결국 Autocad 사용시와 같이 추가적인 작업을 요구하게 됩니다. 반면 Catia의 경우는 볼트와 너트하나까지 삼차원화 하고 이를 추가적인 노동없이 fabrication을 위한 도면화를 할수 있다는 커다란 장점을 가지고 있으나 비싼 라이센스 가격 및 건축쪽에서 얕은 유저층으로 인해 범용화 되지 않았다는 점이 걸림돌이 되고 있다고 할 수 있겠습니다.

이상으로 최근 가장 많은 조명을 받고 있는 몇가지 parametric tool에 대해 살펴 보았습니다. 첫 회부터 지금까지 조금 지루하셨을지 모르겠는데요, 다음 회부터는 현재 미국내의 건축사사무소들과 건축 대학들이 어떻게 parametric tool을 이용하여 작업을 하는지 사례들에 대해 자세히 알아보도록 하겠습니다.

---

2) Grasshopper의 Graphical Editor