

예측과 번역 사이에서 난잡하게 진화하는 조형

남웅(인권활동가, 미술평론가)

소프트웨어와 예술(가)

이 텍스트는 한글과 구글 드라이브를 왕래하며 작성했다. 기본적으로는 집에 있는 데스크탑 컴퓨터를, 이동 중에는 노트북과 태블릿, 스마트폰을 이용한다. 그러니까 이 문장은 몇 개의 프로그램과 기기들의 상호 호환 속에 쓰였다.

똑똑한 기기들은 사용자에게 멀티태스킹을 유혹한다. 말인즉 글을 쓰면서도 다른 용무를 처리한다. 급하게 알고 싶은 내용을 검색하거나 간간히 온라인 북을 읽기도 하며 SNS를 하고 메일을 체크하다가 맛집을 검색하고 알고리즘에 걸리는 상품들에 눈을 뺏기고 넘어가기가 부지기수다. 일상에 프로그램을 이용하는 순간보다 이용하지 않는 순간을 생각하는 것이 좀 더 빠를 것이다. 하지만 익숙하다고 내가 온전한 사용자라고 인지하는 것은 착시일지 모른다. 그러니까 글을 쓰는 일은 소프트웨어에 어김없이 예측되어 있음을 인지하는 일이다. 이 푸념마저 프로그램이 없으면 이렇게 남길 수도 없다.

사용자라면 한 번쯤 의심했을 고민이 작가도 예외는 아니었다고 한다. 그는 사용하는 소프트웨어가 이미 주어진 구조 아래서만 사용할 수 있다는 무력함을 느꼈다고 한다. 소프트웨어는 일상의 리듬뿐 아니라 작업의 형식과 그 환경을 재구성하고, 예의 프로세스에 몸의 리듬을 적용시켜 체화한다. 이에 작가는 소프트웨어 프로그램을 조형의 대상이자 소재로 삼으며 응전도 모한다. 다만 방법에 있어서는 새로운 도구를 활용하여 구조와 질서를 창안해서 개입하기보다, 대상으로 삼은 소프트웨어의 구조를 추출하고 그 코드 값을 유전자 염기서열로 간주하여 몸체를 재구성하기를 택한다.

생명의 기둥이라 부르는 DNA 염기서열의 구성이 코드화한 단위임을 고려하면, 프로그램 코드에 염기서열을 유비하는 일이 새롭거나 어색할 것은 없다. 주지할 점이라면 작가는 직접적인 연결고리가 없는 소프트웨어와 유전자의 구조를 조형의 자원이자 문법으로 삼으며 자의적으로 포개는 점이다. 가령 그는 html 태그에 단백질을 만드는 20가지의 아미노산과 대응시키는 번역 체계를 세운다. 화학식으로 변환한 웹사이트 태그 데이터를 분자생물학에서 사용하는 분자구조 계산프로그램에 집어넣어 3차원 태그의 배열을 입체 형태로 구현한다. 그렇게 코드로 이루어진 도면에 상응하는 웹사이트의 입체 구조를 추적할 방법을 고안한 것이 <웹사이트 구조의 편집 방법 개발>(2020)이다.

서로 다른 두 정보 값을 매칭하는 일은 사용자의 주도권을 뺏은 프로그램을 뼈대부터 분석하여 갈취하는 해킹의 조형을 떠올리게 한다. 프로그램을 코드 값으로 해체하고 유전자 문법에 대입한 결과를 3D 프린팅 프로그램을 통해 산출하는 공정은, 각각의 소프트웨어를 배치하고 구동시키며 번역과 오역을 동시적으로 수행하는 가운데 이뤄진다. 이는 프로그램을 선별하고 배치하는 이른바 비평적 메타 프로그램을 설계하는 창조자로서의 예술가 주체를 떠올리게 만든다.

소프트웨어 프로그램은 그 자체로 독립적이지만, 완결되지 않은 채로 사용자들의 피드백을 받으며 오류를 수정하고 업데이트하며 주어진 용처를 늘린다. 프로그램을 사용하는 이들이 발견하는 오류와 그에 대한 피드백은 프로그램의 코드 값을 수정하고 업데이트 하도록 한다. 이에 작가는 수정 과정을 포함한 데이터의 증식을 근육과 살의 성장에 빗대 건축프로그램과 그래픽 툴을 활용해 그 골격을 체화한다. <소프트웨어의 성장과 형태에 대해>(2021)는 그렇게 살을 얻으며 조형적 갱신을 거듭한다.

## 응전을 갱신하는 예술

소프트웨어를 뿌리부터 파헤쳐 뼈대를 구성하고 이를 다시 소프트웨어를 통해 살을 입힌 오브제는 그리드에 걸쳐 있을지라도 현실에서는 좌대와 작품의 전통적인 분리에 놓인다. 피스를 프레임에 고정하거나 좌대 위에 올려놓는 모습은, 조립식 모형 장난감의 러너(플라스틱 틀)처럼 비계와 좌대 위에 놓여 여전히 자율적이지 못한 위상을 갖는다. 이에 작가는 자동차 제조 과정에서 차체를 만드는 뼈대로 사용하는 ‘모노코크’를 참조한다. 작가는 알루미늄을 원료로 주물 제작하는 모노코크 공법을 통해 골조와 몸체를 하나의 구조로 제작한다. 기존 지지대와 조립 부품, 거치대와 사물의 위계를 두거나 구분하지 않고, 주변적인 것과 중심적인 기능을 하나의 덩어리로 조망한다. 여기에 작가는 이커머스 플랫폼 구조를 끌어와 둘의 유사성(Similarities)을 부각한다. 말하자면 기계장치를 보관하는 프레임과 차체를 통합시킨 모노코크의 모습이 개인 판매자와 중개 사업자, 배달노동자와 구매자의 생태계를 한데 모아낸 이커머스 플랫폼에 조응하는 것이다. 조형성의 비평적 갱신은 소프트웨어 프로그램뿐 아니라 그것에 기반한 산업과 외부 환경에 연결하며 공정의 맥락을 확장한다.

단일 오브제로서 개체를 단위로 삼은 작업은 이제 자연의 원소 단위를 향한다. 2023년 <모노코크: 정원의 원리>에서 그가 대상으로 삼는 것은 소프트웨어의 무덤이라 불리는 어밴던웨어(Abandonware)의 프로그램들이다. 작가는 생애주기를 지나 저작권이 소멸하고 폐기된 소프트웨어, 무덤에 버려진 소프트웨어를 조형으로 소생시킨다. 그는 운영체제에서 선별한 구름(Gooroom), 해(Sun), 붉은별(Red Star)의 형태를 금속성 물질에 새기고, 토끼(Hare), 폭포(Cascade), 덩굴식물(Creeper), 돌맹이(Stone) 이름의 바이러스를 취해 바다로부터 채취한 물질들을 거푸집 삼아 그 형태를 찍어낸다. 음각과 양각으로 산출된 형태들이 전시공간의 바닥과 벽면에 설치된 모습은 정원의 생태계에 비견하는 경관을 형성한다. 자연물의 이름을 기표 삼아 물질을 바이러스 코드에 매칭하는 방식은 다분히 주관적인 고대 원자론적 사고를 떠올리게 한다. 이는 다시금 소프트웨어 무덤에서 버려진 것들을 조형으로 소생시킴으로써 프로그램의 생태적 순환이라는 조형의 변칙적 화용론을 가능케 한다.

유전자학의 논리를 적용하여 소프트웨어의 몸을 생성하는 구자명의 작업은 SF를 방불케 한다. 소프트웨어에 유사 유기체의 위상을 부여하는 시도는, 유전자를 일종의 소프트웨어로 접근하도록 한다. 둘을 이접하기 위해서는 소프트웨어가 몸체를 가진 생명의 프로그램이라고 설정하는 동시에, 유전자에 기반 한 인간을 소프트웨어의 총합으로 간주하는 사고의 확장이 필요하다. 일련의 사고를 확장하면 재차 예술가의 주체성을 재고하기에 이른다. 단적으로 그는

소프트웨어와 유전자의 중개인이자 상이한 프로그램을 선별하고 교란하는 해커이고 번역자다. 유전자와 소프트웨어 프로그램을 설계도 삼아 조형적 DNA를 탐구하는 면모는 르네상스 이후 조명되었던 창조자로서의 자율적인 주체 상을 떠올리게 하지만, 여전히 환기해야 하는 것은 작가가 소프트웨어를 사용하며 느꼈던 무력감이다. 소프트웨어를 소재로 삼고, 조형성을 부여하기 위해 사용하는 것도 소프트웨어 프로그램이며, 예의 조형에 원소의 속성을 분류하여 의미를 매기는 작업의 원천 또한 프로그래밍 된 자연의 비유들임을 환기하면, 그가 소프트웨어 프로그램으로부터 자유로울 수는 없다.

프로그램을 전유하여 그 방법적 대상이자 도구로 활용하는 작가의 태도는 니콜라 부리오의 '포스트 프로덕션'체제의 예술가를 떠올리게 한다. 이른바 문화적 참조와 전유의 노동을 업으로 삼는 이는 정보의 과포화와 유통으로부터 자신의 논리를 디제잉 하듯 만들어 낸다고 볼 수 있다. 하지만 다양한 영역의 소재와 도구를 활용하는 동시대 예술가의 심층에 자리한 무력감을 고려하면, 주체적으로 방법론을 다변화 해온 예술가의 이면에 전도된 주객의 위치에서 주도권을 회복하고자 하는 고군분투를 조명할 수 있다. 이미 완결적인 소프트웨어 프로그램에 느꼈던 무력감은 프로그램을 해체하여 그 몸을 설계하고 시각화하는 모티프로 작동한 것은 아니었을까.

이를 염두에 두면 소프트웨어의 설계 도면을 DNA에 유비하여 조형의 문법으로 삼는 작업의 열개를 그저 형식 실험을 위한 도구로만 소환한다고 보는 것은 좁은 독법일 것이다. 오브제들은 소프트웨어를 상호 교차하여 구현한 산출물에 가깝지만, 실상은 다분히 프로그램 의존적인 주관성에 바탕하는 혼성적 살들이다. 소외당하고 의존적이며 주변적이고 부차적인 예술가의 위상을 인지하면서도 주체성을 발휘하여 상이한 프로그램을 선별하고 포낸다. 주요한 살들 또한 그를 소외시키는 프로그램을 통해 만들어 낸 결과인바, 프로그램에 종속되고 의존할 수밖에 없는 객체로 남을지라도 주체적 객체성을 행사하는 이로서의 역전사를 시도하기를 멈추지 않을 것이다.<sup>1)</sup>

## 조형의 난잡한 진화

지금까지 설명한 그의 방법론은 소프트웨어의 코드 값과 분자생물학의 유전자 구조에 이어 이커머스 플랫폼 산업과 고대 원소론으로 교차의 대상을 확장하며 조형의 진화를 거듭한다. 여

---

1) 변칙적인 배치와 이접, 번역에 대한 그의 실험이 분자생물학의 관점에서 바이러스의 역전사를 소환하는 것은 어색한 적용이 아니다. 가령 RNA는 DNA의 정보 값을 전사하며 외부로 나가 단백질과 합성하며 생명 현상이 유지되도록 한다. 그것이 HIV 바이러스의 경우에는 RNA가 세포 속으로 들어가 DNA에 역전사하며 증식하는 과정을 거친다. 수잔 손택의 통찰처럼, 파괴와 손상으로 점철하는 감염병의 은유는 바이러스가 인체에 침입하고 변형하며 파괴한다는 식의 군사적인 비유를 만연하게 만든다. 하지만 관점을 달리하면 인체는 언제나 열려있음을 생각할 수 있다. 몸은 바이러스 앞에 객체이자 피해자이고 희생자일 수만은 없다. 다시 말해 몸은 그저 바이러스에 무차별적으로 침투당한 것이 아니라 몸을 열어 맞이한다. 인류학자 서보경의 제안처럼, 몸은 바이러스에 일방적으로 침범당하고 파괴되기보다 서로 감응하면서 변형을 겪고 진화를 거듭하며 현재에 이르게 된다. 이를 참고하면서, 우리는 구자명의 작업이 DNA의 정보 값을 전사하면서 생명의 유지와 재생산에 기여하는 역할에 그쳤던 RNA의 역전사와 공명하고 있음을 음미할 수 있다. DNA와 RNA, 바이러스와 인체에 관한 자세한 이야기는 서보경, 『휘말린 날들-HIV, 감염 그리고 질병과 함께 미래 짓기』, 반비, 2024. 를 참고할 수 있다.

기에 최근 작가가 북한의 홈페이지들을 해킹하여 그 페이지마다 구조를 엮기 서열화하고 조형하는 시도는, 그의 조형적 실험에 외부의 요소들이 거듭 소환되고 있음을 확인할 수 있다. 국내에서는 공식적으로 접속이 금지되어 있지만, 이미 VPN 등을 통해 변칙적으로 접속이 가능한 홈페이지는 편법으로라도 들어가는 순간 국정원의 표적이 되기 쉽다. 대척에 있는 두 집단은 적대하지만 오히려 적대를 활용하며 체제를 단속하고 유지해 왔다. 근거리에서 서로가 필요한 만큼 이용하거나 서로에게 걸쳐 있는 관계는, 또한 그의 작업이 두는 상호 수탈과 협력의 관계를 포괄한다. 지독한 상호 적대적 의존관계에 작가는 어떤 골격과 살을 붙여갈 것인가. 미시공학적인 접근 아래 그가 산출하는 조형은 다분히 정치적인 맥락 또한 배경으로 삼기에 이른다.

하지만 코드 값을 단위로 삼는 그의 방법론이 프로그램의 기능과 활용을 비롯한 외부 맥락과 해석을 애초에 배제하고 있음을 안다면, 각각의 오브제가 보이는 시각적 속성을 앞서 참조한 프로그램의 내용과 쓰임에 직접적으로 연결하는 것은 자의적인 해석이 될 것이다. 철저하게 프로그램의 주관적이고 유동적인 의미와 쓰임을 배제한 코드 값으로 제작한 형상은 내용과 외양이 어떤 접점도 가질 수 없는 괴형태의 위상을 갖는바, 이 낯설고 무용한 에일리언은 프로그램들의 난잡한 유성생식에 가까운 조형성을 확보함에도 프로그램의 쓰임과 의미와는 관련성을 갖지 않는 회색지대의 위상을 갖는다. 적어도 확실한 것은 앞서 작가가 적용했던 고대 원소론을 연상케 하는 자연적 질료의 배치와 같은 자의적인 적용은 온전한 개연성을 줄 수 없다는 점이다. 정합적인 조형의 방법론이 배치의 자의적이고 우연적인 속성을 또한 품을 수밖에 없다면, 이 틈새에서 재차 탄생할 오브제들은 어떻게 다른 물성을 요구하거나 변칙적인 도약을 감행할 수 있을까.

여기에 다른 관점의 접근도 가능하다. 상이한 소프트웨어를 해체적으로 전유하는 작가의 방법론은 그 자체로 메타적인 소프트웨어라고 가정할 수 있다. 그렇다면 소프트웨어 프로그램으로서 작가의 방법론은 어떻게 자가 전유를 통해 비평적 조형을 획득할 수 있을까. 외려 이러한 가정은 상이한 소프트웨어들 간 번역에 의해 산출하는 조형의 난잡한 갱신이 아닌 무성생식의 자가 복제에 지나지 않을까. 여기에 조형의 문법을 탈구하는 변이나 우연성은 어떻게 개입할 수 있을까. 이는 예술가가 프로그램에 맞서 조형성을 부여한 재응전의 궤적에 있으면서도, 프로그램과 작가 사이의 관계를 초과한 난잡한 관계성의 변수가 되지는 않을까.