

개조

2장에서 이야기한 전통적인 인터랙션 디자인에서 사용자는 알려지지 않은 목적으로 제품을 사용할 수가 없었다. 대부분 잘 디자인된 제품은 정해진 특정 방법만으로 쓸 수 있다. 그러나 제품이 출시된 이후에 사용자들은 예기치 않은 목적과 방법으로 제품을 사용하기도 한다. 이것이 바로 개조 hacking(때에 따라 영문 뜻 그대로 해킹이라고 부를 수도 있다. - 옮김이다).

개조란 당초 계획된 기능 외의 방법으로 사용하도록 제품을 용도 변경하는 것을 말한다(그림 7.8). 라이터로 병을 따는 사람도 있고, 전화 시스템을 조작해서 무료 통화를 하려고 복잡한 시도를 하는 사람도 있다.

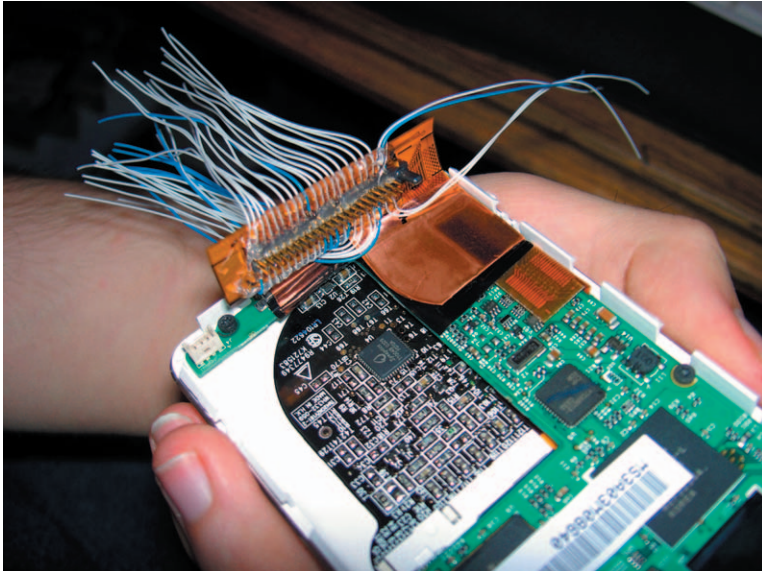


그림 7.8 콜린 앨런은 아이팟을 개조해서 외장 하드 드라이브로 연결했다.

개조가 꼭 파괴적일 필요는 없다. 때로 개조를 통해 개인이 제작한 전화 번호부를 휴대폰에 추가하는 등의 새로운 기능을 삽입하기도 한다. 때로 개조는 단순히 정치적이거나 사회적인 요소를 반영하기도 한다. 예를 들어, 몇 년 전 게릴라 예술 프로젝트에서 바비 인형과 지아이 조 G.I. Joe의 목소리 칩을 바꿔서, 지아이 조가 “쇼핑하고 싶어!”라고 말하거나 바비 인형이 “부대원, 전진!” 같은 명령을 내리게 만들기도 했다.

개조 혹은 해킹은 사용자가 제품을 ‘소유’하고 있든 아니든 간에, 난폭하게 제품을 변형해 자신의 의지에 맞춰 조형하는 개인화의 강한 표현이라고 생각할 수도 있다. OS권 정부 조직이든 심지어 아이들 인형이건 간에, 통제권이 없는 사람이 시스템을 통제하는 수단이 되기도 한다.

서비스 등 모든 제품은 개조 혹은 해킹될 수 있으니, 인터랙션 디자이너가 이를 장려해도 될까? 괜찮다면 어떤 방식으로 잘 하게 만들 수 있을까? 아직도 답을 찾지 못한 문제다.

비공식적으로나마 개조(해킹)가 장려되는 한 가지 이유는 개조(해킹)를 통해 제품의 새로운 사용법을 만들어낼 수 있고, 이 사용법이 공식적으로 디자인되고 상품화될 가능성이 있다는 것이다. 스프레드시트 프로그램에 사람들이 추가한 새 기능이 이 애플리케이션을 쓰는 모든 사용자에게 유용할 수도 있다. 할아버지가 보행기에 덧댄 찢어진 테니스 공이 앞으로 보행기가 개선될 방향을 제시하거나 혹은 심지어 완전히 새로운 제품으로 태어나게 할 수도 있다(그림 7.9).

MOBILITY BALLS

THE JOURNAL OF AMERICAN PHYSICAL THERAPY ASSOCIATION'S STUDY DETERMINED MOBILITY BALLS-TYPE TENNIS BALLS, PLACED ON THE TIPS OF WALKERS ARE THE #1 PREFERENCE OF USERS OF WALKERS.

Home Our Products Customer Care Order Form Contact Us

"I've got to say... your Mobility Balls were a welcomed addition to my walkers' appearance."

J. Titone
Broken Arrow, OK

ASSEMBLED & OWNED BY MEMBERS OF THE DISABLED COMMUNITY

Packaged and sold in the U.S.A. by the disabled community for the disabled community

DIGNIFY YOUR WALKER WITH MOBILITY BALLS

Welcome to Mobility Balls, the home for perhaps one of the most unique mobility walker accessories on the market today. Our **Mobility Balls** are specifically designed for physically challenged or disabled walking aids. Mobility Balls eliminate the constant scuffing of floors or the difficult awkward movements over rough surfaces caused by balance walking aids. We have created our Mobility Balls with the intention of making your disability aids as colorful and vivacious as you are. Traditional tennis balls on walker legs are unbelievably unattractive and boring. Replace those unsightly green and yellow tennis balls on your wheeled walking aids with one of our unique mobility balls. Mobility Balls are available in a variety of bright appealing colors and patterns; your walker or disability aids can be transformed into a fun appealing accessory.

- Replace those unsightly green and yellow tennis balls
- Give character to your walker
- Protect your floors against scuffing
- Enhance walking on rough surfaces
- Pre-cut for easy placement on walker legs
- Fun to have during the holidays

MORE INFO PURCHASE

"SHOES FOR YOUR WALKER"

그림 7.9 '모빌리티 볼'이라는 상품은 개조품(보행기에 댄 테니스 공)으로 만들어진 실제 제품이다.

당연하게도 파괴적인 개조(해킹)를 원하는 회사는 없다. 최소한 사생활과 공적인 관계에 대한 논란을 불러일으킬 수도 있다. 마텔 사가 바비 인형과 지아이 조 인형의 불법 개조를 불쾌하게 생각했을 거라는 건 쉽게 짐작할 수 있다. 일부 기업, 특히 자동차 회사에서는 사용자가 차량을 임의로 개조하면 보증 기간 내에도 보증 수리의 의무가 없으므로 이런 개조를 모른 체 하기도 하며, 혼다의 1990년대 시빅과 같이 튜닝하기 쉬운 차에 대한 수요가 생기기도 한다.

레이어

개조 가능한 디자인을 위한 한 가지 방법은 건축 이론가인 스투어트 브랜드가 창안하고 IBM 엔지니어인 톰 모란(그림 7.10)이 채택한 개념인 레이어를 쓰는 것이다. 상단 혹은 ‘빠른^{fast}’ 레이어는 빠르게 변화가 일어나는 곳이다. 하단 혹은 ‘느린^{slow}’ 레이어는 좀 더 안정적이며 기본적인이다. 제품의 상단 레이어가 조금 더 잘 보이도록 디자인된다면 해커들은 궁극적으로 안정성을 제공하고 가능한 변화를 받쳐줄 하단 레이어를 놓아둔 채 상단의 레이어를 가지고 놀 수 있다.



그림 7.10 스투어트 브랜드와 톰 모란의 연구에 기초한 다이어그램. 상단의 ‘빠른’ 레이어는 좀 더 빨리 변화하고, 하단의 ‘느린’ 레이어는 안정성을 제공한다. 상단 레이어에서 만들어진 변화는 시간이 지나면 아랫단으로 전파될 수 있다.

예를 들어 전형적으로 비^非 플래시나 플렉스 웹 애플리케이션들의 최상위 레이어(전통적으로 프레젠테이션 레이어, 또는 인터페이스라 불린다)는 대개 HTML, XML, CSS 등으로 구성된다. 그 아래의 레이어는 논리 레이어로, 많은 기능을 포함하며 종종 미들웨어라고 불린다. 좀 더 아랫쪽 레이어에는 데이터베이스, 데이터 소스, 서버 소프트웨어 등이 위치한다. 그보다 더 낮은 레이어들은 물리적 서버 등이 될 수 있다.

만약 이미 만들어진 애플리케이션의 레이어들 사이에 틈새가 존재한다면 (예를 들어 볼 수 있는 소스코드에 감춰진 명령어) ‘선수’ 들은 이 부분을 해킹해서 자신이 원하는 내용으로 바꿔칠 것이다. 이들은 별로 많은 것을 망가뜨리지 않고 상위 레이어(특히 CSS)를 해킹함으로써 애플리케이션을 자신이 원하는 대로 바꿀 수 있다. 하지만 어떤 회사도 해커가 그 데이터베이스나 서버에 손대기 원치 않으므로 그것들은 하위 레벨에 위치시킨다.

한편, 필요 이상의 레이어를 공개해 해킹할 여유를 주는 회사도 있다. 마이크로소프트와 애플은 모두 중간 레이어를 공개함으로써 써드 파티 개발자들이 OS 위에서 돌아가는 애플리케이션을 만들 수 있는 애플리케이션 프로그램 인터페이스 API를 제공한다. 구글은 구글 맵 API를 통해서 개발자와 디자이너, 해커 등이 지도를 만들고 새로운 애플리케이션을 만들 수 있는 플랫폼을 개발할 수 있도록 중간 레이어 사용을 장려한다.

참고 API란 용어는 프로그래머가 시스템과 상호작용하기 위해 미리 정의된 함수를 쓸 수 있게 하는 명령어와 함수, 프로토콜의 집합체다.

제품 개조와 적응성: 댄 힐 인터뷰



댄 힐은 인터랙티브 산업의 선두주자이고 BBC 라디오 & 뮤직 인터랙티브의 디자이너다. 그는 BBC 라디오와 웹, 디지털 TV, 모바일 플랫폼에서 음악에 기반을 둔 서비스를 제공하는 일을 맡았다.

제품이 개조되도록 만들 수 있을까요? 아니면 모든 제품이 개조할 수 있는 것일까요?

사실 모든 제품은 개조할 수 있습니다. 사용자가 수정할 수 있는 제품을 개조 가능한 제품이라고 정의한다면, 거의 모든 제품이 개조되고 있음을 알 수 있습니다.

하지만 이 정의는 너무 광범위해서 유용하지 않고, 개조되는 제품의 범위를 크기, 유연성, 복잡성을 기준으로 제한을 두는 편이 적절할 것입니다. 자동차는 튜닝이라는 형태로 활발히 개조되어 왔고 빌딩은 처음부터 끝없이 리모델링되고 커스터마이징됩니다. 옷도 종종 수선해 입고, 악기도 자신에 맞게 고쳐서 연주합니다. 심지어 디터 람스가 제작한 브라운 메이커의 하이파이 시스템 같은 걸보기엔 분명히 완벽한 현대식 제품도 엄청나게 많이 수정됩니다. 애플의 아이팟조차도 외관이 밀봉된 채로 출시되지만 이내 해킹됩니다.

좀 더 개조가 잘 되는 디자인을 하려면 어떻게 해야 할까요?

내게 있어 이 질문은 전통적 디자인 관념을 뒤집는 것으로 들리는군요. “잘 디자인되어 있다면 눈에 띄지 않는다”라는 금언은 제품 개조에 대한 생각에는 별로 유용하지 않습니다. 만약 사용자를 초대할 거라면 그들이 발견할 수 있도록 제품에 이런저런 틈새와 흔적, 디자인 과정에 대한 힌트 등을 남겨줘야 합니다. “좋은 디자인이란 디자이너의 흔적을 남기지 않는 것이다”라는 나오토 후카사와의 생각은 디자이너의 겸손함을 되돌아보게 하는 좋은 금언이지만 사용자에게는 디자인의 흔적이 남아있는 쪽이 매우 유용할 것입니다.

매튜 찰머스는 제품의 틈새를 남겨둬으로써 어떻게 제품을 바꾸고 적응할지 알려줄 수 있다고 제안합니다. 이것은 미리 그 사용방법을 제안하는 어포던스를 넘어섭니다. 이런 생각은 제품의 짜임새가 어떻게 각 구성 요소들과 긴밀하게 커뮤니케이션하는지, 또한 ‘보이지 않는 컴퓨팅 환경(invisible computing)이나 UI 자체가 사라지는 날이 와야 한다는 사용자 중심 디자인의 기존 사상에 반해 제품을 만들어야 하는지 등에 대해 생각하게 합니다. 찰머스의 ‘(아름다운) 틈새가 가득한 시스템’에 대한 주장은 임의 개조를 가능하게 한다는 컨텍스트로 참고할 만합니다.

개조 가능한 제품을 만드는 데는 다음 기법들이 필요할 것입니다.

- 어포던스와 틈새가 명확하고 순응성이 있어야 한다.
- 특정 레이어의 조사, 복제, 조작이 가능하게 한다.
- 순응성, 객체 지향적 본질, 코드의 사회적 행동으로부터 배운다.
- 행동/사용의 관점에서 자신을 스스로 감지하고 이를 다시 사용자에게 되돌리는 제품을 만든다.
- 디자인 연구뿐 아니라 이런 행동으로부터 제품이 나오도록 한다.
- 제품을 둘러싼 사회적 인터랙션이 가능하게 한다.

적응하는 디자인의 가치는 무엇인가요?

개조에 대한 용어는 종종 ‘훅’, ‘소켓’, ‘플러그’, ‘핸들’ 등의 단어로 범벅이 됩니다. 적응하는 디자인에서는 코딩보다는 건축 쪽의 용어를 차용해 ‘빠른 레이어의 변화 가능성’, ‘느린 레이어의 안정성 구축’, ‘점진적 변화를 위한 공간 설계’, ‘시간이 최고의 디자이너다’ 등의 좀 더 우아하고 정제된 어휘를 씁니다. 따라서 적응하는 디자인이란 더 조심스럽게 분절과 진화를 디자인하는 과정이라는 점에서, 자유분방한 개조와 차별화됩니다.

그러나 어쨌든 이 둘의 기본 개념은 비슷합니다. 디자인이란 디자이너와 사용자 사이에서 진행되는 사회적 과정이라는 것입니다. 제품을 시간에 걸쳐 점진적으로 개량하고, 느슨하게 결합된 레이어 구조를 통해 시스템이 학습할 수 있게 합니다.

적응하는 디자인에서는 사용자의 경험 혹은 오브젝트가 학습할 수 있어야 하며, 사용자는 이런 경험/오브젝트를 가르칠 수 있어야 합니다. 그러므로 양방향 인터랙션이며, 이것을 통해 사용자는 자신에게 유용하도록 제품을 개조하길 원합니다. 따라서 디자이너는 그들에게 직접적으로 유용한 경험을 제공하기보다는 유용한 경험을 달성하는 데 집중하기 위한 적응성에 집중해야 합니다. 디자이너는 제어보다는 가능성을 열어두는 데 집중해야 합니다.

‘디자이너가 아닌 자들의 손에 창조성’을 부과하는 과제가 떨어졌습니다. 어떻게 인터랙션 디자이너가 이 과제를 달성할 건가요?

우선, 적응하는 제품을 만들기 위해 인터랙션 디자이너는 여러 분야에 걸친 환경에서 작업하고 소프트웨어 개발자나 다른 전문가들과 밀접하게 의사소통할 필요가 있습니다. 다시 말하면 코드를 정말로 이해해야 한다는 말입니다. 그렇다고 실제 코딩을 할 필요까지는 없습니다.