



[장려상] 소프트웨어 특허의 진화 - 유저 인터페이스(UI)의 특허법적 보호

저자 (Authors)	이승현
출처 (Source)	IT와 법 연구 7 , 2013.2, 331-354 (24 pages) IT & LAW REVIEW 7 , 2013.2, 331-354 (24 pages)
발행처 (Publisher)	경북대학교 IT와 법 연구소 IT & Law Research Institute
URL	http://www.dbpia.co.kr/Article/NODE02152551
APA Style	이승현 (2013). [장려상] 소프트웨어 특허의 진화. IT와 법 연구, 7, 331-354.
이용정보 (Accessed)	고려대학교 163.152.3.*** 2018/12/13 11:14 (KST)

저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

소프트웨어 특허의 진화
: 유저 인터페이스(UI)의 특허법적 보호

이승현*

《 목 차 》

I. 서론	한 인식
1. 연구의 배경 및 목적	3. 유저인터페이스 기술의 기능성
2. 그래픽 유저 인터페이스의 정의	4. UI개발
II. 소프트웨어와 특허	IV. 소프트웨어의 보호
1. 소프트웨어의 정의	1. 소프트웨어 지적재산권을 보호하기 위한 노력의 필요성
2. 소프트웨어에 대한 특허의 허여	2. 소프트웨어 특허권의 그늘
III. 그래픽유저인터페이스와 특허	3. 소프트웨어 특허의 진화
1. GUI의 역사	V. 결론
2. 우리 특허법에서의 GUI에 대	

I. 서론

1. 연구의 배경 및 목적

2010년도에 시작된 글로벌 기업들 사이의 디지털 특허분쟁으로 이 소송의 대상이었던 그래픽 유저 인터페이스(GUI)와 같은 소프트웨어 특허에 대한 관심이 사뭇 높아졌다. 특히 삼성과 애플의 특허소송은 기존의 하드웨어 기술특허 위주로 진행된 소송과는 다르게 유저 인터페이스 특허와 다

* 경북대학교 법학전문대학원 3기

자인보호에 관한 소송이 포함되어 지식재산권 전반의 내용을 다루는 소송으로 진행되고 있어 관련업종의 이목을 끌고 있다.

유저 인터페이스 특허는 기본적으로 소프트웨어에 대한 특허이면서 동시에 회화적 성격을 가지고 있어서 디자인보호법에 의한 보호도 받을 수 있는 독특한 성격을 가지고 있다. 과거에는 소프트웨어는 수학적이며 추상적 아이디어라는 이유로 특허의 대상이 될 수 없지만 컴퓨터프로그램 기술의 발전으로 인해서 미국을 중심으로 소프트웨어에 대한 특허의 허용범위가 점차 넓어지고 있는 상황이다. 하지만 우리나라 특허법에 의하면 소프트웨어는 자연법칙을 이용한 것이 아니라는 이유로 발명이 될 수 없고 그에 따라 산업상 이용가능성이 없으므로 특허의 대상이 되지 못한다. 유저 인터페이스의 특허성을 판단하기 위해, 먼저 소프트웨어가 발명의 대상이 될 수 있는가를 판단해 보고 유저 인터페이스가 기술적 아이디어로 평가될 때 어떠한 요소가 고려 될 수 있는지에 대해 알아보겠다.

2. 그래픽 유저 인터페이스의 정의

GUI는 Graphical User Interface의 약어로 컴퓨터를 사용하면서, 그림으로 된 화면 위의 물체나 틀, 색상과 같은 그래픽 요소들을 어떠한 기능과 용도를 나타내기 위해 고안된 사용자를 위한 컴퓨터 인터페이스이다.¹⁾ GUI가 이용되기 전의 컴퓨터 이용자는 키보드를 통한 문자입력 방식을 토대로 작업을 해왔지만 직관적이지 못한 사용자 환경으로 인해서 텍스트 명령어 시스템에 익숙하지 못한 사용자는 운영체제의 명령어를 익힐 때까지 컴퓨터사용법에 적응하기 어려웠다. 하지만 GUI가 컴퓨터 운영체제에 점차 대중적으로 활용되어 지게 되자, 컴퓨터를 처음 사용하는 초보자도 단순히 마우스의 컨트롤만으로 쉽게 컴퓨터를 사용하는 시대가 열리게 됐다.

최근에는 스마트폰의 발명으로 손가락을 이용한 터치 기반의 인터페이스가 대중화되어 사용되고 있다. 하지만 글로벌 특허소송에서 많은 사람들이

1) 그래픽 사용자 인터페이스, ko.wikipedia.org/wiki/GUI, 2012. 9. 1.

사용자 인터페이스에 대해 특허를 부여하는 것에 대해 부정적 의견을 제시하고 있다는 점에서 인터페이스 기술의 중요성에 대한 인식이 부족함을 엿볼 수 있다.

II. 소프트웨어와 특허

1. 소프트웨어의 정의

"소프트웨어"라 함은 컴퓨터·통신·자동화 등의 장비와 그 주변장치에 대하여 명령·제어·입력·처리·저장·출력·상호작용이 가능하도록 하게 하는 지시·명령(음성이나 영상정보 등을 포함한다)의 집합과 이를 작성하기 위하여 사용된 기술서 기타 관련 자료를 말한다.²⁾ 그래픽 유저 인터페이스의 경우는 컴퓨터 장비와 그 주변장치에 대하여 입력과 출력 그리고 상호작용을 가능하게 하는 지시의 집합이라 볼 수 있기 때문에 소프트웨어에 속하게 된다.

2. 소프트웨어에 대한 특허의 허여

1) 이론적 논의

특허법 제2조 제1호는 발명을 '자연법칙을 이용한 기술적 사상의 창작으로서 고도한 것'으로 정의하고 있다. 그래서 순수한 컴퓨터 프로그램의 경우, 자연법칙을 이용한 것이 아니기 때문에 발명이 아니기 때문에 특허법 제29조 1항의 '산업상 이용 가능성'을 충족하지 못하기 때문에 특허의 대상이 되지 못한다.³⁾ 그러나 특허청 심사지침에 의하면 컴퓨터 프로그램이 컴

2) 소프트웨어산업진흥법 제2조 제1호

3) 대법원 1998. 9. 4. 선고 98후744 판결

퓨터를 실행하는 명령에 불과할 경우에는 프로그램 자체는 발명이 될 수 없지만 컴퓨터 프로그램에 의한 정보처리가 하드웨어를 이용해 구체적으로 실현되는 경우에는 해당 프로그램과 연동해 동작하는 정보처리장치(기계), 그 동작 방법 및 해당 프로그램을 기록한 컴퓨터로 읽을 수 있는 매체는 자연법칙을 이용한 기술적 사상의 창작으로 발명에 해당한다.⁴⁾고 판단하고 있어 컴퓨터프로그램이 하드웨어를 이용해 구체적으로 실현되는 경우에는 프로그램이 연동해 동작하는 기계는 특허의 대상이 되고 있다.

2) 소프트웨어의 특허성에 관한 학설대립⁵⁾

(1) 부정설

부정설은 과거에 컴퓨터 소프트웨어가 그랬듯이 소프트웨어는 일종의 계산방법에 지나지 않고 소프트웨어는 정신적, 지능적 수단이나 과정과 동일한 것이므로 자연법칙을 이용한 것이 아니라고 하는 설이다.

(2) 긍정설

긍정설은 웨어하우스(warehouse)설이라고 한다. 프로그램이 설치되기 전의 하드웨어는 단순한 창고에 불과하지만 프로그램이 설치되면 프로그램과 하드웨어는 물리적, 기능적으로 일체가 되어 구체적 장치와 동일시 할 수 있게 되므로 자연법칙을 이용한 것이라고 하는 설이다.

(3) 절충설

컴퓨터 소프트웨어 중에서도 특허성을 인정 할 수 있는 종류가 있고 그

4) 특허청, 『컴퓨터 관련 발명 심사기준 해설』, 특허청, 2007, p.63

5) 최재원, “소프트웨어 특허 범위 확대되어야 한다.”, 월간 마이크로소프트웨어, 1997년 8월

렇지 않은 종류가 있다는 설이다. 순수한 컴퓨터 프로그램은 특허성이 없다고 보지만 기술적 과정에서 기계를 제어하는 방법처럼 제어방법에 대해서 기술적 특성을 가지고 있다면 특허성이 있는 것으로 한다.

3) 각국의 소프트웨어 특허 허여⁶⁾

(1) 유럽의 경우

2005년에 유럽에서는 소프트웨어 특허법안이 유럽의회에서 압도적인 차이로 부결됐다. 회원국 간의 결렬한 찬반 논쟁이 있었지만 미국의 특정 업체들에 의한 특허권의 독점이 일어날 것이란 의견으로 인한 기각이었다. 그래서 유럽 특허협약은 컴퓨터 프로그램을 특허대상에서 명시적으로 제외하고 있다. 하지만 실무에서는 컴퓨터 프로그램이 소프트웨어와 하드웨어 사이에서 일어나는 통상 범위내의 물리적 상호작용을 넘는 기술적 특성을 가진다면 특허성이 있다고 판단한다. 유럽특허조약과 유럽특허청의 실무심사에는 소프트웨어의 특허성 여부에 차이가 존재하는 것이다. 따라서 유럽에서 컴퓨터프로그램을 청구하는 것 자체가 거절이유가 되지 않는다.

(2) 일본의 경우

일본의 컴퓨터 프로그램 심사운영 지침의 자연법칙을 이용하고 있다고 할 수 있는 예로는,⁷⁾ 하드웨어 자원에 대한 제어 또는 제어에 따른 처리, 대상의 물리적 성질 또는 기술적 성질에 의한 정보처리, 하드웨어 자원을 이용해서 처리하는 것을 예로 들고 있다. 이와 반대로 자연법칙을 이용한 것이라 볼 수 없는 예로는 해결수단이 수학적 해법 그자체인 경우 또는 그 수단이 인문과학인 경우를 들고 있다. 과제 해결 수단으로 자연법칙을 이용하더라도 컴퓨터만을 이용한 처리, 기록매체에 프로그램을 저장하는 것

6) 조영선, 『특허법』, 박영사, 2011, p.14

7) 김원학, 「SW의 효과적 보호방안」, IT와 법연구(1), 2009, pp.9~15.

만으로는 발명으로 인정하고 있지 않다. 특히 일본의 경우에는 법개정을 통해서 컴퓨터프로그램을 물건으로 간주하는 규정을 특허법에 명문화하여 컴퓨터프로그램을 보호하고 있다.

(3) 미국의 경우

1972년 미국 연방대법원은 *Gottschalk v. Benson* 판결에서 2진화 10진법을 단순히 2진법으로 변환하는 프로그램은 추상적 아이디어인 수학에 불과하므로 특허의 대상이 될 수 없다고 판시했다. 그 후 1981년 *Diamond v. Diehr* 판결에서는 출원자가 고무의 경화 시점을 수학을 이용한 계산을 통해서 찾아내서 처리하는 방법을 특허로 낸 것이다. 처음에는 이 청구항에 수학적 알고리즘(아레니우스 방정식)이 포함되었다는 판단에 의해서 거절되었다. 하지만 단순히 수학적 알고리즘이 포함되어 있다고 해서 전체발명이 특허성이 없다고 할 수 없으며 고무를 경화하는 알고리즘이 이용되어 실제적인 효과를 나타내었으므로 특허의 대상이 될 수 있다고 하였다. 1999년에는 이보다 더 나아가 *AT&T v. Excel Communications, Inc.* 사건에서 *Diamond v. Diehr* 사건에서와 같이 물리적 조건이 필수적인 요건이 아니며, 유용하고 구체적이며 유형의 결과를 만들어 낼 수 있다면 컴퓨터 프로그램 그 자체도 특허의 대상이 될 수 있다고 하였다. 이 후 미국 특허청은 대부분의 컴퓨터 프로그램이 위 조건을 충족하였기 때문에 컴퓨터 프로그램에 대하여 특허를 인정해 오고 있다.⁸⁾ 그러므로 미국에서도 추상적인 수학기공식을 포함하더라도 전체적으로 볼 때 수학을 이용하여 특허법이 보호하는 기능을 한다면 특허의 대상으로 된다.

(4) 우리나라의 경우

우리나라 특허법 제2조 1항에 의하면 발명이란 ‘자연법칙을 이용한 기술

8) Roger E. Schechter 외 1인 공저, *Intellectual Property the law of Copyrights, Patents and Trademarks*, Tohmson West(2003), 360p.

적 사상의 창작으로서 고도의 것'을 말한다. 소프트웨어 발명의 경우에는 프로그래밍 언어로 작성되어 하드웨어에서 작동되는데 이 때 '자연법칙을 이용'하였는가가 문제된다. 우리나라의 특허청에서는 컴퓨터 프로그램은 하드웨어를 실행하기 위한 명령에 불과한 것으로 판단하기 때문에 순수한 컴퓨터 프로그램 자체의 특허성은 부정하고 있다. 하지만 기계장치를 제어하는 컴퓨터 프로그램의 경우처럼 소프트웨어가 하드웨어와 결합하여 구체적인 수단으로 실현되는 경우에는 컴퓨터 프로그램에 발명으로서의 성립성을 인정하고 있다. 그래서 특정 하드웨어를 위해서 작성된 컴퓨터 프로그램은 발명으로 인정된다. 그런데 컴퓨터 프로그램이 범용컴퓨터에서 실행되는 경우에는 '정보처리가 하드웨어를 이용해 구체적으로 실현되는'이라는 요건을 충족하지 못한다고 판단하고 있기도 하다. 그런데 실제로 범용컴퓨터와 특정 목적을 위한 컴퓨터는 공학적으로 동일하여 구분할 방법도 없기 때문에 특허 허용에 대한 명확한 이론적 판단 기준을 제시하고 있지 못하다.

대법원 판결⁹⁾에 의하면 '수치제어입력포맷을 사용하여 소프트웨어인 서브워드 부가 가공프로그램을 구동시켜 하드웨어인 수치제어장치에 의하여 기계식별·제어·작동을 하게 하는 것일 뿐만 아니라, 하드웨어 외부에서의 물리적 변환을 야기하는 것이며 그 물리적 변환으로 인한 실제적 이용가능성이 명세서의 개시되어 있는 이상, 이를 자연법칙을 이용하지 않은 순수한 정신적 활동에 불과하다고 볼 수는 없다'¹⁰⁾고 판시하고 있다. 이 판결은 컴퓨터 프로그램을 하드웨어와 연결하여 외부에서의 물리적 변환을 일으키며 그 변환이 실제적 이용가능성이 있는 경우에는 특허의 대상이 된다는 것이다. 따라서 컴퓨터 프로그램이 외부에서의 실제적 이용가능성이 있는 물리적 변환을 야기하는 경우에는 특허의 대상이 된다는 것은 명확하다. 그런데 외부에서의 물리적 변환을 야기하는 경우에는 특허의 대상이 된다는 판결의 해석이 외부에서의 물리적 변환을 일으키지 않으면 특허의 대상이 되지 않는다는 명제를 도출하지는 못하므로 반드시 외부에서의 물리적 변환을 필요로 한다고 해석할 수는 없다.¹¹⁾ 그러므로 외부에 물리적 변환

9) 대법원 2001. 11. 30. 선고 97후2507 판결

10) 조영선, 『특허법』, 박영사, 2011, pp.10~16

을 일으키지 않는 컴퓨터 프로그램에 특허를 부여하더라도 기존 판례에 위배되는 것은 아니다.

4) 컴퓨터 프로그램에 특허 인정을 위한 조문해석

특허법 2조에서 말하는 발명의 정의¹²⁾는 독일의 법학자 코올러(Kohler)의 발명의 정의를 수용하여 규정한 것이다. 이러한 정의는 19세기의 기계 기술을 중심으로 한 개념으로 눈에 보이는 기계 산업사회와는 달리 무형의 재산이 강조되므로 정보기술사회에 과거의 개념을 그대로 적용하는 것은 부적절하다. 정보기술 사회에 적절한 지식재산권 보호를 위해서는 기존의 발명에 대한 정의를 개정하여 구체적이고 명백하게 보호하는 것이 효과적이다. 그러나 개정이 이루어지지 못한 경우에도 컴퓨터프로그램의 경우에는 하드웨어인 마이크로프로세서 안에서 작용하는 컴퓨터프로그램의 물리적 상태를 해석함으로써 기존의 발명의 정의를 개정하지 않고도 컴퓨터프로그램이 특허의 대상인 발명이 될 수 있는 방법이 있다.

대부분의 특정한 목적을 위해서 작성된 프로그램은 하드웨어에서의 작동을 전제하고 있다. 작성된 컴퓨터프로그램이 하드웨어에서 작동하게 되면, 마이크로프로세서 상에서 전기적 신호를 주고받게 되고, 전기적 신호는 주변장치를 제어하는 역할을 한다. 이때 주변장치를 제어한다는 측면은 물리적 장치에 변환을 일으킨 것이고, 이러한 프로그램은 자연법칙을 이용한 것으로 볼 수 있다.¹³⁾

거시적으로 봤을 때는 소프트웨어가 하드웨어와 구분된 추상적 개념으로 보이지만 실제로 마이크로칩 내부에서는 프로그램이 전기적 신호의 흐름이라는 물질적 속성을 가지게 된다.

물론 이 해석방법에서도 컴퓨터프로그램이 프로그래밍용 언어, 알고리즘

11) 명제 $A \rightarrow B$ 라고 해서 역인 $B \rightarrow A$ 가 반드시 참으로 성립하는 것은 아니다.

12) 자연법칙을 이용한 기술적 사상의 창작으로서 고도한 것

13) 김관식, 「컴퓨터프로그램의 특허법적 보호」, 정보화정책 제13권 제4호, 2005, p.112

과 같은 수학적 해법 그리고 통신수단인 프로토콜인 경우에는 추상적 아이디어이기 때문에 특허의 대상이 되지 못한다.

5) 하드웨어와 소프트웨어의 경계가 사라졌다.

컴퓨터 프로그램이 약전기의 신호로서 마이크로프로세서 안에서 작동하게 되면 프로그램이 없을 때에는 다양하게 활용될 수 있는 가능성만이 존재하는 프로세서에서 특정한 목적에 맞는 구체적인 배선을 가진 전기 회로와 동일한 방식으로 작동하게 된다. 이것은 특정 목적을 위해 전선으로 이어진 전기장치와 물리적으로 동일하다. 즉 물리적 차원에서 전기회선장치와 마이크로프로세서에서 작동하는 전자적 프로그램은 크기만 차이가 날 뿐 본질을 다르게 볼 수 없으므로 법 해석의 차원에서 컴퓨터프로그램에 특허인정을 하지 못할 이유는 없다.

예를 들면, 컴퓨터 기술이 발전하기 전의 작곡가들은 물리적 피아노를 사용하여 창작하고 악보에 음표를 그리면서 곡을 창작했다. 그런데 전자기가 발달하게 되자, 겉모습은 나무로 제작된 피아노와 유사하지만 속은 전자장비로 설계된 전자건반악기가 만들어졌다. 현재에는 개인용 컴퓨터가 발달하게 되자 클래식피아노나 신디사이저 같은 과거의 악기들이 VSTi¹⁴⁾라는 이름으로 소프트웨어로 변화했다. 주목할 점은 디스플레이상에서 가상의 피아노를 마우스나 키보드를 이용해서 연주를 하면 물리적 피아노와 동일한 소리가 스피커를 통해 흘러나오게 된다는 것이다. 소프트웨어 악기를 컴퓨터에 설치하게 되면 작곡가는 그 소리를 울리게 하기 위해 클래식 피아노나 신디사이저를 구입할 필요가 없다. 다시 말해서 소프트웨어 악기는 전자악기와 완전히 동일하게 구체적이며 유용한 결과를 만들어낸다. 이와 같은 상황에서 나무로 만들어진 피아노나 전기적 장치로 이루어진 신디사이저는 특허의 대상이 되는데 컴퓨터프로그램으로 작성된 가상악기가 특허의 대상이 되지 못한다면 이용자를 더 편리하게 해주는 장치에는 특허가

14) Virtual Studio Technology instrument

부여되지 못하고 반대의 경우에는 특허의 대상이 되기에 불합리한 상황이 생기게 된다.

이와 같은 논리는 모뎀과 같은 기술특허에도 동일하게 적용된다. 모뎀은 컴퓨터와 단말기와의 사이를 전화 회선을 이용하여 데이터 교환을 할 때 그 데이터 신호를 변조하는 장치이다. 과거의 모뎀은 전자회로로 설계되어 하드웨어적으로 설계되어 작동했지만 현재에는 컴퓨터 자체의 처리능력 향상으로 하드웨어적 모뎀이 소프트웨어로 구현되어 컴퓨터의 중앙처리장치(CPU)에서 데이터 송수신에 필요한 정보를 처리한다. 모뎀이 소프트웨어로 구동되게 되면 시스템 변경과 같은 업그레이드를 손쉽게 할 수 있어 간편하고 상대적으로 저렴하기까지 해서 훨씬 효율적이다. 만약 소프트웨어 모뎀이 특허의 대상이 되지 못한다면 발명가들은 필요하지 않더라도 하드웨어적 요소를 추가하여 특허를 받으려 할 것이고 이것은 기술의 발전을 촉진하여 산업발전에 이바지함을 목적으로 하는 특허법¹⁵⁾의 취지에 반하는 것이 명백하다. 현재와 같은 상황에서 소프트웨어가 특허의 대상이 되지 말아야 한다는 발언은 하드웨어 기술에도 특허를 주지 말자는 주장과 다르지 않다.

III. 그래픽유저인터페이스와 특허

1. GUI의 역사

그래픽 유저 인터페이스(GUI)는 1972년 앨런 케이가 제록스사의 팔로 알토 연구센터에서 GUI에 대한 개념과 기술 구현에 대해 연구하면서 시작됐다. 앨런 케이는 맥루한의 미디어 이론과 제롬 부루너의 인지이론을 연구하며 컴퓨터 전문가가 아닌 어린이가 컴퓨터와 경험을 공유할 수 있도록

15) 특허법 제1조 「이 법은 발명을 보호·장려하고 그 이용을 도모함으로써 기술의 발전을 촉진하여 산업발전에 이바지함을 목적으로 한다.」

GUI를 제안했다. 최초의 GUI는 더글라스 엥겔바흐가 개발한 마우스를 사용하여 문자명령 대신에 아이콘과 윈도우를 조작하는 시각적 방식으로 컴퓨터를 쉽게 조작하도록 도와줬다. 앨런 케이가 개발한 제록스사의 운영체제는 성공을 거두지 못했지만 애플사의 매킨토시 컴퓨터와 마이크로소프트가 개발한 운영체제인 윈도우 시리즈가 성공하면서 대중에게 널리 활용되었다.¹⁶⁾ 하지만 기존의 GUI가 일부 복잡한 유틸리티에서는 명령단추가 지나치게 늘어나게 되었고 인터페이스는 다양한 변화를 모색하게 되었다. 또 스마트폰 기술의 발달과 함께 트랙 휠과 가속장치, 멀티터치 스크린이 가능해짐에 따라 다양한 입력 값들이 추가되어 GUI에 변화를 가져왔다.

2. 우리 특허법에서의 GUI에 대한 인식

특허청의 컴퓨터프로그램 심사지침에 의하면 ‘GUI(Graphical User Interface)에 의해 입력이 용이화 하는 것도 기존의 명령어 방식의 입력 수단을 보다 편리하게 하기 위해 GUI 방식에 의한 입력 수단이 개발되었고 이를 해결과제로 제시하는 것은 당업자 수준에서 통상의 창작 능력에 불과한 것으로 판단한다.¹⁷⁾’라고 한다. 이렇게 GUI에 의한 입력을 용이화 하는 것이 통상의 창작능력에 불과한 것으로 보게 된다면 GUI는 저작권에 의해서 보호될 수밖에 없는데 저작권은 아이디어가 아닌 표현만을 보호하므로 그 보호의 범위가 좁을 수밖에 없고, 침해소송에서도 표현이 우연이 아닌 고의로 침해되었다는 증명은 하기가 쉽지 않다. 무엇보다 GUI의 경우에는 디자인적 창작성 외에도 입력방식의 문제를 해결하는 아이디어도 포함한다는 측면에서 저작권에 의한 보호만으로는 부족하다.

3. 유저인터페이스 기술의 기능성

16) http://en.wikipedia.org/wiki/Graphical_user_interface

17) 특허청, 『컴퓨터 관련 발명 심사기준 해설』, 특허청, 2007, p.105

1) 인터페이스 기술적 판단기준

최근의 스마트폰과 태블릿 컴퓨터는 물리적 입력장치가 거의 없는 상태에서 디스플레이에 터치기반의 메인입력장치가 접합되어 하나로 이루어져 있기 때문에 소프트웨어 인터페이스의 역할이 중요하다. 멀티터치 기반의 소프트웨어 인터페이스는 기존의 마우스 기반의 GUI에 익숙해져 있는 사람들에게는 작은 화면에서 아이콘 버튼이 없는 다소 생소한 입력방식이므로 처음 이용하는 사람이라도 직관적으로 자연스럽게 인터페이스를 작동하도록 유도하여야 한다. 또 멀티터치 기반의 인터페이스는 물리적 인터페이스에 비해 입력의 확실성이 부족하므로 신뢰성 높은 인터페이스 설계가 요구된다. 뛰어난 인터페이스 디자인으로 유명한 애플의 경우에는 직관적인 제스처 기반의 인터페이스 디자인으로 순식간에 스마트폰 세계시장을 석권하였다.

인터페이스의 기술성에 대해 알아보기 위해 글로벌 스마트폰 제조기업간의 특허소송으로 대중적으로 알려진 slice to unlock, bouncing back, Editing UI 에 대해 분석해 보며 소프트웨어 인터페이스의 기술적 판단요건에 대해 알아보겠다.

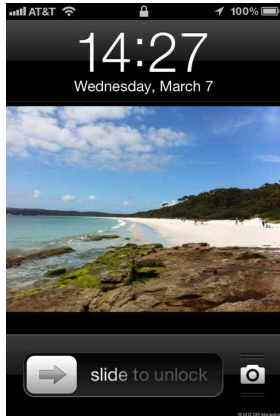
(1) slice to unlock¹⁸⁾ UI

‘밀어서 잠금 해제’라는 이름으로 불리는 이 UI는 iOS¹⁹⁾상에서 잠금 상태를 해제할 때 사용된다. 밀어서 잠금 해제 UI는 국내외를 막론하고 애플과 안드로이드 진영에 대한 소송에서 각종 언론과 대중에게 가장 많은 비판을 받았던 UI이기도 하다.

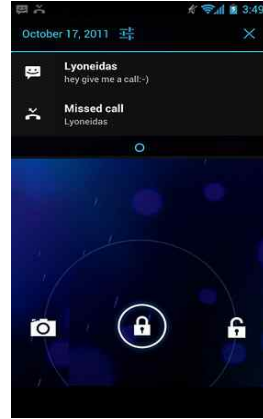
밀어서 잠금해제 UI의 디자인을 자세히 관찰해 보면 다음과 같은 구성으로 이루어져 있음을 알 수 있다.

18) U.S. Patent No. 5,946,647

19) iPhone, iPad에서 사용되는 Apple이 개발한 운영체제이다.



iPhone 의 '밀어서 잠금해제' UI



google 의 Android 잠금해제 UI

'밀어서 잠금해제'라는 글자가 좌측에서 우측으로 갈수록 밝기가 변하며 주목을 끄는 요소, '밀어서 잠금해제'라는 글자의 하이라이트 부위가 왼쪽에서 오른쪽으로 움직이고 있어 밝은 화살표 아이콘을 오른쪽으로 밀어야 한다는 것을 암시하는 요소, 화살표의 아이콘이 움직이는 가이드라인이 존재하고, 가이드라인은 명암을 오목하게 처리하여 아이콘이 움직일 수 있는 길임을 암시하는 요소, 어두운 색의 가이드라인에서 좌측에는 이와는 대비되게 밝은 색인 아이콘 안에는 화살표가 그려져 있어서 오른쪽으로 아이콘을 밀어야 한다는 것을 인식할 수 있는 요소, 화살표 아이콘의 라운드값²⁰⁾과 가이드라인의 라운드값이 유사하여 모서리 끝까지 밀어야 함을 암시하는 요소, 아이콘을 터치하여 오른쪽으로 조금이라도 움직이게 되면 그 움직이는 정도에 따라 '밀어서 잠금해제'라는 글자가 희미해지며 조금씩 사라지는 애니메이션을 디스플레이하여 밀어서 잠금해제라는 명령을 올바르게 수행하고 있음을 인식할 수 있는 요소, 주머니 속에서나 의도하지 않게 아이콘을 미는 경우를 예방하기 위해서 화살표 아이콘이 가이드의 끝부분에 닿기 전에 화살표 아이콘으로부터 손가락이 일정거리 이상 멀어지게 되면 아이콘이 원래의 자리로 복귀하는 요소, 복귀 할 때의 움직임은 마치 용수

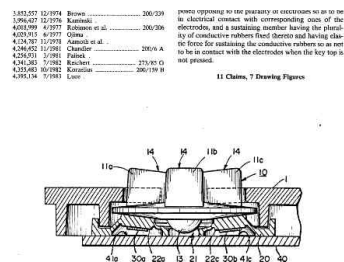
20) 모서리의 둥근정도로 원의 반지름 길이로 표시한다.

철이나 고무줄처럼 탄력 있게 복귀하여 입력이 제대로 완성되지 않았음을 인식하게 해주는 요소, 화살표 아이콘이 가이드라인의 끝에 닿으면 바로 잠금해제가 되는 것이 아니라 아이콘을 밀었던 손가락을 가이드라인 끝부분에서 일정거리 안에서 화면으로부터 떼야 잠금해제가 되는 요소, 잠금해제가 디스플레이의 하단에 위치하고 있음을 전제로 하면 손가락이 디스플레이 화면 바깥으로 미끌어져 정밀한 터치를 하지 못한 경우에는 잠금해제가 이루어지도록 된 요소로 이루어져 있다. 이런 요소들은 입력의 신뢰도 높은 잠금해제를 가능하게 해준다. 하나하나의 요소들은 별것 아닌 것 같지만 전체적으로 조화를 이루어 신뢰도 높은 입력이 이루어지는 것이다.

아이폰의 경우에는 물리적 홈버튼까지 음각으로 처리되어 있어 주머니 안에서 버튼이 눌려질 가능성도 적으면서 밀어서 잠금해제 UI가 위의 요소들로 섬세하게 디자인 되어 있어 우연히 장치를 잠금해제 시킬 가능성이 적다. 대부분이 정전식 터치방식으로 되어 있는 스마트폰의 경우에는 아무리 약한 힘을 주더라도 손가락과 디스플레이의 접촉이 있기만 하면 인식되기 때문에 만약 버튼방식으로 입력방식을 디자인해 놓는다면 몸을 움직이다가 실수로 피부에 살짝 닿기만 해도 장치가 작동된다. 최근에는 스마트폰을 이용한 직접 결제시스템이 개발되고 있는데 결제대기 상태에서 한 번의 터치로 이루어지는 버튼방식을 사용하게 되면 이용자의 불안을 가중시키게 된다. 수많은 요소로 이루어진 ‘밀어서 잠금해제’의 입력방식을 이용하면 결제 시에 입력의 신뢰도가 높으면서도 간편하게 단 한 번의 제스처로도 입력이 가능한 장점이 있다.



Nintendo의 십자키 인터페이스 특허21)



위의 그림은 닌텐도에서 개발한 물리적 십자키 인터페이스이다. 닌텐도의 십자키 외의 4개의 버튼으로 이루어진 방향키 인터페이스는 대각선방향으로 명령을 입력하기 위해서 한 개의 엄지손가락으로 위쪽 버튼과 오른쪽 버튼을 동시에 눌러줘야 해서 불편하다. 이에 반해서 닌텐도의 십자키 인터페이스는 상단의 버튼과 오른쪽의 버튼이 이어져 있어서 엄지손가락으로 대각선 방향으로 눌러주기만 하면 대각선방향의 입력이 가능하다. 이렇게 신뢰도 높은 입력방식으로 닌텐도는 십자키 특허를 등록받았다. 이와 비교해서 봤을 때 ‘밀어서 잠금해제’UI가 우리나라 법원에서 기술의 진보성 부족으로 인정받지 못한 것은 쉽게 납득되지 않는 부분이 있다. 물리적 인터페이스에서 플라스틱의 형태와 고무재료가 하는 완충작용을 소프트웨어 인터페이스에서는 시각적요소가 대신하고 있는 것이다. 인터페이스 디자인은 시각적 미감을 느끼게 하는 역할만 하는 게 아니라 밝기와 색과 형태에 따른 인식을 활용해서 사용자가 시각적 인지를 통해 효과적으로 입력할 수 있도록 도와주는 기술적 역할을 한다.

(2) bouncing back UI²²⁾

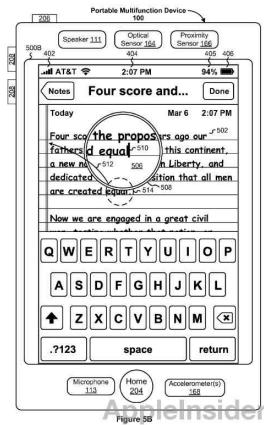
고무줄 UI로도 불리는 이 특허는 스마트폰 장치에서 화면아래의 자료를 보기 위해 화면을 올릴 때 하단에 더 이상의 자료가 없으면 고무줄이 늘어나다 다시 회복하는 듯하게 반동효과를 화면에 보여줌으로써 현재 페이지가 마지막 페이지임을 알려주는 인터페이스이다. 화면을 위로 올려도 화면이 고무줄처럼 탄성력 있게 튕김으로써 아래쪽에는 더 이상 페이지가 없다는 내용을 자연스럽게 인식시켜 주는 기능을 한다. 만약 리버밴드 UI가 없다면 사용자는 화면이 시스템 딜레이 때문에 아래로 넘어가지 않는 건지 또는 페이지가 더 이상 없는지를 판단하지 못한다. 안드로이드 운영체제의 경우에는 이 특허를 피하기 위해 마지막 페이지가 되면 불빛효과를 주는데 이것은 우리가 일상생활에서 자연스럽게 느낄 수 있는 방식이 아니다. 또

21) U.S. Patent No. 4,687,200 , USPTO

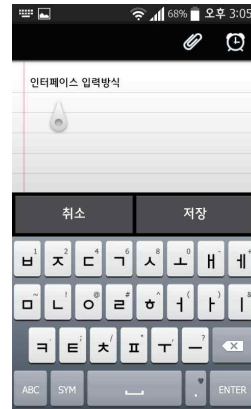
22) U.S. Patent No. 7,469,381

위에서 본 slice to unlock 특허와 마찬가지로 화면을 회복할 때에 반동효과를 주기 때문에 전체 사용자 경험의 일체성도 높인다.

(3) Magnifier Editing UI ²³⁾



U.S. Patent No. 8,255,830



Android 4.0.4 cursor editing.

돋보기 인터페이스로 알려져 있는 이 특허는 편집 상태에서 문자열을 터치한 상태로 잠시 있으면 손가락 터치부위 상단에 돋보기가 나타나고 터치한 상태에서 손가락의 위치를 이동시켜 주면 커서가 글자사이의 공간을 선택가능하게 해주는 방식에 대한 특허다. 기존의 마우스를 사용한 문서편집에서는 문자열을 편집하고 싶을 때 마우스를 이동시켜 클릭하면 되지만 상대적으로 두꺼운 손가락을 사용하여 작은 화면의 문자열 사이를 선택하는 건 쉬운 일이 아니다. 하지만 이 인터페이스를 이용하면 편집하고자 하는 부근에 터치하면 손가락으로 인해 터치한 글자의 가려진 부분이 상단에 돋보기로 확대되어 보여 지고 이 상태에서 손가락을 움직이면 그에 따라 커서가 이동하여 정밀한 포인팅이 가능하다. 안드로이드 운영체제를 사용하

23) Methods and graphical user interfaces for editing on a multifunction device with a touch screen display, U.S. Patent No. 8,255,830

는 스마트폰²⁴⁾의 경우에는 돋보기를 사용하지 않아 손가락으로 터치 한 후 손가락을 떼야 글자사이를 선택할 수 있는 커서가 생기고 이동가능하기 때문에 iOS에 비해 손가락을 붙였다 떼는 동작을 한번 더 요구한다. 작은 화면에서 작은 글자를 손가락으로 편집해야 하는 스마트폰에서 돋보기 편집 방식은 간단하면서 효과적인 인터페이스다.

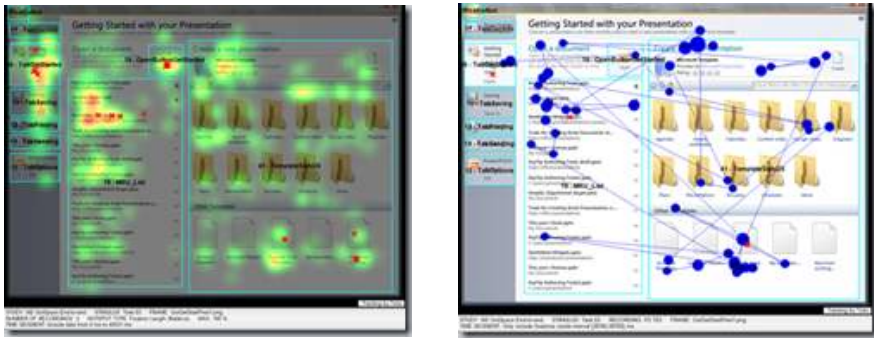
4. UI개발

앞에서 예를 든 인터페이스 외에도 터치 방식의 스마트폰에는 pinch to zoom, scroll indicator, unified search와 같은 다양한 소프트웨어 인터페이스가 이용되어지고 있다. 각종의 개별적 인터페이스가 효율적으로 작동하고 있으면서 유사한 느낌을 주도록 설계되어 최종적으로는 사용자가 기계에 대해 느끼는 감각이 총체적 경험을 결정하게 된다. 위와 같은 인터페이스가 없어도 문자명령 입력 방식이나 과거에 이용하던 GUI방식으로도 시스템은 작동은 하지만 입력의 불편함을 감수하여야 한다. 소비자들은 작동하기 편리하고 신뢰도 높은 제품을 원한다. 그래서 기업들은 인터페이스 개발에 열을 올리고 있다.

최근에는 쾌적한 UI설계를 위해 첨단 기술을 도입하고 있다. 아래의 그림은 마이크로소프트에서 판매하는 MS오피스 프로그램의 인터페이스 설계를 위한 안구 위치 추적 자료이다. 인간의 눈이 직접적으로 인식하는 부분은 디스플레이의 광활한 영역에 비해 협소하므로 디스플레이에 화상이 주어졌을 때 눈동자의 이동에 잦은 영역을 확인하여 그 영역에 주요한 아이콘을 두는 방법이다. 마이크로소프트는 인터페이스에 대한 연구투자로 오피스 프로그램에서 사용자가 선택한 작업에 따라 인터페이스가 변화하는 리본메뉴 인터페이스를 제작하였다. 기존의 고정툴바메뉴는 시스템이 고도화됨에 따라 아이콘들이 늘어나서 사용이 불편해졌는데 리본메뉴는 작업내용에 따라 메뉴바가 적절하게 변화하기 때문에 작업 시간을 대폭 줄일 수

24) LG-KU5400, Android 4.0.4 version

있다.



안구 위치 추적기술을 통한 UI 디자인²⁵⁾

이처럼 기술기업들은 인터페이스 디자인을 위해 많은 투자와 연구를 하고 있다. 그럼에도 불구하고 이런 인터페이스 보호에 대한 인식은 미흡하다. 많은 사람들이 이처럼 당연하고 쉬운 것에 특허를 주는 것을 이해할 수 없다고 말한다. 뛰어난 인터페이스일수록 사용법에 대한 학습을 최대한 줄이고 이용할 수 있도록 디자인하므로 소비자에게는 그 인터페이스가 당연하고 직관적으로만 보이게 된다. 하지만 그러한 직관성과 당연함을 얻기 위해서 소프트웨어 회사는 많은 시간과 연구비를 투자한다. 인터페이스 기술에 대한 사회적 무관심으로 인해서 참신한 아이디어로 정교하게 설계된 인터페이스들이 명백하고 당연한 것이라 여겨지고 있지만 이런 모든 것들 뒤에는 기업의 투자와 개발자들의 땀과 노력이 숨어있다.

IV. 소프트웨어의 보호

1. 소프트웨어 지적재산권을 보호하기 위한 노력의 필요성

25) [http://en.wikipedia.org/wiki/Ribbon_\(computing\)](http://en.wikipedia.org/wiki/Ribbon_(computing))

전반적으로 소프트웨어 산업의 가장 큰 문제점은 후발업체들이 선발업체의 혁신적인 제품을 따라잡기 위해 투자한 시간과 비용이 선발업체가 제품개발을 위해 투자한 시간과 비용에 비해 훨씬 적다는 것이다. 미국의 사례로 스프레드시트 소프트웨어의 대명사였던 로터스 1-2-3 제품은 약 300불에 판매되었다. 하지만 로터스 1-2-3을 클로닝한 소프트웨어인 페이퍼백은 69달러에 판매될 수 있었다.²⁶⁾ 유저 인터페이스 디자인에 투자하고 싶은 회사가 있어도 경쟁 업체가 적은 투자비용으로 동일한 제품을 쉽게 제작할 수 있다면 선두 업체는 투자비용을 회수하기 못할 위험을 감수해야 한다. 시장의 소비자들은 동일한 기능을 하면서 저렴한 제품에 대해 환영할 것이기 때문에 시장의 선택에 맡겼을 때의 결과는 참담할 것이다. 이런 환경에서는 UI와 같은 컴퓨터프로그램이 법적으로 보호받지 못한다면 엄청난 시간과 돈을 투자할 사람은 아무도 없을 것이다. 산업의 중심이 지식 재산에 가치로 변해가고 있는 현재의 상황을 고려해 볼 때 소프트웨어에 대한 우리의 법적체제를 정비해야 할 필요가 절실하다.

우리나라의 컴퓨터 프로그램은 대부분 저작권법에 의해서 보호된다. 하지만 저작권에 의한 보호는 권리의 범위가 협소하고 침해의 증명이 어렵다는 문제가 존재한다. 그밖에도 영업비밀에 의한 보호가 제시 될 수 있겠지만 소프트웨어는 특성상 리버스 엔지니어링에 의한 알고리즘이 쉽게 드러나기 때문에 영업비밀에 의한 보호도 큰 의미는 없다. UI의 경우에는 우리나라의 디자인 보호법에 의한 화상디자인으로 등록받아 보호받을 수 있지만 이 경우에는 UI의 회화적 요소만 보호받을 수 있을 뿐 기능적 아이디어에 대한 측면을 보호받을 수 없다는 단점이 있다.

컴퓨터 UI의 경우에는 너무나 쉽게 복제가 가능하므로 시장에 출시된 제품에 대해 일정기간 동안 복제를 금지할 수 있어야 한다. 그러므로 일정기간동안 배타적으로 복제를 금지할 수 있는 특허권에 의한 보호가 요구된다.

26) 최민석, 「소프트웨어 지적재산권에 대한 새로운 시각」, 2008, p.48

2. 소프트웨어 특허권의 그늘

미 백악관은 웹사이트에 청원을 올릴 수 있도록 했는데 상위 10개의 청원 중에 소프트웨어 특허금지가 포함되어 있다. 이들은 ‘소프트웨어 특허가 혁신과 경쟁을 지원하기 보다는 혁신을 지체시키고 경쟁을 방해하는 방법이 됐다’고 주장한다. 또 소프트웨어 특허는 대기업이 중소기업에 대한 독점 도구로 이용되고 있다고 말한다. 특허를 내는 것도 그리고 유지한 비용도 개인과 중소기업이 감당하기는 어려우므로 결국 대기업이 아니면 특허제도의 이점을 활용하기 어려우며 소프트웨어의 경우에는 공장이 필요 없이 개인이 머릿속에 떠올라서 쓸 수 있는 것이므로 수 없이 양산될 수 있고 그 결과 특허권이 없는 사람은 아무것도 할 수 없다고 한다. 하지만 이러한 주장은 더 많은 자본을 가진 자가 더 유리할 수밖에 없는 자본주의 체제가 전체가 가진 단점일 뿐이다.

컴퓨터 잡지 컬럼리스트인 Katherine Noyes는 소프트웨어 특허가 사라져야 하는 것이 마땅하다고 주장한다.²⁷⁾ 그녀는 소프트웨어는 수학과 같은 것이어서 알고리즘 대해서 특허를 받는 것은 수학에 대해 특허를 받는 것과 같다고 한다. 하지만 현행 특허법에서는 추상적 수학이나 알고리즘만은 특허를 부여하고 있지 않으며 프로그래머는 알고리즘을 이용하여 컴퓨터프로그램을 제작하는 것이지 프로그램이 알고리즘 그 자체인 것은 아니다. 이 주장은 음악이 보편적인 7음계로 이루어지기 때문에 작곡가가 창작한 음악에 저작권을 줄 수 없다는 말과 같다. 컴퓨터프로그램이 추상적 수학이나 알고리즘을 이용해서 만든 것이기 때문에 특허권을 부여할 수 없다는 주장은 컴퓨터프로그램에 대한 이해의 부족 때문이다. 알고리즘과 컴퓨터프로그램의 기능은 독립적인 측면이 존재한다. 각기 다른 알고리즘으로도 동일한 기능을 하는 컴퓨터프로그램을 제작할 수도 있다. 컴퓨터프로그램은 추상적 수학과 알고리즘 해법을 이용하는 것이지 수학과 알고리즘 그 자체는 아니다. 현대의 컴퓨터프로그램은 기술적 아이디어를 실현하기 위

27) Katherine Noyes, It's clear why software patents need to disappear, www.PcWorld.com Aug 17, 2011

해 알고리즘을 사용하는 것이다.

또 소프트웨어로 작성되는 UI 특허의 경우에는 특허등록으로 인해 각 회사가 각기 다른 인터페이스를 가지게 된다면 사용자는 다른 회사에서 제작된 프로그램을 사용하게 될 때마다 새롭게 인터페이스에 익숙해지는 시간을 허비해야 된다는 주장을 한다. 그러나 소비자들이 다양한 인터페이스를 사용하게 됨에 따른 불편함이 인터페이스를 개발하기 위해 시간과 노력을 들인 개발자에의 성과를 없앨 만큼 크다고 할 수 없다. 개발자가 이러한 인터페이스를 개발할 동기가 없었다면 애초에 우수한 인터페이스가 생기지도 않았을 것이기 때문이다. 그리고 UI 특허는 소프트웨어 특허로 완전하게 동일한 모방이 아니라면 특허청구항에 변화를 주면 조금 성능이 떨어지더라도 회피가 가능하다. iOS²⁸⁾의 대부분의 인터페이스는 안드로이드가 특허청구항을 조금씩 변화주면서 비슷하게 개발할 수 있고 실제로도 그렇게 사용하고 있다.²⁹⁾ 또한 독과점의 문제는 소프트웨어 특허만의 문제가 아니라 자본주의 체제의 어디서나 볼 수 있는 전체의 문제이다. 시스템 전체의 문제에 대해서 그 일부인 소프트웨어 특허를 비난하는 것을 올바른 지적이 아니다.

3. 소프트웨어 특허의 진화

과거의 컴퓨터프로그램은 단순히 컴퓨터에 명령을 내리는 방식이었다. 이것은 추상적 수학의 방식이었고 특허의 대상이 아니었다. 그 이후에는 수학을 이용하더라도 청구항 전체로 보았을 때 외부에 유용한 물리적 결과를 만들어내면 특허의 대상이 되었다. 그 후에는 컴퓨터프로그램이 하드웨어를 이용해 구체적으로 실현되면 자연법칙을 이용한 기술적 사상의 창

28) iOS (previously iPhone OS) is a mobile operating system developed and distributed by Apple Inc., <http://en.wikipedia.org/wiki/IOS>

29) 구글 대변인은 애플의 핀치-투-줌 특허(US 7,844,915)에 대해 청구항 중 하나인 ‘단일 입력이 있을 경우에는 스크롤로 해석하고’라는 부분이 안드로이드 젤리빈과 다르므로 특허를 침해하지 않았다고 주장한다.

작으로서 발명에 해당되었다. 그리고 현재에는 정보산업의 발달로 인해 기존의 하드웨어가 소프트웨어로 대체되고 있어 하드웨어와 소프트웨어의 구분이 무의미한 시점까지 이르게 되었다. 최근에는 UI에 대한 기술이 발전하면서 기술의 성능이 단순히 기계적인 성능만을 나타내는 것이 아니라 인간에 대한 이해가 바탕이 된 컴퓨터프로그램들이 탄생하고 있다. 그 예로 최근의 사용자 환경에는 인공지능 음성인식 시스템이 도입되어있다. 인공지능 음성인식 시스템은 기존의 UI에 결합하여 다양한 컴퓨터프로그램들의 새로운 진화를 보여줄 것으로 기대된다. 시시각각 변해가는 기술의 특성으로 볼 때 우리의 지식재산 법체제 역시 지속적으로 변화하며 시대에 적응할 것이 요구된다.

V. 결론

글로벌 특허분쟁으로 인해서 유저 인터페이스와 같은 소프트웨어 특허에 대한 관심이 많아졌다. 전자기기에 사용되는 소프트웨어 인터페이스는 사소해 보이지만 개발자들의 시간과 기업의 많은 자금이 투자되어 개발되어진다. 하지만 소프트웨어 인터페이스에 대한 사람들의 인식은 그리 좋지 못하다. 많은 사람들이 소프트웨어를 무료라고 생각하기도 한다. 소프트웨어 인터페이스를 제작하기 위해서는 많은 시간과 노력이 들어가지만 이미 만들어진 UI를 복제하는 것은 쉬운 편이다. 시장에서 성공한 UI와 동일한 기능을 수행하도록 복제UI를 제작하는 경우에는 프로그램을 제작하는데 있어 기존의 UI로부터 아이디어를 얻었기 때문에 최초 개발에 따른 시간과 비용을 절약하게 되고 결국 최초 개발자만 손해의 위험을 감수해야 한다.

복제한 인터페이스가 표현을 베낀 것이 아니라 아이디어를 베낀 경우에는 저작권법과 디자인보호법에 의한 처벌은 불가능하다. 왜냐하면 저작권법은 표현을 보호하고 디자인보호법은 출원된 화상만을 보호하기 때문이

다. 그래서 소프트웨어 인터페이스에 대해 특허권의 적극적인 보호가 더욱 요구되는 것이다.

소프트웨어 인터페이스 기술의 중요성에 대한 이해의 부족으로 인하여 인터페이스에 특허권을 부여하는 것에 대해 사회적 비판의 목소리가 높다. 하지만 새로운 인터페이스를 개발하기 위해서는 많은 투자가 필요하고 개발된 기술이 보호받지 못하면 그 기술에 대해서는 누구도 투자하지 않을 것이다. 그러므로 지금이라도 유저인터페이스의 중요성을 인식하고 그 보호를 위해 법제를 정비하여 UI에 대한 사회적 인식을 높인다면 인터페이스 기술에 대한 많은 연구와 그에 따른 발전이 있을 것이다.

【참고문헌】

조영선, 『특허법』, 박영사, 2012.

프레드 원쇼프스키, 『특허전쟁』, 세종서적, 1996.

계승균외 4인공저, 『로스쿨 지적재산권법』, 법문사, 2010.

김관식, “컴퓨터프로그램의 특허법적 보호”, 『정보화정책 제13권 제4호』, 2006년 겨울, pp.108~127

김원학, “SW의 효과적 보호방안”, IT와 법연구(1), 2009, pp.1~30

김현호, “애플 VS 삼성, 두 거인의 특허 전쟁”, 『invention & patent』, 2012. 2, pp.19~22

배대헌, “대중요법을 채용한 Bilski v. Kappos 사건판결”, 재산법연구, 2011, pp.509~538