

오픈 이노베이션 전략의 법적 함의

- SCO 대 IBM 소송을 중심으로 -

박 성 필* · 김 용 길**

목 차

- | | |
|-------------------------|------------------------------|
| I. 서론 | IV. SCO v. IBM 소송 |
| II. 공개 소프트웨어의 일반론 | 1. 유닉스(UNIX)와 리눅스(GNU/Linux) |
| 1. 공개 소프트웨어의 개념 | 2. SCO의 주장 |
| 2. FOSS 라이선스의 종류 | 3. IBM의 법적 대응 |
| 3. 공개 소프트웨어의 권리 · 의무 관계 | 4. 소송의 경과 |
| 4. 공개 소프트웨어의 장단점 | 5. SCO v. Novell 소송 |
| 5. 공개 소프트웨어 산업 | 6. SCO v. IBM 소송의 시사점 |
| III. IBM의 오픈 이노베이션 전략 | V. 결론 |
| 1. IBM의 지식재산 경영전략 | |
| 2. IBM의 공개 소프트웨어 지원정책 | |

* 한국과학기술원 교수, 법학박사, 미국변호사.

** 원광대학교 법학전문대학원 교수, 법학박사.

초록

오늘날 IT 산업을 중심으로 오픈 이노베이션은 기업의 생존과 성장을 위한 선택이 아닌 필수불가결의 전략으로 자리매김하고 있다. 한편 공개 소프트웨어 운동은 오픈 이노베이션과 그 기원 및 접근방법에서 본질적인 차이가 있지만, 최근 오픈 이노베이션 전략을 채택한 기업들 중 상당수가 공개 소프트웨어를 활용하고 있으며, 오픈 이노베이션의 기본 모델 자체를 자신의 모델의 중요한 일부로 여기고 있다. 공개 소프트웨어 진영에서도 상업적 목적의 소프트웨어 업체들과도 협력하는 사례가 지속적으로 증가하면서 이를 뒷받침하는 라이선싱 규약들이 개발되어 왔다. 세계 굴지의 하드웨어 기업으로서, 최근 소프트웨어 중심의 혁신을 주도하고 있는 IBM은 이러한 오픈 이노베이션과 공개 소프트웨어 운동이 어떻게 조화될 수 있는지 보여주는 대표적인 기업이다. 본고는 IBM의 오픈 이노베이션 전략, 그리고 SCO와의 소송 관련 법적 쟁점과 그 시사점들을 분석하였다. 아직까지 종결되지 않은 이 소송은 IBM과 같이 오픈 이노베이션을 주도하는 기업들이 지식재산권 분쟁에 휘말릴 경우 문제될 수 있는 법적 쟁점들을 보여주며, 그에 대한 적절한 대응전략을 모색할 수 있게 해 주는 좋은 사례이다. 이에 따라 본고에서는 SCO 대 IBM 소송의 사실관계와 절차적 진행과정, 논점별 정리를 통해 공개 소프트웨어의 다양한 관점과 효율적인 소송전략, 지식재산 포트폴리오 관리의 중요성 등을 분석하였다.

주제어

오픈 이노베이션, 자유/오픈소스 소프트웨어, SCO v. IBM, 지식재산 포트폴리오, 리눅스, 지식재산권

I. 서론

오픈 이노베이션(open innovation), 즉 “개방형 혁신”은 체스브로(Henry Chesbrough)¹⁾ 교수가 제안한 신조어이다. 그에 따르면 오픈 이노베이션이란 기업이 회사 내부와 외부의 아이디어, 시장에 진입하기 위한 내부와 외부의 경로를 모두 활용하여 기술혁신을 달성할 수 있고, 또한 그래야만 한다는 혁신의 패러다임을 뜻한다고 한다.²⁾ 한 기업이 지식을 독점하기 어려운 상황에서 가장 효율적으로 혁신을 달성할 수 있는 방법은 결국 기업의 내부와 외부의 자원을 총체적으로 활용하는 것이라는 의미이다. 한편 “자유/오픈소스 소프트웨어(Free/Open Source Software, 이하 ‘FOSS’라 한다)”³⁾ 운동은 소프트웨어 사용자가 자유롭게 소스코드에 접근하고, 수정하고, 배포할 수 있도록 그 소스코드를 공개하는 것을 의미한다. 이러한 개념을 처음 주창한 스톨만(Richard Matthew Stallman)과 “자유소프트웨어재단(Free Software Foundation, 이하 ‘FSF’라 한다)”은 소프트웨어가 기업의 이윤추구를 위해 사유화되는 것을 거부하고, 개인 프로그래머와 기업으로 구성된 사용자들이 자유로운 공유와 협력을 추구하는 이상적인 목표를 가지고 있었다.

오픈 이노베이션과 오픈소스는 이처럼 출발점이 서로 달랐으며, 어떻게 보면 서로 연관성이 없었던 개념이라고도 할 수 있다. 오픈 이노베이션은 산업을 구별하지 않지만, 오픈소스 운동은 용어 자체에서 소프트웨어 산업에 특화된 개념

1) Chesbrough, Henry William, *Open Innovation: The New Imperative for Creating and Profiting from Technology*, Harvard Business Press, 2003. 체스브로는 이 책의 서론에서 과거의 혁신은 “닫힌 혁신(closed innovation)”으로서 “성공적인 혁신은 통제를 요구한다(successful innovation requires control).”고 말하는 입장이었는데, 노동시장에서 경험과 기술이 뛰어난 인력의 유동성이 점점 커지고, 기술을 사업화하는 데 필요한 시간이 점차 줄어들었으며, 공급자들과 소비자들의 지식이 점점 증가하여 자신의 역량만으로 수익을 창출하기가 힘들어진 결과 패러다임 변화가 일어나 열린 혁신(open innovation)이 제시되었다고 설명하고 있다.

2) 전개서의 원문은 다음과 같다. “Open innovation is a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as the firms look to advance their technology.”

3) 본고에서 자유/오픈소스 소프트웨어를 지칭할 때는 약어인 FOSS라고 하거나 정부에서 사용하고 있는 “공개 소프트웨어”라고 하기도 한다.

임을 드러내고 있다. 오픈 이노베이션이 거래(transaction)와 계약(contract)에 기초한 기술의 혁신을, 오픈소스 운동은 협력(collaboration)과 공유(sharing)의 정신으로 소프트웨어를 개발하고 사용하는 방식이라고도 할 수도 있다.⁴⁾ 혁신 활동의 구조를 기준으로 한다면, 오픈 이노베이션은 본래 혁신하는 기업이 중심이 되어 방사형으로 둘러싼 외부 자원들이 집중되어 가치를 창출하는 방식으로, 오픈소스 운동은 사용자 공동체가 서로 신경망처럼 연결되어 정보를 공유하고 협력하는 방식으로 고안된 것이다.⁵⁾ 이러한 구분에 따르면, 단기적으로는 혁신의 주체인 기업이 프로젝트의 방향을 설정하고 프로세스를 주도해 갈 수 있으므로 오픈 이노베이션이 유익할 수 있지만, 장기적으로는 오픈소스 방식이 훨씬 유용하고 또 리스크가 적을 수 있다.

그러나 이들은 최소한 ‘개방성(openness)’을 지향한다는 점에서 공통점이 있다. 또한 오픈 이노베이션의 주창자인 체스브로 교수는 혁신을 창조하는 네 가지 유형의 조직 중 하나로서 오픈소스 커뮤니티를 “혁신 선교사(innovation missionaries)”로 명명하면서 오픈 이노베이션의 개념 속에 오픈소스 운동을 포함시켰다.⁶⁾ 체스브로 교수에 따르면 혁신 선교사들은 어떤 명분(cause)을 살리기 위해 기술을 창조하고 발전시키며, 혁신 상인들이나 혁신 건축가들과는 달리 자신들의 일에서 금전적인 이익을 추구하지 않는다. 그들이 추구하는 미션이 그들에게 동기를 부여한다.⁷⁾

최근 OpenPower Consortium을 통한 파워칩(power chip) 관련 하드웨어와 소프트웨어 라이선싱을 시작한 데서 단적으로 알 수 있듯이, IBM은 오픈 이

4) <<http://opensource.com/business/10/10/open-innovation-and-open-source-innovation-what-do-they-share-and-where-do-they-differ>>, 검색일: 2013. 4. 20.

5) 위 웹사이트 <<http://opensource.com>>의 기사는 Stefan Lindegaard와 Chris Grams의 좌담 형식으로 오픈 이노베이션과 FOSS의 차이를 설명하고 있다. 이에 따르면 소프트웨어 산업 밖에서는 오픈소스 방식의 혁신이 어렵거나 혹은 불가능하다. 다만 최근에는 오픈소스 하드웨어의 가능성에 대한 논의도 이루어지고 있는 것으로 보인다. 검색일: 2013. 5. 1.

6) <<http://sloanreview.mit.edu/article/the-era-of-open-innovation>>, 검색일: 2013. 5. 1. 나머지 세 가지 혁신조직 유형은 혁신 탐험자(innovation explorers), 혁신 상인(innovation merchants), 혁신 건축가(innovation architects)이다.

7) FOSS 운동이 반드시 금전적 보상을 배제하는 것은 아니다. 대표적인 비즈니스 모델은 FOSS 사용자에게 대한 다양한 서비스 제공의 대가를 수입원으로 삼는 것이다.

노베이션을 적극적으로 실천해 온 기업이다.⁸⁾ IBM은 전통적으로 하드웨어의 기술이 핵심역량이었지만, 하드웨어를 운용하는 데 필수적인 운영체제(operating system, 'OS') 공급업체들, 특히 윈도우즈(Windows)를 공급하는 마이크로소프트(Microsoft)에 대한 의존성을 탈피하기 위해 일찍부터 오픈소스 기반의 리눅스(Linux) 커뮤니티를 오픈 이노베이션 형태로 지원해 왔다.⁹⁾ 본고는 IBM의 이러한 오픈 이노베이션 전략을 중심으로 공개 소프트웨어의 기본 개념과 발전 방향, 특히 상업적 이윤을 추구하는 기업들과 FOSS 운동이 만나는 지점에서 지식재산권과 계약상 이슈들을 고찰해 보고자 한다.

본고는 특히 IBM의 이러한 오픈 이노베이션 전략에 대한 반작용으로 SCO(Santa Cruz Operation)가 IBM을 제소한 사건을 분석하고자 한다. 이 사건을 둘러싼 기업들의 갈등과 지식재산권법과 계약법적 이슈들이 FOSS에 대한 심층적인 이해에 도움을 주며, 또한 오픈 이노베이션과 FOSS 운동이 어떻게 조화될 수 있는지에 대한 단초를 제공하기 때문이다.

II. 공개 소프트웨어의 일반론

1. 공개 소프트웨어의 개념

소프트웨어가 개발된 초창기에는 소프트웨어를 사유재산으로 취급하는 경우가 드물었다. 그러나 1980년대 이후 개인용 컴퓨터가 널리 보급되면서 소프트웨어 산업이 급성장했고, 그 상업적 가치가 주목받기 시작하였다. 재산권적인 관점에서는, 소프트웨어를 가치 있는 영업비밀로서 보호하는 외에, 소스코드에 대해 저작권을 주장하거나, 소프트웨어상 알고리즘을 산업에 적용하는 경우에

8) <www.pcworld.com/article/2046044/ibm-to-license-power-chips-to-keep-them-relevant.html>
참조. 이는 고객의 요구를 충족할 수 있는 클라우드 컴퓨팅을 가능하게 하려는 것이며, 현재 구글(Google), NVIDIA, Tyan, Mellanox 등이 참여하기로 한 상황이다.

9) 자유 소프트웨어 진영에서의 리눅스(Linux)의 정확한 용례는 "GNU/Linux"라고 한다.
<www.gnu.org/gnu/why-gnu-linux.html>, 검색일: 2013. 5. 1.

방법발명으로 특허등록을 하거나, 소프트웨어와 하드웨어와의 결합을 특허등록하는 방법이 활용되기 시작하였다. 나아가 이러한 지식재산권을 근거로 라이선싱 계약을 통해 사용자의 자유로운 사용·복제·배포·수정을 통제하면서 로열티 수익을 확보하는 비즈니스 모델이 각광을 받았다. 하버드 대학에서 물리학을 전공한 스톨만은 대학의 신입생 시절부터 MIT의 인공지능연구소(Artificial Intelligence Lab)에서 OS 개발에 참여하였다.¹⁰⁾ 그 과정에서 스톨만은 소프트웨어 업계가 초창기의 공유정신을 잃어버리고 점차 상업화되어 가는 현실을 비판했고, 결국 1983년 GNU 프로젝트를 시작하였다. “GNU”는 일종의 언어유희처럼 들리지만, “GNU는 유닉스가 아니다(GNU is Not Unix)”라는 뜻의 재귀적 약어(recursive acronym)이다.¹¹⁾ 소프트웨어 판매업자들이 사용자들을 분리하고 정복해서 공유하지 않겠다는 동의를 받아내는데, 스톨만은 이런 식으로 다른 사용자들과 연대를 깨는 것을 거부하며 자신의 선한 양심으로는 비밀유지약정이나 소프트웨어 라이선싱의 계약에 서명할 수 없었다고 하였다.¹²⁾

공개 소프트웨어는 컴퓨터의 소스코드를 공개하여 사용자가 자유롭게 사용, 복제, 검토, 수정, 배포할 수 있도록 한 소프트웨어를 의미한다.¹³⁾ 사용자들이 소스코드에 접근하고, 각종 버그(bug)¹⁴⁾를 수정하고, 기능을 추가할 수 있다는

10) <<http://stallman.org/index.html#serious>>, 검색일: 2013. 11. 20. 리처드 스톨만은 자신의 이름이 약자인 “RMS”로 불리기를 선호한다고 밝히고 있다.

11) <www.gnu.org/pronunciation> 검색일: 2013. 11. 20. 스톨만의 설명에 따르면 GNU는 “그누”로 발음한다.

12) <www.gnu.org/gnu/manifesto.html>, 검색일: 2013. 11. 20. 원문은 다음과 같다: “I consider that the Golden Rule requires that if I like a program I must share it with other people who like it. Software sellers want to divide the users and conquer them, making each user agree not to share with others. I refuse to break solidarity with other users in this way. I cannot in good conscience sign a nondisclosure agreement or a software license agreement.” 또한 스톨만은 자신이 수년간 MIT의 인공지능연구소(Artificial Intelligence Lab)에서 일하면서 이러한 현실에 저항했으나 계속 그런 일이 자행되는 것을 보고 연구소를 사임했으며, 컴퓨터를 불명예스럽게 않게 계속 사용하고자 자유 소프트웨어들을 모아 이 운동을 시작했음을 밝히고 있다.

13) Ilardi, Terry J., “Open Source Software,” *Patents, Copyrights, Trademarks, and Literary Property Course, Handbook Series, PLI/Pat*, No.911(September–October, 2007), Practising Law Institute, p.521.

14) 버그(Bug)란 소프트웨어의 오류나 결함을 의미하며 미 해군 준장을 지낸 컴퓨터 과학자인 Grace Murray Hopper가 컴퓨터와 관련하여 사용한 것이 시초로 알려져 있다. 자세한 내용은 Taylor III, Alexander L., “The Wizard inside the Machine”, TIME, April 16, 1984 참조.

점에서 기존의 “독점 소프트웨어(proprietary software)”와 근본적인 차이가 있다.¹⁵⁾ 공개 소프트웨어는 무료(zero cost) 소프트웨어가 아니며,¹⁶⁾ 아무런 “의무가 없는(without obligation)” 소프트웨어라고 하는 것도 잘못이다.¹⁷⁾ 이러한 오해는 1984년 스톨만이 설립한 자유 소프트웨어재단(Free Software Foundation, “FSF”)이 주도해 온 자유 소프트웨어(free software) 운동에 대한 선입견에서 기인하는 측면이 있다. 여기서의 “자유(free)”의 개념이 “무료(free)”로 이해되는 경우가 많기 때문이다.¹⁸⁾ 여기서의 자유는 소프트웨어를 공유(share)하고, 학습(study)하고, 수정(modify)하는 자유이며, 이를 자유 소프트웨어라고 부르는 것은 “사용자들이 자유롭다(the user is free)”는 것을 강조하기 위한 것이다.¹⁹⁾ 대표적인 공개 소프트웨어로 리눅스 커널(Linux Kernel),²⁰⁾ 아파치(Apache),²¹⁾ 파이어폭스(FireFox),²²⁾ MySQL²³⁾ 등이 있다.

15) “proprietary software”는 직역하자면 “재산권적 소프트웨어”이겠지만, 지식재산권의 독점적 성격에 착안하여 공개 소프트웨어와의 이념적 차이를 강조하기 위해 흔히 독점 소프트웨어라 표현하고 있다. 본고에서도 이를 독점 소프트웨어라 칭하기로 한다.

16) 김병일, “오픈소스 소프트웨어와 법적 위험”, 산업재산권, 제25호(2008), 한국산업재산권법학회, 358면.

17) Rosen, Lawrence, “Open Source and Free Software 2009: Where We’ve Been and Where We’re Going,” *Patents, Copyrights, Trademarks, and Literary Property, Course Handbook Series, PLI/Pat*, No.989(November– December, 2009), Practising Law Institute, p.382.

18) 그러나 “자유 소프트웨어”와 “오픈소스 소프트웨어”는 역사 및 추구하는 이념에 있어서 미묘한 차이가 있다. 소스코드의 자유로운 사용이라는 이상을 강조하는 자유 소프트웨어와 달리, 오픈소스 소프트웨어는 사회적 실용성과 경제성에 중점을 두고 소스코드의 공개를 강조하고 있다. 문화체육관광부, 저작권위원회를 비롯한 정부 및 공공기관에서는 자유 소프트웨어를 포함한 오픈소스 소프트웨어를 대개 “공개 소프트웨어” 또는 “공개 SW”로 칭하고 있으며, 따라서 본고도 특히 어느 한 쪽 입장을 기술할 경우 외에는 양자를 모두 포함하여 “공개 소프트웨어”라 하거나, 영문 약자로 “FOSS(Free/Open Source Software)”라 칭하기도 한다.

19) <www.fsf.org/about/what-is-free-software> 검색일: 2013. 11. 20.

20) 1991년 핀란드의 대학생 토발즈(Linus Torvalds)가 인텔 CPU인 80386 칩에 최적화된 리눅스 커널을 개발했는데, 토발즈는 그 소스코드에 GPL 라이선스를 부여하여 GNU/Linux라는 명칭으로 공개하였다. 이후 리눅스는 인텔 CPU를 장착한 컴퓨터에 한정되지 않은 OS로 발전하였다.

21) 아파치 웹 서버의 라이선스는 사용자가 자유롭게 아파치 소프트웨어를 다운로드 받아서 부분 또는 전체를 개인적 또는 상업적으로 이용하도록 허용하고 있다.

22) 파이어폭스는 “Mozilla Public License(MPL)”로 배포되고 있으며, GPL이나 LGPL도 선택이 가능하다.

23) MySQL은 관계형 데이터베이스 관리 시스템인데, 그 배포는 공개 소프트웨어의 GPL 혹은 독점 소프트웨어의 “end user license agreement(EULA)” 중에서 선택할 수 있다. GPL 조항에 따라 무료 라이선스를 취득할 수 있는데도 어느 사용자가 EULA를 선택한다면 그것은 GPL을 선택할 경우 그 감염성으로 인해 공개를 원치 않는 자신의 독점 소프트웨어 소스코드도 공개해야 하기 때문이다. GPL은 이처럼

그런데 초기에는 자유 소프트웨어가 엄격한 GPL 조항으로 인하여 상용 소프트웨어와 결합하기 어려웠기 때문에 많은 기업들이 적극적인 참여를 꺼렸으나, 1990년대 인터넷 사용자가 급증하면서 GNU/Linux가 널리 보급되었고, 마이크로소프트의 인터넷 익스플로러(internet explorer)에 시장을 잠식당했던 네스케이프(Netscape)가 웹브라우저의 소스코드를 전격 공개하면서 IBM, Sun 등이 자유 소프트웨어에 대한 지원을 시작하였다. 1998년 네스케이프의 소스코드 공개 선언에 뒤이어 개최된 캘리포니아 주의 팔로알토(Palo Alto) 미팅에서 피터슨(Christine Peterson)의 제안에 따라 작명된 ‘오픈소스(Open Source)’라는 말이 사용되기 시작했고, 레이몬드(Eric Raymond)와 페렌스(Bruce Perens)는 자유 소프트웨어 운동의 지나친 철학적, 정치적 색채를 넘어서 공개 소프트웨어 운동을 주도할 기관의 필요성을 제시하며 Open Source Initiative(“OSI”)를 설립하였다.²⁴⁾

2. FOSS 라이선스의 종류

공개 소프트웨어 개발자와 사용자 사이에 그 사용방법 및 조건을 명시한 계약이 “FOSS license(공개 소프트웨어 사용허가권)”이다.²⁵⁾ 특정 OSS를 사용하기 위해서는 그 개발자가 설정한 사용방법 및 조건에 따라야 하며, 이를 위반할 경우 계약 위반 또는 저작권의 침해가 문제될 수 있다.²⁶⁾ FOSS 라이선스는 계약 당사자의 의지에 따라 매우 다양한 유형이 가능하지만, 크게 보면 FSF의

소스코드의 창작자라면 MySQL의 경우처럼 일반 상용 라이선스를 병행적으로 배포하는 것도 허용한다. 현재 MySQL은 오라클(Oracle)이 소유하고 있다.

24) <<http://opensource.org/history>> 검색일 : 2013. 11. 20. OSI에서는 오픈소스에 해당하는 다양한 종류의 라이선스가 지켜야 할 최소한의 기준을 정의(Open Source Definition, “OSD”)해 놓고 이 정의의 규정에 따라 인증, 관리 및 촉진시키는 일을 한다. 이에 대해서는 후술한다.

25) FOSS 라이선스의 법적 성격에 대해 검토한 중요한 판례로서 *Jacobsen v. Katzer*, 535 F.3d 1373 (Fed. Cir. 2008) 참조. 이 사건은 결국 당사자 사이의 화해로 마무리되었지만, 연방순회항소법원이 FOSS 라이선스가 저작권법에 따른 보호를 받음을 분명히 하였다는 점에서 큰 의의가 있다. 결국 독점 소프트웨어와 공개 소프트웨어가 모두 저작권법의 보호대상이 되는 것이다.

26) 고재중, “오픈 소스 소프트웨어의 법적 문제에 대한 검토”, *상사판례연구*, 제25권 제3호(2012), 한국상사판례학회, 169면.

General Public License("GPL")와 Lesser General Public License("LGPL"), 이른바 "학술 라이선스(academic license)"라 불리는 Berkeley Software Distribution("BSD"), 파이어폭스의 Mozilla Public License("MPL") 등이 있다.²⁷⁾ FOSS 라이선스는 원칙적으로 사용자들이 개발자가 공개한 소프트웨어를 획득하여 자유롭게 사용·수정·배포를 할 수 있도록 하고 있다. OSI 홈페이지에는 2013년 12월 현재 약 70개의 OSS 라이선스가 승인을 받은 것으로 등재되어 있는데,²⁸⁾ 이 중 산업적으로 널리 사용되고 있는 것은 몇 개에 불과하다.²⁹⁾

그러나 세계적으로는 여전히 오픈소스 프로젝트가 여러 곳에서 자유롭게 진행되고 있으며, 라이선스의 숫자가 늘어나는 것을 막을 수는 없다. 예를 들어, 오픈소스 프로젝트 포털사이트 Freshmeat에 등록되어 있는 개발 프로젝트들 중에서 오픈소스로 분류되는 약 43,722개 프로젝트가 진행 중에 있는데, Freshmeat에 있는 "오픈소스 소프트웨어 라이선스 분포"에 따르면 개발 프로젝트 중 약 72% 정도가 GPL과 LGPL 라이선스를 사용하고 있으며, 그 뒤를 BSD가 차지하고 있다.³⁰⁾ 다만 FOSS는 셰어웨어(shareware)나 프리웨어(freeware)와 구별되어야 하며, 또한 공공영역(public domain)에 속하는 것도 아니다.³¹⁾ 라이선스별 소프트웨어를 비교하면 다음 <표 1>과 같다.

한편 이윤추구적인 비즈니스 환경하에서 공개 소프트웨어에 대한 수요가 점점 커지고 있으며, 나아가 상업적인 제품에 결합하는 공개 소프트웨어의 숫자도 점차 늘어가고 있다. 이것은 공개 소프트웨어 커뮤니티에는 일종의 위험요인이

27) Fontana, Richard E., "Open Source License Enforcement and Compliance," *Patents, Copyrights, Trademarks, and Literary Property, Course Handbook Series, PLI/Pat*, No.989(2009), Practising Law Institute, pp.79-91.

28) <<http://opensource.org/licenses/alphabetical>>, 검색일: 2013. 12. 1.

29) 개발자들이 자신이 직접 개발한 소프트웨어에 집착하거나, 새 버전의 소프트웨어를 설치하는 것이 귀찮아 기능적 열등성을 감수하려는 경향이 있기 때문이다. FOSS 라이선스의 확산(Proliferation) 이슈를 다루고 있는 문헌으로 Gomulkiewicz, Robert W., "Open Source License Proliferation: Helpful Diversity Or Hopeless Confusion?," *Washington University Journal of Law and Policy*, Vol.30(2009), pp.261-291.

30) 정보통신부, 오픈소스 SW 라이선스 가이드, 2007, 12면. 한편 기존의 Freshmeat 웹사이트는 현재는 <<http://freecode.com>> 도메인으로 변경되었다. 검색일: 2013. 11. 20.

31) Frazer, Brad, "Open Source Is Not Public Domain: Evolving Licensing Philosophies," *Idaho Law Review*, Vol.45(2009), p.373.

〈표 1〉 라이선스별 소프트웨어의 비교³²⁾

유형		가격	소스공개 및 접근	라이선스의 특징	실례
상용 SW		유료	X	허용 범위 내 사용	MS Office, Adobe Photoshop
Shareware		무료/유료	X	실행일 · 실행회수 · 사용목적 · 기능 제한	V3+, 알집
Freeware		무료	X	사용, 소스코드 접근 불가	뽕집, Winamp, Acrobat Reader
Bundle/OEM		유료	X	SW가격이 HW가격 에 포함	Windows, Power DVD
공개 SW	GPL	무료/유료	O (사용 · 수정 · 배포 可)	2차적저작물 재공개 의무(O) 독점 SW와 결합(X)	GNU/Linux, Ogle
	LGPL	무료/유료	O (사용 · 수정 · 배포 可)	2차적저작물 재공개 의무(O) 독점 SW와 결합(O)	Ming, JBoss
	MPL	무료/유료	O (사용 · 수정 · 배포 可)	2차적저작물 재공개 의무(O) 독점 SW와 결합(O)	Firefox (GPL, LGPL도 가능)
	BSD	무료/유료	O (사용 · 수정 · 배포 可)	2차적저작물 재공개 의무(X) 독점 SW와 결합(O)	FreeBSD, Tcpdump
	Apache	무료/유료	O (사용 · 수정 · 배포 可)	2차적저작물 재공개 의무(X) 독점 SW와 결합(O)	Apache, Tomcat
Public Domain		무료	O (사용 · 수정 · 배포 可)	2차적저작물 재공개 의무(X) 독점 SW와 결합(O)	저작권 보호기간 만료, 저자의 헌정 (dedication)

될 수 있다. 오픈 이노베이션에 FOSS를 “유기적으로 결합”하는 전략적인 프로세스의 구축은 소프트웨어를 개발하는 프로그래머나 또는 관련 제품개발의 프

32) 이 표는 다음 자료들을 종합하여 필자가 작성하였다. Bruce Perens, “The Open Source Definition,” *Open Sources: Voices from the Open Source Revolution*, 1999

〈<http://oreilly.com/openbook/opensources/book/perens.html>〉; 문화체육관광부 · 한국저작권위원회, 소프트웨어 관리 가이드, 2013, 27면; 〈www.gnu.org/philosophy/categories.html〉, 검색일: 2013. 11. 20.

로젝트를 지휘하는 경영자의 전문성이 필요한 영역일 수 있지만, 여기서 간과해서는 안 될 것은 라이선싱 계약의 구조와 그 법적 리스크를 이해하고 기업의 가치를 극대화하는 판단을 내릴 줄 아는 법률 전문가의 개입이 필요하다는 점이다. 현실적으로 독점 소프트웨어를 택할 것인지 아니면 공개 소프트웨어를 택할 것인지에 대한 의사결정이 제품개발의 시점에 쫓기고 있는 프로그래머의 단순한 감각이나 관행에 의존하여 결정되거나, 라이선싱 계약의 구조를 정확히 이해하지 못하지만 시장의 흐름과 프로젝트 수행의 경험은 충분히 가지고 있는 경영자의 판단에 의해 ‘과감하게’ 결정되는 경우가 많다는 사실이다. 이에 대해서는 필립스(Philips)에서 근무하던 특허 변호사이자 컨설턴트인 앵겔프리트(Arnoud Engelfriet)가 나름대로의 의사결정 프로세스를 제시한 바 있다.³³⁾ 그는 먼저 무수히 많은 유형의 FOSS를 크게 다음의 세 개의 영역으로 구분한다.

〈표 2〉 FOSS 라이선스의 유형³⁴⁾

라이선스 유형	2차적저작물 (derivative works)	결합저작물 (oeuvre composite)	라이선스 사례
Free-for-all Licenses (academic license)	공개 SW (독점 SW 가능)	공개 SW (독점 SW 가능)	BSD, MIT, Apache
Keep-open Licenses	공개 SW	공개 SW (독점 SW 가능)	LGPL, MPL
Share-alike Licenses (copyleft)	공개 SW	공개 SW	GPL, Open Software License ("OSL")

이러한 분류법은 첫째, 2차적 저작물의 재공개의무가 존재하는지 여부, 둘째, 독점 소프트웨어와 결합하여 만들어지는 결합저작물이 공개 소프트웨어가 되는지 여부를 기준으로 한 것이다. 요약하자면 GPL은 Share-alike, LGPL 및

33) Engelfriet, Arnoud, "The Best of Both Worlds," *Intellectual Asset Management*, Aug/Sep(2006).
 (www.iam-magazine.com), 검색일: 2013. 11. 20.

34) 이 표는 Engelfriet의 상계 논문의 내용을 기초로 필자가 작성하였다.

MPL은 Keep-open, BSD와 Apache는 Free-for-all 유형이라 할 수 있다. 엔젤 프리트가 제안하는 프로세스는 서로 다른 라이선스를 소프트웨어 제품의 각기 다른 특성에 따라 전략적으로 선택할 수 있게 한다. 이를 위하여 그는 다시 제품의 특성을 세 개의 영역으로 구분한다. 그가 정확히 명시하고 있지는 않지만, 이러한 분류는 기본적으로 결합저작물인 소프트웨어의 각 구성부분을 대상으로 한 것으로 이해해야 할 것이다.

〈표 3〉 소프트웨어 제품의 특성에 따른 라이선스 유형³⁵⁾

SW 특성	제품 차별화	소비자 반응	전략 옵션	라이선스
Differentiator	차별화(value added)	구매동기유발	Make	독점 SW
Baseline	무차별	기대	case-by-case	case-by-case
Commodity	무차별	무관심	Open	공개 SW

특히 결합저작물인 소프트웨어에 대해 이러한 특성별 분류를 거친다면, 경영 전략 이론에서 전통적으로 사용하는 “make-or-buy” 의사결정에 “open”을 추가한 형태가 될 것이다. 엔젤프리트 프로세스는 직감에 의존하거나 아무런 기준 없이 결정될 수 있는 FOSS 라이선스의 선택에 나름대로의 기준을 제시한 점에서 의의가 있다. 가령 본고에서 관심을 가지는 IBM의 리눅스를 통한 오픈 이노베이션은 왜 GPL 규정을 따르고 있을까 하는 질문에 대해 의미 있는 답을 주는 것이다. IBM의 하드웨어 제품에 임베디드(embedded) 상태로 결합하는 리눅스의 경우는 위의 분류상 “commodity”에 해당하며, 이는 IBM의 차별화 역량과는 거리가 있는 것으로서 공개전략이 타당하고, 공개 소프트웨어 라이선스(이 경우 GPL)를 채택하는 것이 가장 타당하다는 것이다. 다만 엔젤프리트 프로세스도 아직은 충분히 정교한 전략적 의사결정을 내릴 수 있는 개념 도구는 아니다. 특히 Baseline 소프트웨어의 경우에는 기술적, 경영적, 법적인 고려에 따라 다양한 옵션이 나올 수 있는데, 이에 대한 구체적인 답을 주지는 못하기 때문이다.

35) 이 표는 Engelfriet의 전제 논문의 내용을 기초로 필자가 작성하였다.

3. 공개 소프트웨어의 권리 · 의무 관계

FOSS 라이선스는 라이선시(Licensee)에게 FOSS의 소스코드를 자유롭게 사용, 복제, 수정할 수 있도록 하며, 일정한 조건하에서 복제, 수정된 내용을 재배포할 수 있도록 한다. 다른 한편 FOSS 라이선스는 사용자가 지켜야 할 일정한 의무를 부가하는데, 이는 FOSS와 함께 배포되는 라이선스의 소스코드나 관련 웹사이트 등에 그 내용이 명시되어 있다.³⁶⁾ FOSS 라이선스에서 요구하고 있는 준수사항을 그 이용자가 이행하지 않는 경우에는 권리자로부터 계약 위반이나 저작권 위반으로 소송을 당하거나 해당 소프트웨어의 배포가 금지될 수도 있다. 따라서 FOSS를 사용하고자 하는 업체는 반드시 사전에 라이선서(Licenser)의 요구 사항을 확인해야 한다. OSI는 공개 소프트웨어의 개념을 10가지 조항으로 정의하였는데, 이러한 정의는 FOSS의 전반적인 내용 파악에 도움을 준다.³⁷⁾ 본 고에서 오픈소스 소프트웨어를 FOSS라 칭하고 있으나, 여기서는 OSI의 원문을 살리기 위해 OSS로 칭한다.

1) 자유로운 재배포(Free Redistribution)

OSS 라이선스는 ‘집합 소프트웨어(aggregate software)’의 구성부분(component)으로서 해당 소프트웨어를 판매 혹은 무상 배포하는 것을 제한할 수 없다. 서로 다른 소스로부터 제작된 프로그램들의 집합도 이러한 집합 소프트웨어의 개념에 포함된다. OSS 라이선스는 이러한 판매에 대해 로열티나 다른 수수료를 부과할 수 없다. 이처럼 OSS 라이선스가 무상의 재배포를 요구하도록 강제하는 것은 협력자들이 단기이익(short-term gains)의 유혹 때문에 장기이익(long-term gains)을 포기하는 것을 막기 위해서이다.

2) 소스코드의 공개(Source Code Open)

36) 소스코드에서는 주로 최상위 디렉토리에 ‘COPYING’이라는 독립된 파일에 라이선스 조항을 기재하기도 하며, 각각의 소스코드 파일 상단에 명시해 두기도 한다. 정보통신부, 전게서, 12면.

37) <<http://opensource.org/osd-annotated>>, 검색일: 2013. 11. 20. 참조. 여기서는 OSI가 제시한 10개 OSS 개념정의 항목에 대한 설명과 주석을 중심으로 FOSS 라이선스의 개념을 분석해 보고자 한다.

프로그램은 소스코드를 포함해야 하는데, 컴파일된 형식뿐 아니라 소스코드 형식으로도 배포가 허용되어야 한다. 제품이 소스코드와 함께 배포되지 않는 경우에는 합리적인 복제비용 이하로 소스코드를 얻을 수 있는 널리 알려진 방법이 있어야 하는데, 가장 선호되는 것은 인터넷에서 무료로 다운로드하는 방식이다. 소스코드는 프로그래머가 쉽게 수정할 수 있는 형식이어야 하며, 의도적으로 난해하게 만드는 것은 허용되지 않는다. 전처리기(pre-processor)³⁸⁾나 번역기(translator)³⁹⁾의 산출물 같은 중간형태도 허용되지 않는다.

3) 2차적 저작물(Derived Works)

OSS 라이선스는 수정 및 2차적 저작물을 허용해야 하며, 원 소프트웨어와 동일한 라이선싱 조건으로 배포되는 것을 허용해야 한다.

4) 저자의 소스코드의 완전성(Integrity of the Author's Source Code)

OSS 라이선스는 소스코드와 함께 패치파일(patch file)의 배포를 허용하는 경우에는 소스코드가 수정된 형식으로 배포되는 것을 제한할 수 있다. 이렇게 하면 빌드타임에 패치파일을 이용해 프로그램을 수정할 수 있기 때문이다. 이런 경우가 아니라면 OSS 라이선스는 수정된 소스코드로 제작된 소프트웨어의 배포를 명시적으로 허용해야 한다. OSS 라이선스는 2차적 저작물(소프트웨어)에 원래 소프트웨어와 다른 이름이나 버전 번호를 붙이도록 요구할 수 있다. 이는 원래 소프트웨어의 저자인 프로그래머의 저작권권을 보호하는 취지로 이해된다.

5) 개인 또는 단체에 대한 차별금지(No Discrimination against Persons or Groups)

38) 전처리기(pre-processor)란 프로그래밍 언어에 유용한 기능들을 추가하여 언어를 확장시키는 지시문을 말하는데, C언어에서 '#define', '#include' 같은 지시문들이 대표적인 예이다.

39) 번역기(translator)란 인간이 이해할 수 있도록 자연어에 가까운 고급 프로그래밍 언어로 작성된 소스코드(원시코드) 문서를 기계가 이해할 수 있는 기계어(machine language) 또는 어셈블리어(assembly language)로 작성된 목적코드(object code) 문서로 변환시키는 프로그램이다.

OSS 라이선스는 어떤 개인이나 단체도 차별해서는 안 된다.

6) 활용분야에 대한 차별금지(No Discrimination against Fields of Endeavor)

OSS 라이선스는 특정 분야에 프로그램이 사용되는 것을 제한해서는 안 된다. 가령 프로그램이 사업에 활용되거나 유전자 연구에 활용되는 것을 제한할 수 없는 것이다.

7) 라이선스의 배포(Distribution of License)

프로그램을 재배포 받은 당사자들은 추가적인 라이선스의 획득 없이도 프로그램에 설정된 모든 권리를 향유해야 한다.

8) 라이선스가 한 제품에 특정되지 않아야 함(License Must Not Be Specific to a Product)

프로그램에 설정된 권리들은 특정 소프트웨어 배포의 한 구성부분이라는 점에 의존해서는 안 된다. 만일 프로그램이 이러한 배포로부터 도출되어 프로그램 라이선스의 범위 내에서 사용 또는 배포된다면, 그 프로그램을 재배포 받는 당사자들은 원래 소프트웨어 배포와 관련하여 부여된 것과 동일한 권리들을 가져야 한다.

9) 라이선스가 다른 소프트웨어를 제한하지 않아야 함(License Must Not Restrict Other Software)

OSS 라이선스는 해당 소프트웨어와 함께 배포되는 다른 소프트웨어에 제한을 가해서는 안 된다. 가령, 동일한 매체로 배포되는 다른 모든 프로그램들도 OSS가 되어야만 한다고 주장할 수 없다.

10) 라이선스가 기술중립적이어야 함(License Must Be Technology-Neutral)

OSS 라이선스의 어느 조항도 특정한 기술이나 인터페이스 스타일을 기반으로 할 수 없다. 이는 라이선싱 계약을 체결하는 데 어떤 명시적인 합의방식을 요구하는 것이 OSS 라이선스의 개념에 적합하지 않을 수 있음을 뜻한다. 가령, 클릭랩(click-wrap)을 강제하는 조항을 두게 되면 이 방식을 따르지 않는 FTP 다운로드, CD-ROM 선집, 웹 미러링(web mirroring) 등 다른 중요한 소프트웨어 배포방식들로는 프로그램을 배포할 수 없게 되기 때문이다.

4. 공개 소프트웨어의 장단점

공개 소프트웨어는 여러 관점에서 전략적인 활용가치가 뛰어나다.⁴⁰⁾ 소프트웨어 개발업체의 입장에서는 공개된 소스코드에 접근하여 수정·재배포가 가능하기 때문에 초기 개발비용이 적게 들고, 빠르고 유연한 개발이 가능하다는 점이 가장 강점이라고 생각할 것이다. 공개 소프트웨어는 또한 전 세계에 흩어져 있는 뛰어난 프로그래머들이 직접 개발과 디버깅에 참여하기 때문에 폐쇄적 방식으로 개발되는 독점 소프트웨어에 비하여 품질이 우수하고 더 안정적으로 동작할 수 있다. 공개 소프트웨어는 서로 다른 소프트웨어 간의 상호 연동성이 뛰어나며, 소위 벤더종속(vendor lock-in)⁴¹⁾이 발생하지 않는 장점이 있다. 나아가 공개 소프트웨어가 대부분의 독점 소프트웨어와의 호환에도 문제될 것은 없기 때문에 다양한 비즈니스 모델의 설계가 가능할 수 있다. 한편 공개 소프트웨어는 중대한 리스크도 내포하고 있다. 특정 기업이 보유한 독점 소프트웨어와

40) Stern, Alan & A. Clifford Allen, "Open Source Licensing", *Patents, Copyrights, Trademarks, and Literary Property, Course Handbook Series, PLI/Pat*, No.985(2009), Practising Law Institute, pp.335-338.

41) 이는 독점 소프트웨어 소비자들이 기존 소프트웨어의 품질에 만족하지 못하여 새로운 소프트웨어로 전환하기 원하지만 추가로 지출할 비용 때문에 실행에 옮기지 못하고 기존 업체에 구속되는 현상을 말한다. 공개 소프트웨어는 제품 자체보다는 서비스 제공을 통한 수익모델이 주를 이루고 있다. 따라서 사용자들은 어느 업체의 서비스에 만족하지 못할 경우 별다른 전환비용(switch cost) 없이 새로운 서비스 공급업체를 선택할 수 있다.

공개 소프트웨어와 결합함으로써 개발비용과 시간을 절감하기 원할 경우에, 그 FOSS 라이선스의 유형에 따라서는 제품 전체가 공개 소프트웨어가 될 가능성이 있다. 이는 특히 GPL 코드를 활용하는 업체의 경우 경험할 수 있는 문제인데, 전체 FOSS 라이선스 중 가장 큰 비중을 차지하고 있는 것이 GPL임을 감안할 때 상당한 법적 리스크라고 할 것이다.⁴²⁾ 이 외에도 특정 기업이 보유한 소프트웨어 특허권이 FOSS에 포함되는 경우에, 대부분의 공개 소프트웨어 규약은 특허권에 대한 로열티 없이 배포할 것을 요구하고 있는 점도 주의해야 한다.

5. 공개 소프트웨어 산업

글로벌 IT 시장조사 기관인 IDC에 따르면, 2013년 상반기 세계 소프트웨어 시장규모가 1,790억불 규모에 달했으며,⁴³⁾ 2017년까지 빅데이터(big data) 및 분석 소프트웨어, 다양한 기업용 어플리케이션의 수요 증가로 연평균 6% 성장할 것으로 전망하였다.⁴⁴⁾ 본고에서 자주 거론되는 리눅스의 경우 세계 시장에서 이 OS를 탑재한 서버의 신규 판매는 2014년까지 꾸준히 증가할 것으로 예상된다.⁴⁵⁾ FOSS는 국내외 업계에 많이 도입되었지만, 그에 대한 비즈니스 모델은 상대적으로 제한적이어서 다양하게 개발되지 못하였다. 2011년 국내 공개 소프트웨어 시장 규모는 243억 원 규모로 매우 작은 편이다.⁴⁶⁾ 그러나 소프트웨어 시장 전체 규모가 11조 원 정도로 작지만 소프트웨어가 융복합을 통해 산업고도화에 기여하는 바가 크므로 국가적으로 중요한 산업으로 인식되는 것처럼,⁴⁷⁾ 공개 소프트웨어도 시장규모와 상관없이 기업의 비용절감과 제품개발기간 단축에

42) 이런 관점에서 GPL과 같은 라이선스를 전염성(viral) 라이선스라 하거나 오염성(contaminating) 라이선스라 부르기도 한다. 즉 GPL과 결합하는 어떤 독점 소프트웨어도 GPL 코드에 감염이 되어 FOSS가 되어버리는 것이다.

43) <<http://www.idc.com/getdoc.jsp?containerId=prUS24418213>>, 검색일: 2013. 11. 20.

44) <<http://www.idckorea.com/product/Getdoc.asp?idx=581&field=PressRelease>>, 검색일: 2013. 12. 6.

45) 정보통신산업진흥원, 공개소프트웨어 백서(Open Source Software white Paper korea), 2012, 17면. 리눅스 서버가 점차 활용 증가 추세를 보이겠으나 유상 구매하는 리눅스 비중 감소로 전체 리눅스 매출은 감소할 수 가능성도 있다. 이는 무상으로 탑재하는 리눅스OS 비중이 적지 않기 때문이기도 하다.

46) 상계서, 21면.

47) 상계서, 24면.

기여하는 바가 크기 때문에 무시할 수는 없다.⁴⁸⁾ 본고에서 다루어질 IBM의 사례와 같이 OS 등의 소프트웨어를 하드웨어에 설치해 판매하는 소위 임베디드(embedded) 소프트웨어의 경우 하드웨어 가격에 소프트웨어 가격이 포함되어 있는 경우가 많아 시장규모를 정확히 파악하기 어렵지만, 이러한 제품에 대한 수요가 큰 만큼 공개 소프트웨어의 비중은 높다는 것은 분명하다. 구글, 아마존, 페이스북, 트위터 등 소셜네트워크서비스(SNS) 기업들도 공개 소프트웨어를 활용하여 기업의 가치를 높이고 있다. 공개 소프트웨어 개발공급 업체의 확대와 국내 소프트웨어 산업 생태계에 수준 높은 개발자들의 커뮤니티가 활성화되는 것은 산업 활성화를 위해 매우 중요한 요소라 하겠다.

III. IBM의 오픈 이노베이션 전략

1. IBM의 지식재산 경영전략

IBM의 전신은 1911년 미국 뉴욕 주에 설립된 “Computing-Tabulating-Recording Company(CTR)”로서, 19세기 말부터 시작된 몇 개의 사업을 통합한 것이었다.⁴⁹⁾ “International Business Machines(IBM)”이라는 이름을 쓰기 시작한 것은 1917년 캐나다 시장에 진출한 때부터이며, 미국에서 회사명 변경은 1924년에 이루어졌다.⁵⁰⁾ IBM은 1950년대에는 핵심역량을 가진 대형컴퓨터 시장의 강세로 고속 성장을 지속하다가, 1980년대 이후 PC와 워크스테이션 등 새로운 형태의 컴퓨터 시장의 비중이 커지자 어려움을 겪었다. 그러나 1993년 루이스 거스너(Louis Gerstner)가 CEO에 취임한 이후,⁵¹⁾ IBM은 특허권을 비롯한 지식재산 포트폴리오를 전략적으로 관리하고 사업화하여 경영상황을 크게 개선

48) Engelfriet, Arnoud “The Best of Both Worlds,” *Intellectual Asset Management*, 2006. <www.iam-magazine.com>, 검색일: 2013. 11. 20.

49) <www-03.ibm.com/ibm/history/history/decade_1880.html>, 검색일: 2013. 10. 1.

50) <www-03.ibm.com/ibm/history/history/year_1924.html>, 검색일: 2013. 10. 2.

51) <www-03.ibm.com/ibm/history/history/year_1993.html>, 검색일: 2013. 10. 2.

할 수 있었다. IBM의 로열티 수입이 연간 약 10억 달러에 달하는 것은 지식재산 포트폴리오 전략의 성공을 대변해 준다.⁵²⁾ 자사의 핵심기술과 1등급 기술을 독자적으로 사업화하여 직접 수익을 창출하고, 주변기술과 2~3 등급 기술은 라이선싱(이른바 “licensing-out”)하여 불필요한 비용을 줄이고 로열티를 확보하는 것은 포트폴리오 관리의 기본이다. 아울러 IBM은 고도한 지식재산 경영전략을 다양하게 구사하여 수익을 창출하고 있는데,⁵³⁾ 그 중에서 가장 혁신적인 것은 공개 소프트웨어를 적극 활용한 오픈 이노베이션 전략이다. 이는 또한 공개 소프트웨어가 영리추구를 기본철학으로 하는 기업들에게 어떻게 활용될 수 있는지를 보여주는 좋은 사례이기도 하다.

2. IBM의 공개 소프트웨어 지원정책

1990년대 후반 이후 IBM은 막대한 비용과 인력을 투입하여 리눅스,⁵⁴⁾ 이클립스(Eclipse), 아파치(Apache) 등 공개 소프트웨어 개발 커뮤니티를 지속적으로 후원하여 왔다. IBM이 2005년 1월에 500건의 소프트웨어 특허를 무상으로 공개한 것도 이러한 전략의 일환이다.⁵⁵⁾ 소프트웨어 산업에서 IBM의 전략은 지식재산권 포트폴리오를 통한 강력한 독점배타적 지위의 확보와 공개 소프트웨어 커뮤니티에 대한 지원 및 협력을 동시에 추구하는 것이 큰 특징이다. IBM이 결코 지식재산권에 기초한 시장지배자의 지위를 놓치고자 하지 않는 것은 2012

52) <www.ibm.com/ibm/gjo/us/en/world.harvestmoon.html>, 검색일: 2013. 11. 20.

53) IBM은 핵심이 아닌 주변기술에 대해서는 소위 울타리치기(picket fencing) 전략의 일환으로 특허를 출원하지 않고 공공영역(public domain)에 둬으로써 경쟁업체의 특허 등록 의지를 꺾거나, 적극적으로 크로스 라이선싱(cross licensing)을 유도하여 분쟁의 소지를 차단하거나, 직무발명에 대한 다양한 보상 패키지를 개발하여 특허 포트폴리오를 강화하는 등 다양한 전략을 구사해 왔다. IBM의 이러한 지식재산 경영전략의 유형은 다음 웹사이트 기사를 참조하라.

Melba Kurman, "What Universities Can Learn from IBM's IP Licensing Strategies", 2011, <www.innovationexcellence.com/blog/2011/07/09/what-universities-can-learn-from-ibm%E2%80%99s-ip-licensing-strategies/#sthash.QusT5cGu.dpuf>, 검색일: 2013. 11. 20.

54) IBM이 리눅스에 투자하게 된 것은 이 OS가 작은 서버에서 잘 동작하고, 확장성이 좋으며, 고객 업체들의 소프트웨어 비용을 획기적으로 절감해 주기 때문이었다. IBM은 자체적인 OS 개발에 투자할 자원을 서비스와 솔루션 역량 차별화에 투자하여 오픈 이노베이션을 통한 이종의 상승효과를 누리고 있다.

55) <www.nytimes.com/2005/01/11/technology/11soft.html?_r=0>, 검색일: 2013. 11. 20.

년까지 IBM이 20년 연속 미국 특허출원 1위 기업이었던 것만 보아도 알 수 있다.⁵⁶⁾ 막강한 독점적 지식재산권 포트폴리오를 적극 활용하던 IBM이 공개 소프트웨어를 자사 하드웨어에 결합하는 혁신적 정책을 채택함으로써 회사의 조직과 문화에도 많은 변화를 가져왔다. 특히 지식재산권의 독점적인 소유를 통해 이윤을 추구하던 전략을 품질 향상과 지속 성장을 위한 전략으로 전환하기 시작한 것이다.⁵⁷⁾ IBM이 독점적으로 보유하던 노하우가 공개 소프트웨어 생태계에 공유되어 거대한 지식생태계(ecosystem)가 형성되었다. 스톨만은 한 언론 인터뷰에서 리눅스 커뮤니티를 지원하는 IBM은 “더 이상 100% 독점 소프트웨어 회사가 아니다(not a 100 percent proprietary software company anymore).”라고 언급하여 IBM의 기업문화가 변하고 있음을 암시하기도 하였다.⁵⁸⁾ 현재 IBM 홈페이지에 Developer Works⁵⁹⁾와 Alpha Works⁶⁰⁾가 운영되고 있는 것도 독점이윤 창출에 주력하던 IBM의 전통적인 기업문화가 오픈 이노베이션을 통해 공유와 협력으로 변해 가는 것을 알 수 있다.

IV. SCO v. IBM 소송

1. 유닉스(UNIX)와 리눅스(GNU/Linux)

56) <www.bloomberg.com/news/2013-01-10/ibm-granted-most-u-s-patents-for-20th-straight-year.html>, 검색일: 2013. 11. 20. IBM의 2012년 연차보고서에 따르면 2012년 한 해 동안 빅데이터(big data), 사이버보안(cyber security), 클라우드(cloud), 모바일(mobile) 등 첨단 분야를 중심으로 미국 특허를 등록했는데, 등록건수가 총 6,478건이었다. <www.ibm.com/annualreport/2012>, 검색일: 2013. 11. 20. 이는 액센츄어, 아마존, 애플, EMC, HP, 인텔, 오라클, 시만텍 등 8개 기업의 특허건수를 합친 것보다 많은 숫자이다.

57) 최근에는 IBM 외에도 Computer Associate, Ericsson, Nokia, RedHat 등 공개 소프트웨어의 개발과 이용에 적극적으로 참여하고 있는 기업들이 자사가 가진 특허를 오픈소스 공동체에 기부하거나 자유롭게 사용할 수 있도록 하겠다는 선언을 계속하고 있다. 이철남, “오픈소스 소프트웨어에 대한 소프트웨어 특허의 법적함의”, 경영법률, 제17권 제1-2호(2006), 한국경영법률학회, 933-934면.

58) <www.linux-mag.com/id/255>, 검색일: 2013. 11. 20.

59) 개발자들이 겪는 문제점을 서로 토론하고 연구하는 사이트로 <www.ibm.com/developerworks>.

60) 소위 “얼리 어답터(early adopter)”들이 최근 부각되고 있는 기술들을 체험하고 연구할 수 있는 사이트로 <www.alphaworks.ibm.com>.

유닉스는 AT&T Bell Laboratories(“AT&T”)가 처음 개발했는데, AT&T는 이를 노벨(Novell)에게 매각했으며, 그 후 노벨은 이를 다시 SCO에 매각하였다.⁶¹⁾ 한편 리눅스는 유닉스와는 전혀 별개의 코드에 기반을 둔 유닉스의 파생 운영체제이다. 현재 리눅스는 모든 주요 서버 업체와 대다수 소프트웨어 업체에게 지지를 받고 있다.⁶²⁾ 리눅스는 한 회사가 소유하거나 컨트롤할 수 없는 공개 플랫폼이다. 따라서 리눅스 사용자들은 스스로 제품 출시나 발표 스케줄을 결정하는 등 플랫폼 발전에 적극적으로 참여할 수 있었다.

2. SCO의 주장

SCO v. IBM 소송은 2003년 SCO의 제소 이후 2007년 SCO의 파산⁶³⁾으로 소송절차가 정지될 때까지 치열한 공방을 펼치며 진행되었고, 그 후 6년 만인 2013년 절차가 다시 재개되었다. 그록로(Groklaw)라는 웹로그(weblog)가 이 소송에 대한 방대한 기록을 빠짐없이 수집하여 공개하고 있다.⁶⁴⁾ 본고에서 참고하는 소송기록들은 대부분 그록로 사이트에서 원문을 확인할 수 있다.⁶⁵⁾ SCO는 2003년 3월 6일 유타(Utah)주 솔트레이크(Salt Lake)시 제3지방법원⁶⁶⁾에서

61) SCO는 미국 캘리포니아 주 산타크루즈에 본사를 둔 회사인 Santa Cruz Operation의 약어로 쓰였었다. 2001년 SCO는 칼데라(Caldera Systems)에 유닉스 및 관련 사업을 매각하였다. 칼데라는 이후 사명이 Caldera International이 되었다가 다시 SCO로 바뀌었으며, 그 후에는 현재와 같은 “SCO Group”이 되었다. 본고에서는 현재 이 회사의 명칭을 간단히 SCO로 칭하기로 한다.

62) SCO가 IBM에게 소송을 제기했던 2003년 당시 IDC 분석에 따르면 2002년 출하된 서버 중 13%가 리눅스를 운영체제로 탑재했으며, 2007년에는 25%를 상회하고 금액 기준으로는 15%를 차지할 것으로 예측되고 있었다.

63) 미국 연방파산법 chapter 11에 의한 “reorganization”이었다.

64) (www.groklaw.net) 참조. 이 사이트의 운영자는 paralegal로 일한 경력이 있는 Pamela Jones(“PJ”로 약칭)인데, 그 글의 논조가 IBM과 공개 소프트웨어 커뮤니티를 대변하며 SCO를 비판하는 것이라고 느낀 SCO는 SCO v. Novell 소송 중에 PJ에게 선서증언(deposition)을 받고자 시도하기도 하였다. 현재 Groklaw는 PJ가 사용하던 이메일 업체인 Lavabit이 2013년 8월 폐쇄되면서 업데이트가 중단된 상태이다. (www.forbes.com/2007/02/13/groklaw-sco-ibm-tech-enter-cz_dl_0213sco.html), 검색일: 2013. 11. 20.

65) SCO v. IBM 소송 관련 보도 자료들은 (www.fact-index.com)의 “SCO v. IBM” 및 “SCO v. IBM Linux lawsuit: Press coverage”에 정리되어 있다. 검색일: 2013. 11. 21.

66) SCO가 2003년 Utah州 제3지방법원에 제기한 소송은 이후 연방지방법원으로 이송(removal)되었다.

IBM에 대해 영업비밀(trade secret)의 침해, 불공정 경쟁, 계약 불이행, 영업 방해 등을 이유로 10억 달러의 손해배상을 청구했는데,⁶⁷⁾ SCO의 이러한 주장은 소송절차가 진행되면서 계속 변경되었으며, SCO의 주장은 2004년 2월 제출된 “제2차 수정訴狀(second amended complaint)”에 요약되어 있다.⁶⁸⁾ SCO의 주장에 따르면 IBM이 ① AT&T와의 사이에 체결한 “UNIX System V와 그 개량 소프트웨어”의 소스코드에 대한 소프트웨어 계약인 “SOFT-00015”를 위반하여 제3자에게 해당 소스코드를 공개, 배포, 사용하게 하는 등 계약을 불이행하였고, ② IBM이 AT&T와의 사이에 체결한 서버 라이선싱 계약인 “SUB-00015A”를 위반하여 “UNIX System V와 그 개량 소프트웨어” 목적코드(object code)를 배포함으로써 계약을 불이행하였으며, ③ 시퀀트(Sequent Computer Systems, Inc.)가 AT&T와 체결한 소프트웨어 계약인 “SOFT-000321”를 위반하여 제3자에게 해당 소스코드를 공개, 배포, 사용하게 하는 등 계약을 불이행하였고, ④ 시퀀트가 AT&T와 체결한 서버 라이선싱 계약인 “SUB-000321A”를 위반하여 제3자에게 해당 소스코드를 공개, 배포, 사용하게 하는 등 계약을 불이행하였다. 또한 SCO는 IBM에게 ⑤ 저작권 침해, ⑥ 부정경쟁(unfair competition), ⑦ 세계의 소비자에 대한 계약 방해, ⑧ 노벨(Novell)과 SCO 사이의 자산매입계약(Asset Purchase Agreement)에 대한 방해, ⑨ 기타 사업관계에 대한 방해 등을 하였다고 주장하였다.

Utah주는 州 지방법원이 District Court라고 불리고, 연방지방법원도 US District Court라 불리므로 혼동되는 경우가 많은데, SCO v. IBM 소송이 시작된 제3지방법원과 연방지방법원이 모두 Salt Lake City에 있어서 더욱 혼동될 수 있다. <www.linux.org/news/sco/timeline.html>, 검색일: 2013. 11. 21.

67) SCO는 최초로 제출한 소장에서 다음과 같이 주장하였다: 유닉스는 컴퓨터 OS 및 이와 관련된 소프트웨어로서 AT&T Bell Laboratories(“AT&T”)가 처음 개발한 것이었다. SCO는 유닉스를 AT&T로부터 인수하고 이후 수억 불을 투자하여 자신의 독자적인 개량 버전인 SCO/UNIX도 개발하여 보유하고 있었다. SCO는 유닉스와 SCO/UNIX를 제한적인 라이선싱 조건으로 소프트웨어 업체인 IBM과 최종 소비자인 맥도널드(McDonald’s)를 포함한 많은 업체들에게 라이선싱하였다. 그런데 IBM은 SCO에 대한 계약을 불이행하고, 제3자가 SCO에 대한 계약을 불이행하도록 유도(induce)하고 조장(encourage)했으며, SCO의 영업을 방해(interfere)하였고, SCO와의 부정경쟁(unfair competition)에 개입되었다. 특히 IBM은 SCO의 독점적 소프트웨어를 남용(misusing) 및 유용(misappropriating)하고 제3자의 남용이나 유용을 유도, 조장, 지원(enabling)하였으며, SCO의 독점적 소프트웨어를 공개소프트웨어에 포함시켜 제공하거나 제3자가 제공하도록 유도, 조장, 지원하였다.

68) SCO v. IBM, Second Amended Complaint, Case No. 03-CV-0294, The US District Court for the District of Utah, 2004. 2. 27.

3. IBM의 법적 대응

IBM은 SCO에 대한 2003년 8월 반소장(counterclaim)⁶⁹⁾에서 SCO가 GPL 규정에 따라 리눅스 커널 2.4 버전을 배포한 적이 있으므로 그에 대한 독점권을 행사할 수 없다고 주장하였다. 또한 SCO의 소프트웨어가 IBM의 4개 특허를 침해하였고, IBM의 유닉스인 AIX의 판매권을 침해하여 IBM의 영업을 방해하였다고 주장하였다. 현재 IBM의 주장은 2004년 3월 제출된 “제2차 수정反訴狀(the second amended counterclaim)”에 요약되어 있다.⁷⁰⁾

IBM에 따르면 ① IBM은 UNIX System V와 관련한 “SOFT-00015”, “SUB-00015A”, “SOFT-000321”, “SUB-000321A” 등 모든 계약에 대해 정당한 권리자인데, SCO가 IBM의 계약상 권리를 부정함으로써 자신의 계약상 의무를 불이행하였고, ② SCO가 IBM의 AIX, Dynix, 기타 리눅스 관련제품들의 법적 정당성에 대해 공개적으로 허위진술을 함으로써 연방상표법(Lanham Act)을 침해하였으며, ③ SCO가 고의로, 인식하면서, 허위로, 그리고 악의(bad faith)로 IBM의 AIX, Dynix, 기타 리눅스 관련제품들의 가치를 박탈한 것은 부정경쟁(unfair competition)에 해당하고, ④ SCO가 고의로 IBM의 AIX, Dynix, 기타 리눅스 관련제품들의 잠재적인 경제적 관계형성을 방해하였으며, ⑤ SCO가 IBM의 권리에 대한 허위표시를 통해 불공정하고 기망적인 거래를 실행하였다. 본고의 논의와 관련하여 가장 의미있는 IBM의 주장은 ⑥ SCO가 자신의 리눅스 소스코드를 GPL 규정대로 배포하고서도 이에 반하는 방식으로 복제, 수정, 라이선싱, 배포함으로써 GPL 규정을 위반했고, ⑦ GPL 규정을 준수한다는 SCO의 약속을 신뢰하고 중대한 사업상 결정을 많이 내린 IBM에 대해 SCO가 그 약속에 반하는 행동을 하는 것은 금반언원칙(promissory estoppel)⁷¹⁾에 반하며,

69) SCO v. IBM, Plaintiff IBM's Counterclaims against SCO, Case No. 03-CV-0294, The US District Court for the District of Utah, 2003. 8. 6.

70) SCO v. IBM, Plaintiff IBM's Second Amended Counterclaims against SCO, Case No. 03-CV-0294, The US District Court for the District of Utah, 2004. 3. 29.

71) 당사자가 특정한 의사표시를 한 이상 그 의사표시를 부정하는 주장을 할 수 없다는 “형평의 원칙(equitable principle)”의 하나이다. 보통법(Common Law)에서 계약 이행을 강제하려면 원칙적으로 계약의 성립(formation)이 필요하고, 이를 위해서는 유효한 청약(offer)과 승낙(acceptance) 외에도 그 둘

⑧ IBM이 등록한 리눅스 관련 저작권들을 SCO가 사용, 수정, 라이선싱, 배포하는 것은 GPL을 따르지 않는 한 저작권 침해를 구성한다는 것이다. IBM은 이외에도 자신이 SCO의 저작권들을 침해하지 않는다는 것과, SCO의 주장들이 근거가 없고 오히려 IBM이 주장하는 권리들을 SCO가 침해하였다는 사실에 대한 확인판결(declaratory judgment)을 구했으며, 또한 SCO가 세 개의 IBM특허를 침해하였다고 주장하였다.

4. 소송의 경과

SCO가 FOSS 라이선스로 배포된 리눅스를 근거로 소송을 제기한 것은 IBM 한 기업에 대한 것이 아니라 사실상 리눅스를 OS로 채택한 세계 모든 기업에 대한 위협이었다.⁷²⁾ 실제로 SCO는 IBM에 이어 노벨, 레드햇, 오토존 및 다임러 크라이슬러 등 리눅스 진영의 대표 기업들과 소송을 수행하였다. 그러던 중 2007년 9월 14일에 SCO는 델라웨어 주에서 연방파산법(Bankruptcy Code) 제 11장에 따른 파산보호를 신청해야 하는 상황이 되었다. 이는 단기간에 소송전을 승리로 이끌지 못하면서도 무리하게 여러 거대 기업들을 상대로 소송을 계속 진행한 결과였다. 이로 인해 약 3년 반 동안 진행된 SCO와 IBM 소송은 절차적으로는 소송의 정지 상태로 약 6년을 휴전 상태로 보냈다. 2013년 3월에 SCO는 유타 연방지방법원에 IBM과의 절차의 재개를 신청하였고, 2013년 6월 너퍼(David O. Nuffer) 판사의 결정으로 결국 소송절차가 재개되었다. 표면적으로는 SCO의 청구를 받아들여 준 것으로 보이지만, 법원은 사실 IBM이 제시한 가이드라인을 따르고 있는 상황이어서 SCO에 유리한 방향으로 소송이 전개되지 않으리라는 분석이 많다.⁷³⁾

사이에 "대가관계(bargained-for exchange 또는 consideration)"가 성립해야 한다. 그러나 금반언의 원칙이 적용된다면 consideration의 존재를 입증하지 못하더라도 법원의 판단에 의해 형평의 관점에서 필요한 한도 내에서 약속을 이행할 것을 명할 수 있다.

72) Potenza, Joseph M., & Steve Chang, "the SCO Group, Linux, and Life after SCO's Setback against Novell," *Intellectual Property & Technology Law Journal*, Vol.20 No.1(2008). SCO는 리눅스를 채택하고 있는 1,500여 기업에게 경고장(warning letter)을 발송하여 업계에 파문을 일으킴으로써 법정 외에서도 IBM을 압박하였다.

5. SCO v. Novell 소송

IBM과의 소송절차가 정지된 동안에도 SCO는 노벨(Novell)을 비롯한 리눅스 진영과 계속되는 소송전을 벌였다. 2004년 1월 20일 SCO는 유타주 제3지방법원에 노벨이 SCO에 유닉스 시스템을 매각할 때 유닉스 OS의 모든 저작권도 함께 매각하였다는 주장을 골자로 하는 “소유권 명예훼손(slander of title)”의 소를 제기하였다.⁷⁴⁾ SCO의 주장은 ① 1995년 9월 19일에 체결된 자산구매계약(Asset Purchase Agreement)을 통해 노벨은 당시 1억 달러 이상의 가치가 있었던 SCO 보통주 6,100,000주를 취득했고, SCO는 그 대가로 UNIX와 UnixWare 사업, OS, 소스코드, 기타 모든 저작권 및 사업과 관련된 모든 재산적 권리 일체를 취득했는데, ② 노벨은 악의(bad faith)로 UNIX와 UnixWare에 대한 저작권 행사를 방해하고 기자회견을 통해 공개적으로 이들 소프트웨어 저작권을 자신이 보유하고 있다고 주장하여 SCO의 저작권에 대한 명예훼손을 하였다는 것이다. SCO는 법원이 예비적 및 영구적인 금지명령을 통해 노벨이 위 자산구매계약 이후 UNIX와 UnixWare에 대해 등록한 모든 저작권을 SCO에 이전하고, 이들에 대한 저작권 주장을 금지하며, 그러한 저작권 주장이 반영된 출판물을 회수할 것과, 노벨의 명예훼손으로 인한 실질손해 및 특별손해에 대한 배상과 징벌적 손해배상을 구하였다.⁷⁵⁾

2010년 3월 30일에 유타연방지방법원(The US District Court for the District of Utah)은 노벨이 소유권에 대한 명예훼손을 한 바가 없으며, 소유하고 있는 저작권을 이전할 필요도 없고, 따라서 자신이 저작권자라는 사실에 기초하여 IBM에 허락한 저작권 포기(waiver)를 인정하는 등 모든 쟁점에 대해 노벨의 주장을 수용하였다. 2011년 8월 30일 제10연방순회항소법원(The US Court of

73) <<http://arstechnica.com/tech-policy/2013/06/its-back--district-court-judge-revives-sco-v-ibm>>, 검색일: 2013. 11. 20.

74) “slander of title”은 원래 부동산의 소유자가 아닌 자가 마치 소유자인 것처럼 문서상 주장하는 것을 말하며, 이는 보통법상 “고의의 불법행위(intentional torts)” 유형에 해당한다. SCO는 이 소인(訴因)을 저작권 소유에 대한 허위주장에 적용한 것이다. 다만 “slander”는 보통법상 구두에 의한 명예훼손의 뜻으로 사용되는 것이 보통인데, 이 경우에는 출판물에 의한 명예훼손도 포함한다는 점을 주의해야 한다.

75) SCO v. Novell, Complaint, Civil No. 040900936, 2004. 1. 20.

Appeals for the Tenth Circuit)은 연방지방법원의 결정을 확정하였다.⁷⁶⁾ 이러한 판결이 IBM 및 그와 협력하고 있는 리눅스 커뮤니티에게 유리한 이유는 SCO의 소송상의 주장들이 이 판결로 인하여 사실상 그 근거를 대부분 상실하였기 때문이다.

6. SCO v. IBM 소송의 시사점

1) 지식재산 포트폴리오 전략의 중요성

IBM과 같이 지식재산권 침해주장을 당한 기업이 방어적인 자세로만 일관할 것인지, 아니면 공세적으로 대응할 수 것인지를 판단하는 것이 쉬운 일은 아니다. 관련 업계에서는 처음 SCO가 소를 제기한 의도가 IBM을 협상 테이블로 이끌어 내서 합의금을 받는 데 있다고 보았다. 그러나 IBM은 공격적으로 대응하여 사실상 SCO에 대해 완승을 거두고 있다. IBM이 강한 반격을 할 수 있었던 이유는 여러 가지이겠지만, 그 동안 꾸준히 축적한 지식재산권 포트폴리오가 기반이 되었음은 분명하다. SCO도 IBM이 보유하고 있는 지식재산권을 회피해서 사업을 수행할 수 없었기 때문에, 소송 전략적으로도 충분하게 IBM을 공략하기에는 역부족이었다. 공유와 협력이 강조되는 FOSS 커뮤니티가 확장되고 있지만, IBM처럼 독점적 권리인 지식재산 포트폴리오를 전략적으로 관리하는 것에 대한 중요성은 여전히 간과될 수 없다.

2) 오픈 이노베이션과 공개 소프트웨어 운동의 조화 가능성

공개 소프트웨어 커뮤니티에서 가장 커다란 위협은 IBM, 레드햇과 같이 이윤을 추구하는 기업들이 지식재산권을 행사할 가능성이 상존하는 것이었다. IBM이 SCO에 대한 반소에서 SCO의 지식재산권 침해를 주장한 것은 알고리즘과 비즈니스 프로세스를 자신이 독점할 수 있다는 판단에서 비롯된 것이다. 이

76) SCO v. Novell, D.C. No. 2:04-CV-00139-TS, The US Court of Appeals for the Tenth Circuit, 2011. 8. 30.

는 SCO가 IBM과 리눅스 커뮤니티를 공격할 때 품었던 생각과 사실상 동일한 것이다. 그러나 SCO와의 소송이 계속 중이던 2005년 7월 4일 IBM이 자사의 특허 포트폴리오를 리눅스에 불리한 방향으로 사용하지 않겠다고 선언하면서, 공개 소프트웨어 진영에 대한 큰 위협 중의 하나가 해결되었다.⁷⁷⁾ IBM이 보유한 엄청난 양의 특허 포트폴리오가 글로벌 특허 전쟁에서 위협이 되지 않고, 오히려 SCO와 같은 무차별적 소송공세에 대한 전쟁 억지력으로 작용한 것이다. IBM은 이를 통해 오픈 이노베이션에 대한 전략적 선택이 옳았음을 확인해 주었고, 공개 소프트웨어 진영과의 협력도 지속적으로 강화해 나갈 기반을 마련하였다. 마이크로소프트가 IBM의 방향에 공감하여 공개 소프트웨어 커뮤니티 지원에 동참할 수밖에 없었던 이유는 단순한 동료압력(peer pressure)때문이 아니라 그러한 전략적인 선택이 자사의 장기적인 생존과 발전에도 필수불가결하다는 판단 때문이었을 것이다.⁷⁸⁾

3) FOSS 라이선스 선택을 위한 전략적 의사결정 프로세스

SCO 대 IBM 소송이 보여주는 또 하나의 교훈은 FOSS 라이선싱 규정에 대한 정확한 이해와 전략적 활용 능력이 중요하다는 사실이다. 본고의 취지는 FOSS의 다양한 라이선싱 규정들을 자세히 분석하는 것이 아니었다. 다만 현존하는 FOSS의 무수히 많은 유형들과 각 라이선싱 조항의 정확한 의미를 파악하기 어려운 점, 특히 FOSS와 전통적인 독점 소프트웨어가 다양한 방식으로 결합하는 때에 권리·의무관계가 불명확한 경우가 많다는 점 등이 기업의 전략적인 선택을 어렵게 하고 있다. IBM의 성공적인 FOSS 활용은 향후 많은 기업들이 FOSS 라이선싱을 시도할 때 모범적인 사례가 될 것이다. 그러나 FOSS 라이선싱에는 전략적인 장점과 리스크가 동시에 존재하기 때문에 항상 그 전략적 선택

77) <http://news.cnet.com/2100-7344_3-5296787.html>, 검색일: 2013. 11. 21. 당시 IBM 부사장 도노프리오(Nick Donofrio)는 샌프란시스코에서 개최된 리눅스 월드 박람회에서 "IBM은 리눅스 커널에 불리한 방향으로 특허 포트폴리오를 사용하지 않겠다."고 선언하였다. 리눅스 이용자들에게 소송보해를 판매하는 "Open Source Risk Management(OSRM)"는 당시 리눅스 커뮤니티에서 문제의 소지가 있는 특허 중 60건을 IBM이 보유하고 있다고 밝힌 바 있다.

78) <www.microsoft.com/en-us/news/features/2009/jul09/07-20linuxqa.aspx>, 검색일: 2013. 11. 20.

이 긍정적인 결과로 돌아오리라는 법은 없다. SCO v. IBM 소송은 FOSS 라이선싱 규정의 역할이 IBM과 리눅스 커뮤니티의 권리를 보호하고 그 경쟁력을 강화해 주는 방향으로 나타났지만, 앞으로 벌어질지 모르는 잠재적인 분쟁들을 고려할 때 항상 유사한 결과가 나올 것으로 단정할 수 없다. FOSS의 유형에 따라서는 SCO 같은 기업들의 완승으로 끝나는 분쟁들도 얼마든지 상정할 수 있기 때문이다. 앵겔프리트가 제안한 전략적 의사결정 프로세스는 “make-buy-open”의 의사결정을 해야 하는 기업에게 어느 정도의 가이드라인을 제공해 준다. 다만 아직까지 충분한 수준으로 정교화 된 것은 아니어서, 이에 대한 추가적인 연구가 필요하да 할 것이다.

V. 결론

본고는 오픈 이노베이션이 공개 소프트웨어 운동을 포용하여 전개될 수 있는지, 또는 반대로 공개 소프트웨어 운동이 이념적이고 정치적인 색채를 벗고 전통적인 이윤추구 기업들을 포용하여 발전할 수 있는지에 대한 질문을 던지고, 이에 대한 해답을 찾기 위한 기초적인 연구를 시도하였다. 오픈 이노베이션은 다분히 경영학적인 개념이고, 공개 소프트웨어 운동은 기술적인 측면을 제외한다면 매우 정교한 법적 개념들을 내포하고 있다. 전자는 기업의 이윤추구 방식의 새로운 패러다임이라 한다면, 후자는 소프트웨어 업체들의 이윤추구 행태에 대한 반성과 비판을 통해 형성된 운동이다. 본 연구를 통해서 오픈 이노베이션의 구성요소인 거래와 계약이 결국은 공개 소프트웨어 운동의 이상인 공유와 협력을 위해서도 현실적으로 필요한 요소라는 사실과, 공개 소프트웨어 운동도 추상적인 공유와 협력을 구상하는 것이 아니라, 까다로운 라이선싱 계약을 도구로 삼아 발전해 왔다는 사실을 알게 되었다. 본고는 사실상 오픈 이노베이션과 공개 소프트웨어 운동의 어느 한 쪽을 깊이 분석하기보다는, 둘 사이의 접근과 조화를 모색하는 데 초점을 맞추었다. 이를 위해 오픈 이노베이션을 실천하고 있는 IBM이 어떻게 공개 소프트웨어 커뮤니티와 공존하고 있는지, 어디에서 기업의 생존과 지속가능성을 위한 자원을 발견하고 있는지 살펴보고자 하였다. 세계

굴지의 하드웨어 기업이자, 강력한 소프트웨어 기업으로 자리매김하고 있는 IBM과 그 후원을 받은 공개 소프트웨어 진영이 SCO와의 긴 분쟁을 승리로 마무리하는 것으로 보인다. 그러나 단순한 소송의 승패를 떠나, 앞으로 독점 소프트웨어와 공개 소프트웨어의 결합이 만들어 낼 수 있는 분쟁상황의 복잡성과 그에 관련된 법적 쟁점들, 다양한 FOSS 라이선스의 유형과 이를 전략적으로 활용할 수 있는 전문성의 습득이 특히 글로벌 IT 산업에서 우리 기업들에게 매우 중요한 차별화 전략이 될 것이라는 사실을 중시해야 할 것이다. 앵겔프리트의 전략적 의사결정 방법론이 아직 초기 단계의 모형이지만, 기존의 전략경영 모델과 법적 고려를 통합한 의사결정 프로세스를 구상하기 위한 본격적인 연구가 의미 있는 결과를 가져올 수 있음을 시사하고 있다.

참고문헌

〈국내 단행본〉

문화체육관광부 · 한국저작권위원회, 소프트웨어 관리 가이드, 2013.

사법연수원, 저작권법, 2006.

송영식 · 이상정 · 김병일, 지적재산법, 8정판, 세창출판사, 2007.

오승종 · 이해완, 저작권법, 박영사, 2001.

윤선희, 특허법, 법문사, 2003.

정상조, 지적재산권법, 홍문사, 2004.

정보통신부, 오픈소스 SW 라이선스 가이드, 2007.

정보통신산업진흥원, 공개소프트웨어 백서, 2012.

〈국내 학술지〉

고재종, “오픈소스 소프트웨어의 법적 문제에 대한 검토”, 상사판례연구, 제25권 제3호(2012), 한국상사판례학회.

김병일, “오픈소스 소프트웨어와 법적 위험”, 산업재산권, 제25호(2008), 한국산업재산권법학회.

이철남, “오픈소스 소프트웨어에 대한 소프트웨어 특허의 법적합의”, 경영법률, 제17권 제1-2호(2006), 한국경영법률학회.

정진근, “스마트기술의 함의와 저작권법의 과제”, 강원법학, 제33권(2011), 강원대학교 비교법학연구소.

〈해외 문헌〉

Chesbrough, Henry William, *Open Innovation: The New Imperative for Creating and Profiting from Technology*, Harvard Business Press, 2003.

Engelfriet, Arnoud, “The Best of Both Worlds”, *Intellectual Asset Management*, Aug/Sep(2006).

Fontana, Richard E., “Open Source License Enforcement and Compliance”, *Patents, Copyrights, Trademarks, and Literary Property, Course Handbook Series, PLI/Pat*, No.989(2009), Practising Law Institute.

- Frazer, Brad, "Open Source Is Not Public Domain: Evolving Licensing Philosophies", *Idaho Law Review*, Vol.45(2009).
- Gomulkiewicz, Robert W., "Open Source License Proliferation: Helpful Diversity Or Hopeless Confusion?", *Washington University Journal of Law and Policy*, Vol.30(2009).
- Ilardi, Terry J., "Open Source Software", *Patents, Copyrights, Trademarks, and Literary Property, Course Handbook Series, PLI/Pat*, No.911(2007), Practising Law Institute.
- Potenza, Joseph M., and Steve Chang, "the SCO Group, Linux, and Life after SCO's Setback against Novell", *Intellectual Property & Technology Law Journal*, Vol.20 No.1(2008).
- Rosen, Lawrence, "Open Source and Free Software 2009: Where We've Been and Where We're Going", *Patents, Copyrights, Trademarks, and Literary Property Course, Handbook Series, PLI/Pat*, No.989(2009), Practising Law Institute.
- Stern, Alan and A. Clifford Allen, "Open Source Licensing", *Patents, Copyrights, Trademarks, and Literary Property Course, Handbook Series, PLI/Pat*, No.985(2009), Practising Law Institute.
- Taylor III, Alexander L., "The Wizard inside the Machine", *TIME*, April 16, 1984.

Legal Implications of Open Innovation Strategy - With Focus on “SCO v. IBM” Litigation-

Park, Sung-Pil & Kim, Yong-kil

Abstract

“Open Innovation” is an industry buzzword today. It is becoming a prerequisite for a firm’s survival and growth, especially in the IT industry. Free/Open Source Software (“FOSS”) movement is different from Open Innovation in terms of its origin and philosophical foundation. But it also should be noted that more and more companies embedded with the Open Innovation concept are now actively utilizing FOSS to make the most of their product value. Furthermore, Henry Chesbrough’s Open Innovation model itself contains an element that requires participation of FOSS developers and labels them as “innovation missionaries.” The FOSS community is also experiencing limitless collaboration opportunities with commercial software developers and increase in relevant licensing guidelines.

IBM, as a leading hardware and software company in the global market, is located at the harmonization point where the Open Innovation strategy and the FOSS movement can be merged. We examined IBM’s Open Innovation strategy and SCO’s legal battle against this company, so that we can obtain practical implications on how to analyze complex legal issues arising out of such disputes and obtain valuable lessons for entrepreneurs and legal experts working for technology driven IT enterprises. The litigation between SCO and IBM, which is still pending before the Utah federal court, challenges us to design appropriate strategy options for innovation leaders that can

potentially be involved in IP litigations which are becoming more and more complicated in case software products would become key components of the dispute. This paper tried to review and explain factual background, procedural history, courts' analyses on FOSS issues, litigation strategies, and IP portfolio management practices found in the course of this litigation.

Keywords

open innovation, free/open source software, SCO v. IBM, IP portfolio, Linux, intellectual property

