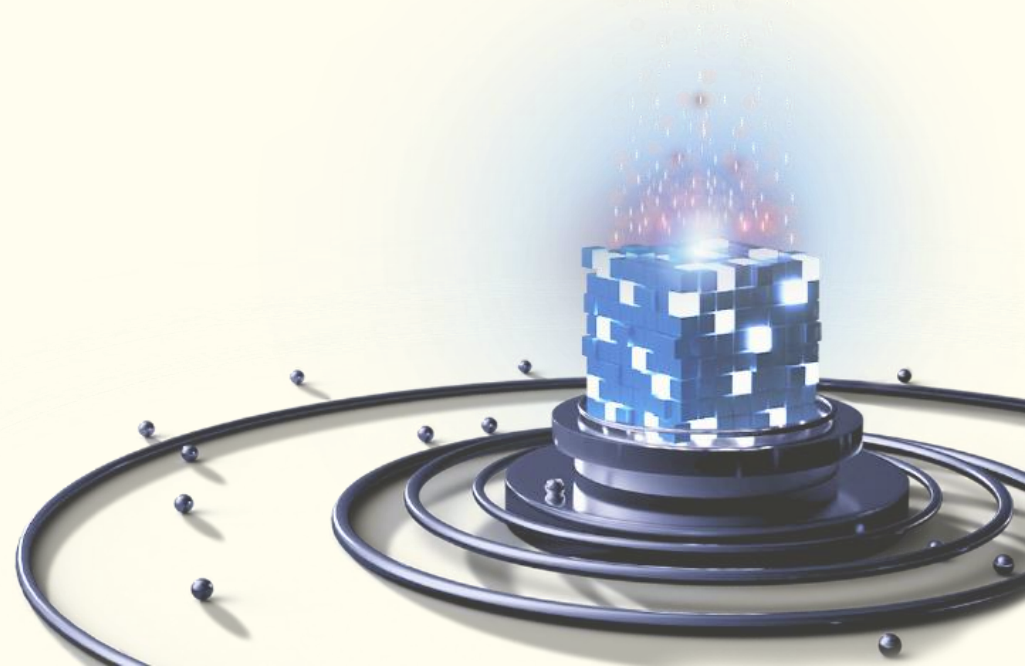


저작권 소송에서 오픈소스의 지위와 쟁점

The status and issues of open source in copyright disputes

김시열

전주대학교 로컬벤처학부 교수



CONTENTS

- 01 들어가며
- 02 프로그램의 저작권 침해 입증 구조
- 03 실질적 유사성 판단 시 정량적 유사도의 활용
- 04 저작권 소송상 오픈소스에 관한 문제
- 05 정리





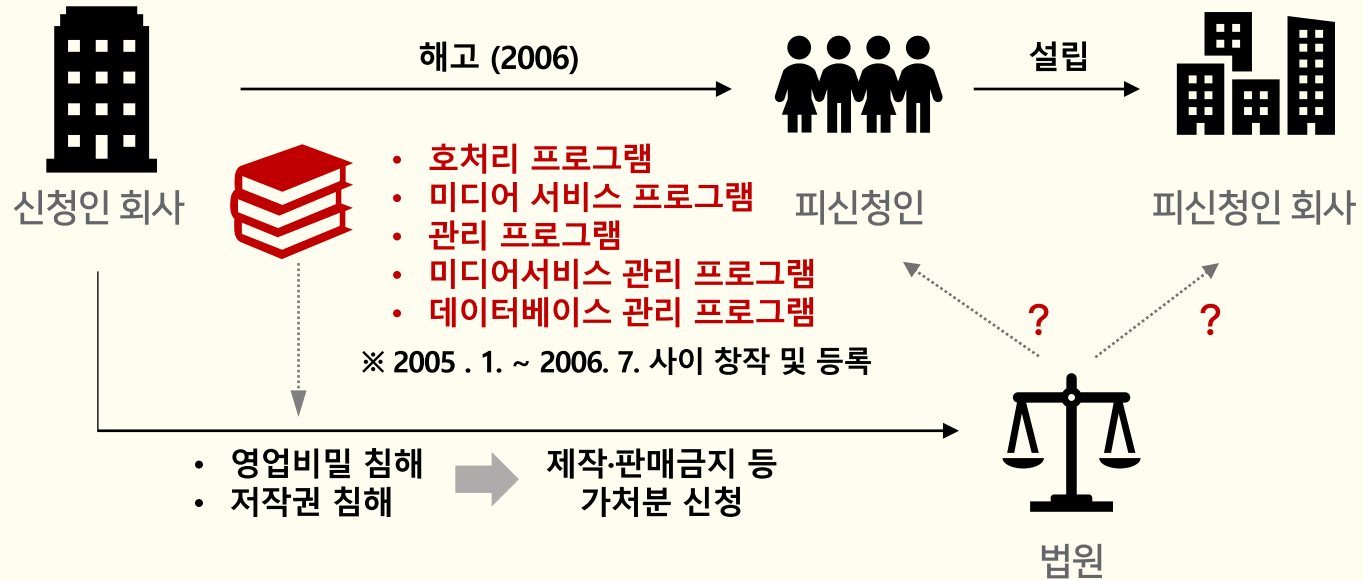
01

들어가며

01

들어가며

➤ 인터넷교환기 프로그램 사건(서울중앙지법, 2007)



- A社 : 인터넷교환기 제조 회사
- B社 등 : 인터넷교환기 판매 회사



들어가며

➤ 법원의 판결

“살피건대, 기록에 의하여 소명되는 다음의 각 사정을 종합하면 이 사건 저작물과 피신청인 프로그램 사이에는 실질적 유사성이 인정된다고 선뜻 단정하기는 어렵다고 할 것이다.

즉, 1. 피신청인 제품에는 IP-PBX(HEXUS) 호처리 프로그램, IP-PBX(HEXUS) 미디어 서비스 프로그램, IP-PBX(HEXUS) 관리 프로그램, IP-PBX(HEXUS) 미디어서비스 관리 프로그램, IP-PBX(HEXUS) 데이터베이스 관리 프로그램 등 5개의 컴퓨터프로그램이 사용되고 있음에도 불구하고, 신청인은 그중 2개의 프로그램과 이 사건 저작물 사이의 유사성 여부에 관하여만 감정을 신청하여 감정이 진행되었으므로, 이 사건 저작물 전체와 피신청인 제품에 사용된 위 5개의 컴퓨터프로그램 전체에 관한 유사성 여부가 소명되지 아니하였다.

2. 이 법원의 컴퓨터프로그램보호위원회에 대한 감정촉탁결과는, **공개된 프로그램 소스 일부를 제외하는 절차를 거치기는 하였으나**, 이 사건 저작물 중 인터넷전화 교환기의 표준적인 기능을 구현하기 위하여 필수적으로 사용되어야 하거나, 그와 결합하여 사용될 것으로 예상되는 다른 프로그램과의 호환성을 위하여 필요하거나, 기타 일반적으로 받아들여지는 프로그래밍 관행 등으로 인하여 **표현이 유사할 수 밖에 없는 부분을 여과하는 절차를 거치지 아니하고 소스코드의 유사성을 비교한 것이다.**

3. 이 법원의 컴퓨터프로그램보호위원회에 대한 감정촉탁결과에 의하더라도, 소스코드의 유사성에 관한 정량적 비교를 하였을 때, 피신청인 프로그램 중 IP-PBX(HEXUS) 호처리 프로그램의 33.2%에 이 사건 저작물과 유사한 소스코드가 존재하고, IP-PBX(HEXUS) 미디어 서비스 프로그램의 48.0%에 이 사건 저작물과 유사한 소스코드가 존재하며, 이 사건 저작물을 기준으로 위 각 프로그램의 소스코드와 유사한 부분을 살펴보면 그 유사도가 21.4%이다. 또한 개별 모듈 및 파일의 유사성을 살펴보면, 피신청인들이 공개된 프로그램 소스를 사용하였다고 주장하는 부분과 기능적으로 유사한 부분에 있어서 높은 유사도를, 그렇지 않은 부분에서는 매우 낮은 유사도를 보이고 있어, 기능의 유사함으로 인하여 불가피하게 유사한 소스코드가 만들어졌을 가능성을 배제할 수 없다.

따라서, 이 사건 저작물과 피신청인 프로그램 사이의 **실질적 유사성이 고도로 소명되었다고 보기 어려우므로...** (후략)”



02

프로그램의 저작권 침해 입증 구조



프로그램의 저작권 침해 입증 구조

➤ 저작권을 침해한 행위

- 침해자의 작성물이 저작권자의 작성물(저작물)을 얼마나 가져다 이용하였는지 → 이용한 정도가 실질적이라면 침해
- 저작권자의 작성물 가운데 보호받는 부분이 얼마나 침해자에 의해 이용되었는지 → 이용한 정도가 실질적이라면 침해

➤ 침해 여부의 판단 구조 { 주로 '복제권' 및 '2차적저작물작성권'에 적용 ①→②→③ (O) / ③→②→① (X)

- [1단계] 이용된 작성물이 저작권법에 의한 보호받는 저작물일 것 (피침해자의 유효한 저작물 보유)
- [2단계] 침해자가 저작권자의 저작물에 의거하여 자신의 것을 만들었을 것 (피침해자의 이용행위 존재)
- [3단계] 침해물과 피해 저작물 사이에 동일성 내지 실질적 유사성이 존재할 것 (침해로 볼 수 있을 정도의 이용)

➤ 다툼 상 주요 쟁점 사항

- 피침해자의 SW가 진정한 보호법익이 있는 대상인지 여부
- 비교 대상 간 유사한 정도가 실질적으로 유사한 것인지 여부



프로그램의 저작권 침해 입증 구조

➤ 저작권 침해 여부 판단의 핵심 : 실질적 유사성(substantial similarity)

- 개념 : 침해물이 피침해 저작물에 대한 종속성이 인정될 수 있는 정도의 유사 정도
 - 실질적 유사 여부는 저작권법상 보호받는 표현(저작물에 해당하는 부분)만을 대상으로 비교하여 판단
- ‘실질적으로 유사하다’의 의미
 - 복제행위가 있었다는 사실적 문제에 더하여 저작권법에 의해 보호받는 권리를 침해한 정도로 유사하다는 것
- 저작권 침해 여부의 판단 시 [3단계]에 해당하며, 실무적으로는 저작권 침해 판단에 있어 가장 핵심적인 요소로 기능
- 개념적 본질 상 객관적 기준을 정립할 수 없으므로, 실질적으로 유사하다는 정도는 개별 사안마다 상이하게 존재

➤ 추상화-여과-비교 모델 (3 Step Test)

- 아이디어와 표현을 분리(추상화)→ 보호받지 못하는 표현(아이디어 포함) 제거(여과)→ 표현의 핵심만을 비교(비교)
- 美 제2연방항소법원의 Altai 판결(1992)을 통해 제시
- 우리 법원은 이 모델을 명시적 도입하지 않았으나, 저작권법 기본원리에 따라 이 모델과 유사한 형태의 판단 방식이 활용



프로그램의 저작권 침해 입증 구조

➤ 오픈소스의 저작권적 지위

- 오픈소스 소프트웨어는 개발자가 자신이 개발한 소스코드를 공개하여 일정한 조건 하에서 누구나 자유롭게 사용·복제·수정 및 배포할 수 있도록 개방한 소프트웨어로 정의
- (OSS 창작자 관점) 공개되어 누구나 이용할 수 있도록 되어 있지만, 해당 소스코드가 창작성이 있는 경우에는 창작자인 개발자가 저작권을 보유 → OSS 라이선스를 통해 자신의 저작권 행사를 제한
- (OSS 이용자 관점) 오픈소스 소프트웨어를 이용하여 특정한 소스코드를 작성(프로그램 개발)한 경우, 작성한 부분에 오픈소스 이외에 달리 창작성 있는 부분이 없다면 오픈소스의 이용자는 저작권자(2차적저작물에 대한 저작권자)가 될 수 없음 → 통상의 저작권 침해 소송에서 오픈소스가 고려되는 경우
 - 이 경우 오픈소스를 사용한 부분은 개발자(OSS이용자)에게 부여되는 저작권 보호범위에 해당X → 소송상 문제의 출발점

➤ 오픈소스의 저작권 침해 판단 구조 상의 작용

- 논리적으로는 [1단계]에서 오픈소스의 존재(사용)가 고려될 수 있지만, 실무적으로는 고려하지 않음 (3단계에서 일괄 고려)
- [3단계]의 실질적 유사성 판단 과정에서, 오픈소스는 유사성 비교 대상에 포함되지 않도록 **비교대상에서 제거**(여과)



03

실질적 유사성 판단 시 정량적 유사도의 활용



실질적 유사성 판단 시 정량적 유사도의 활용

➤ 정량적 유사도 도출을 통한 실질적 유사성 판단에 활용

- 정량적 유사도는 실질적 유사성이라는 질적(質的) 가치를 숫자라는 표현 수단을 이용하여 양적(量的) 가치로 나타낸 것
→ 다만, 무엇이 유사한 것인지는 결국 이를 판단하는 사람의 주관적 의사에 의할 수 밖에 없기 때문에 단순히 양적인 상태에서 바로 실질적 유사 여부가 도출될 수는 없음
- 정량적 유사도는 논리적으로는 실질적 유사성 판단 시 참고할 여러 요소 가운데 하나에 불과하나, 실무상 실질적 유사성 판단에 가장 핵심적인 요소로 기능하는 경우가 많음 → 비교 대상 프로그램의 관계를 명확하게 보여준다는 장점

➤ 정량적 유사도 산출의 방식

- 원칙적으로 특별한 산출 방식이 정해진 것은 아니나(산출방식 다양성 인정), 적용된 산출방식은 유사도 산출 목적에 비추어 합리성을 확보할 수 있어야 함
- 법적 가치인 실질적 유사성 개념에 최대한 근접한 실체적 수치(분석값)의 제시를 목적으로 함에 따라 다음의 구체적 요건을 충족해야 함
 - ① 저작권법에 의해 보호받는 표현만을 대상으로 유사도 산출이 이루어져야 함
 - ② 유사 여부 판단의 객관적 기준이 설정되어야 하며, 그 양적 기준의 합리성이 확보되어야 함

실질적 유사성 판단 시 정량적 유사도의 활용

➤ 재판에서 활용된 정량적 유사도 산출 유형

※ 표 출처
한국저작권위원회

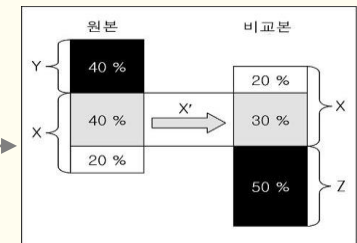
구분	원본기준방식		비교본기준방식	
	수식	유사도	수식	유사도
전체감정영역 기준 방식	$\{X' / (X + Y)\} \cdot 100$	40%	$\{X' / (X + Z)\} \cdot 100$	30%
공통감정영역 기준 방식	$\{X' / X\} \cdot 100$	67%	$\{X' / X\} \cdot 100$	60%

유사도 표현방식 기준		유사도 계산방식 기준		비교대상 설정방식 기준	
방식1	방식2	방식3	방식4	방식5	방식6
소스코드 및 DB 등을 파일 쌍 등으로 구분하여 각각의 유사도를 산출하고 그 결과를 그대로 제시하는 형태	각 구성요소 별로 유사한 정도를 도출하고 이를 합하여 단일의 종합유사도를 산출하여 제시하는 형태	소스코드 등을 전체 줄(라인) 대비 유사 줄(라인)의 정도를 비교하여 유사도를 산출하는 형태	구성요소별로 구분하고 각 요소를 유사 혹은 비유사로 판단, 이를 종합하여 전체의 유사도 계산 및 산출 형태	어느 쪽의 프로그램을 기준으로 유사도를 계산하느냐에 따라 원본기준방식, 비교본기준방식으로 구분	기준영역을 어떻게 산정하느냐에 따라 전체영역기준방식, 공통영역기준방식으로 구분

[방식1] 사례

(서울고법 2011라457)

양 당사자의 소스코드를 비교한 결과 채권자 프로그램 기준 12.11%, 채무자 프로그램 기준 19.52%의 유사도가 인정되었고, 양 당사자의 데이터베이스를 비교한 결과 데이터베이스 스키마에 대한 유사도는 채권자 프로그램을 기준으로 하였을 때 99.09%, 채무자 프로그램을 기준으로 하였을 때 89.76%를 인정하였다. 한편 사용자정의 프로시저에 대한 유사도는 채권자 프로그램 기준으로 하였을 때 99.84%, 채무자 프로그램을 기준으로 하였을 때 8.76%를 인정 한 사례



※ 그림 출처
한국저작권위원회

[방식2] 사례

(서울남부 2005가합3367)

이 사건 제1, 2프로그램 사이의 비교항목별 유사도는 소스코드의 경우 0.55%, 사용자 인터페이스의 경우 14.99%, 데이터베이스의 경우 4.62%이고, 위 3가지 비교항목의 유사도에 가중치를 부여하여 이를 합산하여 산정된 이 사건 제1, 2프로그램 사이의 종합유사도는 3.22%인 사실을 인정할 수 있는바..

[방식3] 사례

(서울고법 2009나71420)

Win98용 프로그램의 소스코드 45,842 라인 중 37,319 라인이 이 사건 프로그램의 소스코드와 동일하거나 유사하여 종합유사도는 81.41%를 나타냈다.

[방식4] 사례

(서울고법 2006나113835)

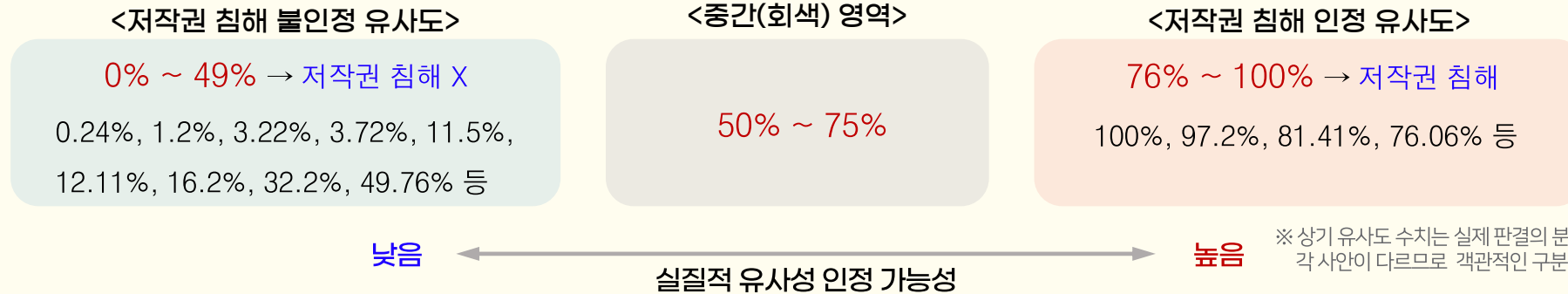
이들을 파일의 개수, 줄 수, 함수 수에 따라 정량적으로 비교한 결과, 파일의 수를 비교할 때 Bancs 파일들 중에서 호출관계가 50% 이상 유사한 ProBank 파일들이 포함된 파일 수는 전체 Bancs 파일 수의 71.84%(681개/948개)이고, 파일 안에 들어있는 줄 수를 비교할 때 Bancs 파일들에 포함된 줄 중에서 함수 호출관계가 50% 이상 유사한 ProBank 파일들에 포함된 줄 수는 Bancs 전체 줄 수의 41.25%(635,809/1,541,174)이며, 함수 수를 비교할 때 Bancs 파일들에 포함된 함수 중에서 함수 호출관계가 50% 이상 유사한 ProBank 파일들에 포함된 함수 수는 Bancs 전체 함수 수의 15.12%(9,763/64,574)이다. 따라서 COBOL로 구성된 Bancs의 소스코드 중에서 ProBank와 유사성이 50% 이상인 소스코드에 관한 정량적 분석결과는 전체의 42.74%(다음에서 보는 평균값)를 차지한다.

03

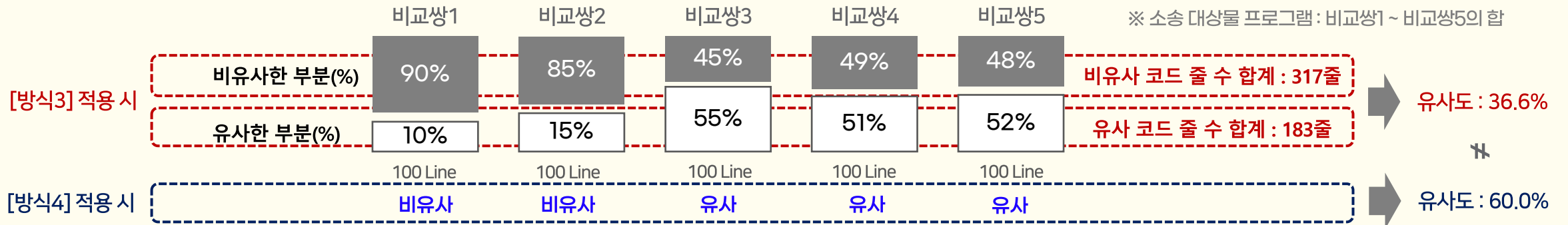
실질적 유사성 판단 시 정량적 유사도의 활용

➤ 정량적 유사도 활용상의 특징·한계

- 유사도 스펙트럼 상 중간영역의 회색지대 존재 → 해석의 어려움 및 실무상 가장 난해한 지점



- 산출방식에 따라 결과값의 큰 차이 발생 가능





04

저작권 소송상 오픈소스에 관한 문제



저작권 소송상 오픈소스에 관한 문제

➤ 실질적 유사성 비교 과정에서 오픈소스의 충실한 제거 문제

- 재판 과정에서 제3자에 의한 오픈소스의 비교대상 배제 어려움
 - 실무상 당사자의 협조를 얻어 사용된 오픈소스를 특정하고 이를 비교대상에서 배제하는 절차를 적용하나, 협조가 되지 않는 경우 사용된 오픈소스를 탐지하는 것 자체가 상당한 장애 요인 → 당사자의 비협조에 따른 불충실 제거 시 비교 결과의 의미?
- 개발 프로젝트의 특정한 요구, 혹은 성능 최적화 등의 여러 이유로 오픈소스를 수정하여 사용한 소스코드에서 오픈소스를 제거할 경우 구체적인 제거 범위를 어디까지 할 것인지에 대한 검토가 필요 → 수정된 부분의 저작물성, 줄 단위의 특정 등
- 실질적 유사성에 대한 비교가 아닐 경우 오픈소스의 제거 불필요 → 명확한 목적 인지 필요

➤ 수정된 오픈소스의 성격 문제

- 단순히 부수적인 일부 표현을 수정한 경우 : 이용자의 저작물 아님(오픈소스 개발자의 저작물에 해당)
- 창작성 있는 주요한 표현 변경으로 수정한 경우 : 새롭게 변경(추가)된 표현에 따라 이용자의 저작물(2차적저작물)로 인정될 여지가 있음 → 인정 시 실질적 유사성 판단을 위한 비교 시 제거 대상에서 제외(보호받는 표현이 되므로)
 - 이에 따라 저작권이 발생할 수 있으나, 이용한 오픈소스의 라이선스로 인하여 수정으로 발생한 저작권을 행사하는 것은 제한됨



저작권 소송상 오픈소스에 관한 문제

➤ 여과에 따른 대상 프로그램의 형해화

- SW 개발 시 오픈소스 사용이 많은 경우, 여과 과정에서 이들을 모두 제거하고 남는 일부의 소스코드만을 저작권법상의 컴퓨터프로그램이라 할 수 있는지에 대한 의문
- 비교대상에서 오픈소스 여과 후 극히 일부만 남은 소스코드(core에 해당하는 부분) 가운데 일정한 부분이 침해자의 것과 유사한 것으로 판단되었다고 하여 이를 저작권 침해로 인정할 수 있겠는가에 대한 문제
- 저작권법이 보호하려는 대상이 과연 무엇인가에 관한 고민으로 연결

➤ 정량적 유사도의 실질적 의미 차이

- 오픈소스의 활용이 이루어진 프로그램의 경우 그렇지 않은 경우와 동일한 유사도를 갖더라도 그 유사도의 실질적 의미는 완전하게 다르게 나타남
 - 100줄 소스코드의 유사도가 60%라고 가정할 때, 모두 직접 창작한 것이라면 침해된 소스코드가 60줄이나, 오픈소스를 50줄 사용한 것이라면 같은 유사도에도 불구하고 침해된 소스코드는 30줄에 불과
- 비율의 표현을 수단으로 하는 유사도의 특징 상 유사 정도의 해석 시 실제 침해된 소스코드의 양을 별도로 고려하여야 할 필요가 있음



05

정리



정리

➤ 개발한 프로그램에서 타인이 창작한 오픈소스 소프트웨어는 침해 증거 될 수 없음

- 오픈소스 개발자(OSS창작자)의 저작권과 별개로, 그 오픈소스를 이용하여 프로그램을 창작한 경우에는 그 프로그램 내에서 오픈소스 부분은 그 프로그램 개발자(OSS이용자)의 저작물로 인정되지 않음 → 프로그램 개발자 권리보호 대상 X
- 이에 프로그램 개발자(OSS이용자)의 저작권 보호를 목적으로 한 저작권 소송에서 오픈소스 소프트웨어는 실질적 유사성 판단 과정에서 제외되어야 하는 존재

➤ 불완전한 제외 시, 창작 결과(SW)에 대한 적절한 법적 보호가 곤란

- 재판에서 실질적 유사성에 대한 판단을 위한 비교 분석 시, 오픈소스를 비롯한 비교대상에서 제외되어야 할 요소들이 적절하게 제거되지 않았을 경우, 침해에 대한 증거(유사성 비교·분석 결과)를 인정받지 못할 수 있다는 점에 주의 필요
- 침해자의 경우 오픈소스의 불완전 제거가 오히려 침해를 감추는데 사용(침해증거의 불인정으로)될 수 있으므로, 이러한 부정적 결과를 초래하지 않도록 해야 함

➤ 복잡한 오픈소스의 법적 문제들을 이해하고, 개발 과정에서 이를 고려한 적절한 관리 (예를 들어 사용한 오픈소스를 목록화 하는 등) 노력이 필요

감사합니다

저작권 소송에서 오픈소스의 지위와 쟁점

