
컴퓨터프로그램의 실질적 유사성 판단을 위한 정량적 분석에 관한 연구

김시열*

목 차

- I. 서론
- II. 컴퓨터프로그램에서의 실질적 유사성
 - 1. 컴퓨터프로그램의 기술적 배경
 - 2. 실질적 유사성의 개념과 그 보호대상
- III. 우리나라 판례에서의 실질적 유사성 판단
 - 1. 실질적 유사성이 인정된 경우
 - 2. 실질적 유사성이 부정된 경우
 - 3. 정리
- IV. 실질적 유사성 입증 요소로서의 정량적 유사도 검토
 - 1. 컴퓨터프로그램의 실질적 유사성 판단 기준에 관한 미국에서의 논의
 - 2. 정량적 유사도의 개념 및 특징
 - 3. 정량적 유사도의 산출
 - 4. 실질적 유사성에 대한 적합성을 위한 검토
- V. 결론

초록

컴퓨터프로그램 저작권 침해 소송에서 실질적 유사성은 침해를 입증하기 위하여 많은 중요성을 갖고 있다. 추상성을 특징으로 하는 실질적 유사성은 명확한 기준을 제시하기가 쉽지 않으며, 컴퓨터프로그램에 관하여도 이는 동일하다. 따라서 컴퓨터프로그램의 실질적 유사성을 판단하기 위한 기준을 정립하기 위하여 지속적인 논의가 이루어지고 있다.

컴퓨터프로그램을 구성하는 요소들 중 창작자의 표현의 다양성이 가장 넓게 나타나는 것은 문언적 형태로 작성된 소스코드이다. 기능성 및 효율성이라는 목적으로 인하여 표현이 제한된 다른 요건들과는 달리 표현의 다양성이 넓은 소스코드는 컴퓨터프로그램 저작권 침해 판단에 있어서 가장 핵심적인 요건으로 기능하고 있다. 우리나라의 판례를 살펴봐도 소스코드를 가장 중요한 비교대상으로 삼고 이에 대한 복제의 정도를 실질적 유사성 판단을 위한 근거로 삼는 경우가 상당히 발견된다.

본 연구는 소스코드 등에 대한 정량적 유사도가 컴퓨터프로그램의 저작권 침해 입증을 위한 실질적 유사성 판단에 중요한 역할을 하고 있는 현실에서, 기술적 개념으로 볼 수 있는 소스코드 등의 정량적 유사도가 법적 개념인 실질적 유사성을 위하여 적절한 의미를 가질 수 있도록 관련 문제들을 살펴보고자 하였다.

주제어

컴퓨터프로그램, 저작권 침해, 실질적 유사성, 3단계테스트, 감정, 소스코드 비교

I. 서론

저작권법은 컴퓨터프로그램을 저작물로 규정하고 이를 보호하고 있다.¹⁾ 컴퓨터프로그램은 본래 어문저작물의 한 유형으로 보호되었으나,²⁾ 컴퓨터프로그램이 내재하고 있는 특성인 효율성 및 외부적 요인 등에 의하여 표현이 제한될 수 밖에 없는 등의 특징으로 인하여 예술적 특성을 기본으로 하는 일반적인 어문저작물의 경우와 동일하게 보호할 수가 없다. 특히 저작권법은 창작성 있는 표현만을 보호대상으로 하고 단순히 아이디어에 해당하는 부분은 보호대상에서 제외하는 ‘아이디어 · 표현 이분법’을 기본 원칙으로 채택하고 있으며, 이는 저작권법상의 보호대상을 확정하고 저작권의 침해 여부를 판단하는 데 매우 중요한 역할을 하는 개념으로 자리잡고 있다.³⁾ 이에 따라 컴퓨터프로그램을 구성하는 요소들도 표현에 해당하는 요소와 아이디어에 해당하는 요소로 구분된다. 컴퓨터프로그램은 단일의 표현 형식이 아닌 여러 가지 형태의 표현 등이 결합된 것으로서, 소스코드(source code), 목적코드(object code), 마이크로코드(microcode)⁴⁾, 흐름도(flow chart), 사용자매뉴얼(user manual), 알고리즘(algorithm) 및 스크립트(script) 등이 일체 혹은 다른 형태로 하나의 컴퓨터프로그램을 이루고 있다. 컴퓨터프로그램을 저작권으로 보호함에 있어서 아이디어 · 표현 이분법 체계에서 각각의 세부적 구성요소들이 보호대상으로 인정될 수 있는지에 관한 논의가 계속되고 있는 실정이다.⁵⁾

1) 저작권법 제2조 제16호 “컴퓨터프로그램저작물은 특정한 결과를 얻기 위하여 컴퓨터 등 정보처리능력을 가진 장치(이하 “컴퓨터”라 한다) 내에서 직접 또는 간접으로 사용되는 일련의 지시·명령으로 표현된 창작물을 말한다.”

2) 컴퓨터프로그램이 미국저작권법 제102조(a)의 보호 저작물의 분류 중 어디에 해당하는지에 대해서는 하원 보고서에서 “‘어문저작물’이란 용어는 문학적인 품격이나 질적 가치의 기준을 암시하지 않는다. 이에는 카탈로그, 디렉토리 및 사실에 관한 저작물, 참고서 또는 지도서, 그리고 자료 편집물이 포함된다. 또한 컴퓨터 데이터베이스와 컴퓨터프로그램도 사상 그 자체와 구별되는 것으로서, 프로그래머의 독창적 사상을 포함한 것인 한 포함된다.”라고 명시함으로써 컴퓨터프로그램을 어문저작물로 보았다. 최경수 번역, 저작물의 새로운 기술적 이용에 관한 국립위원회의 최종보고서(CONTU), 저작권심의조정위원회, 1994, 46면.

3) 이해완, 저작권법, 박영사, 2007, 873면.

4) 마이크로프로세서(CPU)를 작동시키는 것을 목적으로 하드웨어적으로 롬(ROM) 등에 저장된 컴퓨터프로그램이다.

5) ‘2세대(second generation) 논의’로서 문언적으로 표현되지 않은 비문언적 부분을 어디까지 저작권으로

여러 컴퓨터프로그램의 구성 요소 가운데 사람이 직접 작성한 소스코드와 이를 컴파일하여 생성한 목적코드는 저작권법에서 이론 및 판례를 통하여 명확하게 보호대상으로 삼고 있는 요소들이다. 소스코드는 인간이 읽고 이해할 수 있는 문자의 형태로 구성되며, 프로그래머가 직접 작성한다는 점에서 컴퓨터프로그램 중 창작성 및 표현의 다양성이 높게 인정된다. 따라서 실제적인 저작권 침해소송에서는 보호대상 여부에 대하여 논란이 있는 다른 요소 보다 소스코드에 대한 복제여부를 확인하여 이를 중심으로 저작권 침해 여부를 판단하고 있는 것으로 보인다. 특히 소스코드 등은 유사한 정도를 정량적으로 나타낼 수 있다는 장점으로 인하여 소송과정에서도 정량적인 비교 결과를 중요하게 활용하고 있다.

본 논문에서는 컴퓨터프로그램 저작권 침해 소송에서 가장 핵심적인 요건인 실질적 유사성을 판단함에 있어서 소스코드 등의 정량적 유사도의 활용 형태를 판례를 통하여 살펴보고, 아직까지는 기술적·공학적 개념으로 볼 수 있는 소스코드 등의 정량적 유사도가 법적 개념으로서의 실질적 유사성을 위한 적절한 의미를 가질 수 있도록 정량적 유사도에 관한 전반적인 내용과 활용에 있어서의 고려사항을 살펴보고자 한다.

II. 컴퓨터프로그램의 실질적 유사성

1. 컴퓨터프로그램의 기술적 배경

컴퓨터는 각종 데이터를 처리하는 것을 목적으로 하는 매우 정교화된 전자적 장치이나, 그 자체로는 어떠한 기능도 수행할 수 없고 이는 사람이 작성한 프로그램에 의해서만 기능 수행이 가능하다. 이때 컴퓨터가 읽고 이해할 수 있는 언

보호할지에 관한 논의이다. 이는 컴퓨터프로그램의 보호범위와 관련하여 아이디어·표현 이분법의 구체적인 기준 및 실질적 유사성의 판단 기준에 관한 문제라고 할 수 있다. 반면 '1세대(first generation) 논의'는 컴퓨터프로그램을 저작권으로 보호할 수 있는지 여부에 대한 것으로서 이 시기(1970년대~1980년대)에 소스코드와 목적코드 및 ROM에 저장된 프로그램, 운영체제프로그램 등도 보호대상이 된다는 것이 명확하게 되었다. 이해완, 전게서, 904면.

어를 기계어라고 한다. 기계어는 인간의 언어와는 달리 2진언어(binary code)⁶⁾로 이루어져 있으며 사람이 2진언어를 이해하는 것은 매우 어렵다. 그러므로 프로그래머는 고급언어⁷⁾인 소스코드(source code)를 작성하고 이를 기계어로 변환(컴파일, compile)하여 컴퓨터가 이해할 수 있는 소위 목적코드(object code)라고 하는 2진언어를 만들게 된다.

컴퓨터프로그램을 작성하는 것을 프로그래밍(programming)이라고 한다. 이는 특정한 문제의 해결 및 작업의 수행을 목적으로 프로그래밍 언어를 사용하여 프로그래머가 갖고 있는 아이디어 등을 최종적인 제품으로 구현하는 것으로서⁸⁾ 다음의 여러 단계를 거쳐 이루어지게 된다. 첫째, 프로그래머는 작성될 프로그램이 이용자에게 어떠한 서비스의 제공을 목적으로 하는지를 결정하게 된다. 둘째, 프로그래머는 이용자의 요구를 충족시키는 데 필요한 기능을 결정하여야 한다. 셋째, 프로그래머는 작성하고자 하는 프로그램의 전반적인 구조를 디자인하여야 한다. 일반적으로 세 번째 단계에서 흐름도 등의 산출물들이 도출되고 코딩과정에서 활용된다. 넷째, 프로그래머는 실제적인 소스코드를 작성함으로써 디자인된 프로그램을 구체화시키게 된다. 마지막으로 작성된 프로그램에서 오류 등이 발생하지 않는지 검토하는 단계를 거쳐 프로그램이 완성되게 된다. 다만 일련의 제작 과정 중 각 단계별로 동일한 노력이 들어가는 것은 아니다. 일반적으로는 소스코드를 직접 작성하는 과정에 많은 노력이 수반되는 것으로 생각될 수 있으나, 실제로 프로그램을 작성함에 있어서 가장 많은 노력이 들어가는 부분은 컴퓨터프로그램을 기획하고 구조 등을 디자인하는 과정이다.⁹⁾

컴퓨터프로그램을 개발하는 데 많은 시간과 노력이 소요되는 것과는 달리 컴퓨터프로그램은 전자적 형태로 이루어져 있기 때문에 거의 추가적인 노력없이

6) 수치 데이터, 문자 데이터, 음성 및 영상 등의 디지털 정보를 나타내기 위한 0과 1로 나타내는 방식이다. 컴퓨터는 소스코드 자체를 읽을 수 없어 소스코드를 컴파일하여 생성된 목적코드, 즉 2진언어를 통하여 작동이 가능하다.

7) 인간이 이해할 수 있는 명령문이나 기호를 사용하여 프로그램의 작성과 판독이 쉽도록 만들어진 프로그래밍 언어이다.

8) 이해완, 전게서, 851면.

9) 따라서 소스코드에 대한 보호뿐만 아니라 컴퓨터프로그램을 디자인하는 단계에서의 산출물에도 저작권을 부여하여야 한다는 주장이 있다. Julian Velasco, "The copyrightability of nonliteral elements of computer programs", *Columbia Law Review*, January 1994, pp.2-3.

이를 원본과 동일한 수준으로 복제하는 것이 가능하다는 문제로 인하여 권리자들은 컴퓨터프로그램 저작권의 강력한 보호를 주장하고 있다.¹⁰⁾ 다만, 그 구체적인 보호의 대상은 실질적 유사성 판단 대상에 관한 문제와 연결된다. 저작권 침해 소송에 있어서 실질적 유사성 판단은 저작권에 의하여 보호받는 표현으로만 그 대상을 한정하고, 두 비교의 대상물이 실질적으로 유사한 정도에 이르는지 여부를 판단하는 것이기 때문이다.

2. 실질적 유사성의 개념과 그 보호대상

저작권 제도의 핵심은 저작권에 의한 보호범위의 설정이며, 이에 큰 영향을 미치는 것이 저작권 침해의 범위 설정 문제와 저작물의 공정이용 범위의 설정 문제이다.¹¹⁾ 그러나 현재 우리나라의 저작권법은 저작물의 이용이 어떤 경우 공정이용에 해당하는지에 관하여는 구체적으로 열거하고 있으나,¹²⁾ 저작권 침해 문제와 관련해서는 저작권의 내용을 열거한 후 그에 따른 권리가 침해되었을 경우의 구제수단을 규정하고 있을 뿐이고, 저작권이 침해되지 않은 경우와 침해된 경우를 구분할 수 있는 어떠한 기준도 제시하지 않고 있다. 이는 완전한 복제(dead copy)의 경우에 있어서는 큰 어려움이 없으나, 침해물에 어느 정도의 수정이 가해진 경우에는 명문화된 기준이 없어 침해 여부에 대한 판단에 어려움이 있게 된다. 이러한 입법 형태는 실질적 유사성을 판단하기 위한 기준을 법규정에 일률적으로 정하는 것이 사실상 불가능하기 때문으로 볼 수 있다.¹³⁾ 이러한 점으로 인하여 저작권 침해를 판단하는 문제는 결국 법원의 영역으로 옮겨져¹⁴⁾ 판례를 통하여 그 기준과 방법이 논의되게 되었다.¹⁵⁾

10) 이러한 취지에서 기업의 무형자산인 컴퓨터프로그램의 저작권의 엄격한 보호의 목적과 더불어, 컴퓨터프로그램 개발도구 및 언어를 이용한 개작 및 증거인멸을 시도하는 등 점차 침해 여부를 밝혀내는 것이 어려워지는 문제를 해결하기 위해 소프트웨어저작권 전문과학수사기법의 개발이 필요하다고 하는 주장도 있다. "SW지식재산권의 중요성", <<http://www.etnews.com/2011032900087/>>, 검색일 : 2011.9.29.

11) 권영준, 저작권 침해판단론(실질적 유사성을 중심으로), 박영사, 2007, 75면.

12) 저작권법 제23조 내지 제38조 참조.

13) 권영준, 전거서, 75-76면.

14) 이러한 방식의 논의는 각 사건이 발생할 때마다 법원에 의하여 해결되어야만 하기 때문에 소송 진행에 따른 시간과 비용이 상당히 소요된다는 것이 단점이다.

컴퓨터프로그램의 구성은 문언적 요소(literal elements)와 비문언적 요소(non-literal elements)로 구분될 수 있다. 문언적 요소는 어문저작물의 텍스트와 유사하게 볼 수 있는데, 소스코드 및 목적코드와 같이 문자를 매개로 하여 작성된 부분이 이에 해당한다. 반면 비문언적 요소는 어문저작물에서의 등장인물의 관계, 구체적 줄거리 등 문언적 형태로 나타나지 않은 부분, 즉 문언적 요소 이면에서 나타나는 표현 부분을 말하는 것으로 컴퓨터프로그램에서는 호출관계 그래프, 기능구조 및 데이터베이스 구조 등이 이에 해당될 수 있다. 다만, 이 두 가지 요소는 서로 별개의 것으로 볼 수 없다. 예를 들면 문언적 요소인 소스코드 내에 비문언적 요소인 기능 구조가 녹아 있는 것과 같이, 결국 컴퓨터프로그램은 문언적 요소와 비문언적 요소가 유기적으로 결합하여 단일의 목적을 위해 기능하는 하나의 물(物)이라 볼 수 있기 때문이다.¹⁵⁾

컴퓨터프로그램 등의 저작권 침해소송에서 원고가 피고의 저작권 침해를 입증하기 위해서는 ① 원고가 유효한 저작권을 보유하고 있을 것, ② 피고가 원고의 저작물에 의거하여 만들었을 것, ③ 피고의 저작물이 원고의 저작물과 동일성 또는 종속성¹⁷⁾ 즉, 실질적 유사성을 갖고 있을 것이 입증되어야 한다.¹⁸⁾ 저작권 침해의 입증 요건 중 의거는 주관적 요건이라고도 하는데, 의거는 직접적으로 이를 입증하기 용이하지 않아 간접적인 방법에 의하는 경향이 짙다. 따라서 이를 입증하기 위해서는 피고가 원고의 저작물에 접근할 수 있었다는 가능성

15) 반면, 저작물의 공정이용에 대한 문제는 입법영역에서 다루어진다는 점이 대조적이다.

16) 이러한 구조는 어문저작물에 있어서도 유사한데, 서울중앙지법 2006. 6. 30. 선고 2005가단197078 판결에서는 “어문저작물에 대해서는 서로 두 가지 형태의 유사성을 생각해 볼 수 있다. 하나는 부분적·문언적 유사성(fragmented literal similarity)이고, 다른 하나는 포괄적·비문언적 유사성(comprehensive non-literal similarity)인데, 전자는 원고의 저작물 속의 특정한 행이나 절 또는 기타 세부적인 부분이 피고의 저작물에 복제된 경우를 말함에 비해, 후자는 피고가 원고의 저작물 속의 근본적인 본질 또는 구조를 복제함으로써 원·피고의 저작물 사이에 비록 문장 대 문장으로 대응되는 유사성은 없어도 전체적으로 포괄적인 유사성이 있다고 할 수 있는 경우를 말하는 바, 위 두 가지 유사성 중 어느 하나가 있는 경우에는 실질적 유사성이 있는 경우에 해당한다.”라고 하여 두 가지 요소로 구분하여 실질적 유사성을 판단하였다.

17) 동일성과 종속성(실질적 유사성)이 개념적으로는 구분되지만, 두 저작물이 동일할 경우에는 일단 객관적 요건의 입증이 문제되지 않는다는 점에서, 본 글에서는 이를 유사성의 범주로 포함하여 실질적 유사성에 대해 초점을 맞춰 논의하고자 한다.

18) 이해완, 전거서, 664면; 오승중, 저작권법, 박영사, 2007, 960면.

(접근가능성) 및 실제 원고의 저작물을 복제하여 이용¹⁹⁾하였을 것(유사성)이라는 요건의 충족이 요구된다. 이러한 요건에 의하여 원고와 피고의 저작물이 거의 동일하나 피고가 원고 저작물에 접근할 수 있는 가능성이 없었던 경우 또는 접근 가능성은 있었으나 사실상 두 저작물이 상당히 상이한 경우에는 의거가 있는 것으로 보기 어렵다. 그러나 두 저작물이 갖는 유사성이 우연의 일치나 공통의 소재 등으로는 설명되기 어렵고 피고가 원고의 저작물에 의한 것만으로 가능한 정도라면 이는 현저하게 유사(striking similarity)한 것으로 볼 수 있어 피고가 원고의 저작물에 의거하였다는 사실이 추정될 수 있다.²⁰⁾ 또한 상당한 정도의 유사성 이외에 공통적인 오류 등도 현저한 유사성을 입증하는 요소로 활용될 수 있다. 반면 실질적 유사성은 객관적 요건이라고도 하는데, 피고의 저작물이 원고의 저작물과 질적 또는 양적으로 사소한 정도를 넘어서서 유사한 정도가 실질적인 정도에 이르러야 두 작품의 실질적 유사성이 인정될 수 있다. 실질적 유사성은 의거요건에서 본 유사성(또는 현저한 유사성)과는 구별되는 개념인데, 의거요건에서의 유사성이 아이디어를 포함하는 저작물 전체를 대상으로 한다면, 실질적 유사성은 그 중에서 저작권법에 의해 보호받을 수 있는 창작적 표현만을 대상으로 하고 있다는 점에 차이가 있다.

완전히 동일하게 복제한 것이 아닌 원저작물에 수정을 가한 침해물에 대하여 과연 어느 정도 유사한 것을 실질적으로 유사하다고 할 수 있는지를 판단하는 것은 쉬운 일이 아니다. 실질적 유사성의 문제는 극단적으로 유사하지 않은 한 쪽 끝에서부터 완전히 문자적으로 동일한 다른 한 쪽 사이에 실질적 유사성의 경계를 긋는 선을 그리는 것이며,²¹⁾ 이 경계선은 저작권의 보호와 공정한 이용의 도모 사이에서 균형을 이루어야 한다는 점을 전제한다.²²⁾ 따라서 실질적 유사성의 판단, 즉 두 저작물이 어느 정도 유사한 것을 실질적으로 유사하다고 할 수

19) 여기서 복제하여 이용하였다는 것은 유사성을 비교해봄으로써 입증할 수 있으나, 이것은 실질적 유사성과는 구별되는 개념이다.

20) 오승중, 전제서, 970면.

21) Melville B. Nimmer & David Nimmer, *Nimmer on Copyright*, LexisNexis, 2010, pp.13-38.

22) Nicholas R. Monlux, "An Invitation for Infringement: How the ninth circuit's extrinsic and intrinsic similarity tests encourage infringement: An Analysis Using REECE v. ISLAND TREASURES ART GALLERY", *Journal of the Copyright Society of the USA*, vol.56(2009), pp.543-544.

있겠는가라는 질문에 대해 명확히 경계선을 긋는 것은 매우 추상적인 작업이며, 저작권법 체계 안에서 가장 어려운 문제 중 하나이다.²³⁾ 이러한 추상적 난해함은 실질적 유사성을 판단하는 방법에 대한 많은 논의를 가져오게 하는 원인으로 작용한다.²⁴⁾

Ⅲ. 우리나라 판례에서의 실질적 유사성 판단

1. 실질적 유사성이 인정된 경우

업무용통합프로그램 사건²⁵⁾에서 법원은 실질적 유사성을 판단하기 위한 구체적 기준을 제시하지는 않았으며, 감정결과를 근거로 실질적 유사 여부를 판단하였는데, 감정은 ‘접수·발송 프로그램의 소스코드 유사도’, ‘감정평가서 작성 프로그램의 소스코드 유사도’, ‘회계프로그램에 대한 소스코드 유사도’ 및 ‘회계프로그램에 대한 MSSQL DB 프로시저 유사도’에 대하여 감정을 수행하였다. 따라서 별지의 감정결과²⁶⁾를 바탕으로 원고의 컴퓨터프로그램과 피고가 배포한 컴퓨터프로그램은 실질적으로 유사한 것으로 판단하였다.

유통관리프로그램 사건²⁷⁾에서 법원은 컴퓨터프로그램저작권 침해에 해당하기 위해서는 ① 피고가 원고의 컴퓨터프로그램에 의거하여 개발하였을 것, ②

23) Melville B. Nimmer & David Nimmer, *supra* note 21, pp.13-37.

24) Nicholas R. Monlux, *supra* note 22, p.544; 정경석, “실질적 유사성론”, *계간저작권*, 2009. 3, 74면. 특히 논란이 있는 부분은 비문언적 요소들로 이루어진 저작물에 관한 ‘포괄적·비문언적 유사성’에 대한 것이다. 이는 컴퓨터프로그램저작물의 실질적 유사성을 판단하는 데 있어서 구조·흐름 및 조직 등의 유사성을 실질적 유사성을 판단하는 범주에 넣을 것인지와 관련하여 계속적인 논란이 존재한다. Roger E. Schechter & John R. Thomas, *Principles of Copyright Law*, Thomson Reuters, 2010, pp.371-382.

25) 서울중앙지방법원 2010. 12. 9. 선고 2009가합7224 판결.

26) 접수·발송 프로그램의 소스코드 유사도는 38.5%, 감정평가서 작성 프로그램의 소스코드 유사도는 96.9%, 회계프로그램에 대한 소스코드 유사도는 3.2% 및 회계프로그램에 대한 MSSQL DB 프로시저 유사도는 32.9%로 산출되었으며, 유사도는 $\frac{[(\text{전체 동일줄수} + \text{전체 유사줄수}) / \text{전체 대상줄수}] \times 100}{}$ 의 방식으로 산정되었다. 한국저작권위원회 감정임치팀-824(‘10.11.03) 참조.

27) 서울고등법원 2010. 6. 9. 선고 2009나47052 판결.

두 컴퓨터프로그램이 실질적으로 유사할 것을 요건으로 제시하고 있으며, 실질적 유사성은 감정을 통하여 제시된 정량적 유사도 등을 활용하여 판단되었다. 따라서 원고의 제1프로그램과 피고의 제1프로그램은 88.24%가 유사(의거 인정으로 저작권 침해 인정), 원고의 제2프로그램과 피고의 제2프로그램은 87.96%가 유사(의거 불인정으로 저작권 침해 불인정)하고, 원고의 제3프로그램과 피고의 제3프로그램은 92.01%가 유사(의거 인정으로 저작권 침해 인정)하여 실질적 유사성이 인정되었다.

마스터라우터프로그램 사건²⁸⁾에서 법원은 컴퓨터프로그램저작권 침해를 인정하기 위한 요건을 제시하지는 않았으며, 피고인이 3.04버전 컴퓨터프로그램을 사용함에 따라 4.0버전 컴퓨터프로그램의 저작권을 갖고 있는 고소인에 대한 저작권 침해가 인정된다고 판시하였다. 이때 두 컴퓨터프로그램 간의 실질적 유사성은 감정을 통해 제시된 97.19%의 정량적 유사도를 근거로 활용하였다.

코어뱅크프로그램 사건²⁹⁾에서 법원은 컴퓨터프로그램저작권 침해를 인정하기 위한 요건으로 의거성과 실질적 유사성을 제시하고 있다. 이에 피고가 원고의 컴퓨터프로그램인 Bancs에 접근한 사실로 인하여 의거요건을 인정하였고, 원고의 Bancs와 피고의 ProBank이 실질적 유사성이 있다고 판단하여 저작권 침해를 인정하였다. 실질적 유사성을 입증하기 위해서는 감정이 활용되었으며, 두 컴퓨터프로그램이 서로 다른 언어로 구현되어 있어 소스코드를 직접 문언적으로 비교하지 못하고 각 컴퓨터프로그램의 호출관계그래프(call graph)에 대한 정량적 유사성(각 소스코드 파일 중 호출관계그래프가 50% 이상의 유사성을 보이는 파일의 비율이 전체의 42.74%로 산출됨)³⁰⁾과 피고의 컴퓨터프로그램의

28) 인천지방법원 2011. 5. 13. 선고 2011노20 판결.

29) 대법원 2011. 6. 9. 선고 2009다52304, 52311 판결; 그러나 본 판결은 소스코드에 대한 정량적 비교가 이용된 것이 아니라는 점에서 대법원 판결임에도 불구하고 이를 정형화된 정량적 비교방식으로 보기에 는 다소 무리가 있다고 생각된다.

30) 호출관계그래프(call graph)는 실행 중에 소스코드에서 정의한 함수들이 서로 어느 함수를 언제 호출하는지를 표현한다. 따라서 소스코드의 구성이 동일하다면 정의된 함수사이의 호출관계는 유사하게 된다. 호출관계그래프의 유사도를 측정하기 위해서는 노드(node)의 깊이, 노드로 들어오는 간선(edge)의 개수, 노드(node)로부터 나가는 간선(edge)의 개수라는 호출관계그래프의 특성을 활용하여 이 세 가지 특성이 얼마나 많이 존재하는 것으로 유사도를 측정하게 된다. 이를 수식으로 나타내면 다음과 같다.

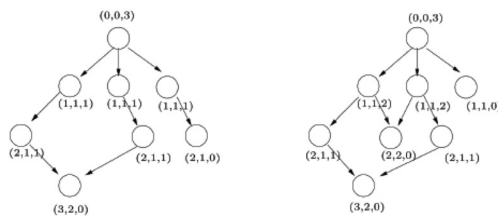
$$\text{유사도}(\%) = \frac{\text{그래프A와 그래프B의 노드 중 세가지 특성이 같은 노드의 수}}{\text{그래프A의 노드 개수} + \text{그래프B의 노드 개수}} \times 100$$

소스코드에 사용된 함수들의 이름과 명령문, 그리고 주석 내용으로 보아 이미 존재하는 COBOL 소스코드를 일정하게 정해진 규칙에 따라 사람 혹은 기계를 사용하여 번역한 것으로 판단³¹⁾된 정성적 결과를 종합하여 실질적 유사성이 인정되었다.

환경오염계측프로그램 사건³²⁾에서 법원은 “채무자는 채권자가 프로그램저작권을 갖고 있는 것으로 추정되는 이 사건 프로그램에 의거하여 그 표현형식을 복제한 구동 프로그램을 제작한 다음, 위 구동 프로그램을 채무자 제품에 장착하여 이를 제조, 판매하고 있다고 봄이 상당하므로, 특별한 사정이 없는 한 채무자는 채권자의 프로그램저작권을 침해하고 있다.”고 판시하였다. 이때 유사도는 감정에 의해 “채무자 제품의 롬(ROM)에 수록된 기계어를 분석한 바이너리(binary, 2진법) 파일과 채권자의 이 사건 프로그램의 바이너리 파일은 97.47%의 유사도”를 갖는다고 산출되었으며, 창작성에 대한 것은 고려하지 않았다.

건설장비통합관리프로그램 사건³³⁾에서 법원은 저작권 침해가 인정되기 위한 구체적인 기준을 제시하지는 않았으나, 감정을 통하여 도출된 채권자의 컴퓨터프로그램과 채무자의 컴퓨터프로그램 간 비교 결과를 통하여 실질적 유사성을 인정하였다. 이는 두 프로그램의 메인화면에서 메뉴구성 및 사용자인터페이

예를들면, 다음과 같은 호출구조그래프가 있다고 하면,



양쪽 그래프의 각 노드에 대해, 세 가지의 특성을 나타내는 순서쌍(노드의 깊이, 노드로 들어오는 간선의 수, 노드에서 나가는 간선의 수)을 비교해보면, (0, 0, 3)에 해당하는 노드가 각각 하나씩 존재하고, (2, 1, 1)에 해당하는 노드는 각각 둘씩 존재하며, (3, 2, 0)에 해당하는 노드는 각각 하나씩 존재한다. 따라서 위 호출구조그래프의 유사도는 다음과 같이 산출된다.

$$\text{유사도}(\%) = \frac{1 \times 2 + 2 \times 2 + 1 \times 2}{8 + 8} \times 100 = 50.0\%.$$

31) 다만 번역의 대상이 원고의 프로그램인지 여부는 확인되지 못하였다. 제2심 판결인 서울고등법원 2009. 5. 27. 선고 2006나113835,113842 판결 참조.

32) 서울고등법원 2008. 4. 11. 자 2006라124 결정.

33) 부산고등법원 2010. 2. 19. 자 2007라210 결정.

스가 매우 유사하며, 서브메뉴의 메뉴명이 거의 동일한 점, 클래스 파일의 경우 39.41% 유사한 점, 폼 파일의 경우 76.20% 유사한 점, 전체 소스코드가 62.17%의 유사도를 갖는다는 점 및 채무자 컴퓨터프로그램이 채권자의 컴퓨터프로그램을 기반으로 하여 추가적 개발로 작성된 사실 등³⁴⁾을 인정할 수 있다는 점을 근거로 하였다.

2. 실질적 유사성이 부정된 경우

밝기·색상보정프로그램 사건³⁵⁾에서 법원은 구체적으로 저작권 침해가 인정되기 위한 기준을 제시하지는 않았으나, 저작권 침해가 성립되기 위해서는 피해회사의 컴퓨터프로그램에 창작성있는 부분이 존재하고, 피고인의 컴퓨터프로그램이 피해회사의 창작성 있는 부분과 유사성³⁶⁾이 있어야 한다는 것을 판시하였다. 즉, 피고인의 컴퓨터프로그램이 피해회사의 컴퓨터프로그램과 상당한 정도의 유사한 면이 있음에도 불구하고³⁷⁾ 피해회사의 컴퓨터프로그램이 만들어질 당시 참조한 제3의 컴퓨터프로그램과 비교해 특별히 창작성 있는 부분이 추가되지 않음으로 판단되어 결과적으로 피고인이 침해한 것은 제3의 컴퓨터프로그램인 것이고, 피해회사의 컴퓨터프로그램의 저작권을 침해한 것은 아니라고 판단하였다.

34) '채무자 컴퓨터프로그램이 채권자의 컴퓨터프로그램을 기반으로 하여 추가적 개발로 작성된 사실 등' 이 본 판결에서는 실질적 유사성의 입증을 위한 근거로 제시되었으나, 오히려 의거요건을 입증할 수 있는 근거로 활용될 수 있는 것임을 볼 때, 본 판결에서 컴퓨터프로그램저작권 침해 입증을 실질적 유사성의 판단만으로 제시하는 것 같지만 이는 의거요건을 포함한 것으로 볼 수 있을 것이다.

35) 서울남부지방법원 2007. 10. 5. 선고 2004노1493 판결; 본 판결에서 컴퓨터프로그램저작권 침해와 관련된 사항은 상고가 이루어지지 않았고, 상고심(대법원 2008. 4. 24. 선고 2007도8814 판결)에서는 영업비밀 및 업무상 배임에 관한 문제가 다루어 졌다.

36) 실질적 유사성에 대한 언급은 없으나 창작성 있는 부분과의 유사성이라는 점은 개념상 실질적 유사성을 의미하는 것으로 생각된다.

37) 각 컴퓨터프로그램 사이의 유사성을 판단하기 위해서는 감정을 활용하였는데, 감정결과 제3의 컴퓨터프로그램과 피고인 컴퓨터프로그램은 94%, 제3의 컴퓨터프로그램과 피해회사 컴퓨터프로그램은 95% 그리고 피해회사와 피고인 컴퓨터프로그램은 90%가 유사한 것으로 보았으며, 특히 세 컴퓨터프로그램 간 주석은 45.3% 내지 68.2%가 유사한 것으로 나타났다. 컴퓨터프로그램보호위원회 감정평가팀-313('07. 8. 9) 참조.

버스회사ERP프로그램 사건³⁸⁾에서 법원은 컴퓨터프로그램저작권 등의 침해가 성립하기 위해서는 ① 저작권자의 동의나 허락없이 그의 저작물에 의거하여 침해저작물이 작성되어야 할 것, ② 침해저작물이 저작권자의 저작물 중 저작권의 보호대상이 되는 표현 부분을 이용하여야 할 것, ③ 저작권자의 저작물 중 저작권의 보호대상이 되는 표현과 침해저작물 사이에 동일성 내지 실질적 유사성이 인정될 것이라는 기준을 제시하였다. 이에 실질적 유사성을 판단하기 위해서는 구체적으로 외부에 표현된 창작적 표현에 해당되는 부분만을 가지고 비교되어야 한다고 전제하면서, “어문저작물 등의 경우와 마찬가지로 먼저 저작권이 침해되었다고 주장되는 컴퓨터프로그램 소스코드 중 창작성이 인정되는 부분을 특정한 다음, 그 부분과 저작권을 침해하였다고 주장되는 컴퓨터프로그램의 소스코드를 대비하여 양자 사이에 실질적 유사성이 인정되는지 여부를 살펴보아야 하며”, 컴퓨터프로그램의 특성상 외부적 제약요소를 고려해 표현의 다양성이 축소될 수 밖에 없는 점을 고려해야 한다고 하였다. 판단결과 두 컴퓨터프로그램의 소스코드가 0.23%만이 유사³⁹⁾하며, 유사한 부분도 프로그래밍의 관행에 해당하는 부분 및 국세청에서 제시한 계산방법에 해당하는 등 창작성을 인정할 수 없다고 하였다. 이외에도 몇 가지 사항에 대해 살펴본 결과⁴⁰⁾ 실질적 유사성을 인정할 수 없다고 판단하였다.

인터넷교환기프로그램 사건⁴¹⁾에서 법원은 컴퓨터프로그램저작권 등의 침해가 성립하기 위해서는 ① 채무자들 프로그램이 채권자 프로그램에 의거하여 제작되었는지 여부와 ② 실질적 유사성이 인정되는지 여부를 입증하여야 한다고 전제하였다. 여기서 의거여부는 상당한 접근의 기회가 있었는지 여부로 입증될

38) 서울고등법원 2010. 10. 13. 선고 2010나10261 판결.

39) 소스코드 및 데이터베이스의 유사도 등은 감정을 통하여 도출하였다.

40) 데이터베이스의 개별 칼럼은 데이터베이스의 기능적 요소에 불과하여 창작성을 인정하지 않았으며, 원고가 자신의 컴퓨터프로그램에서의 고유한 부분이라 주장하는, 데이터베이스의 테이블이나 소스코드의 함수들 중 사전정보를 Key_Code와 Key_Seq의 조합 구성으로 인식한다는 부분은 업무의 효율적 처리를 위한 아이디어에 속하며, 개별 테이블은 데이터베이스의 기능적 요소에 불과하여 창작성을 인정하지 않고, 차량일련번호 칼럼(CAR_SEQ) 또한 버스운수회사 업무 특성을 반영한 기능적 요소에 불과하여 창작성을 인정하지 않았다. 또한 날짜자동완성 함수도 매우 간단하여 누가 작성하여도 비슷할 수 밖에 없으므로 창작성이 인정되지 않았다.

41) 서울고등법원 2009. 8. 5.자 2008라1199 결정.

수 있으며, 실질적 유사성은 ① 저작권으로 보호받지 못하는 요소들을 제외한 다음, ② 남아있는 핵심적 요소(core)인 창작적 표현만을 비교하여 판단되어야 한다고 하였다. 다만 컴퓨터프로그램은 기능성과 논리성 때문에 표현의 다양성이 낮아 실질적 유사성의 범위를 다른 저작물에 비하여 좁게 보아야 한다고 전제하였다. 실질적 유사성의 입증을 위해서 감정이 활용되었는데, 감정결과 ‘호처리프로그램 부분’은 33.2%의 유사도, ‘미디어서비스프로그램 부분’은 48.0%의 유사도가 산출되었다.⁴²⁾ 이에 법원은 이러한 정량적 유사도에 관하여 공개소스코드를 제외하는 절차를 거쳤음에도 표준적 기능을 구현하기 위한 필수적 부분, 호환성을 위하여 필수적인 부분, 프로그래밍 관행으로 인한 부분 등으로 인하여 유사할 수 밖에 없는 소스코드를 여과하지 않았다는 점 등을 이유로 감정결과에서 제시된 유사도만으로 실질적 유사성을 인정할 수 없다고 판시하였다.

콘텐츠보안프로그램 사건⁴³⁾에서 법원은 컴퓨터프로그램저작권 침해를 입증하기 위해서는 원고 프로그램과 피고 프로그램의 창작적 표현형식 사이에 실질적 유사성이 인정되어야 한다고 전제하며,⁴⁴⁾ 실질적 유사성을 판단하기 위하여 감정을 활용한 결과 두 컴퓨터프로그램 간 실질적 유사성이 인정되지 않는다고 판단하였다. 이때 근거로는 구현방식의 차이, 구성형태, 기능의 세부적 구현방식, 유사 기능 간 소스코드의 평균 유사도가 10.7%인 점, 파일명이 동일한 파일 간 소스코드의 유사도가 20.58%인 점, 자동생성 부분을 제외한 소스코드는 불일치한 점, 논리적 복제도가 0%인 점, 가중치를 고려한 전체 유사도는 11.5%인 점 등을 고려하여 실질적 유사성을 인정하지 않았다.

형상관리솔루션 사건⁴⁵⁾에서 법원은 컴퓨터프로그램의 저작권 침해를 입증하기 위한 요건을 별도로 제시하고 있지 않다. 다만 감정결과에서 제시한 가중치를 고려한 종합유사도가 3.22%인 점 및 소스코드·사용자인터페이스·데이

42) 컴퓨터프로그램보호위원회 SW기술침해대응팀-56('08. 2. 1) 참조.

43) 서울중앙지방법원 2006. 9. 7. 선고 2004가합76119 판결.

44) 의거요건에 대해서는 명시적인 언급은 없었으나, 판결 전체의 내용을 보면 사실관계를 통하여 접근가능성이 충분히 인정될 수 있었기 때문에 이를 다시 언급하지 않은 것으로 생각된다.

45) 서울남부지방법원 2005. 11. 18. 선고 2005가합3367 판결.

터베이스 등에 있어서 일부 동일·유사한 부분이 존재하긴 하나 그 정도가 미약하여 비교대상인 두 컴퓨터프로그램은 상이⁴⁶⁾한 것이므로 저작권을 침해한 것이 아니라고 판단하였다.

SMS프로그램 사건⁴⁷⁾에서 법원은 구체적으로 저작권 침해가 인정되기 위한 기준을 제시하지는 않았으나, 신청인과 피신청인 양측의 컴퓨터프로그램의 저작권 침해 여부를 판단하기 위해 가장 중요한 기준이 되는 소스코드가 완전히 상이한 점(감정결과 유사도 0%), 환경설정 파일에 일부 유사한 점이 있기는 하나 그 유사도가 그리 높지 않은 점(유사도 26%), 호환성이 필수적인 요소인 점, 데이터베이스 구조 및 전송 프로토콜이 상당 부분 유사할 수 밖에 없는 점 등을 근거로 저작권 침해를 인정하지 않았다.

3. 정리

앞에서와 같이 컴퓨터프로그램 저작권 침해 사건의 판례를 살펴봄으로써 최근 우리나라 법원에서 컴퓨터프로그램의 실질적 유사성을 어떠한 방식을 통하여 판단하여 왔는지 그 흐름을 확인해 볼 수 있었다. 우리나라의 법원은 대체적으로 소스코드 등 문언적 요소를 서로 직접 비교할 수 있는 경우에는 정량적 유사도를 도출한 후 기타 표현요소들에 대한 검토를 종합하여 피고가 원고의 컴퓨터프로그램을 실질적으로 유사한 정도로 복제하였는지 여부를 판단하고 있는 것으로 보인다. 이때는 프로그래밍 언어의 표현이 다양하므로 문언적인 외형상 다르게 보이는 소스코드라도 실제로는 동일한 표현 범주에 해당하는 경우가 있을 수 있어 어문저작물에 비하여 판단의 방법이 어려우며 매우 분석적일 수 밖에 없다. 반면 상이한 프로그래밍 언어로 구현된 경우 혹은 침해당시의 컴퓨터프로그램의 확보가 용이하지 않아 소스코드 등의 직접적인 비교가 어려운 경우에는 미국의 3단계테스트를 차용하고 이를 일부 변형하여 실질적 유사성을 판단해온 것을 알 수 있다. 결국 우리나라는 컴퓨터프로그램의 실질적 유사성 판

46) '상이한'이라는 용어를 사용하였지만, 전체적 의미를 고려할 때 저작권을 침해하지 않았다는 의미로 이해된다.

47) 서울중앙지방법원 2005. 10. 13. 자 2004카합3789 결정.

단을 위해 문언적 표현에 대한 정량적 비교방식을 중심으로 정성적 요소들을 고려하고 있는 것으로 보이며, 비문언적 표현의 일부를 저작권법상 보호범위로 포함시켜 그에 대한 비교를 보완적으로 적용하고 있는 것으로 생각된다. 그러나 이 두가지 방식을 서로 상이한 것으로 볼 수는 없으며, 실제로는 3단계테스트를 광범위하게 활용하고 있는 것으로 볼 수 있다. 특히 비문언적 요소의 실질적 유사성의 판단은 물론이고, 정량적 유사도를 산정하는 과정에서도 추상화-여과-비교 과정을 충족시키도록 하고 있으며 여과가 충실히 이루어지지 않는 경우 정량적 유사도를 실질적 유사성 판단근거로 활용하지 않는 점에서 이를 알 수 있다. 다만 이러한 방법을 정형화된 기준으로 정립된 것이라고 보기에는 다소 어려움이 있는 것으로 생각된다. 판례에서는 구체적인 사건에 따라 각기 다른 방식을 통하여 실질적 유사성의 입증이 시도되기도 하며, 미국의 이론을 적용함에 따라 용어의 통일이 이루어지지 않았거나 각 요건을 변경하는 과정에서 구체적인 내용을 제시하지 못하는 경우 등의 이유가 존재하기 때문이다. 이러한 비정형성은 실질적 유사성 판단의 궁극적인 목표인 저작권 침해 여부 판단에 소요되는 비용과 시간의 증가, 법적 안정성 및 예측가능성을 저해할 수 있다는 점에 문제가 된다.

IV. 실질적 유사성 입증을 위한 정량적 유사도 검토

1. 컴퓨터프로그램의 실질적 유사성 판단 기준에 관한 미국에서의 논의

미국 연방법원은 컴퓨터프로그램의 실질적 유사성을 결정하기 위한 방법으로 여러 가지 방식을 적용⁴⁸⁾하여 왔으나 Whelan 판결⁴⁹⁾에서 제시된 기준과 Altai 판결⁵⁰⁾에서 제시된 기준이 대표적이다.⁵¹⁾

48) 이는 모두 아이디어·표현 이분법의 적용을 받는다.

49) Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 122(3d Cir. 1986).

50) Computer Associates International, Inc. v. Altai, Inc., 982 F.2d 693 (2nd Cir. 1992).

51) 이외에도 Lotus Development Corp. v. Borland International, Inc.(49 F.3d 807 (1st Cir. 1995)) 판결에서의 기준과 Brown Bag Software v. Symantec Corp.(960 F.2d 1465 (9th Cir. 1992) 판결에서의

1986년 제3항소법원의 Whelan 판결에서는 컴퓨터프로그램의 비문언적 요소를 면밀히 분석하여 실질적 유사성의 입증근거로 활용하는 방식을 처음 제시하였다. 본 사건에서 지방법원은 ‘파일구조, 화면출력절차 및 서브루틴의 일부’에 대한 두 프로그램 사이의 유사성이 있다는 원고측 전문가의 증언을 받아들여 두 프로그램의 실질적 유사성을 인정하였고, 피고가 원고의 프로그램에 접근하였었다는 사실에 기초하여 피고가 원고 프로그램의 공통요소를 복제하여 저작권을 침해하였다고 판단하였다. 이때 지방법원은 컴퓨터프로그램에서 저작권법에 의해 보호가능한 표현에는 소스코드만 있는 것이 아니라 컴퓨터프로그램을 작동하고 통제 및 관리하는 방법 역시 보호가능한 표현에 속한다고 판시하였다. 이에 피고는 지방법원의 판단은 소스코드를 비교하여 실질적 유사여부를 확인한 것이 아니기 때문에 법적 문제인 실질적 유사성이 인정된 것이 아니라고 주장하였으나, 항소법원은 이를 받아들이지 않고, 컴퓨터프로그램에 대하여 아이디어·표현 이분법을 적용함에 있어 저작권으로 보호받을 수 없는 아이디어는 컴퓨터프로그램의 목적 혹은 기능 그 자체가 해당할 뿐이고 컴퓨터프로그램에서 저작권으로 보호받을 수 있는 표현은 소스코드 뿐만이 아니라 그 컴퓨터프로그램의 목적이나 기능을 구현하기 위해 필수적인 것이 아닌 모든 것이 해당한다고 하였다.⁵²⁾ 따라서 원고의 컴퓨터프로그램의 목적은 치과 연구소의 조직을 효과적으로 운영하는 것이고, 원고의 컴퓨터프로그램의 목적을 구현하는 데 필수적인 것이 아닌 구체적 구조는 보호가능한 표현이라고 전제하고, 구조·처리의 흐름 및 구성을 보호대상으로 하여 평균적 일반인의 관점에서 직관과 느낌에 의하는 소위 ‘look and feel’ 방식을 통하여 실질적 유사성을 인정하였다.⁵³⁾

기준을 함께 제시하는 견해도 있다. Julian Velasco, *supra* note 9, pp.9-12. 이러한 기준들은 본래 컴퓨터프로그램 저작권 침해 소송에서 비문언적 표현 요소에 대한 실질적 유사성을 입증하기 위한 것들이나, 특히 Altai 기준 등은 후술하는 바와 같이 비문언적 표현 요소에 관한 것 뿐만 아니라 컴퓨터프로그램의 실질적 유사성 입증에 모두 활용될 수 있다고 보인다.

52) 즉 목적을 달성하는 다양한 방법이 있는 경우에는 그 가운데 특별히 선정된 방법이 그 목적을 위하여 필연적인 것이 아니므로 이는 아이디어가 아닌 표현에 속한다고 볼 수 있다. 김정원, “컴퓨터소프트웨어의 법적 보호”, 전남대학교 박사학위논문, 1990, 84면.

53) Whelan 판결에서의 기준은 컴퓨터프로그램에 대하여 가장 강력한 보호주의적인 경향을 보여주는 이론으로서 효율성 및 외부조건에 의하여 표현이 제한되는 컴퓨터프로그램을 창작자의 개성이 중시되고 광범위한 표현이 가능한 소설 등의 문헌저작물과 혼동하고 있다는 비판이 있다. 이해완, 전거서, 906-907면.

제3항소법원의 Whelan 판결의 기준은 컴퓨터프로그램에 대한 실질적 유사성 입증에 있어서 비문언적 요소를 어떻게 바라볼지에 대한 명확한 지침을 제공⁵⁴⁾하였다는 점에 의의가 있으나,⁵⁵⁾ 이후 제2항소법원의 Altai 판결에서 3단계 테스트(3-step test) 방식⁵⁶⁾이 Walker 판사에 의하여 제시됨에 따라 Whelan 판결에서의 기준인 look and feel 방식을 대체하게 된다.⁵⁷⁾

Altai 판결에서 Walker 판사는 Baker 판결⁵⁸⁾에서와 같이 컴퓨터프로그램의 기능에 필연적으로 수반되는 요소들은 보호대상이 아니라는 것을 전제하며, 컴퓨터프로그램 구조 등에 대한 실질적 유사성을 추상화(abstraction), 여과(filtration) 및 비교(comparison) 과정을 거쳐 판단하도록 제시하였다. 추상화 과정⁵⁹⁾은 프로그램의 소스코드로부터 시작하여 프로그램의 궁극적 기능에서 끝나는 것으로 역분석(reverse engineering)과 유사한 방식으로 비문언적 표현 요소를 구분하고, 아이디어와 표현을 구분하는 것이다. 여과 과정은 보호받지 못하는 표현으로부터 보호받는 표현을 분리하기 위한 것인데, 추상화를 거친 구조적 요소(structural components)들을 효율성에 의해서 지배되는 요소들(elements dictated by efficiency)인지, 외부요인에 의하여 지배되는 요소들(elements dictated by external factors)인지 및 공유영역으로부터 가져온 요소

54) "copyright protection of copmputer programs may extend beyond the programs' literal code to their structure, sequence and orgarnization," Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc., 797 F.2d 122(3d Cir. 1986).

55) Dun & Bradstreet Software Servs., Inc., v. Grace Consulting, Inc. 307 F3d 197, 206 & 214-16 (3d Cir. 2002); Lotus Dev. Corp. v. Borland Int'l, Inc., 799 F. Supp. 203, 215 (D. Mass. 1992).

56) 이는 '연속여과이론'이라는 용어(이해완, 전게서, 688면)로 칭하기도 하며, '추상화-여과-비교 테스트'로 칭하기도 하는(오승중, 전게서, 1020면) 등 정리된 번역용어가 없으므로 본 글에서는 원문의 표현을 고려하여 '3단계테스트'라는 용어로 사용하기로 한다.

57) Whelan 판결에서의 기준은 제2항소법원의 Altai 판결 이후 제3항소법원 이외의 대부분의 법원에서는 적용되지 않고 있으며(Gate Rubber Co. v. Bando Chem. Indus., 9 F3d 823, 840 (10th Cir. 1993); Sega Enters, Ltd. v. Accolade, Inc., 977 F2d (9th Cir. 1992); Plains Cotton Co-Op Ass'n v. Goodpasture Computer Serv., Inc. 807 F2d (5th Cir. 1987)), 많은 비판을 받고 있다, Andrew Chin, "Antitrust analysis in software product markets: A first principles approach", *Harvard Journal of Law & Technology*, vol.18(Fall 2004), p.49.

58) Baker v. Selden, 101 U.S. 99, 25 L.Ed. 841 (1879).

59) '추상화'는 Nichols 판결(Nichols v. Universal Pictures Co., 45 F.2d 119, 121 (2d, Cir. 1930))에서 Learnedhand 판사에 의해 처음 적용된 것으로서, 주로 어문저작물과 같이 문언적 표현에 대해 적용되는 것이었으나, Altai 판결에서 이를 컴퓨터프로그램저작물까지 확장하여 적용하였다.

들(elements taken from the public domain)인지 여부를 살펴보고 이를 제거한다. 이러한 요소들을 여과(제거)하는 과정을 거치게 되면 컴퓨터프로그램에서 보호받는 표현의 핵심(core)만 남게 된다.⁶⁰⁾ 이에 비교 과정에서는 핵심(core)에 해당하는 표현에 대해 피고가 보호받는 표현의 실질적인 부분을 복제 하였는지에 대해 판단하게 된다.⁶¹⁾

하지만 3단계테스트도 실제 사건에 적용함에 있어 드러나는 문제점이 있는데, 이는 철저한 여과(filtering) 과정의 수행으로 인하여 발생하게 된다. 만약 프로그래머가 보호받지 못하는 표현 요소들을 결합하여 컴퓨터프로그램을 작성한 경우에는 여과 과정을 거치면서 사용된 표현 요소가 모두 제거되어 결국 보호대상이 남지 않는 결과를 가져온다. 이것은 컴퓨터프로그램을 개발함에 있어서 오픈소스를 주로 사용하는 최근의 현상에 비추어 보면 앞으로 문제의 소지가 있다고 보인다. 왜냐하면 오픈소스도 타인이 창작한 것이므로 여과 과정을 거치게 되면 모두 제거되어야 하는 대상인데, 그렇게 되면 결국 보호받는 컴퓨터프로그램이 존재하지 않게 되기 때문이다.⁶²⁾ 그러나 이러한 단점에도 불구하고 이 이론은 외부적 제약요인, 효율성 및 공유영역에 의하여 저작권으로 보호받을 수 없는 부분들을 정밀하게 제거함으로써 컴퓨터프로그램 산업분야의 발전을 해하지 않을 수 있도록 하면서, 비문언적 표현 요소에 대한 보호뿐만 아니라⁶³⁾ 문언적 표현 요소 등 모든 표현적 요소에 적용할 수 있는 정밀한 이론이라는 점에서 상당히 합리적인 이론으로 보여진다.⁶⁴⁾

60) 이를 '컴퓨터프로그램의 golden nugget'으로 표현하기도 한다. David I Bainbridge, *Intellectual Property*, 5th edition, Pearson Education, 2002, p.201.

61) 일반적으로 컴퓨터프로그램을 제작하는 과정에서는 대체로 공유영역에 속하는 부분이 많을 수 밖에 없고, 점차 이러한 경향이 강해지고 있기 때문에 3단계테스트 방식이 컴퓨터프로그램의 저작권에 대해 약한 수준의 보호를 제공할 수 밖에 없다는 비판이 있다. *id.* p.202.

62) 이 문제를 해결하기 위해 3단계테스트의 시행 이후, 여과된 표현요소들을 다시 살펴보아 '편집물'로서 보호할 수 있는 가능성이 있는지를 검토하는 방법이 제기되기도 하였다. Julian Velasco, *supra* note 9, pp.18-19.

63) David I Bainbridge, *supra* note 60, p.201.

64) 이해완, 전제서, 909면; 오승중, 전제서, 1021면. 다만, 이 이론이 합리적이라는 점이 우리나라에 그대로 적용될 수 있다는 것을 의미하는 것은 아니며, 해당 이론을 우리나라에서 적용하기 위해서는 법적 배경 및 상황에 맞추어 조절이 필요할 수 있다고 생각된다.

2. 정량적 유사도의 개념 및 특징

정량적 유사도란, 두 컴퓨터프로그램의 소스코드 등을 분석하여 일정한 기준을 적용하여 비교한 결과를 수치로 나타낸 것을 의미한다. 정량적 유사도를 산정하기 위해서는 기본적으로 문자의 형태로 표현된 부분을 기본적 대상으로 하며, 컴퓨터프로그램의 경우에는 소스코드와 목적코드가 주요 대상이 된다.⁶⁵⁾ 이 방식은 두 컴퓨터프로그램 간 유사한 정도, 즉 피고가 원고의 프로그램을 어느 정도 복제하여 이용했는지를 객관화된 수치로 표현해준다는 점과 이러한 객관적인 근거의 제시에 법원이 저작권 침해 여부의 판단을 함에 있어서 불필요한 논쟁을 줄일 수 있고 어느 정도 일반화된 양적 기준을 도출하여 법적 안정성을 도모할 수 있다는 등의 장점이 있어서 실질적 유사 여부를 판단하는 근거로서 자주 사용되고 있다. 다만, 소프트웨어 공학적 관점에서 보면, 정량적 유사도를 산정하는 방식은 다양하게 존재할 수 있으나,⁶⁶⁾ 실질적 유사성을 판단하기 위해서는 그 속에 내포되어 있는 법적 개념과 요건들을 반영하여 기준 및 방법을 정립하고 정량적인 수치를 산출하는 것이 필요한데, 이를 적절히 반영하지 못한 상태에서 산출된 정량적 유사도는 실질적 유사성 판단에 대한 적절한 근거로 삼기 어렵게 된다.⁶⁷⁾ 따라서 이러한 경우에는 저작권 침해 요건 중 의거여부를 입증하기 위한 요건인 유사성의 근거가 될 뿐이지, 실질적 유사성의 근거로 직접 활용되지는 못한다.

이러한 특징을 갖는 정량적 유사도는 비전문가인 일반인에 의하여 산출되는 것이 현실적으로 불가능하므로, 컴퓨터프로그램 분야의 전문가에 의하여 수행되어지는 것이 적절하고,⁶⁸⁾ 우리나라 판례 역시 이러한 경향을 유지하고 있

65) 다만, 최근에는 문언적 표현 요소뿐만 아니라, 비문언적 표현 요소까지도 정량적으로 분석하려는 시도가 있으며, 대법원 2011. 6. 9. 선고 2009다52304, 52311 판결에서도 호출관계그래프를 대상으로 한 비문언적 표현 요소의 정량적 유사도를 인정하기도 하였다.

66) 소프트웨어 공학 분야에서 컴퓨터프로그램의 복제(유사)도는 컴퓨터프로그램 개발 과정에서 버전관리 차원에서 논의되었다. 이러한 단계에서는 유사 여부 보다는 단순히 기존 컴퓨터프로그램과 새로운 컴퓨터프로그램이 동일한지, 동일하지 않은 부분이 있다면 어느 부분인지를 구분하는 정도로 충분하였으나, 실질적 유사성을 판단하기 위한 정량적 비교에서는 두 컴퓨터프로그램이 동일하지 않다면 유사한 것에 해당하는지 여부가 고려되어야 한다는 점에 차이가 있다.

67) 서울고등법원 2009. 8. 5.자 2008라1199 결정 참조.

68) 전문가에 의한 방식의 가장 큰 장점은 침해자의 침해은닉 행위를 찾아낼 수 있다는 점이다. 최근에는 컴

다.⁶⁹⁾ 따라서 컴퓨터프로그램 분야의 전문가에 의해 수행되는 정량적 유사도의 산출은 일반적으로 소프트웨어 공학적 관점에서 이루어지게 되는 경향이 있다. 소프트웨어 감정⁷⁰⁾에서는 ‘두 프로그램 간의 상호 비교 및 분석을 통하여 두 프로그램이 어느 정도의 유사성을 가지는가를 판단하여 복제여부의 판단 근거를 제시’ 하는 것을 정량적 유사도를 판단할 수 있는 ‘복제(유사)도 감정’이라 정의하고 있다.⁷²⁾ 복제(유사)도 감정을 통하여 컴퓨터프로그램의 소스코드 레벨에서의 유사도, 자료구조의 유사도, 데이터베이스 유사도, 디렉토리 및 주요 파일 이름의 유사도 등 다양하게 선정된 항목의 유사도가 각각 백분율로 산출되며, 이때는 일반적으로 원본기준 방식을 채택한다.⁷²⁾ 그리고 산출된 각각의 유사도에 전체 컴퓨터프로그램에서 해당 부분이 차지하는 중요도를 가중치로 반영하여 종합 유사도를 산출하고, 그 결과를 컴퓨터프로그램의 복제여부 판단의 근거로 제시하게 된다.⁷³⁾ 복제(유사)도 감정은 물리적인 면과 논리적인 면으로 구분하여 이루어져야 한다고 한다. 물리적인 비교는 ‘구문(syntax)비교 방식’이라고도 하는데, 일반적으로 소스코드에서 토큰 및 라인(줄)을 추출하여 유사정도를 산출하는 방식이다.⁷⁴⁾ 논리적인 비교는 단순한 구문의 비교가 아닌 컴퓨터프로그

퓨터프로그램 저작권 침해에 있어서 단순한 복제 형태를 벗어나 ① 특정함수, 모듈을 여러 개로 분할하여 소스코드를 생성, ② 여러 함수, 모듈을 통합하여 소스코드 생성, ③ 기본 알고리즘은 같으나 유사한 기능을 하는 함수 호출, ④ 작성언어의 변경, ⑤ 컴퓨터프로그램의 중요 알고리즘은 같으나 화면 인터페이스 처리 부분의 변경 및 ⑥ 객체지향(object-oriented), 시각적(visual) 언어를 사용한 경우에 클래스 이름만 변경하여 사용하는 등의 방식이 이용되고 있다. 소정, S/W 감정도구 비교 및 활용방안에 관한 연구, 프로그램심의조정위원회, 2003, 16면.

69) 서울고등법원 2009. 5. 27. 선고 2006나113835,113842 판결; 서울고등법원 2010. 10. 13. 선고 2010나10261 판결; 서울고등법원 2009. 8. 5.자 2008라1199 결정; 부산고등법원 2010. 2. 19. 자 2007라210 결정; 인천지방법원 2011. 5. 13. 선고 2011노20 판결; 서울중앙지방법원 2010. 12. 9. 선고 2009가합7224 판결; 서울남부지방법원 2007. 10. 5. 선고 2004노1493 판결; 서울중앙지방법원 2006. 9. 7. 선고 2004가합76119 판결; 서울남부지방법원 2005. 11. 18. 선고 2005가합3367 판결; 서울중앙지방법원 2005. 10. 13. 자 2004가합3789 결정 등 참조.

70) 소프트웨어 감정은 컴퓨터프로그램의 저작권 침해 여부 등에 대한 분석과 판단을 통한 컴퓨터프로그램 저작권 분쟁에서 효과적인 증거제출을 목적으로 하고 있으며, 현재 한국저작권위원회 및 한국SW감정평가학회 등이 중심이 되어 이루어지고 있다.

71) 프로그램심의조정위원회, 유형별 감정기법 표준지침, 2005, 5면.

72) 원본기준방식은 원본의 얼마나 많은 부분이 복제되었는지를 기준으로 하는 방식이다.

73) 프로그램심의조정위원회, 전게서(2005), 5면.

74) 컴퓨터프로그램보호위원회, 2007 SW감정위킹그룹 연구결과보고서, 2007, 165면. 토큰비교방식은 토큰

램의 논리적 측면, 즉 논리구현 · 자료구조 및 메모리 구조 · 파일 입출력 양식 · 메시지 전달 양식 · 데이터베이스 구조 등을 비교하는 방식으로 ‘주관비교 방식’이라고도 한다.⁷⁵⁾

3. 정량적 유사도의 산출

컴퓨터프로그램을 분석 · 비교하기 위한 방법은 매우 다양하게 존재한다. 소스코드를 비교하는 경우에도 프로그램 언어 · 구현 기능 · 바이너리 형태 등 여러 환경에 따라 정량적 유사도를 산정하는 방법이 상이할 수 밖에 없다.⁷⁶⁾ 소프트웨어 감정 분야에서는 소스코드에 대한 정량적 비교는 일반적으로 ① 감정항목의 설정, ② 각 항목별 중요도 및 가중치의 산출, ③ 외부적 요소 및 공유영역에 있는 소스코드 제거, ④ 소스코드 라인(줄)을 기준으로 복제(유사)도 계산의 방식으로 이루어진다.⁷⁷⁾

첫째, 감정항목의 설정은 두 비교대상 컴퓨터프로그램을 어떻게 놓을 것인지를 정하는 것인데, 이때 어느 쪽을 기준으로 둘 것이냐에 따라 정량적 유사도는 달라지게 된다. 원고의 컴퓨터프로그램을 기준으로 하는 것을 원본기준 방식이라 하는데 이는 원고의 컴퓨터프로그램 중 어느 정도가 피고에 의해 복제되었는지를 산출하는 방식이고, 피고의 컴퓨터프로그램을 기준으로 하는 비교본기준 방식에서는 피고의 컴퓨터프로그램 중 어느 정도가 원고의 것을 복제한 것인지를 산출하는 방식이다. 따라서 원고의 컴퓨터프로그램이 피고의 것에 비해 양적

을 단위로 유사도를 산출하는 것으로서 각 토큰을 구성하는 문자별 유사도를 기준으로 유사 및 동일한 토큰을 판단하나 토큰간의 상관관계를 고려하기 어렵다는 단점이 있다. 반면 라인(줄)비교방식은 라인(줄)을 단위로 유사도를 산출하는 것으로서 각 라인(줄)에 포함된 토큰의 전체 유사도를 기준으로 유사 라인을 판정하며, 라인(줄)간의 상관관계나 토큰간의 상관관계를 고려할 수 있다는 장점이 있다.

75) 컴퓨터프로그램보호위원회, 상게서(2007), 170면.

76) C. K. Roy와 J. R. Cordy는 소스코드의 유사도를 산정하는 방식을 ① 텍스트 · 토큰비교, ② 매트릭스 비교, ③ 문법트리비교, ④ 의존도그래프비교, ⑤ 융합비교방식 및 ⑥ 데이터마이닝을 이용한 비교방식 등 6가지로 구분하고 있다. C. K. Roy & J. R. Cordy, *A Survey on Software Clone Detection Research, Technical Memorandum 2007-541*, School of Computing, Queen's University, 2007(이옥세, SW소스코드 유사성 비교 프로그램 개선에 관한 연구, 한국저작권위원회, 2010, 9-10면에서 재인용).

77) 여기서의 정량적 유사도 산정 방식은 한국저작권위원회에서 수행하고 있는 소프트웨어 감정을 기준으로 살펴보았다.

으로 크다면 원본기준 방식에서의 유사도는 적을 것이고, 피고의 컴퓨터프로그램이 원고의 것을 복제하고 별도로 추가한 부분을 상당히 포함시킨다면 비교본 기준 방식에서의 유사도가 적을 것이다. 따라서 정량적 유사도를 산출함에 있어서 어느 방식을 사용할 것인지가 적절히 고려되어야 할 것이다. 이에 관하여 생각해보면 저작권 침해소송에서 피고가 원고의 저작권을 침해하였는지에 대한 관점에서는 비교본기준 방식을 적용하는 것이 타당할 것으로 생각된다. 왜냐하면 저작권 침해의 개념이 피고가 자신의 작품을 작성함에 있어서 원고의 저작물을 침해로 볼 수 있을 정도로 복제하였는지를 판단하는 것이므로, 피고의 컴퓨터프로그램을 작성하는 과정에서 원고의 것을 복제한 정도가 어느 정도인지를 판단하는 방식이 이러한 개념에 부합하는 것으로 생각되기 때문이다. 다만, 저작권 침해로 인한 손해배상을 청구하는 경우에는 달리 생각될 수 있는데, 이때는 원고의 컴퓨터프로그램 중 피고가 복제한 부분이 어느 정도인지를 산출하는 원본기준 방식이 의미가 있을 것으로도 생각된다.⁷⁸⁾

둘째, 항목별 중요도 및 가중치를 산출하여 반영한다. 이는 실질적 유사성을 단순히 양적인 문제만이 아니라 질적으로도 상당한 부분을 복제한 경우를 고려하기 위함이다. 가중치에 대해서는 다음 절에서 상세히 살펴보기로 한다.

셋째, 비교대상이 되는 소스코드 등에서 외부적 요소 및 공유영역에 있는 표현을 제거한다. 이는 실질적 유사성의 판단 대상이 되는 보호받는 표현을 확정하는 과정으로서, 당해 컴퓨터프로그램이 목적으로 하는 기능을 구현하기 위해 반드시 동일하게 표현될 수 밖에 없는 부분 및 하드웨어 등의 환경에 의하여 동일할 수 밖에 없는 부분과 같이 외부적 요소에 의한 것과, 개발도구로 인하여 자동적으로 생성되는 부분, 오픈소스를 사용한 부분 및 프로그래밍 관행 혹은 언어적 특성으로 인하여 동일하게 표현될 수 밖에 없는 부분 등 공유영역에 해당하는 것을 비교대상에서 제외시키는 것이다.

마지막으로는 위의 과정을 거친 후 남은 부분만을 대상으로 하여 서로 비교하는 과정을 수행한다. 이 과정에서는 완전한 복제(dead copy)로 보이는 경우에는 큰 어려움이 없으나, 침해물에 어느 정도의 수정이 가해진 경우 이를 유사

78) 최경진, SW지적재산권 침해시 손해배상액 산정에 관한 연구, 컴퓨터프로그램보호위원회, 2007, 65-72면.

한 것으로 판단하기 위한 기준을 정하는 것이 쉽지 않다. 따라서 유사한 정도에 대한 정량적 계산을 위하여 적절한 기준을 정할 필요가 있다.⁷⁹⁾ 특히 컴퓨터프로그램의 소스코드는 어문저작물과 유사한 형태로 표현되어 있으나, 그 특성을 고려하여 비교 과정에서 라인(줄)을 단위로 비교할지 혹은 토큰을 단위로 비교할지 여부, 라인(줄)이 유사한 것으로 보기 위해서는 각 라인(줄)이 어느 정도 서로 유사해야 하는지 여부 등이 세부적으로 검토되어 정량적 유사도 산출이 이루어져야 할 것이다.

위와 같은 산출방식은 감정항목의 설정을 통하여 비교대상이 되는 컴퓨터프로그램 중 비교할 대상을 개발자의 창작적 표현의 범위가 넓은 것으로 특정하는 것, 외부적 요소에 종속적이거나 공유영역에 있는 소스코드를 제거하는 것 및 이 과정을 거쳐 남겨진 부분만을 비교의 대상으로 하여 비교하는 등 일련의 과정을 살펴보면, 미국의 제2항소법원의 Altai 판결에서 제시된 3단계테스트를 컴퓨터프로그램의 실질적 유사성을 위한 정량적 유사도 산출 방식에 적용한 것으로 보인다. 앞에서 살펴본바와 같이 우리나라 판례 역시 이러한 방식의 정량적 유사도 산출 방식에 대해서는 문제를 제기하지 않는 것으로 보여진다.⁸⁰⁾ 다만, 우리나라 판례에서는 주로 외부적 요소 및 공유영역에 있는 표현에 대한 제가 충분하게 이루어지지 않았다는 점이 정량적 유사도를 통한 실질적 유사성을 인정할 수 없게 되는 주요 원인이 되고 있다.⁸¹⁾

4. 실질적 유사성에 대한 적합성을 위한 검토

정량적 유사도는 기본적으로 컴퓨터프로그램 공학적 개념과 방식을 통하여

79) 이러한 기준에 관하여 ALC(Adjacent Line Comparison) 알고리즘을 활용하기도 하는데, 이는 인접라인의 유사율을 측정해서, 유사한 라인이 연속적으로 나타날 때, 가장 최대 블록의 라인끼리 유사하다고 맵핑해주는 알고리즘이다.

80) 서울중앙지방법원 2010. 12. 9. 선고 2009가합7224 판결; 서울고등법원 2010. 6. 9. 선고 2009나47052 판결; 인천지방법원 2011. 5. 13. 선고 2011노20 판결; 서울고등법원 2008. 4. 11. 자 2006라124 결정; 부산고등법원 2010. 2. 19. 자 2007라210 결정; 서울남부지방법원 2007. 10. 5. 선고 2004노1493 판결; 서울중앙지방법원 2006. 9. 7. 선고 2004가합76119 판결; 서울중앙지방법원 2005. 10. 13. 자 2004가합3789 결정 등 참조.

81) 서울고등법원 2009. 8. 5.자 2008라199 결정 등 참조.

이루어진다는 내재적 한계로 인하여 실질적 유사성을 판단하는 데 적절치 못한 정보를 제공하는 경우가 있을 수 있다. 이때 그 한계 및 문제점을 분명히 인지하지 못하고 정량적 유사도를 실질적 유사성 판단과정에 단순히 적용하게 되면 실제의 현상을 왜곡하는 결과를 초래할 수도 있게 된다. 이에 몇 가지 사항에 대하여 살펴해보도록 한다.

첫째, 정량적 유사도의 산정 방식에서 차용하고 있는 3단계테스트에 내재된 문제점으로 인한 것으로써, 정량적 유사도의 실질적 의미의 감소이다. 최근 컴퓨터프로그램 개발환경이 급속히 변화하고 있음으로 인하여 예전과 같이 프로그래머가 컴퓨터프로그램의 소스코드 등을 모두 직접 창작하지 않는 현상이 증가하고 있는데, 이는 프로그래밍의 자동화와 오픈소스(open source)의 사용이 큰 원인으로 나타난다. Visual Basic 등과 같이 사람이 직접 소스코드를 작성하지 않고도 프로그래밍이 가능한 프로그래밍 언어가 개발되고 널리 사용됨에 따라 소스코드에는 해당 도구에 의해 자동적으로 생성되는 소스코드가 많아지고 있다. 또한 오픈소스⁸²⁾는 개발자가 직접 창작한 것이 아니라 타인이 창작한 소스코드를 일정한 라이선스 정책에 따라 활용한 것으로서 최근에는 그 활용폭이 점차 넓어져 OS분야나 클라우드 컴퓨팅 분야에까지 그 적용이 확대되고 있다. 자동 생성된 소스코드 및 오픈소스는 이를 사용하여 컴퓨터프로그램을 작성한 사람에게 저작권을 인정할 수 없는 부분, 즉 공유의 영역에 해당하는 것으로서 보호받지 못하는 표현이다. 따라서 정량적인 비교를 수행하기 위해서는 반드시 제거(여과)되어야 하는 부분에 해당한다. 그런데 이러한 부분의 비중이 급격히 증가함에 따라 정량적 유사도의 실제 의미가 왜곡될 수 있는 위험이 발생한다. 예를 들어 100라인(줄)의 소스코드가 있다고 할 때, 이것이 모두 직접 창작된 것이라면 60%의 정량적 유사도는 100라인(줄) 중 60라인(줄)을 복제한 것으로서 볼 수 있다. 그러나 자동생성코드 및 오픈소스 등 제거될 부분이 50라인(줄)에 달한다면, 60%의 유사성이 있을 경우 실제 복제된 소스코드는 30라인(줄)에 해당하게 된다. 즉 동일한 정량적 유사도를 갖는 경우에도 실제 복제된 소스코드

82) 오픈 소스(open source)는 소프트웨어 혹은 하드웨어의 제작자의 권리를 지키면서 원시 코드를 누구나 열람할 수 있도록 한 소프트웨어 혹은 오픈 소스 라이선스에 준하는 모든 통칭을 일컫는다.
<<http://ko.wikipedia.org/>>, 검색일: 2011.9.24.

의 양은 상이하게 달라질 수 있다는 것이다. 따라서 같은 100라인(줄)의 규모를 갖는 컴퓨터프로그램에서 정량적 유사도가 모두 60%로 같다고 하더라도 이 두 가지 경우를 동일하게 판단할 수는 없을 것이다.

둘째, 소스코드의 양에 따른 왜곡 현상이다. 실무적으로 가장 범하기 쉬운 잘못이 정량적으로 몇 퍼센트 유사한 것이 실질적으로 유사한 것인가 라는 정량적 기준을 세우려는 행동이다. 정량적 유사도는 비율로서 제시되기 때문에 컴퓨터 프로그램에서의 실제 실질적으로 유사한 것으로 판단되는 소스코드의 양을 나타내는 것이 아니고, 단순히 그 비율을 제시하는 것에 불과하다. 결국 유사도만을 갖고는 피고가 복제한 소스코드의 실제 양을 알 수 없으며, 이는 별도로 검토되어야 하는 사항이다. 실질적 유사성을 판단하기 위해서는 실제 복제된 양적인 부분도 고려가 되어야 한다. 왜냐하면 같은 10%의 정량적 유사도를 갖는다고 하더라도 총 100라인(줄)의 소스코드에서의 10%와 총 10,000라인(줄)의 소스코드에서의 10%는 복제된 양에 상당한 차이가 있기 때문이다. 따라서 아무리 많은 양을 복제하였다 하더라도 당해 컴퓨터프로그램의 양적 규모가 크다면 상대적으로 유사도가 낮게 나와 실제 복제행위의 정도를 왜곡할 수 있는 것이다.

셋째, 가중치에 관한 문제이다. 실질적 유사성은 양과 질의 유사성이 실질적인가 하는 것으로서⁸³⁾ 복제된 부분이 전체 컴퓨터프로그램에서 어느 정도의 중요성을 갖는지도 고려되어야 한다. 컴퓨터프로그램의 정량적 유사도를 산출하기 위해서는 그러한 중요성을 가중치로 나타내어 이를 종합 유사도에 반영하게 된다. 그러나 가중치를 적용하는 점은 복제된 부분이 상당성을 충족하여야 하는지에 대한 문제로서 논란이 존재한다. 이 점에 있어서 Nimmer 교수는 실질적 유사성을 판단함에 있어 양과 질을 고려하여야 하고, 해당 저작물의 핵심부분을 복제하여 사용하면 그 복제된 양이 적더라도 이는 상당한 복제에 해당한다고 주장한다.⁸⁴⁾ 반면에 Goldstein 교수는 저작권 침해의 책임이 비중에 따라 영향을 받을 수는 없다고 하며, 단지 원고의 저작물이 중요성을 덜 갖는다고 하여 많은 부분을 복제한 피고에게 용서를 주는 것은 잘못된 것이라 주장한다.⁸⁵⁾ 컴퓨터프

83) 이해완, 전거서, 666면.

84) Roger E. Schechter & John R. Thomas, *supra* note 24, p.377.

85) Paul Goldstein, *Copyright*, 3rd ed, Aspen Pub, 2005, pp.7:28-29.

로그래밍은 기능성을 주요한 목적으로 하는 저작물이므로 질적인 고려라 함은 기능적 혹은 기술적 측면에서 중요한 부분을 지칭하는 것으로 생각하기 쉽다. 그러나 저작권 침해 판단 과정에 있어서 이러한 기능적 혹은 기술적 중요성이 큰 의미가 있는지는 다소 의문이다. 저작권은 창작성 있는 표현을 보호할 뿐이지 이를 창작하기 위한 노력을 보호하는 것은 아니라는 점과,⁸⁶⁾ 타인에 의하여 재산이 도난당했다면 이는 절도 행위 자체에 의하여 처벌이 되는 것이지 그 도난물의 가치가 낮다고 하여 절도한 사람의 책임이 감경되는 것이 아닌 점, 그리고 '질적'이라는 용어가 구체적으로 어떠한 것을 의미하는지가 분명하지 못한 점 등에 비추어 볼 때, Goldstein 교수의 견해가 좀 더 타당한 것으로 생각된다. 따라서 실질적 유사성에 있어 질을 고려한다는 점은 단순히 양적인 측면에서 유사 여부를 판단할 것은 아니라는 의미인 것이며, 정량적 유사도를 산출함에 있어서 기능적 중요도로서의 가중치를 반영하여 적용하는 것은 신중하게 접근해야 할 문제로 생각된다.

V. 결론

지금까지의 논의를 종합해 보면, 우리나라의 판례는 컴퓨터프로그램 저작권 침해 소송에서 두 컴퓨터프로그램의 문언적 표현요소인 소스코드 간 유사한 정도를 산술적으로 나타낸 정량적 유사도를 중요한 근거 중 하나로서 실질적 유사성 판단에 활용하고 있는 것으로 보인다. 또한 정량적 유사도는 추상화-여과-비교 과정으로 이루어진 3단계테스트 기준을 적용하여 산출되고 있는 것으로 보여지며, 이때 컴퓨터프로그램을 작성하는 데 영향을 미치는 외부적 효과 및 오픈소스 등과 같은 공유 영역의 표현 요소들의 제거가 무엇보다 중요하게 이루어진다. 하지만 일부 판례에서 볼 수 있는 바와 같이 컴퓨터프로그램에 대한 정

86) 미국에서는 Feist 판결(Feist Publications, Inc. v. Rural Telephone Service Co., 499 U.S. 340, 18 U.S. Q. 2d 1275(1991)) 이후 창작의 노력이 아닌 창작적 표현 자체를 보호하고자 하였으며, 우리나라도 노동이론의 적용 보다는 최소한의 창작성을 요건으로 하는 태도를 갖는 것으로 보인다(대법원 1995. 11. 14. 선고 94도2238 판결, 대법원 2010. 12. 23. 선고2008다44542판결, 대법원 2011. 2. 10. 선고2009도291판결 등 참조).

량적 유사도는 실질적 유사성을 입증하는 데 있어 일부 불완전한 점이 있을 수 있다. 이는 공학적 개념하에 산출된 정량적 유사도가 법적 개념인 실질적 유사성의 요건을 충족하지 못하는 경우가 있기 때문이다. 이에 실질적 유사성을 판단하기 위한 주요한 근거로서 효율적인 적용이 가능하기 위해서는 몇 가지 기초적 사항에 대해 검토가 이루어져야 했다.

따라서 본 논문에서는 최근의 컴퓨터프로그램 개발 추세와 여과의 진행으로 인하여 파생되는 정량적 유사도의 실질적 의미의 감소, 비율로서의 한계를 갖고 있는 정량적 유사도에 발생할 수 있는 양적 변화에 따른 유사도의 왜곡 현상 및 가중치의 적용에 관한 문제 등에 관하여 살펴보았다. 그러나 이러한 사항은 정량적 유사도를 활용한 컴퓨터프로그램의 실질적 유사성 판단을 위한 논의의 시작점에 해당한다. 따라서 향후에는 정량적 유사도 산정 결과가 활용되는 각 요건에서의 역할, 유사도가 의미하는 구체적인 법적 의미 및 정량적 유사도의 효과적인 활용방안 등에 대한 세부적인 논의가 필요할 것으로 생각된다. 이와 같은 논의가 컴퓨터프로그램의 저작권 침해 문제에 있어서 공학적 영역에서의 컴퓨터프로그램에 대한 문제를 법적 체계 안으로 적절하게 녹아들게 할 것으로 생각된다.

참고문헌

- 권영준, 저작권 침해판단론(실질적유사성을 중심으로), 박영사, 2007.
- 소 정, S/W 감정도구 비교 및 활용방안에 관한 연구, 프로그램심의조정위원회, 2003.
- 오승종, 저작권법, 박영사, 2007.
- 이육세, SW소스코드 유사성 비교 프로그램 개선에 관한 연구, 한국저작권위원회, 2010.
- 이해완, 저작권법, 박영사, 2007.
- 최경진, SW지적재산권 침해시 손해배상액 산정에 관한 연구, 컴퓨터프로그램보호위원회, 2007.
- 최경수 역, 저작물의 새로운 기술적 이용에 관한 국립위원회의 최종보고서(CONTU), 저작권심의조정위원회, 1994.
- 컴퓨터프로그램보호위원회, 2007 SW감정위킹그룹 연구결과보고서, 2007.
- 프로그램심의조정위원회, 유형별 감정기법 표준지침, 2005.
- David I Bainbridge, *Intellectual Property*, 5th edition, Pearson Education, 2002.
- Melville B. Nimmer & David Nimmer, *Nimmer on Copyright*, LexisNexis, 2010.
- Paul Goldstein, *Copyright*, 3rd ed, Aspen Pub, 2005.
- Roger E. Schechter & John R. Thomas, *Principles of Copyright Law*, Tomson Reuters, 2010.
- 정경석, “실질적 유사성론”, 계간저작권, 통권85권(2009. 3).
- 김정완, “컴퓨터소프트웨어의 법적 보호”, 전남대학교, 박사학위논문, 1990.
- Andrew Chin, “Antitrust analysis in software product markets: A first principles approach”, *Harvard Journal of Law & Technology*, vol.18(2004).
- Julian Velasco, “The copyrightability of nonliteral elements of computer programs”, *Columbia Law Review*, January(1994).
- Nicholas R. Monlux, “An Invitation for Infringement: How the ninth

circuit's extrinsic and intrinsic similarity tests e courage infringement:
An Analysis Using REECE v. ISLAND TREASURES ART GALLERY",
Journal of the Copyright Society of the USA, vol.56(2009).

Study on Quantitative Analysis for Substantial Similarity Determination of Computer Programs

Kim Si Yeol

Abstract

Substantial similarity plays a significant role in proving infringement in copyright infringement suits involving computer programs. A clear standard for demonstrating substantial similarity is difficult for things that are abstract such as computer programs. Thus continued discussions are taking place in order to establish the standard for determining substantial similarity when it comes to computer programs. Among elements that form a computer program the element that illustrates diversity in expression in the broadest sense is the composed source code. Contrary to other elements where expression has been limited due to pursuit of the highest functionality and efficiency, source code with diverse expression functions is the core element when it comes to determining infringement of computer programs. Korean cases emphasize source code as the most important comparison subject and there are substantial cases where a level of duplication involving source codes is used for determining substantial similarity. Quantitative similarity regarding source codes serves a pivotal role in determining substantial similarity for proving infringement of intellectual property rights involving computer programs. Thus this research studies problems so that quantitative similarity of source codes can have an adequate meaning for substantial similarity that serves as a legal concept.

Keywords

computer program, copyright infringement, substantial similarity, 3-step test, software appraisal(assessment), source code