

논문 2021-1-2 <http://dx.doi.org/10.29056/jsav.2021.06.02>

대용량 파일 전송 소프트웨어의 동일성 감정 방법

전병태*†

Appraisal Method for Similarity of Large File Transfer Software

Byung-Tae Chun*†

요 약

정보통신의 발달로 인하여 소프트웨어의 중요성이 증대되고 있으며, 이에 따른 소프트웨어 저작권 분쟁도 증가하는 추세에 있다. 본 논문은 제출된 프로그램들의 소스와 관련하여 프로그램 수행에 필요한 파일들을 감정범위로 하였다. 분석 대상인 대용량 파일 전송 솔루션 프로그램은 데이터에 대한 전자서명 및 암호화를 통하여 기밀성, 무결성, 사용자 인증, 부인방지 기능 등의 부가 기능을 제공하고 있다.

본 논문에서는 프로그램 A, 프로그램 B, 프로그램 C 3개에 대하여 분석을 수행한다. 프로그램 유사율을 산출하기 위하여 다음과 같은 내용을 분석한다. 패키지의 구조, 패키지 이름, 각 패키지 내 소스파일 이름, 소스파일 내 변수명, 함수명, 함수구현 소스코드, 제품의 환경변수 정보에 대하여 유사 여부를 분석하고 프로그램의 전체 유사율을 산출한다. 패키지 구조 및 패키지 이름이 일치되는 정도를 확인하기 위해, 폴더 구조를 비교하여 유사도 판단을 하였다. 또한 패키지 구조 및 패키지 이름이 어느 정도 일치하는지와 각 패키지 내 소스 파일(클래스) 이름이 어느 정도 일치하는지에 대한 분석을 하였다.

Abstract

The importance of software is increasing due to the development of information and communication, and software copyright disputes are also increasing. In this paper, the source of the submitted programs and the files necessary for the execution of the program were taken as the scope of analysis. The large-capacity file transfer solution program to be analyzed provides additional functions such as confidentiality, integrity, user authentication, and non-repudiation functions through digital signature and encryption of data. In this paper, we analyze the program A, program B, and the program C. In order to calculate the program similarity rate, the following contents are analyzed. Analyze the similarity of the package structure, package name, source file name in each package, variable name in source file, function name, function implementation source code, and product environment variable information. It also calculates the overall similarity rate of the program. In order to check the degree of agreement between the package structure and the package name, the similarity was determined by comparing the folder structure. It also analyzes the extent to which the package structure and package name match and the extent to which the source file (class) name within each package matches.

한글키워드 : 대용량 파일, 소프트웨어 감정, 유사성 분석, 패키지 구조 분석, 프로그램 유사성

keywords : large file, software appraisal, similarity analysis, package structure analysis, program similarity

* 한경대학교 컴퓨터응용수학부

접수일자: 2021.05.31. 심사완료: 2021.06.03.

† 교신저자: 전병태(email: chunbt@hknu.ac.kr)

게재확정: 2021.06.20.

1. 서론

정보통신의 발달로 인하여 소프트웨어의 중요성이 증대되고 있으며, 이에 따른 소프트웨어 저작권 분쟁도 증가하는 추세에 있다. 소프트웨어 분쟁은 저작권법으로 보호[1-3]가 되고 있다.

본 논문은 대용량 파일 전송 소프트웨어의 동일성 분석[4][5]에 관한 연구이다. 프로그램 A, 프로그램 B, 프로그램 C(국가기관 제출 프로그램)에 대하여 패키지 구조 및 패키지 이름, 각 패키지 내 소스파일(클래스) 이름, 소스파일 내 변수명, 함수명, 함수구현 소스코드, 제품의 환경변수 정보에 대하여 유사한지 여부를 분석하고, 3개 프로그램의 전체 유사율을 산출하였다. 패키지 구조 및 패키지 이름이 일치되는 정도를 확인하기 위해, 폴더 구조를 비교하여 유사도 판단을 하였으며, 프로그램 소스코드 내에 사용된 패키지 이름을 발췌하여 유사율을 산정하였다.

2. 감정 프로그램의 분석

분석대상 프로그램들은 데이터에 대한 전자서명 및 암호화를 통하여 기밀성, 무결성, 사용자인증, 부인방지 기능 등의 부가기능을 제공하는 대용량 파일 전송 솔루션으로 설계되어 있다.

분석의 대상이 되는 분석 목적물은 모두 3개이며 프로그램과 A와 프로그램 B 및 국가기관에 보관되어 프로그램 C가 제출되었다. 프로그램과 A와 프로그램 C은 class 형식으로, 프로그램 B은 java 파일형태로 되어 있다.

제출된 자료를 살펴보면 표1과 같이 프로그램 A에는 class 파일 148개, 환경변수 1개, 프로그램 설명서 1개가 포함되어 있으며, 프로그램 C에는 155개의 class 파일, 프로그램 B의 경우는 137개의 java 파일이 존재하였다.

표 1. 비교 대상 소프트웨어

Table 1. Compared software

구분	분석대상	크기 (Byte)	폴더 수	파일 수	내용
프로그램 A	AT 프로그램	706,980	42	150	1. AT 프로그램-class 파일 148개 2. 환경변수 3. 프로그램설명
프로그램 C	MT 프로그램	819,722	44	165	1. MT 프로그램-class 파일 155개
프로그램 B	MT 프로그램	1,233,483	86	312	1. MT 프로그램-java 파일 137개

3. 감정 내용 및 방법

3.1 감정 분석 사항

패키지 구조 및 패키지 이름이 어느 정도 일치하는지와 각 패키지 내 소스파일(클래스) 이름이 어느 정도 일치하는지에 대한 분석을 수행 한다 [6-8].

3.2 감정 방법

유사율 산정에 있어서 한 프로그램이 다른 프로그램을 복제했다고 가정할 때, 두 프로그램 중 어느 프로그램을 기준으로 분석을 수행하느냐에 따라 그 결과가 달라질 수 있다[9][10].

프로그램 B가 프로그램A을 복제한 경우를 가정한 것이다. 그림에서 X는 원본과 비교본의 공통기능이 구현된 부분이며, Y는 새로 추가된 부분이다. 그리고 X영역의 진한부분(X')은 비교본이 원본을 복제한 부분을 나타낸다. 이 경우 복제도 계산방식은 그 기준이 원본인지 또는 비교본인지에 따라 다음의 두 가지 방식으로 나누어진다. 전체영역 기준 방식과공통영역 기준 방식

이다. (1) 원본 기준 방식 : 원본의 얼마나 많은 부분이 복제되었는지를 기준으로 한다. 이 방식은 비교본이 독자적으로 추가한 부분이 있더라도 그 많고 적음에 영향을 받지 않지만 비교본의 크기가 작아서 원본의 일부 부분만을 복제한 경우에는 유사도가 낮아질 수 있다. (2) 비교본 기준 방식 : 비교본의 얼마나 많은 부분이 원본으로부터 복제된 것인지를 기준으로 한다. 이 방식은 비교본이 독자적으로 추가한 부분이 많아질수록 유사도는 낮아진다. 본 연구에서는 분석목적물간 요청된 분석사항들에 대하여 각 프로그램들을 비교하여 일치여부와 유사도를 확인하였다. 비교방식은 프로그램 A와 프로그램 C, 프로그램 A와 프로그램 B, 프로그램 C와 프로그램 B를 각각 비교 하였다.

바 소스는 다음과 같이 작성된다.

```

[] java_prog
- bin
- src
  - org
    - openprog
    - javaprogram
      - example
    
```

그림 1. 디렉토리 구조

Fig. 1. directory structure

4. 소프트웨어 일치성 분석

4.1 패키지 구조 및 패키지 이름 일치성 분석

패키지란 윈도우 폴더에서 파일을 분류하듯이 자바 프로그래밍을 하며 클래스를 그룹별로 분리한 것을 말한다. 일반적으로 패키지 이름을 지을 때 “com.XXX.이름”의 형태로 짓는데 패키지를 사용하는 이유는 중복을 피하기 위한 목적이다. 일반적으로, 다른 사람이 개발한 클래스를 사용해야 하는 경우 자신이 만든 클래스와 동일한 이름의 클래스가 존재할 때 중복문제가 발생된다. 따라서 이를 방지하기 위해 패키지 이름 자체를 고유하게 만들어주면 자동으로 패키지 아래에 있는 클래스의 이름도 고유해지게 된다. 패키지 이름이 ‘KKK.com.name’의 형태를 가질 때 그 폴더 구조는 그림 1과 같다.

그림 2의 패키지 구조를 보면 폴더구조와 대응 관계가 있는데 패키지의 구조대로 src 하위에 폴더를 만들고 해당 폴더에 ‘aaa.java’를 만들면 자

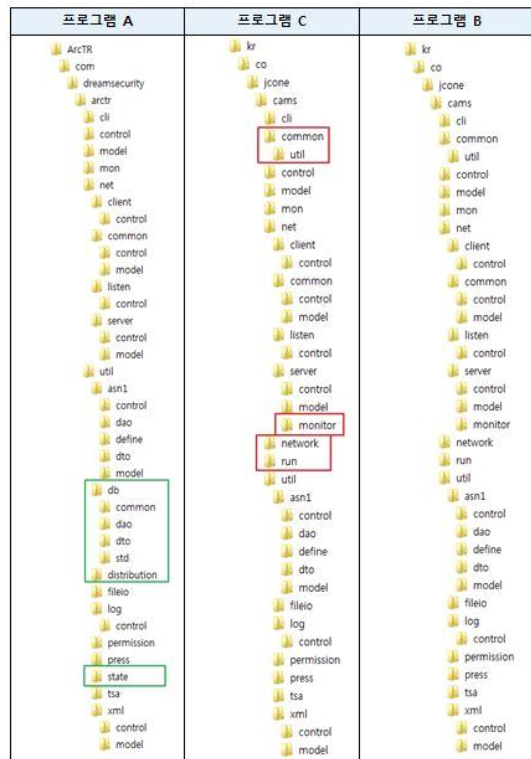


그림 2. 프로그램 A, 프로그램 C, 프로그램 B 패키지 구조 비교

Fig. 2. Comparison of package structure for program A and program C and program B

본 연구에서는 프로그램 A와 프로그램 B의 패키지 구조와 패키지 이름의 일치여부를 확인하였다. 각종 소스나 파일이 저장되는 위치가 패키지의 이름과 큰 연관성을 가지고 있고 해당 자료는 폴더에 저장된다. 따라서 패키지 구조의 유사성을 확인하기 위해 그림 2와 같이 프로그램 폴더 구조를 살펴봄으로써 패키지 구조의 유사도를 판정하였다. 또한 그림 2와 같이 프로그램 A와 프로그램 B의 패키지 구조는 유사한 구조로 되어 있음을 확인할 수 있다.

유사도 산정결과, 표 2와 같이 프로그램 A와 프로그램 C간의 패키지 이름의 유사도는 원본 기준 75.86%, 비교본 기준 81.48%로 산출되었고, 프로그램 A와 프로그램 B와 간의 패키지 이름의 유사도는 원본 기준 75.86%, 비교본 기준 81.48%로 산출되었다. 또한 프로그램 C와 프로그램 B간의 패키지 이름의 유사도는 원본과 비교본 모두 100%로 산출되었다.

표 2. 패키지 이름에 대한 비교 유사도
Table 2. Comparative similarity for package names

원 본	비교본	유사 패키지 수	기 준	유사도 (%)
프로그램 A -패키지 29개	프로그램 C -패키지 27개	22개	원 본	75.86
			비교본	81.48
프로그램 A -패키지 29개	프로그램 B -패키지 27개	22개	원 본	75.86
			비교본	81.48
프로그램 C -패키지 27개	프로그램 B -패키지 27개	27개	원 본	100
			비교본	100

패키지 이름의 유사도는 표 3과 같이 각 프로그램에서 사용된 패키지 이름을 모두 발췌한 후 유사도를 산술적으로 계산하였다. 프로그램 C와 프로그램 B의 패키지 이름은 완전 일치하였다.

표 3. 프로그램 A와 프로그램 B의 이름 유사도 비교

Table 3. Comparison of name similarity between program A and program B

no	Program A	Program C = Program B	유사 여부
1	com.aa.ar.cli:	bb.co.jp.cs.cli:	유사
2	program B 자료에만 존재	bb.co.jp.cs.common.util:	-
3	com.aa.ar.control:	bb.co.jp.cs.control:	유사
4	com.aa.ar.model:	bb.co.jp.cs.model:	유사
5	com.aa.ar.mon:	bb.co.jp.cs.mon:	유사
6	com.aa.ar.net.client.control:	bb.co.jp.cs.net.client.control:	유사
7	com.aa.ar.net.common.control:	bb.co.jp.cs.net.common.control:	유사
8	com.aa.ar.net.common.model:	bb.co.jp.cs.net.common.model:	유사
9	com.aa.ar.net.listen.control:	bb.co.jp.cs.net.listen.control:	유사
10	program B 자료에만 존재	bb.co.jp.cs.net.server.control:	
11	com.aa.ar.net.server.model:	bb.co.jp.cs.net.server.model:	유사
12	com.aa.ar.util.asn1.control:	bb.co.jp.cs.util.asn1.control:	유사
13	com.aa.ar.util.asn1.dao:	bb.co.jp.cs.util.asn1.dao:	유사
14	com.aa.ar.util.asn1.define:	bb.co.jp.cs.util.asn1.define:	유사
15	com.aa.ar.util.asn1.dto:	bb.co.jp.cs.util.asn1.dto:	유사
16	com.aa.ar.util.asn1.model:	bb.co.jp.cs.util.asn1.model:	유사
17	com.aa.ar.util.db.common:	program A 자료에만 존재	-
18	com.aa.ar.util.db.dao:	program A 자료에만 존재	-
19	com.aa.ar.util.db.dto:	program A 자료에만 존재	-
20	com.aa.ar.util.db.std:	program A 자료에만 존재	-
21	com.aa.ar.util.distribution:	program A 자료에만 존재	-
22	com.aa.ar.util.fileio:	bb.co.jp.cs.util.fileio:	유사
23	com.aa.ar.util.log.control:	bb.co.jp.cs.util.log.control:	유사
24	com.aa.ar.util.log:	bb.co.jp.cs.util.log:	유사
25	com.aa.ar.util.permission:	bb.co.jp.cs.util.permission:	유사
26	com.aa.ar.util.press:	bb.co.jp.cs.util.press:	유사
27	com.aa.ar.util.state:	program A 자료에만 존재	-
28	com.aa.ar.util.tea:	bb.co.jp.cs.util.tea:	유사
29	com.aa.ar.util.xml.control:	bb.co.jp.cs.util.xml.control:	유사
30	com.aa.ar.util.xml.model:	bb.co.jp.cs.util.xml.model:	유사
31	com.aa.ar:	program A 자료에만 존재	-
32	program B자료에만 존재	bb.co.jp.cs.net.server.monitor:	-
33	program B자료에만 존재	bb.co.jp.cs.network:	-
34	program B자료에만 존재	bb.co.jp.cs.run:	-

4.2 패키지 내 클래스 이름 일치성 분석

자바 프로그램은 표 4와 같이 클래스(Class) 기반의 언어로 모든 것들이 클래스 위주로 작성되며, 클래스란 유사한 특징을 지닌 객체들의 속성을 묶어 놓은 집합체를 말한다. 본 연구에서는 패키지의 각 폴더별로 소스파일(클래스) 이름의 일치 여부를 확인하였다. 소스파일의 이름이 일치하는 경우와 접두어 등의 명명규칙이 유사한 경우는 유사한 것으로 판정하여 유사도를 계산하였다.

표 4. 패키지 내 클래스 명
Table 4. class name in package

Program A	Program C
<pre> package com.dreamsecurity.actr.net.common.control; import com.dreamsecurity.actr.net.common.control.ABCTCCollection; public class ABCTCConnOut implements ABCTCConnOut, ABCTCConnOut { private String sTargetIp = null; private final int sLEN_RECVBUF_OUTSIDE = 4; private final int sLEN_RECVBUF_IN = 4; private ByteBuffer bbRecvBuf = null; private ByteBuffer bbRecvContentBuf = null; private ByteBuffer bbSendBuf = null; private SocketChannel sockChannel = null; private String sThreadID = null; public ABCTCConnOut() } </pre>	<pre> package kr.co.yoon.cas.net.common.control; import java.io.ByteArrayOutputStream; public class MCTTCConnOut implements ABCTCConnOut, ABCTCConnOut { private String sTargetIp = null; private int sLenRecv = 0; private final int sLEN_RECVBUF_OUTSIDE = 4; private final int sLEN_RECVBUF_IN = 4; private ByteBuffer bbRecvBuf = null; private ByteBuffer bbRecvContentBuf = null; private byte[] recvContentBuf = null; private ByteBuffer bbSendBuf = null; private byte[] sendBuf = null; private SocketChannel sockChannel = null; private String sThreadID = null; private Socket socket = null; private OutputStream os = null; private OutputStream out = null; public MCTTCConnOut() } </pre>

프로그램 A의 클래스는 128개, 프로그램 C 클래스는 132개, 프로그램 B의 클래스는 133개로 구성되어 있었으며, 이들 간의 유사도를 비교한 결과는 표 5와 같이 나타났다.

표 5. 클래스 이름 비교 유사도
Table 5. Compared software

원본	비교본	유사 클래스 수	기준	클래스 이름 유사도
프로그램 A -클래스 128개	프로그램 C -클래스 132개	107개	원본	83.59%
			비교본	82.95%
프로그램 A -클래스 128개	프로그램 B -클래스 133개	107개	원본	83.59%
			비교본	82.31%
프로그램 C -클래스 132개	프로그램 B -클래스 133개	132개	원본	100%
			비교본	99.27%

프로그램 C(국가기관 제출, 132개)과 프로그램 B(제출, 133개)의 경우는 132개 클래스 파일 이름이 동일하였으며 프로그램 B에 1개의 파일(runSendCompleteMove.java)이 추가되어 있음을 확인하였다

5. 결론

본 감정은 프로그램 A와 프로그램 B, 그리고 프로그램 C의 소스코드 동일·유사 정도 및 프로그램 구조·변수 등의 일치 여부 등을 감정하였다. 프로그램 A와 프로그램 C 및 프로그램 B 간의 패키지 구조 및 패키지 이름, 클래스 이름 유사성 비교결과, 대부분이 일치하거나 유사함을 확인하였다.

참고 문헌

- [1] 김시열, 강윤수 “소프트웨어 감정의 간이절차 활용 논의에 대한 검토”, 한국소프트웨어 감정평가학회 논문지 제15권 제1호, pp. 25-34. 6월, 2019년. DOI: <http://dx.doi.org/10.29056/jsav.2019.12.04>
- [2] 김시열, 컴퓨터프로그램 저작권 유사도론, 세창출판사, ISBN : 978-89-8411-754-9 2018.
- [3] 전병태, “SNMP MIB 파일의 유사도 분석에 관한 연구”, 한국소프트웨어 감정평가학회 논문지 제15권 제1호, pp.25-34. 6월, 2019년. DOI: <http://dx.doi.org/10.29056/jsav.2019.06.04>
- [4] Robert C. Osterberg and Eric C. Osterberg, “Substantial Similarity in Copyright Law”, Practising Law Institute New York City, 2018. ISBN-13: 978-1402403415 ISBN-10: 1402403410
- [5] 저작권 소송과 소프트웨어 포렌식, (사)한국소프트웨어감정평가학회, 마지원(출판사), 2021. ISBN : 9791188127856

- [6] Maybury M. (ed). Intelligent multimedia interfaces. AAAI Press, Menlo Park, 1993
- [7] exEyes, <http://www.copyright.or.kr>, 2015.
- [8] Black Duck, “The tenth annual future of open source survey”, Black Duck Software <http://www.blackducksoftware.com/2016-future-open-source>, Oct. 2016.
- [9] 이규대, “유사성 비교에서 세부항목 설정 기준”, 한국소프트웨어감정평가학회 논문지, 12권 1호, pp.21-26, June, 2016. <http://www.i3.or.kr/html/paper/2018-1/2016-1.html>
- [10] 이규대, “임베디드 기기 감정에 디바이스드라이버 유사성 설정”, 한국소프트웨어감정평가학회 논문지, 14권 1호, pp.27-32, June, 2018.

저자 소개



전병태(Byung-Tae Chun)

2001년 2월 고려대학교 박사
 1989~1996년 한국과학기술연구원(KIST) 선임연구원
 1996~2004년 한국전자통신연구원(ETRI) 책임연구원
 2004.2.~현재 국립 한경대학교 컴퓨터응용수학부 교수
 2013.8.~현재 한국저작권위원회(KCC) SW감정전문위원
 2019.1~현재 한국지식정보학회(KKITS) 논문지편집위원장
 2014.1~현재 TTA/TC4/JTC1분과 부위원장
 2011.2~현재 (사)한국SW감정평가학회 이사
 2015.12.10. 제10회 대한민국 로봇 대상 산업통상자원부 장관상

<주관심분야> 영상처리, 로봇 HRI, 인공지능, SW감정분야