논문 2023-1-4 http://dx.doi.org/10.29056/jsav.2023.3.04

소스 코드가 표준기술규격에 의하여 구현된 부분과 비율에 대한 감정

전병태*†

Assesment of the Part and Ratio of the Source Code Implemented in Accordance with the Standard Technical Specification

Byung Tae Chun*†

요 약

표준기술규격서는 설계, 개발 단계의 참고사항이며 개념도 및 절차를 도식화하고 있다. 규격서에는 실제 시스템을 개발할 때 그대로 인용 가능한 소스코드를 포함하고 있지는 않다. 따라서 개발자들은 표준기술규격을 참고하여 프로그램을 개발하지만 프로그램 소스코드는 개발자의 주관과 능력, 경험에 따라 서로 상이하게 작성되고 표현된다. 본 논문은 A측 소스코드 부분 중에서 표준기술규격에 의하여 구현된 부분과 해당 부분의 비율이 어느 정도인지 분석한다. 추가적인 분석으로는 A측 프로그램 소스코드 중에서 선언부를 특정하고 해당 부분이 전체 소스코드에서 차지하는 비율을 분석한다. 분석 결과 선언 부분의 비율은 다음과 같다. A측의 전체 소스코드 선언부 부분은 0.78%, B의 전체 소스코드 선언 부분은 1.72%이다. 표준기술규격서를 의하여 구현된 비율은 0.22%로 분석되었다.

Abstract

The standard technical specification is a reference in the design and development stages and diagrams conceptual diagrams and procedures. Specifications do not include source codes that can be cited as they are when developing actual systems. Therefore, developers develop programs by referring to standard technical specifications, but program source codes are written and expressed differently depending on the developer's subjectivity, ability, and experience. This paper analyzes the ratio of the part implemented by the standard technical specification and the corresponding part among the A-side source code parts. As an additional analysis, the declaration part is specified in the A-side program source code and the proportion of the corresponding part in the entire source code is analyzed. As a result of the analysis, the ratio of the declaration part is as follows. The entire source code declaration part of A side is 0.78%, and the entire source code declaration part of B side is 1.72%. The ratio realized by the standard technical specification was analyzed as 0.22%.

한글키워드 : 표준기술규격, 소스 코드, 소프트웨어, 소프트웨어 감정, 성능

keywords : standard technical specification, source code, software, software appraisal, performance

* 한경국립대학교 컴퓨터응용수학부
† 교신저자: 전병태(email: chunbt@hknu.ac.kr)

접수일자: 2023.03.11. 심사완료: 2023.03.15.
게재확정: 2023.03.20.

1. 서 론

정보통신의 발달로 인하여 소프트웨어의 중요성이 증대되고 있으며, 이에 따른 소프트웨어 저작권 분쟁도 증가하는 추세에 있다. 소프트웨어 분쟁은 저작권법으로 보호가 되고 있다[1-4].

국가기록원의 기록관리 표준은 「기록물관리 표준화 업무 운영규정」에서 규정한 적용범위에 따라 국가표준과 공공표준으로 구분된다[5]. 표준기술규격서는 전자기록물의 온라인 전송을 위한 송ㆍ수신 전송 기술규칙으로, 전자기록물 송ㆍ수신 모듈의 기능 요구사항, 업무시스템과의 연계 인터페이스, 송ㆍ수신 프로토콜 등을 정의하고 있다. 제출된 표준기술규격서는 설계 및 개발 단계의 참고사항이나 개념도 및 절차를 도식화하여 설명하고 있으며, 실제 시스템을 개발할 때 그대로 인용 가능한 소스코드를 포함하고 있지는 않다. 본 논문은 A측 소스코드 부분 중에서 표준기술규격에 의하여 구현된 부분과 해당 부분의 비율을 확인하고, 그 분량이 어느 정도인지 확인하고자한다. 추가적인 분석으로는 A측 프로그램 소스코드 중에서 선언부를 특정하고 해당 부분이 전체 소스코드에서 차지하는 비율을 분석한다. 2장에서는 분석 대상 소프트웨어를 분석하고 3장에서는 소스코드 중 선언부 부분과 그 비율에 대하여 설명한다. 4장은 소스코드 중 표준기술규격에 의하여 구현된 부분과 그 비율에 대하여 분석한다.

2. 분석 대상 소프트웨어 분석

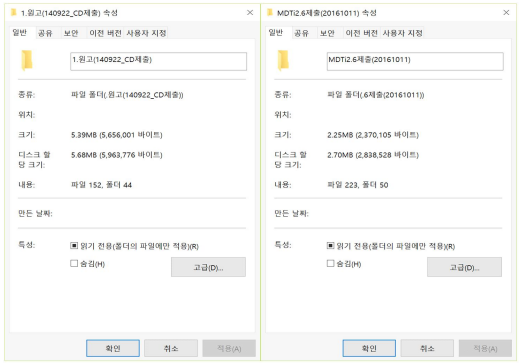
2.1 분석 소프트웨어

분석의 대상은 A 프로그램과 B 프로그램으로, 표1과 같다. A 프로그램은 class 파일로, B의 프로그램은 java 파일로 제출되었다. A 프로그램은 152개 파일(class 파일 148개)이 포함되어 있으며, B 프로그램에는 223개의 java 파일이 포함되어 있다.

표 1. A와 B 측의 프로그램 규모
Table 1. Program size for A and B sides

분석 대상	크기 (Byte)	폴더 수	파일 수	내 용
A	5,656,001	44	152	A 폴더 -class파일 148개
B	2,370,105	50	223	B 폴더 - java 파일 223개

그림1은 프로그램의 상세 정보를 보여주고 있다.



(a) A 프로그램 (b) B 프로그램

그림 1. 프로그램 상세정보
Fig. 1. Program details

2.2 분석 방법

국가기록원의 표준기술규격에 기술된 내용들을 반영해서 구현된 소스코드 부분을 특정하고 해당 부분이 차지하는 비율을 나타내기 위해, B가 제출한 '을7호증 전자기록물 온라인 전송을 위한 기술규격(v1.0)'과 '을15호증 전자기록물 온라인 전송을 위한 기술규격(v1.1)'을 확인한다. 단, 두 프로그램 모두 해당 표준기술규격을 준수하여 프로그램을 구현하여야 함에 따라 본 분석

에서는 개발자가 다르더라도 동일·유사한 소스코드로 작성될 여지가 있는 부분에 대하여만 분석하기로 한다[6-11].

3. A 소스코드 중 선언부 부분과 그 비율

선언부는 개발자의 경험과 주관에 따라 구성, 개발되므로 아래의 목록과 같이 여러 선언부들 중에서 공통적으로 사용되는 일반적인 import문을 대상으로 한정하여 비율을 산정하였다. 그림2는 표준기술규격서의 송·수신 메시지를 보여주고 있다.

import 선언문 제거 목록(java.*, javax.*, org.xml.*, com.jcraft.*)
import java.math.BigInteger; *import java.net.InetSocketAddress;* *import java.net.Socket;* *import java.nio.channels.SocketChannel;* *import java.util.ArrayList;* *import org.w3c.dom.*;* *import org.xml.sax.SAXException;* *import org.xml.sax.SAXParseException;* *import org.xml.sax.helpers.DefaultHandler;* *import java.security.MessageDigest;* *import com.jcraft.jzlib.ZInputStream;* *import com.jcraft.jzlib.ZOutputStream;* 등

그림 2. 송·수신 메시지
Fig. 2. Sent/received message

분석 결과 A의 전체 소스코드 28,206 라인에서 해당되는 import 선언부 부분은 219 라인이었으며, 전체에서 차지하는 해당 선언부 라인 수의 비율은 0.78%로 확인되었다. B의 전체 소스코드 59,235 라인에서 import 선언부 부분은 1,020 라인이었으며, 전체에서 차지하는 해당 선언부 라인 수의 비율은 1.72%로 확인되었다. 표2는 A,B의 분석 결과를 보여주고 있다.

표 2. 선언부 제거 결과(B측)
Table 2. Result of removing the declaration(B)

원본	선언부	선언부 제외 후	비율
59,235	1,020	58,215	1.72%

4. 소스코드 중 표준기술규격에 의하여 구현된 부분과 그 비율

A 소스코드 중에서 국가기록원의 표준기술규격서를 참고하여 구현된 부분을 특정하고 해당 부분이 전체 소스코드에서 차지하는 비율을 산정한다. B 소스코드 분석을 위해 2010년 12월 30일 제정한 NAK/TS 5:2010(v1.0)과 2013년 12월 30일 개정한 NAK/TS 5:2013(v1.1) 두 가지 버전의 기록관리시스템 표준기술규격을 제출하였다.

제출된 표준기술규격서는 설계 및 개발 단계의 참고사항이나 개념도 및 절차를 도식화하여 설명하고 있으며, 실제 시스템을 개발할 때 그대로 인용 가능한 소스코드를 포함하고 있지는 않다. 따라서 개발자들은 표준기술규격을 참고하여 프로그램을 개발하지만 프로그램 소스코드는 개발자의 주관과 능력, 경험에 따라 서로 상이하게 작성되고 표현될 것으로 생각된다.

제출된 B의 소스코드은 표준기술규격을 준수하여 설계하고 동작되도록 개발된 소스코드이다. 표준기술규격을 검토한 결과 '6.3.3 송·수신 메시지 구성, 그림3에서는 송·수신 메시지 종류'와 '6.4.2 오류내역을 위한 로그포맷, 그림4에서는 오류코드'에 기술된 부분은 '메시지 이름'과 '오류코드 설명'을 정의하고 있어 다른 개발자가 작성하여도 동일·유사하게 작성될 수 있는 부분으로 확인하였다.

4.1 송 · 수신 메시지

표준기술규격서 '6.3.3 송·수신 메시지 구성, '송·수신 메시지 종류' 부분에는 그림3과 같이 메시지 이름 '12개'에 대하여 메시지 이름을 정의하고 있다. 해당 부분은 개발자가 다르더라도 소스코드를 작성할 때 내용이 유사·동일할 수 있는 부분이다.

표 · 송 · 수신 메시지 종류

메시지 종류	메시지 이름	설 명
OP0001	접근토큰 요청	로그인을 위한 접근토큰 요청메시지
OP0002	접근토큰 응답	로그인을 위한 접근토큰 응답메시지
OP0003	로그인 요청	로그인 요청을 위한 메시지
OP0004	로그인 응답	로그인 처리결과 응답메시지
OP0005	로그아웃 요청	로그아웃 요청을 위한 메시지
OP0006	로그아웃 응답	로그아웃 처리결과 응답메시지
OP0011	전송통지 요청	파일의 기본정보 상호검사 요청메시지
OP0012	전송통지 응답	파일의 기본정보 상호검사 응답메시지
OP0013	파일전송 요청	전송할 파일정보를 포함한 요청메시지
OP0014	파일전송 응답	수신한 파일정보 처리결과 응답메시지
OP0015	연결해제 요청	연결중인 소켓 연결 해제 요청메시지
OP0016	연결해제 응답	연결중인 소켓 연결 해제 응답메시지

그림 3. 송·수신 메시지
Fig. 3. Sent/received message

4.2 오류 코드

표준기술규격 '6.4.2 오류내역을 위한 로그포맷, 오류코드' 부분에는 오류코드 '61개'에 대하여 설명하고 있다.

오류코드 메시지를 그림 4에서 정의하고 있다. 해당 부분은 개발자가 다르더라도 소스코드를 작성할 때 내용이 유사·동일할 수 있는 부분이다.

표준기술규격을 참고하여 구현하게 되면 그림 5의 A 소스코드와 그림 6의 B 소스코드에서 나타나는 바와 같이 작성된 메시지들의 이름은 개발자가 다르더라도 동일하게 작성되어진다. 다만 A의 소스코드는 디컴파일된 결과로 보여지므로 '한글'이 유니코드로 생성되어 그림7과 같이 코드 형태로 나타나게 된다.

오류코드	오류 설명
E0000	정상 처리되었습니다.
E0101	지정된 경로에 해당파일이 존재하지 않습니다.
E0102	전송목록파일의 XML 구성이 잘못되었습니다.
E0103	지정된 경로 파일에 접근할 수 없습니다. 파일의 권한을 검사하십시오.
E0104	디렉토리 구조가 잘못 구성되었습니다.
E0105	sendlist 경로가 잘못 지정되어 있습니다.
E0106	전자기록물파일의 크기를 설정할 수 없습니다.
E0107	전송목록파일 검증에 실패했습니다.
E1001	로그인 요청메시지의 구문이 잘못 구성되어 있습니다.
E1002	로그인 타입과 로그인 정보가 일치하지 않습니다.
E1003	인증서 또는 개인키 파일이 해당 경로에 존재하지 않습니다.
E1004	로그인 요청메시지의 암호화 시 오류가 발생했습니다.
E1005	개인키 비밀번호가 잘못되었습니다.
E1006	전자서명 값을 생성할 수 없습니다.
E1007	전자서명 값 형식이 잘못 구성되었습니다.
E1008	전자서명 값 검증에 실패했습니다.
E1009	서버인증서 검증에 실패했습니다.

그림 4. 오류 코드
Fig. 4 error code

```
35 ⊖  public static final String sOPCODE_DESC[] = {
36        "\uC0DD\uAE30\uB654", "\uC811\uADFC\uD1A0\uD070 \uC694\uCCAD", "\uC811\uADFC\uD1A0\uD070 \uC751\uB2F5", "\
37        "\uD30C\uC77C\uC804\uC1A1 \uC751\uB2F5", "\uC5F0\uACB0\uB574\uC81C \uC694\uCCAD", "\uC5F0\uACB0\uB574\uC81
38    };
39 }
```

그림 5. 메시지 이름(IARProtocol.java, A)
Fig. 5. message name(IARProtocol.java, A)

```
1024 ⊖  /**
1025     * 송수신 메시지 종류 - 메시지 이름(접근토큰 요청,...)
1026     */
1027 ⊖  public final static String MESSAGE_CODE_VALUE[] = {
1028        // 0~
1029        "초기화", "접근토큰 요청", "접근토큰 응답", "로그인 요청", "로그인 응답", "로그아웃 요청", "로그아웃 응답", "전송통지 요청", "전송통지 응답",
1030        // 10~
1031        "파일전송 응답", "연결해제 요청", "연결해제 응답", "제어메세지 요청", "제어메세지 응답"
1032    };
```

그림 6. 메시지 이름(MtiDef.java, B)
Fig. 6 message name(MtiDef.java, B)

(유니코드) : A 소스코드 생성파일(IARCTRProtocol.java)

```
public static final String sOPCODE_DESC[] = {
        "\uCD08\uAE30\uD654", "\uC811\uADFC\uD1A0\uD070
\uC694\uCCAD", "\uC811\uADFC\uD1A0\uD070 \uC751\uB2F5",
"\uB85C\uADF8\uC778 \uC694\uCCAD", "\uB85C\uADF8\uC778
\uC751\uB2F5", "\uB85C\uADF8\uC544\uC6C3 \uC694\uCCAD",
"\uB85C\uADF8\uC544\uC6C3\uC751\uB2F5",
"\uC804\uC1A1\uD1B5\uC9C0 \uC694\uCCAD",
"\uC804\uC1A1\uD1B5\uC9C0 \uC751\uB2F5",
\uD30C\uC77C\uC804\uC1A1 \uC694\uCCAD",
"\uD30C\uC77C\uC804\uC1A1 \uC751\uB2F5",
"\uC5F0\uACB0\uD574\uC81C \uC694\uCCAD",
"\uC5F0\uACB0\uD574\uC81C \uC751\uB2F5"
 };
(한글코드로 변환)
"초기화", "접근토큰 요청", "접근토큰 응답", "로그인 요청", "로그인 응답", "
로그아웃 요청", "로그아웃응답", "전송통지 요청", "전송통지 응답", 파일전송
요청", "파일전송 응답", "연결해제 요청", "연결해제 응답"
```

그림 7. 한글유니코드 생성
Fig. 7. Generate Korean Unicode

마찬가지로 표준기술규격을 참고하여 구현하면 그림 8의 A 소스코드와 그림 9의 B 소스코드에 나타나는 바와 같이 소스코드에 작성된 오류코드들은 개발자가 다르더라도 동일하게 나타날 수 있다.

그림 8. 오류내역 구현(AR.java, MD v2.1(A))
Fig. 8. Implementation of error history(AR.java, MD v2.1(A))

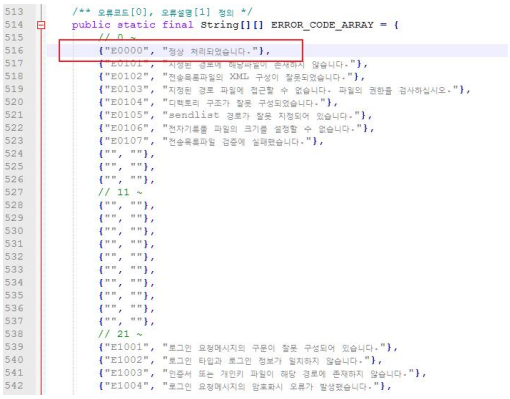


그림 9. 오류내역 구현(MdtiDefine.java, (B))
Fig. 9. Implementation of error history(MdtiDefine.java, (B))

표준기술규격서를 참고하여 개발할 경우, 서로 다른 개발자라도 구현된 소스코드가 동일할 수 있는 부분을 확인한 결과, 표3과 같이 전체에서 차지하는 해당 라인 수의 비율은 0.22%로 확인되었다.

(유니코드) : A 소스코드 생성파일(IARCTREDefineASN1.java)

```
"E0000", "\uC815\uC0C1 \uCC98\uB9AC\uB418\uC5C8\uB2B5\uB2C8\uB2E4."
"E0101", "\uC9C0\uC815\uB41C \uACBD\uB85C\uC5D0 \uD574\uB2F9\uB30C\uC77C\
 uC774 \uC874\uC7AC\uD558\uC9C0 \uC54A\uC2B5\uB2C8\uB2E4."
"E0102", "\uC804\uC1A1\uBAA9\uB85D\uB30C\uC77C\uC758 XML \uAD6C\uC131\uC774
\uC798\uBABB\uB418\uC5C8\uC2B5\uB2C8\uB2E4."
"E0103", "\uC9C0\uC815\uB41C \uACBD\uB85C \uD30C\uC77C\uC5D0
\uC811\uADFC\uD660 \uC218 \uC5C6\uC2B5\uB2C8\uB2E4.
\uD30C\uC77C\uC758 \uAD8C\uD55C\uC744 \uAC80\uC0AC\uD558\uC138\uC694."
"E0104", "\uB514\uB809\uD1A0\uB9AC \uAD6C\uC870\uAC00 \uC798\uBABB
\uAD6C\uC131\uB418\uC5C8\uC2B5\uB2C8\uB2E4."
"E0105", "sendlist \uACBD\uB85C\uAC00 \uC798\uBABB \uC9C0\uC815\uB418\uC5B4
\uC788\uC2B5\uB2C8\uB2E4."
"E0106", "\uC804\uC790\uAE30\uB85D\uB3C \uD30C\uC77C\uC758 \uD06C\uAE30\uB97C
\uC124\uC815\uD660 \uC218 \uC5C6\uC2B5\uB2C8\uB2E4."
"E0107", "\uC804\uC1A1\uBAA9\uB85D\uD30C\uC77C \uAC80\uC99D\uC5D0
\uC2E4\uD328\uD588\uC2B5\uB2C8\uB2E4."

(한글코드로 변환) =>
"E0000", "정상 처리되었습니다."
"E0101", "지정된 경로에 해당파일이 존재하지 않습니다."
"E0102", "전송목록파일의 XML 구성이 잘못되었습니다."
"E0103", "지정된 경로 파일에 접근할 수 없습니다. 파일의 권한을 검사하십시오."
"E0104", "디렉토리 구조가 잘못 구성되었습니다."
"E0105", "sendlist 경로가 잘못 지정되어 있습니다."
"E0106", "전자기록물 파일의 크기를 설정할 수 없습니다."
"E0107", "전송목록파일 검증에 실패했습니다."
```

그림 10. 한글 유니 코드 생성
Fig. 10. Generate Korean Unicode

표 3. 표준규격따른 코드 구현 결과(A측)
Table 3. Result of code implementation according to standard specification (A sides)

순번	파일명	대상 라인 수
1	IARDef.java	61
2	IARProto.java	2
합 계		63 라인 0.22%(63/28,206)

5. 결 론

A의 전체 소스코드 28,206 라인에서 해당되는 import 선언부 부분은 219 라인이었으며, 전체에서 차지하는 해당 선언부 라인 수의 비율은 0.78%로 확인되었다. B의 전체 소스코드 59,235 라인에서 import 선언부 부분은 1,020 라인이었으며, 전체에서 차지하는 해당 선언부 라인 수의 비율은 1.72%로 확인되었다. 표준기술규격서를 참고하여 개발할 경우, 서로 다른 개발자라도 구현된 소스코드가 동일할 수 있는 부분을 확인한 결과, 전체에서 차지하는 해당 라인 수의 비율은 0.22%로 확인되었다.

참 고 문 헌

[1] Baek-Seo Kang, Min-Ji Shin. (2022). A Comparative Study on the Legislation Related to Data Collection and Analysi, Journal of The Korea Software Assessment and Valuation Society. 18(2), 63-75. DOI : http://dx.doi.org/10.29056/jsav.2022.12.07, 2022년

[2] Si-Yeol Kim, Yoon Soo Kang. (2019). Journal of The Korea Software Assessment and Valuation Society. 15(2). DOI://dx.doi.org/10.29056/jsav.2019.12.04.

[3] Si-Yeol Kim. (2018). Computer Program Copyright Similarity Theory, Publisher Sechang. ISBN:9788984117549

[4] Byung Tae Chun, (2019). A Study on Similarity Analysis of SNMP MIB File. Journal of The Korea Software Assessment and Valuation Society. 15(1), 25-34. DOI://dx.doi.org/10.29056/jsav.2019.06.04.

[5] National Archives of Korea. https://www.archives.go.kr/next/newdata/standardCondition.do

[6] Si-Yeol Kim. (2022). A Review of the Improvement of the Expert Evidence System under the Criminal Procedure Act, Journal of The Korea Software Assessment and Valuation Society. 18(2). 87-101.DOI:http://dx.doi.org/10.29056/jsav.2022.12.09

[7] N. Cvejic and T. Seppanen, "Digital AudiWatermar king Techniques and Technologies: Applications and Benchmarks," IGI Global Snippet, 2008.

[8] H. J. Kim, et al. "Audio watermarking techniques," Intelligent Watermarking Techniques 7, pp.185-208, 2004.

[9] Myung-Mook Han*・Yong Hak Ahn,"A DRM System for Streaming Services in Mobile Environment", Journal of information and security v.7 no.2 , 2007, pp.135 - 143

[10] Bogdan C. Popescu, Bruno Crispo, Frank L. A. J. Kamperman, “A DRM Security Architecture for Home Networks”, DRM ’04, pp. 1-10, October, 2004

[11] Kyu-Tae Lee. (2016). A detail item guideline for IT device evaluation, ournal of The Korea Software Assessment and Valuation Society, 12(1). 21-26, http://www.i3.or.kr/html/paper/2018-1/

저 자 소 개

전병태(Byung-Tae Chun)

2001. 2. 20 고려대학교 박사
1989~1996년 한국과학기술연구원(KIST)
선임연구원
1996~2004년 한국전자통신연구원(ETRI)
책임연구원
2004.2.~현재 국립 한경대학교
컴퓨터응용수학부 교수
2013.8.~현재 한국저작권위원(KCC)
SW감정전문위원
2019.1~2021.12 한국지식정보학회(KKITS)
논문지편집위원장
2014..1~현재 TTA/TC4/JTC1 로봇분과
부위원장
2011.2~현재 한국SW감정평가학회
회장 역임 및 이사
1992. 5.20. IR52 장영실 상 수상
(과학기술부 장관상)
2015.12.10. 제10회 대한민국 로봇 대상
(산업통상자원부 장관상)
<주관심분야> 영상처리, 로봇 HRI,
인공지능, SW감정 분야