소프트웨어 소스 코드의 저작권 관리를 위한 디지털 라이센스의 검색

Digital License Searching for Copyright Management of Software Source Code

차병래
호남대학교 컴퓨터공학과

ByungRae Cha(chabr@honam.ac.kr)

요약

지적재산권 제도는 21세기 정보화 사회의 발전에 있어서도 중요한 역할을 하고 있다. 국가 경쟁력 제고를 위해서도 디지털콘텐츠에서 확대하여 소프트웨어 소스 코드에 대한 지적재산권 제도와 기술의 정비는 매우 중요한 의미를 지닌다. 소프트웨어 소스코드의 소유권 분쟁이 발생 시 소유권을 증명하기 위해서는 원본의 소프트웨어 소스코드를 판별해야만 하는 문제점을 갖고 있다. 본 연구에서는 소프트웨어 소스코드의 원본 판별을 지원하기 위한 소프트웨어 소스코드의 디지털 라이센스는 소스코드의 예약어를 파싱하여 계층구조를 갖는 XML 파일로 표현하며, 복잡한 소스코드 대신에 소프트웨어 소스코드의 아키텍처를 트리 구조 형태로 표현할 수 있다. 그리고 디지털 라이센스를 검색하기 위한 색인 및 검색에 대한 연구를 수행한다.

■ 중심어 : | 디지털 라이센스 검색 | 소프트웨어 소스 코드 | 저작권 관리 |

Abstract

The intellectual property system was very important to the past industrial society. It is so important to the 21C information age. It is a leading role to developing these information society. Not only the digital content control but the technology of software source code for the intellectual property is so much mean to international competition. On occurring disputation property, we have to prove the fact, there is a problem to discriminate the original source code.

In this paper, we make a study of the digital licence prototype for discriminate the original source code. Reserved words of software source code by parsing express to XML file that have hierarchical structure. Then, we can express architecture of software source code by tree structure form instead of complex source code. And we make a study of the indexing and searching to search digital license.

■ keyword : | Digital License Searching | Software Source Ccode | Copyright Managements |

I. 서 론

최근 들어서 저작권의 경제적 영향에 대한 학계의 관심과 저작권의 국민경제적 역할의 계량적 측정을 통한 정책담당자들의 관심이 커지고 있다. 전 세계적으로 일상적인 경제생활의 생산, 유통, 소비의 전반에 걸쳐 지속적으로 확대되어 온 저작권의 영향력을 간과할 수준을 넘어섰음을 의미한다. 이러한 저작권의 경제적 가치의

* 본 연구는 호남대학교의 2006학년도 교내연구비를 지원받아 수행되었습니다.
접수번호 : #060914-001
접수일자 : 2006년 09월 14일
심사완료일 : 2006년 12월 22일
교신저자 : 차병래, e-mail : chabr@honam.ac.kr

증대에는 여러 가지 이유가 있다. 첫째, 저작권에 대한 관심은 부분적으로는 비물질적인 생산요소가 보다 주목을 받는 지식기반사회에서 증가하고 있는 지적재산(Intellectual Property : IP)의 역할에 대한 인식이 증대된 결과이다. 저작권은 창의성과 정보에 기반을 둔 산업의 성장과 산업의 생산성, 고용 및 투자를 위하여 중요하다. 둘째는, 디지털기술의 발달로 인하여 저작권보호의 대상물의 범위가 크게 증대되었다는 것이다. 소프트웨어, 멀티미디어 및 기타 기술기반 산물로부터 경제적 이득은 막대해지고 있다. 셋째는 디지털 혁명의 결과, 저작권 보호 대상들이 전자상거래의 주요한 요소가 되었다는 점이다. 과거 산업사회의 태동과 발전과정에서 뿐만 아니라, 21세기 정보화 사회의 발전에 있어서도 이러한 지적재산권 제도는 중요한 역할을 하고 있다. 우리나라의 국가 경쟁력 제고를 위해서도 디지털콘텐츠에서 확대된 소프트웨어의 소스 코드에 대한 지적재산권 제도와 기술의 정비는 매우 중요한 의미를 지닌다.

디지털 콘텐츠의 저작권 보호를 위한 연구에 비해서 소프트웨어의 소스코드에 대한 저작권을 관리하기 위한 기술은 아직도 초기 연구단계에 있다. 소프트웨어 소스코드의 소유권 분쟁이 발생시 소유권을 증명하기 위해서는 원본의 소프트웨어 소스코드를 공개해야만 하는 문제점을 갖고 있다. 본 연구에서는 소프트웨어의 소유권 주장에 하드카피이외에 더 많은 정보를 부여하며, 컴퓨팅 환경에서 자동으로 처리를 위한 프로토타입을 연구하였다. 연구 내용으로는 소프트웨어 소스코드의 원본 판별을 지원하기 위한 소프트웨어 소스코드의 디지털 라이센스는 소스코드의 예약어를 파싱하여 계층구조를 갖는 XML 파일로 표현하며, 복잡한 소스코드 대신에 소프트웨어 소스코드의 아키텍처를 트리 구조 형태로 표현할 수 있다. 그리고 디지털 라이센스를 파일시스템에서 검색하기 위한 색인 및 검색에 대한 연구를 수행한다.

본 논문에서 2장은 소프트웨어 소스코드의 DRM 기술과 소스코드의 디지털라이센스에 대한 관련 연구를, 3장에서는 문제 해결을 위한 비즈니스 모델을 제안한다. 4장에서는 디지털라이센스 발행을 위한 패턴 생성 과정을, 5장에서는 디지털라이센스의 색인/검색 설계 및 파일 크기를 비교하였으며, 마지막으로 결론을 기술한다.

II. 관련 연구

1. 소프트웨어의 소스코드를 위한 DRM

디지털 컨텐츠란 디지털로 되어 있는 문자, 소리, 화상, 영상 등의 형태로 이루어진 정보 내용물이라고 할 수 있다. 본 연구에서는 디지털 컨텐츠의 영역을 확장하여 소프트웨어의 소스코드를 포함하고자 한다. 소프트웨어의 소스코드는 논리적 사고과정을 컴퓨터가 처리할 수 있는 고급 언어 또는 그에 준하는 프로그래밍 언어를 통해 기술해 놓은 일종의 디지털 컨텐츠이다.

DRM(Digital Rights Management)은 디지털 콘텐츠의 보호를 위한 암호화 및 사용자 인증키 관리, 디지털 콘텐츠 유통 환경을 구성하는 주체들 간의 지적재산권 및 거래 규칙, 과금·이용규칙에 대한 표현, 디지털 콘텐츠의 이용 및 분배, 사용 및 접근제어, 권한제어, 워터마킹, 불법복제 축적기술, 투명한 전자상거래를 위한 거래내역의 관리 및 보고, 과금 처리 등을 가능하도록 하는 디지털 저작권 관리에 대한 전반적인 기술이다. 즉, 콘텐츠의 암호화와 권한제어를 통해 콘텐츠에 권한을 가지고 있는 사용자들만이 그 권한에 따라 콘텐츠를 열어 볼 수 있도록 함으로서 콘텐츠의 불법사용 방지, 자동과금, 결제대행, 부가서비스 등 디지털 콘텐츠의 생성, 유통, 사용에 관련된 전 분야의 서비스를 제공하는 것이다[1-3].

대부분의 DRM 기술은 바이너리 파일에 대한 DRM 기술이다. 원래의 디지털 저작물인 바이너리 코드를 패키징을 수행하여 디지털 저작권 관리가 이루어진다. 그러나 디지털 저작물의 원천 요소인 소프트웨어 소스 코드에 대해서는 암호화이외에는 별다른 방법이 존재하지 않으며, 아직은 연구 초기 단계이다. 암호화 방법 역시 소프트웨어 소스 코드 개발자와 소프트웨어 소스 코드 구입자간의 1차적인 거래에만 저작권 관리를 지원할 뿐 그 이후에 발생하는 거래에 대한 저작권을 보호해 주지 못하고 있다.

소프트웨어 소스 코드에 대한 DRM 기술이 연구 초기 단계인 이유는 소스 코드 자체가 아스키 코드나 유니코드로 구성되어 있기 때문이다. 이런 이유로 인하여, 암호화를 통한 1차적 거래에만이 1:1 관계의 디지털 저작권 관리가 이루어지고 있다.

소프트웨어 소스코드의 패턴 매칭에 대한 연구는 Rieger[4]과 Yang[5]에 의해서 연구되어졌다. Rieger는 소프트웨어의 소스코드가 중복되거나 복제된 부분을 시각화하여 탐지하는 연구를 수행하였고, Yang은 두 개의 다른 프로그램을 구문론적으로 동일한 코드를 찾는 연구를 수행하였다. 프로그램 표절 검출 S/W로는 국외는 SIM[6], Dup[7], Plague, YAP[8], YAP3, MOSS[9] 등이 있다. 국내는 KAIST의 clonechecker와 부산대의 LOFC가 있으며[10], 프로그램 심의위원회의 exEyesLight[11]가 있다.

본 연구는 소프트웨어의 소스코드에 대해서 DRM을 지원하기 위한 기초 연구이다. DRM의 저작권 보호 기술보다는 저작권 관리 기술을 제안하며, DRM의 영역을 소프트웨어의 소스코드까지 확장한다. 소프트웨어 소스코드에 대한 DRM 기술은 부재하며, 저작권 보호기술로는 1:1 관계의 거래에서만 임의의 효력을 갖은 암호화 기술뿐이며, 초기연구단계에 머물러 있다. 소프트웨어 소스코드는 제품을 생산하는 공장의 설계도면을 뛰어넘어서 제품생산라인과 같은 아주 중요한 디지털 콘텐츠 자원이다. 그러나 정작 이 자원들을 지원하는 DRM 기술은 너무나도 부족한 상태이다.

2. 소프트웨어의 디지털 라이센스

대부분의 DRM 기술은 바이너리 파일에 대한 DRM 기술이다. 원래의 디지털 저작물인 바이너리 코드를 패키징을 수행하여 디지털 저작권 관리가 이루어진다. 그러나 디지털 저작물의 원천 요소인 소프트웨어 소스 코드에 대해서는 암호화이외에는 별다른 방법이 존재하지 않으며, 아직은 연구 초기 단계이다. 소프트웨어 소스 코드에 대한 DRM 기술이 연구 초기 단계인 이유는 소스코드 자체가 아스키코드나 유니코드로 구성되어 있기 때문이다. 이런 이유로 인하여, 암호화를 통한 1차적 거래에만이 1:1 관계의 디지털 저작권 관리가 이루어지고 있다. 암호화 방법 역시 프로그램 소스 코드 개발자와 프로그램 소스 코드 구입자간의 1차적인 거래에만 저작권 관리를 지원할 뿐 그 이후에 발생하는 거래에 대한 저작권을 보호해 주지 못하고 있다.

일반적으로 소프트웨어 등록 협회에 소프트웨어를 등록하면 하드 카피인 등록증 하나가 배달된다. 최근에는 좀 더 개선되어 웹을 통해서 소프트웨어 등록증을 출력할 수 있다. 그렇지만, 정작 분쟁이 생기면 이를 해결하기 위한 절차가 복잡해진다. 단지, 소프트웨어를 보관하고 있다가 분쟁이 발생하면 복사본을 제공받으며, 먼저 등록한 사람에게 법정에서 우선권을 주장할 수 있는 근거를 제공할 뿐 등록한 소프트웨어에 대한 좀 더 많은 정보를 제공하지는 못하고 있다. 하드 카피의 단순한 등록증에서 탈피하여 하드 카피의 내용에 부가적으로 소프트웨어의 소스코드에 대한 많은 정보를 제공할 수 있는 XML 형태의 디지털 라이센스를 설계 및 구축하였다.

소프트웨어를 이해하기 위해서는 코드와 아키텍처를 보다 면밀하게 구분할 필요가 있다. 코드 또는 소프트웨어는 컴퓨터에 의해서 수행되어질 프로그램을 구성하는 벽돌과 시멘트에 해당한다. 한편 아키텍처는 코드를 벽돌 삼아 쌓아올린 구조물을 의미하는 것이다. 코드는 행위에 제약을 가할 수도 있고 구조에 영향을 미치기도 한다. 그렇지만 모든 것이 오로지 코드에 의해서만 결정되는 것은 아니다. 본 논문에서는 소프트웨어 소스코드의 디지털 라이센스를 생성하는 과정을 그림 1에 나타내었다[12]. 생성된 디지털 라이센스를 이용하여 색인생성과 검색 연구에 이용하였다.

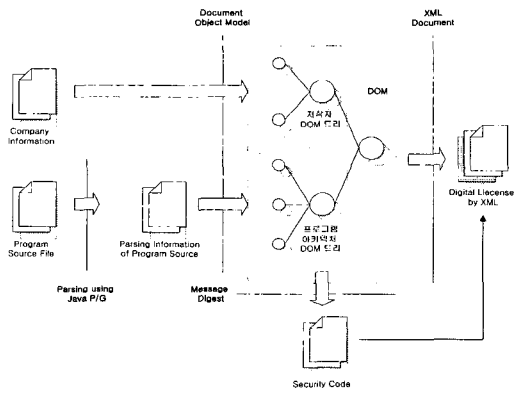

그림 1. 소프트웨어 소스코드의 디지털 라이센스 생성 과정

2.1 디지털 라이센스를 위한 DOM 트리의 구성

DOM 모델은 문서 객체들의 표준 집합을 통하여 응용 프로그램들에 대한 플랫폼 중립적이며 언어 중립적인 인

터페이스를 정의한다. 본 연구에서는 프로그램 소스 코드의 디지털 라이센스 생성에 DOM 모델을 적용한다. 소프트웨어 소스코드의 디지털 라이센스는 두 개의 자노드를 갖는데, 소프트웨어 소스코드의 저작자에 대한 정보를 갖는 저작자 DOM 트리와 소프트웨어 소스코드의 아키텍처를 나타내는 소프트웨어 아키텍처 DOM 트리로 [그림 1]과 같이 이루어진다.

소프트웨어 소스 코드에 대한 암호화 기법은 단지 패키징 기법 중의 하나일 뿐이다. 본 연구에서는 디지털라이센스를 검색 및 색인을 생성하기 위해서 소프트웨어 소스 코드를 파싱하여 32개의 예약어, 라이브러리 그리고 함수 등을 DOM 트리로 [그림 2]와 같이 디지털 라이센스를 생성하였다. C언어 프로그램 소스 코드의 예를 들어, 모든 C 언어의 프로그램은 main() 함수와 서브 함수로 구성된다. 그러므로, main() 함수가 디지털 라이센스의 프로그램 아키텍처 부분의 DOM 트리의 루트 노드가 되며, 서브 함수는 main() 함수의 자노드로 구성된다. 각 노드에는 사용된 라이브러리와 변수들 자료형, 연산자들의 패턴 정보들을 갖는다.

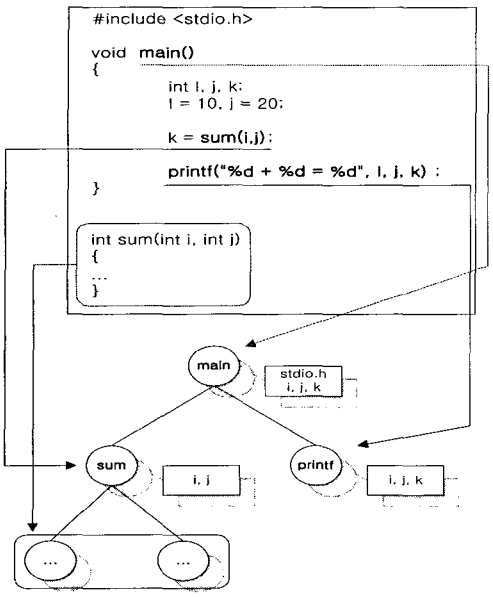

그림 2. 소프트웨어 소스 코드의 소프트웨어 아키텍처 DOM 트리의 변환 과정

2.2 디지털 라이센스를 위한 자노드 패턴, 연산자 그리고 입출력 자료형 패턴생성

소프트웨어 소스코드의 디지털 라이센스를 생성 시에는 최상의 루트 노드는 디지털 라이센스이다. 루트 노드의 밑에는 프로그램 저작자 DOM 트리 노드와 프로그램 아키텍처 DOM 트리 노드를 갖는데, 왼쪽 자노드는 저작자에 대한 정보를 표현하는 DOM 트리로 구성된다. 일반적인 소프트웨어 등록증의 내용과 정보의 무결성을 보장하기 위한 메시지 축약(MD : Message Digest)[13] 코드가 추가된다. 소프트웨어 소스코드의 디지털 라이센스의 핵심 부분은 오른쪽 자노드인 소프트웨어 아키텍처 DOM 트리이다.

이 자노드는 소스코드를 파싱하여 생성된 예약어 토큰과 소프트웨어 구조에 의해서 계층 구조의 아키텍처로 [그림 2]와 같은 형태로 생성된다. 그 외에도 중괄호("{", "}")로 나타내어지는 블록을 이용한 자노드 생성, 제어문인 조건문, 반복문 그리고 분기문에 의한 자노드 생성 그리고 함수의 호출에 의한 자노드를 생성한다.

함수의 코드 중에 연산자가 존재하지 않는 경우에는 자노드를 생성하지 않는다. 연산자를 포함하는 함수만이 입력에 대해 새로운 출력을 생성할 수 있다. 즉, 연산자를 포함하는 함수이어야만 임의의 프로세싱으로 간주하고, 연산자를 포함하지 않는 코드는 단지 화면의 출력이거나, 악의적인 경우의 원본 코드를 속이기 위한 빈 코드의 삽입으로 간주할 수 있다.

그러므로, 이런 코드에 대해 대처할 수 있는 트리의 축약가정이 필요하다. 모든 노드에는 노드만이 갖는 패턴을 생성하게 된다. 노드에 나타내는 패턴은 입력된 자료형과 수, 연산자 그리고 출력된 자료형과 개수로 노드의 패턴을 내부 형식으로 생성한다. 노드의 패턴 데이터를 이용하게 되면, 프로그램의 아키텍처가 동일하더라도 노드의 패턴 데이터로 프로그램의 기능이 다름을 구분할 수 있게 된다.

즉, 노드의 입력 패턴 수와 데이터 형태, 연산자 패턴, 출력 패턴 수와 데이터 형태로 표현되는 노드를 표현함으로 프로그램의 아키텍처만을 비교하는 단점을 극복할 수 있게 되며, 프로그램의 구별 능력을 증대시키게 된다.

2.3 XML 기반의 디지털 라이센스 설계

과거에는 모든 정보를 바이너리 형태로 생성하였다. 그러나 최근에는 암호화 기술 및 통신의 발달은 바이너리 형식보다는 가독성이 높고 메타 데이터를 기술할 수 있는 XML 형식으로 대부분 기술한다. 향후의 개선된 스키마로 기술되더라도 쉽게 변환이 가능하다는 장점을 갖는다. 소프트웨어 소스 코드의 디지털 라이센스는 XML 기반의 소프트웨어 소스 코드에 대한 소프트웨어 작성자, 프로그래밍 언어, 등록 날짜, 등록번호, 버전 번호, 소프트웨어 소스 코드의 DOM 정보 등이 기술되어 있다. 소프트웨어 소스 코드의 디지털 라이센스의 무결성과 비밀성을 보장하기 위해서 XML로 만들어진 디지털 라이센스 코드의 일부분을 암호화를 수행한다. 본 연구에서는 JCE[13]의 패키지를 이용하여 메시지 축약을 수행하며 [그림 3]과 같이 나타난다.

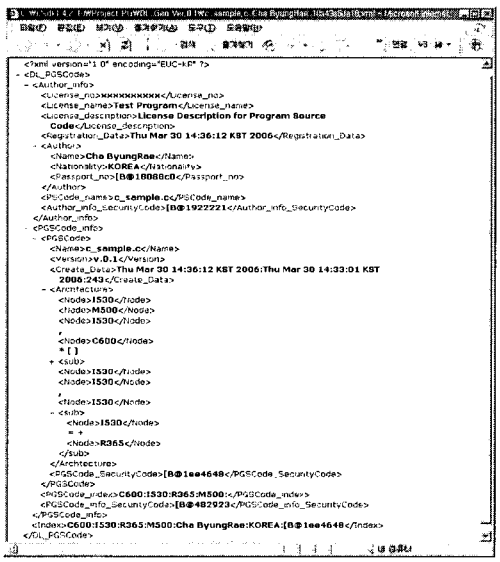

그림 3. 소프트웨어 소스 코드의 디지털 라이센스 프로토타이핑

III. 디지털 라이센스의 비즈니스 모델 제안

1. 소프트웨어 불법 유출에 의한 분쟁

소프트웨어 개발자 A, B, C와 법원 D의 소프트웨어 불법 유출에 의한 문제 사항을 [그림 4]와 같이 제기한다. 소스코드의 원본의 소유자 A와 불법 유출자 B에 의해서 분쟁은 발생한다[그림 4①]. 법원은 이 문제를 해결하기 위해서 개발자 A와 불법 유출자 B의 소스 코드 를 제출할 것을 요구한다[그림 4②]. 그러나 소프트웨어의 소스코드는 전문가가 감별해야 하기 때문에 법원 D는 소프트웨어 전문 개발자 C에게 소프트웨어 소스코드의 원본과 불법 유출본과의 차이점을 의뢰하게 된다[그림 4③]. 소프트웨어 전문 개발자 C는 원본과 불법 유출본을 비교하기 위해서 소스코드를 이해해야 한다. C에 의해서 원본과 불법 유출본을 감별하는 것은 전적으로 C의 주관성 및 소스코드의 이해 능력에 전적으로 의지하게 되어 객관성이 부족하게 된다[그림 4④]. 여기서 제 2의 소스코드 불법유출 문제가 발생할 수도 있게 된다[그림 4⑤]. 의도하지 않은 소스 코드의 지적재산권이 침해되는 문제가 발생하게 된다. 즉 소소코드의 원본과 불법 유출본을 감별하기 위해서 C가 소스코드를 이해하다보니 자동적으로 C에게 제 2의 소스코드가 불법 유출되는 문제가 발생하며, 제 2의 분쟁 소지를 갖게 된다.

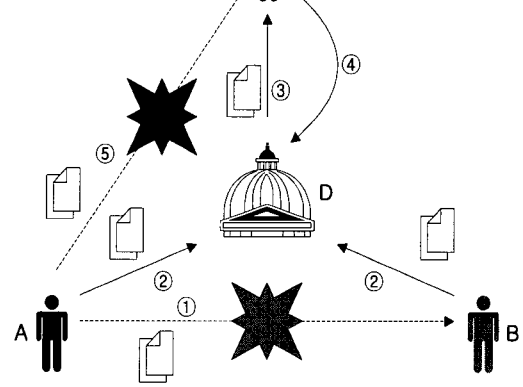

그림 4. 소프트웨어 분쟁에 의한 지적재산권 침해 문제 발생의 다이어그램

[그림 4]의 저작권 침해 문제의 발생을 소프트웨어 소스코드의 디지털 라이센스에 의해서 제 2의 분쟁을 사전에 제거할 수 있다. 소프트웨어 개발자 A, B, C와 법원 D의 소프트웨어 불법 유출에 의한 문제 사항을 소프트

웨어의 소스코드 대신에 디지털 라이센스로 대처하므로 제 2의 분쟁을 사전에 예방할 수 있다. 또한 C에 의한 주관적 감별을 디지털 라이센스의 패턴 분류 프로그램으로 대처함으로서 어느 정도 객관성을 갖출 수 있다.

DRM 기술은 여러 목적으로 제작된 디지털 콘텐츠가 제작·유통·사용되는 과정에서 해당 컨텐츠 저작권이 보호될 수 있도록 관리해 주는 시스템이다. DRM을 이루는 기술은 크게 두 가지로, 저작권 보호 기술과 저작권 관리 기술로 구분지울 수 있다.

대부분의 상업화된 DRM 기술은 저작권 보호기술이 주를 이루고 있다. 프로그램 소스코드에 대한 DRM 기술은 부재하며, 저작권 보호기술로는 1:1 관계의 거래에서만 임의의 효력을 갖은 암호화 기술뿐이며, 초기연구단계에 머물러 있다. 프로그램 소스코드는 제품을 생산하는 공장의 설계도면을 뛰어넘어서 제품생산라인과 같은 아주 중요한 디지털 콘텐츠 자원이다. 그러나 정작 이 자원들을 지원하는 DRM 기술은 너무나도 부족한 상태이다.

2. 디지털라이센스의 비즈니스 모델 제안

[그림 4]의 저작권 침해 발생 문제는 [그림 5]와 같이 디지털 라이센스를 적용함으로써 해결할 수 있다.

A와 B의 분쟁 발생시 소스코드를 법원에 제출하지 않고 A와 B가 갖고 있는 소스코드의 디지털 라이센스를 생성하여 법원에 제출하면 된다[그림 5①]. 법원은 패턴 매칭 프로그램에 의해서 어느 정도 일치하는지를 객관적으로 보고서를 얻을 수 있으며, 추가적으로 C에게 자문을 구함으로써 분쟁 해결에 많은 도움이 될 것이다[그림 5②]. 디지털 라이센스를 이용하면 또한 제 2 분쟁의 불씨를 제거하게 된다.

C는 단지 소스코드가 아닌 디지털 라이센스를 검증하여 유출된 소프트웨어 소스코드가 원본에 어느 정도 일치하는 지에 대해 자문하면 된다[그림 5③].

디지털 라이센스에 의한 비즈니스 모델은 개발자 A의 지적재산권을 보호하고, 소스코드를 유출한 B를 좀 더 쉽게 판별하게 해주며, 또한 소스코드의 판별에 의한 제 2의 소스코드 유출을 사전에 막아서 제 2의 분쟁을 발생시키지 않는다. 더불어 C의 주관성에 의지하지 않고 객관적인 패턴 매칭 프로그램으로 확연한 보고서를 받을 수 있으며, C에 의해서 오랜 분석에 의한 자문에 크게 영향을 받지 않고, 짧은 시간의 패턴 매칭 프로그램 분석으로 분쟁 해결의 시간도 지체하지 않을 것이다.

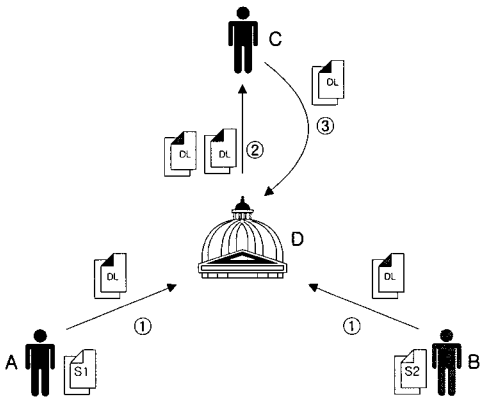

그림 5. 디지털 라이센스에 의한 지적 재산권 침해 방지

Ⅳ. 디지털 라이센스의 색인 및 검색 기법 설계

1. 디지털 라이센스를 위한 DOM 구성 및 파싱

소프트웨어 소스 코드에 대한 암호화 기법은 단지 패키징 기법 중의 하나일 뿐이다. 복호화된 소스코드에 대해서는 어떠한 해결책도 제공하지는 못한다. 소프트웨어 소스코드의 저작권을 표현하는 디지털 라이센스의 프로토타입을 생성하기 위해 필요한 제반기술인 DOM은 상호간에 동작할 수 있는 공통적인 문서 구조를 정의하는 모델로 W3C[14]에서 만들었다. DOM의 처리과정은 XML 문서를 파싱하는 단계와 DOM 트리를 접근하는 단계로 구성된다. DOM에서는 XML 문서가 하나의 트리로 표현되며, 전체 문서는 일종의 중첩된 트리로서 표현된다. 본 연구에서는 소프트웨어 소스 코드의 디지털 라이센스 생성에 DOM 모델을 적용하며, 디지털 라이센스를 바이너리 파일이 아닌 XML 파일로 기술 및 생성하며, XML 문서를 DOM 기술로 처리할 수 있다는 장점을 갖는다[15].

소프트웨어의 소스코드를 분석하기 위해서는 패턴 매칭과 파싱 기술이 필요하다. 소프트웨어 소스코드를 파싱하여 의미를 부여할 수 있는 구분된 컴포넌트를 토큰

이라 한다. 그리고 토큰은 각각 독립적인 유닛을 나타낸다. 소프트웨어 소스코드가 갖는 추상적인 표현을 토큰화를 거쳐야 하며, 구성된 토큰을 이용하여 디지털 라이센스의 정보 및 아키텍처를 구성하게 된다. 토큰은 두 가지 형식을 갖게 되는데, 외부 형식과 내부 형식이다. 외부 형식은 소프트웨어 소스코드를 작성할 때 사용되는 텍스트 형식이다. 외부 형식을 효율적인 처리를 위해서는 내부 형식으로 전환한다. 소프트웨어의 디지털 라이센스의 정보 표현을 생성할 때 내부 형식으로 표현하면 문자열보다는 정수를 사용하는 것이 훨씬 빠르게 수행되기 때문에 내부 형식으로 표현하면 소프트웨어의 디지털 라이센스 생성처리를 위한 성능상의 이점이 있다. 향후 색인 생성기와 검색기에 사용되는 텀(term)은 토큰을 이용한다.

2. 소프트웨어 아키텍처와 노드의 패턴정보

인덱스 패턴이 동일한 경우에는 어느 정도 유사한 프로그램으로 간주할 수 있다. 그래서, 노드로 구성된 프로그램 아키텍처에 의해서 프로그램을 분류할 수 있으며, [그림 2]와 같이 DOM을 이용한 추상 코드 아키텍처(ACA : Abstract Code Architecture)를 생성한다. 코드는 컴퓨터에 의해서 수행되어질 소프트웨어를 구성하는 기본적인 구성 요소에 해당한다. 한편 아키텍처는 코드라는 기본 구성 요소로 이루어진 구조물을 의미한다. 코드는 행위에 제약을 가할 수도 있고 구조에 영향을 미치기도 한다. 그렇지만 모든 것이 오로지 코드에 의해서만 결정되지 않으며, 아키텍처에 의해서도 분류정보를 제공할 수 있다. 인덱스 패턴과 프로그램 아키텍처가 동일한 경우에는 거의 동일한 프로그램이라고 할 수 있다. 그러나 프로그램도 생명주기를 갖기 때문에 과거의 프로그램의 패턴을 상속될 수 있다. 이런 경우를 버전업되었다고 하는데, 노드의 패턴 정보에 의해서 이러한 정보를 제공할 수 있다.

3. 중복 노드 제거와 트리 순회를 이용한 인덱스 패턴 생성

연산자를 포함하지 않은 함수 노드는 DOM 트리의 축소와 실제 소스코드의 분석과정에서 차별화를 위한 어떠한 의미를 제공하지 않는다. 그러므로 연산자 패턴을 포함하지 않은 함수 노드를 제거해도 소스코드를 같은 의미로 볼 수 있다. 그러므로 연산자 패턴을 갖지 않은 함수 노드를 제거할 수 있으며, 더불어 DOM 트리의 크기를 축소한다. 또한 프로그래밍 언어의 예약어에 의한 예약어의 발생 순차 패턴과 순차 패턴의 중복 제거에 의한 고유한 인덱스 패턴을 생성한다. 기본적으로 이 인덱스 패턴에 의해서 소프트웨어를 구분 및 분류 그리고 비교를 수행한다. C 언어의 예약어는 32개이므로 main() 함수를 루트 노드로 설정한 중복 제거한 32!의 가능한 인덱스 패턴이 존재할 수 있다. 프로그램 소스코드의 디지털 라이센스에는 인덱스 패턴을 갖고 있다. 인덱스 패턴에 의해서 디지털 라이센스에 기록된 프로그램에 대한 순차 패턴 정보와 이 정보를 이용한 클러스터링 기능을 제공한다. 만약, 임의의 두 프로그램의 인덱스 패턴을 비교해서 동일한 인덱스 패턴으로 매칭되면, 프로그램의 아키텍처와 노드의 패턴에 의해서 두 프로그램은 정밀한 분석이 이루어진다. 인덱스 패턴의 생성 과정은 DOM 트리의 순회 방법인 중순위(In-order) 방법에 의해서 패턴을 나열하게 되며, 나열된 순차 패턴에서 처음 나타난 다음의 중복된 패턴을 제거하여 인덱스 패턴을 생성하게 된다. 소프트웨어 소스코드의 디지털 라이센스의 정보를 구축하기 위해선 단 하나의 정보를 생성하기 보다는 많은 정보를 제공하기 위한 다양한 정보를 생성하여야 한다. 프로그램의 소스코드를 제공하지 않는 대신에 그에 해당하는 충분한 정보를 제공하여야만 디지털 라이센스의 기능을 수행할 수 있기 때문이다. 본 논문에서는 소프트웨어 소스코드를 이용해서 디지털 라이센스를 구성하는 인덱스 패턴, 소프트웨어 아키텍처, 노드의 패턴 정보 등의 다양한 패턴 정보를 생성하여, 이러한 정보를 이용한 색인생성과 검색을 설계한다.

V. 디지털 라이센스의 색인/ 검색의 프로토타입

1. 디지털 라이센스의 검색 구조 설계

디지털 라이센스를 구성하는 인덱스 패턴, 소프트웨어 아키텍처, 노드의 패턴 정보 그리고 메시지 축약과 암호

화 등의 다양한 이러한 패턴 정보는 소프트웨어의 소스 코드를 분류하거나, 클러스터링 그리고 검색할 수 있는 정보를 제공한다. 더 나아가서는 소프트웨어 버전 관리를 위한 정보도 제공할 수 있다. 소프트웨어 소스코드의 분류 및 검색 단계는 [그림 6]과 같이 3단계로 구성된다. 검색 및 분류를 위한 단계를 진행 할수록 좀 더 세밀한 분류가 이루어진다.

[단계 1] 인덱스 패턴 정보에 의한 분류
[단계 2] 소프트웨어 아키텍처 패턴에 의한 분류
[단계 3] 노드의 패턴 정보에 의한 분류

단계 1의 수준은 인덱스 패턴에 의한 클러스터링 분류 정보를 제공한다. 이를 위해 인덱스 패턴은 소프트웨어 소스코드에 사용된 예약어의 종류와 순차 정보로 구성된다. 인덱스 패턴 정보에 의해서 동일한 예약어를 갖는 소프트웨어 소스코드를 분류가능하다. 그러나 인덱스 패턴 정보도 임의의 정도에서는 한계를 드러낼 것이다. 이에 부가적으로 단계 2의 수준은 거시적인 수준의 계층구조의 아키텍처를 검색한다. DOM을 이용한 추상 코드 아키텍처(ACA) 패턴 정보를 이용하면 같은 예약어 그룹으로 구성된 소프트웨어 소스코드도 계층구조의 소프트웨어 아키텍처에 의해서 그래픽 형태의 거시적인 분류 정보를 제공한다.

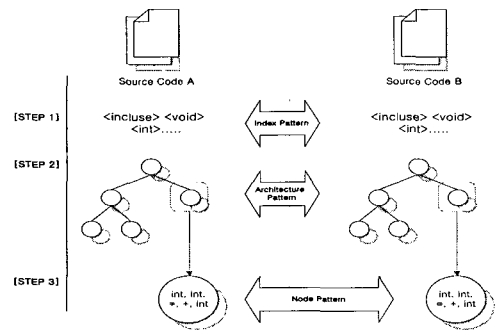

그림 6. 디지털 라이센스에 의한 분류 3단계 과정

한 단계 더 나아가서, 단계 3의 수준은 미시적인 수준의 단말 노드의 패턴 매핑 및 검색한다. 같은 계층 구조의 소프트웨어의 아키텍처를 갖더라도 아키텍처를 이루는 노드의 패턴 정보에 의해서 입력 데이터의 수와 타입, 연산자의 순차나열 그리고 출력 데이터의 수와 타입에 의해서 더 세밀하게 분류 패턴 정보를 제공할 수 있다.

2. 디지털 라이센스 색인 및 검색의 프로토타입

소프트웨어의 소스코드는 특별히 데이터베이스에 관리하지는 않으며, IDE 환경의 프로젝트 디렉터리에 보관하는 게 대부분이다. 유능한 개발자들은 CVS[16] 등을 이용하여 소스 코드를 관리하지만 이것 또한 디렉터리에 관리하게 된다. 그래서 향후 소스 코드와 디지털 라이센스는 같이 디렉터리에 존재하는 경우가 많이 발생할 것이다.

디지털 라이센스를 관리하기 위해서는 디렉터리를 탐색하여 디지털 라이센스의 색인하는 색인기와 역파일로 색인된 색인 파일을 읽어서 텀(term)에 의해 검색하는 검색기를 설계[그림 7][그림 8]하였고, 자바의 루씬(Lucene)[17]을 이용하여 구현하였다. 루씬의 색인구조는 그 자체로 효율성이 높은 자료구조이며 검색엔진의 성능을 높이고 필요한 자원을 줄일 수 있도록 세심하게 만들어졌으며, 다중파일 색인구조와 복합파일 색인 구조를 지원한다. 또한 루씬은 증분색인을 지원하기 때문에 대량의 정보를 처리하면서 문서의 추가/삭제가 빈번한 경우, 즉 매번 재색인을 할 수 없는 경우에 사용하기에 아주 적당하다.

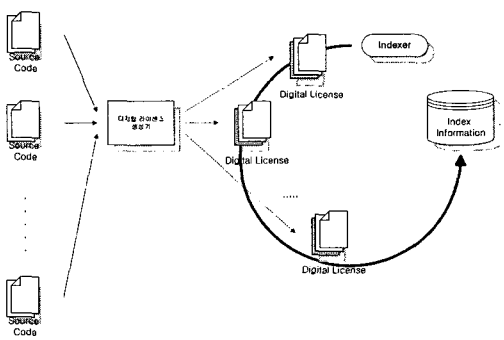

그림 7. 디지털 라이센스의 색인 생성기의 개괄도

본 연구에서는 복합파일 색인구조를 갖으며 증분색인이 가능하도록 구축하였다. 색인된 필드가 여러 개인 경

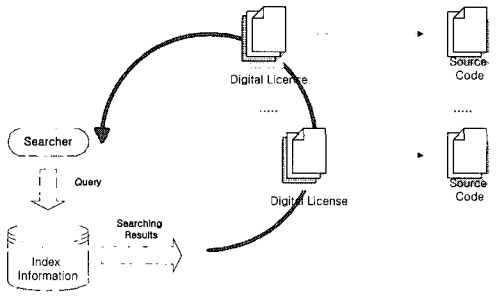

그림 8. 디지털 라이센스의 검색기의 개괄도

우 복합파일 색인을 사용하는 것이 매우 효율적이다.

루씬은 역파일 구조로 색인 문서를 저장한다. cur파일 색인의 구조는 간단하며, 내부적으로 텀이 중심을 이루도록 문서를 재정렬했으며 각 텀은 스스로를 포함하는 여러 개의 문서를 참조하는 구조이다. 색인 내부는 .fnm, .tis, .frq, .prx 등의 확장자를 갖는 파일에 저장한다. .fnm 파일은 세그먼트와 연관되는 모든 필드의 이름을 갖으며, .tis 파일은 세그먼트에 있는 모든 텀이 저장된다. 알파벳 순서로 정렬되며, 텀이 포함된 문서의 개수를 나타내는 문서 빈도수를 포함한다. 각 문서의 텀 빈도 수는 .frq 파일 내에 목록으로 저장된다. 위치 파일(.prx)은 .frq 파일에 목록으로 들어있는 각각의 문서를 기준으로 문서의 모든 텀에 대한 위치값을 저장한다. .prx 파일은 문 서 내 각 텀의 위치를 목록으로 갖는데, 이 위치 정보는 구문 질의나 스팬 질의 등에 사용된다. 그리고 색인에 질의를 보내면 질의와 일치하는 정도에 따라 정렬된 검색 결과의 집합을 얻는다. 여기서 단어 대신 텀을 사용되는데, 텀은 단어 자체와 함께 색인의 필드명과 결합된 값이며, 텀 질의는 대소문자를 구분한다.

2.1 디지털 라이센스의 색인기

프로그램 소스코드를 대처하기 위한 디지털 라이센스를 이용하기 위해서는 먼저, 디지털 라이센스를 파일 시스템에서 검색하고 디지털 라이센스의 정보를 이용한 역파일 색인을 생성한다. 생성된 역파일은 검색기에 의해서 디지털 라이센스의 정보 검색 및 위치를 찾는데 정보로 활용하며 결과는 [그림 9]와 같다.

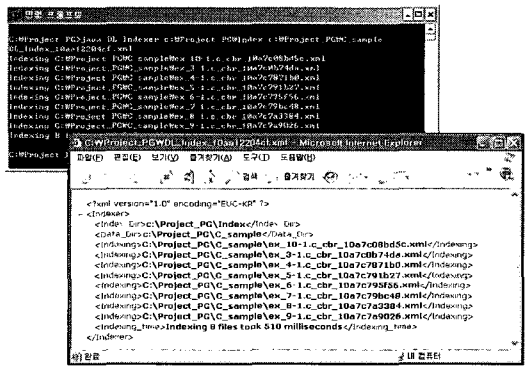

그림 9. 디지털 라이센스의 색인기 시뮬레이션 결과

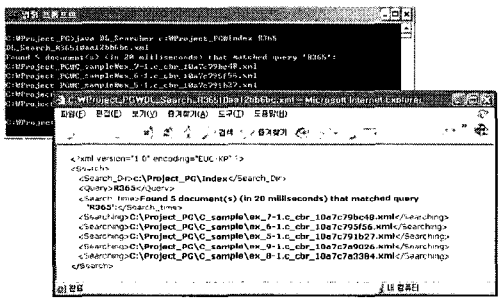

그림 10. 디지털 라이센스의 검색기 시뮬레이션 결과

5.2.2 디지털 라이센스의 검색기

사전에 생성한 디지털 라이센스의 역파일 색인 정보를 이용하여 검색어에 의한 일치되는 디지털 라이센스의 위치와 디지털 라이센스 파일 이름 정보를 생성하며 결과는 [그림 10]과 같다.

3. 디지털 라이센스에 의한 소프트웨어 소스 코드의 비교

먼저, 임의의 A와 B라는 두 개의 C 언어의 소스 코드를 이용하여 A와 B의 디지털 라이센스를 발행하였다. [그림 11]은 A와 B의 소스 코드를 이용하여 발행한 디지털 라이센스를 자체 개발한 XML TreeView에 의해서 나타낸 것이다. A와 B의 두 소스 코드는 거의 같은 트리 구조와 아키텍처 패턴을 가졌으나, 단말 노드 부분에서 두 소스 코드의 디지털 라이센스의 아키텍처가 다르게 표현되었으며, 더불어 노드의 패턴 정보도 다르게 표현되었다.

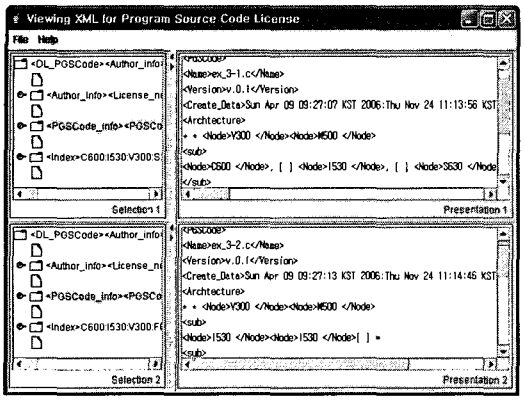

그림 11. 두 개의 디지털 라이센스에 의한 소스 코드의 아키텍처와 노드의 패턴 정보 비교

분쟁 해결을 위해서는 소소 코드의 패턴 매칭에 의한 방법도 있지만, 트리 구조로 표현된 소스코드의 아키텍처를 보면 전반적인 소프트웨어 소스코드의 조감도를 비교할 수 있는 장점을 갖으며, 분쟁 해결에 도움이 될 것이다.

인덱스 패턴은 프로그램을 구분 및 분류하는데도 이용되지만, 프로그램의 버전 관리에도 필요한 정보를 제공하는데 사용될 수 있다. 임의의 프로그램이 개발되어 유지보수되면 버전업 과정이 이루어진다. 버전업 과정을 위해서는 최소한 인덱스 패턴은 동일하더라도 프로그램 아키텍처나 노드의 패턴이 동일해서는 안된다. 프로그램의 버전업은 과거의 프로그램 버전과는 많은 부분은 동일하더라도 일부분에서는 노드 패턴의 변경, 노드 위치의 변동, 새로운 노드의 추가와 전 버전의 노드 제거 등의 과정이 나타나게 될 것이다. 이러한 정보에 의해서 프로그램의 버전업 관리가 이루어지게 된다.

4. 소스코드와 디지털 라이센스의 파일 크기 비교

통신 및 컴퓨팅 환경에서의 파일 크기에 대한 오버헤드 부분의 분석을 위해서 소프트웨어 소스코드의 크기와 디지털 라이센스의 크기 비교는 필요하다. 시뮬레이션 환경은 Pentium 4 3GHz, 512MByte의 메모리, 그리고 J2SDK 1.4.2_10 버전을 이용하였으며, 수행 결과 그래프를 Matlab을 이용하여 나타냈다. 소스 코드의 크기가 커질수록 디지털 라이센스의 크기를 나타내는 데이터 값의 진동의 폭은 더 커지면서 발산하는 경향을 [그림 12]에서 보였다.

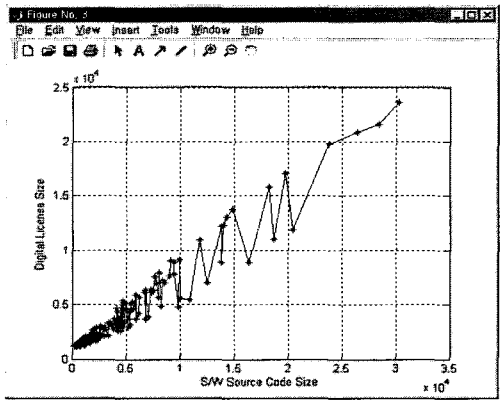

그림 12. 소스 코드와 디지털 라이센스의 파일 크기에 따른 비교 그래프

[그림 12]의 파일의 크기가 커질수록 주기적 진동에서 발산하는 경향을 갖는 이유는 개발자에 의해서 부가되는 코딩 스타일과 주석에 의해서 잡음이 부가된 것으로 추측된다. 같은 크기의 소스 코드도 특징 선택 항목을 무엇으로 결정하느냐와 개발자의 코딩 스타일 등의 소스 코드 개발 환경에 의해서 디지털 라이센스와 소스코드간의 진동에 영향을 미쳤다. 디지털 라이센스의 기초 정보(저작자에 대한 정보)의 payload 부분은 변동이 없으나, 소프트웨어의 아키텍처를 구성하는 부분에서 많은 변동이 발생하였다. 디지털라이센스를 생성할 패턴의 원천이 되는 특징 선택을 어떻게 결정하는가에 의해서 소프트웨어의 아키텍처 부분의 진동에 의한 발산을 줄일 수 있을 것이다.

VI. 결 론

디지털 저작물의 원천 요소인 소프트웨어 소스 코드에 대해서는 암호화이외에는 별다른 방법이 존재하지 않으며, 아직은 연구 초기 단계이다. 암호화 방법 역시 소프트웨어 소스 코드 개발자와 소프트웨어 소스 코드 구입자간의 1차적인 거래에만 저작권 관리를 지원할 뿐 그

이후에 발생하는 거래에 대한 저작권을 보호주지 못하고 있다. 소프트웨어 소스코드의 소유권 분쟁이 발생 시 소유권을 증명하기 위해서는 원본의 소프트웨어 소스코드를 공개해야만 하는 문제점을 갖고 있다. 본 논문에서는 소프트웨어의 소유권 주장에 하드카피이외에 더 많은 정보를 부여하며, 컴퓨팅 환경에서 자동으로 처리를 위한 프로토타입을 연구하였다. 소프트웨어 소스코드의 원본 공개로 인한 저작권 침해와 기술 유출을 막기 위한 소프트웨어 소스코드의 디지털 라이센스의 프로토타입을 제안하였다.

연구 내용으로는 소프트웨어 소스코드의 원본 판별을 지원하기 위한 소프트웨어 소스코드의 디지털 라이센스는 소스코드의 예약어를 파싱하여 계층구조를 갖는 XML 파일로 표현하며, 복잡한 소스코드 대신에 소프트웨어 소스코드의 아키텍처를 트리 구조 형태로 표현할 수 있다. 분쟁 해결에 소소 코드의 패턴 매칭 방법과 트리 구조로 표현된 소스코드의 아키텍처와 노드 정보를 이용하여 거시적인 소프트웨어 소스코드의 조감도를 비교할 수 있으므로 분쟁 해결에 도움이 될 것이다. 그리고 디지털 라이센스를 검색하기 위한 색인 및 검색에 대한 연구를 수행하였다.

향후 연구로는 구조적 프로그래밍 언어와 객체지향 프로그래밍 언어의 구분을 통합 및 소프트웨어의 소스 코드에 대한 정보를 포함 가능한 다양한 패턴 정보의 생성과 생성된 패턴 정보의 검색을 위한 색인기 및 검색기에 대한 연구가 필요하다.

참 고 문 헌

[1] Intel, "Content Protection in the Digital Home," Intel Technology Journal, Vol.6, Issue4, 2002(11).

[2] M. Ripley, "Utilizing Content Protection Technologies," Intel Developer Forum, 2002(9).

[3] A. M. Eskicioglu, "Multimedia Protection in Digital Networks," CNIS 2003, 2003.

[4] M. Rieger and S. Ducasse, "Visual Detection of Duplicated Code," Proceedings ECOOP Workshop on Experiences in Object-Oriented Re-Engineering, 1988.

[5] W. Yang, "Identifying Syntactic Differences Between Two Programs," Software-Practice and Experience, Vol.21, No.7, pp.739-755. July 1991.

[6] http://www.few.vu.nl/~dick/sim.html

[7] http://glimpse.arizona.edu/javadup.html

[8] http://www.ccsr.cam.ac.uk/~mw263/YAP.html

[9] http://www.cs.berkeley.edu/~aikem/moss.html

[10] 조동욱, "디지털 재산권 보호를 위한 S/W 프로그램 표절 감정 기술과 툴의 분석", 한국콘텐츠학회 2003 춘계종합학술대회 논문집, pp.177-184, 2003.

[11] http://www.pdmc.or.kr/

[12] 차병래, 김형종, 이동섭, "프로그램 소스 코드의 원본을 공개하지 않는 소유권 증명을 위한 디지털 라이센스 설계", 한국콘텐츠학회 논문지, pp.29-37, April 2006.

[13] J. Garms and D. Somerfield, "Professional Java Security," Wrox, 2001.

[14] http://www.w3.org/

[15] H. Matuyama, K. Tamura, and N. Uramoto, "XML and Java Developing Web Applications," Addision Wesley, 1999.

[16] 데이비드 토머스, 실용주의 프로그래머를 위한 버전관리 using CVS, 인사이트, 2005.

[17] E. Hatcher and O. Gospodnetic, "루씬 인 액션, 자바 검색 엔진", 에이콘출판사, 2005.

저 자 소 개

차 병 래(ByungRae Cha) 정회원

- 1997년 2월 : 호남대학교 컴퓨터공학과 (공학석사)
- 2004년 2월 : 목포대학교 컴퓨터공학과 (공학박사)
- 2005년 3월 ~ 현재 : 호남대학교 컴퓨터공학과 전임 교수

<관심분야> : 컴퓨터 시스템 보안, 컨텐츠 보호