ETRI

2019
Electronics and
Telecommunications
Trends

OSS 활용 고려사항과 검증 방법

OSS (Open Source Software) Usage Considerations and Verification Method

박정현 (J.-H. Park, jh-park@etri.re.kr) 품질보증연구실 책임연구원
박영식 (Y.-S Park, ysp@etri.re.kr) 품질보증연구실 책임연구원/실장
김현기 (H.-K. Kim, hkk@etri.re.kr) 언어지능연구그룹 책임연구원/PL
김영길 (Y.-K. Kim, kimyk@etri.re.kr) 언어지능연구그룹 책임연구원/그룹장

In this paper, we focus on the process of using open source software (OSS) and factors that should be considered when using project-based OSS. We also elaborate on how to avoid using OSS licenses in an OSS-based technology development process, why dual OSS licenses and security threats should be avoided, and the method of notification after use. In addition, the OSS license verification method and environment are described in the course of project development. In the verification method, the OSS license used for technology development in the course of project execution is validated in advance by the person who decides whether or not to use the OSS, and then additional verification using the tool after technology development. It is expected that this paper will be helpful for establishing the OSS usage consideration and the license verification procedure, and environment in the future.

* DOI: 10.22648/ETRI.2019.J.340113

* 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원[(R7119-16-1001, 지식증강형 실시간 동시통역 원천기술 개발), (2013-2-00131, (엑소브레인-1세부) 휴먼 지식증강 서비스를 위한 지능진화형 WiseQA 플랫폼 기술 개발)]을 받아 수행한 연구 결과임/본 연구 결과는 ETRI 공식 견해와 다를 수 있음.

Ⅰ. 서론

기술 및 제품 개발 시간 단축과 비용 절감, 기술 종속성 탈피, 그 외 자유롭게 사용할 수 있는 권리와 절약된 노력으로 기술과 제품 개발 수준 향상 등 OSS(Open Source Software) 활용 목적은 분명하기에 OSS 활용은 계속 증가할 것으로 기대된다. 실제 현재 70% 이상의 기업에서 기술과 제품을 개발하는데 OSS를 활용하고 있는 것으로 조사되고 있으며[1], 앞으로 4차 산업 혁명 시대를 맞이하면서 IoT, AI, 빅데이터, 클라우드 컴퓨팅, 차세대 모바일 기술 등 신기술 분야에서 오픈소스 활용은 더욱 확대될 것으로 보인다. 이렇듯 빠른 기술 변화와 소프트웨어 패러다임 변화 속에 ICBM(사물인터넷, 클라우드, 빅데이터, 모바일)을 활용한 산업 혁신이 필요한 상황에서 기업에서의 OSS 활용 증대는 더욱 가속화할 것으로 기대된다. 정부 및 공공기관에서도 정보화 사업 수행에 있어 활용되는 OSS 규정 및 지침을 제시하고 정보화사업 단계별 관리 · 점검 가이드 3.0을 기반으로 공개소프트웨어의 관리요소 및 고려사항을 보완한 거버넌스 가이드를 제시하고 있다[2], [3]. ICT 분야에서 다양한 R&D 과제를 수행하는 ETRI에서도 SW 개발이나 아웃 소싱을 통한 기술 개발에서 오픈 소스 사용 비중은 점점 늘어나고 있는 실정이다. 이에 본고에서는 새로운 기술과 제품 개발에서 OSS 활용 증대에 따라 조직에서 더욱 요구되는 OSS 활용 절차와 활용시 사전에 확인해야 할 고려 사항, 그리고 OSS 라이선스 검증 방법과 환경 등을 기술한다. 본고는 Ⅰ장 서론, Ⅱ장 OSS 활용절차와 고려사항, Ⅲ장 OSS 라이선스 검증 방법과 환경, 그리고 Ⅳ장에서 전체 내용을 요약한다.

Ⅱ. OSS 활용 절차와 고려사항

1. OSS 라이선스

SW는 저작권자만이 해당 SW에 대한 사용 권리를 갖는 지적 재산권에 의해 보호받는데 이런 해당 SW의 독점 사용 권리에 대해 SW 개발자와 사용자 간의 이용 방법과 조건을 명시한 대여 규칙을 정해 놓은 허가권을 SW 라이선스라 한다. 이런 OSS 라이선스는 소스 코드의 공개 여부에 따라 강력한 상호 허가권(Strong Reciprocal License)을 갖는 것과 약한 상호 허가권(Weak Reciprocal License)을 갖는 것, 그리고 관대한 허가권(Permissive License)을 갖는 것으로 나눌 수 있다. 강력한 상호 허가권(Strong Reciprocal License)을 갖는 OSS 라이선스는 공개 SW의 일부 혹은 전체를 사용하여 SW를 개발한 경우 사용 혹은 수정한 공개 SW와 더불어 개발 SW 소스 코드 전체를 공개해야 하는 것으로 대표적으로 AGPL-3.0, GPL-2.0, GPL-3.0 등이 있다. 그리고 약한 상호 허가권(Weak Reciprocal License) 갖는 OSS 라이선스는 공개 SW의 일부 혹은 전체를 사용하여 SW를 개발한 경우 사용 혹은 수정한 공개 SW 소스 코드만을 공개하면 되는 것으로 대표적으로 MPL-2.0, LGPL-2.1 등이 있다. 마지막으로 관대한 허가권(Permissive License)을 갖는 OSS 라이선스는 공개 SW의 일부 혹은 전체를 사용하여 SW를 개발한 경우 사용 혹은 수정한 공개 SW 소스 코드와 개발 SW 소스 코드를 공개하지 않아도 되는 것으로 대표적으로 MIT, BSD 3-Clause, Apache-2.0 등이 있다.

〈표 1〉은 주요 OSS에 대한 라이선스 의무 사항을 비교 및 요약한 것이다[1]-[6]. 공개된 OSS는 누구든 자유롭게 사용, 수정, 복제가 가능하다. 다만, OSS 사용하여 개발한 기술이나 제품을 만들어 제 3자에게 배포하거나 기술 이전하는 경우 OSS 활용한 개발 SW 소스 코드 전체를 공개해야 하는 OSS 라이선스 의무사항이 있음을 주의해야 하며 이와 같은 라이선스를 갖는 대표적인 OSS가 GPL 계열과 AGPL 그리고 LGPL과 MPL 및 EPL 등이 있다. 그리고 OSS 를 사용하여 개발한 기술에 특허가 포함된 경우 개발에 사용한 OSS 라이선스 의무사항으로 개발 소스 코드 전체를 공개해야 한다면

〈표 1〉 주요 OSS 라이선스 의무사항

주요 OSS 라이선스 의무사항	강력한 상호 허가권 (Strong Reciprocal License)			약한 상호 허가권 (Weak Reciprocal License)				관대한 허가권 (Permissive License)		
	GPL2	GPL3	AGPL3	LGPL2	LGPL3	MPL1	EPL1	MIT License	BSD3	Apache 2
1. 오픈소스의 자유로운 사용.수정.복사.베포 가능 여부	○	○	○	○	○	○	○	○	○	○
2. 오픈소스 사용 후 사용 및 수정한 오픈소스 및 관련된 개발 SW 소스코드의 제3자 베포시 (기술/제품/서비스 및 기술 이전 등) 수정한 오픈소스 내용 포함 개발한 SW 소스코드의 공개 여부	사용 및 수정된 GPL 소스코드와 링크 [정적/동적]시 관련 모든 소스코드[사용, 수정, 링크된 소스코드]를 GPL에 공개 의무 발생		사용 및 수정된 AGPL 소스코드와 링크 [정적/동적]시 관련 소스코드 [사용, 수정, 링크된 소스코드]를 AGPL에 모두 공개 의무 발생 /그외 네트워크 서비스 형태로 운영되는 경우 관련 소스 코드를 AGPL과 SW 사용자에게 모두 공개 의무 발생	사용 및 수정된 LGPL 소스코드와 정적 링크시 관련 모든 오브젝트 코드[사용, 수정, 정적 링크된 오브젝트 코드: Object Code]를 LGPL에 공개 의무 발생		수정 및 추가된 MPL 코드가 포함된 파일[소스코드]을 MPL에 공개 의무 발생	수정 및 추가된 EPL 코드가 포함된 모듈[소스코드]을 EPL에 공개 의무 발생	어떤 경우에도 소스코드 공개 의무 없음		
3. 오픈소스 사용 후 사용 및 수정한 오픈소스 및 관련된 개발 SW 소스코드의 제 3자 베포시 사용한 오픈 소스 수정 사항 및 관련된 개발 SW 소스코드에 대한 라이선스 [영문] 고지 여부	○	○	○	○	○	○	○	○	○	○
4. 오픈소스 사용 후 변경한 오픈소스 및 개발 SW 소스코드에 특허가 포함되어 제 3자 베포시 해당 특허에 대해 무료서 사용.판매.전송 행위 허락 여부 [특허 실시 허락 여부]	×	○	○	×	○	○	○	×	×	○
5. 오픈소스 사용 후 변경한 오픈소스 및 개발 SW 소스코드에 특허가 포함되어 제 3자 베포시 해당 특허 사용/실시에 소송 제기시 라이선스 종료 여부 [특허 보복 조항 유무]	×	○	○	×	○	○	○	×	×	○
6. 오픈소스 사용 및 복제에 따른 로열티 및 수수료 금지 여부	○	○	○	○	○	○	○	×	×	×
7. 오픈소스 사용 후 사용 및 수정 오픈소스 및 관련된 개발 SW 소스코드의 제 3자 베포시 사용 오픈 소스 수정 사항과 관련 개발한 SW 소스코드 이름, 상표, 상호 사용 금지 여부	×	×	×	×	×	○	×	×	×	×
8. 오픈소스 사용 및 복제에 따른 해당 오픈소스 최초 저자 보증 및 책임 여부	×	×	×	×	×	×	×	×	×	×

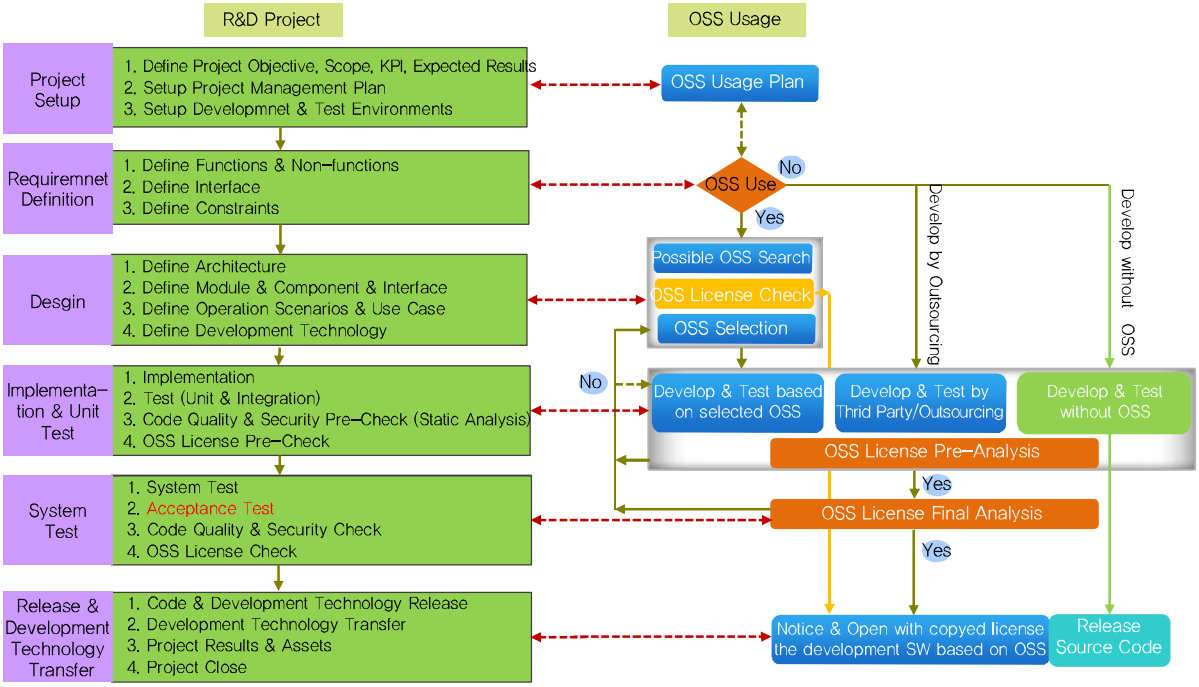
*GPL2/GPL3 (GNU General Public License version 2.0/3.0). *EPL1 (Eclipse Public License version 1.0)
*LGPL2/LGPL3 (GNU Library or Lesser General Public License version 2.0/3.0) *BSD2 (BSD version 2.0)
*AGPL3 (Affero General Public License version 3.0) *Apache2 (Apache version 2.0)
*MPL1 (Mozilla Public License version 1.0)
[출처] Reprinted from 박정현 외, "OSS 거버넌스 동향과 라이선스 분석 툴," 주간기술동향, 2018. 10. 3, pp. 14-25.

개발 기술에 포함된 특허의 권리 주장이 쉽지 않다는 사실도 OSS 사용전에 미리 고려해야 하는 부분이다. 그 외 대부분의 OSS는 사용 후 OSS 수정 내용과 함께 저작권 관련 사항을 해당 OSS 라이선스에 포함하여 고지해야 하는 의무가 있다는 사실도 기억할 필요가 있다.

2. OSS 활용 절차

OSS는 말 그대로 공개된 SW로 누구나 언제든지 사용할 수 있는 SW이다. 다만, OSS를 사용 혹은 수정하여 개발한 소스 코드 OSS를 사용 혹은 수정하여 기술이나 제품을 만들어 기술 이전 혹은 판매 등 제 3자에게 배포 및 사용하게 하는 경우 기술 개발에 사용한 OSS를 포함해 개발 기술 및 제품 전체 소스 코드는 기본적으로 해당 OSS 라이선스를 따르게 된다. 따라서 SW 개발 프로젝트 수행에서 OSS를 사용하여 기술이나 제품 개발을 고려하는 경우 먼저 개발하는 기술의 공개나 기술 이전 혹은 상품화 여부에 따라 어떤 OSS를 활용할 것인지를 해당 OSS 라이선스를 사전 검토하여 판단하는 것이 중요하다.

(그림 1)은 프로젝트 수행 시작 시점부터 프로젝트 완료 단계에 이르기까지 기술 개발 과정에서 OSS 활용 결정부터 활용 후 검증 및 분석, 그리고 라이선스 공지 단계까지의 절차와 연관 관계를 나타내고 있다. 중요한 것은 프로젝트 시작 시점에 기술 개발에 OSS 활용 여부를 결정하는 부분이고, OSS를 이용하여 기술 개발을 진행하는 것으로 결정한 경우 개발할 기술의 공개 혹은 제 3자 배포 여부 등 활용 목적에 따라 어떤 OSS를 사용하고 라이선스 따를 것인지 결정하는 부분이다. 또한, SW 개발 프로젝트 수행에서 OSS를 이용하여 기술 개발을 진행할 때 기술 개발 과정에서 개발자에 의해 사용한 OSS 라이선스의 확인 및 검증은 기본적으로 진행할 필요가 있으며, 기술 개발 완료 시점에서 사용한 OSS 라이선스에 대한 최종적인 확인 및 검증을 추가적으로 진행할 필요가 있다. 아울러 기술 개발 과정에서 진행한 OSS 라이선스 분석 및 검증 내용에 대해서는 개발 기술과 함께 별도 관리할 필요가 있으며, 마지막으로 기술



(그림 1) OSS 활용 절차

개발 과정에서 활용한 OSS 라이선스를 참조하여 OSS 수정 내용과 함께 저작권 관련 사항을 라이선스에 포함하여 고지하는 부분도 OSS 활용 단계에서 꼭 필요한 부분이다.

3. OSS 활용시 고려사항

가. 개발소스 코드 공개 의무 여부

OSS는 기본적으로 누구나 자유롭게 사용·수정·배포가 가능하다. 이는 가능한 많은 사람이 OSS를 이용할 수 있도록 하는 것이고, OSS 이용 후에도 OSS 커뮤니티에 또 다른 기여가 계속 가능하도록 하는 목적일 것이다. 따라서 OSS를 이용해 SW를 개발하는 경우는 기본적으로 개발한 SW를 공개한다는 생각을 갖고 OSS를 이용하는 것이 무리가 없을 것이다. 그럼에도 OSS 이용자는 활용 목적에 따라 자신이 개발한 SW의 공개를 원치 않을 수 있기에 OSS 사용전 OSS라이선스 공개 의무 여부를 반드시 확인할 필요가 있다. 물론 OSS를 이용해 개발한 SW를 제 3자에게 배포하지 않고 자신의 연구 목적으로만 사용하는 경우는 개발 SW의 공개 의무 사항을 고려하지 않아도 된다. 따라서 OSS를 이용해 개발한 기술을 이전하거나 제 3자 배포시 개발 기술 전체 소스 코드 공개 의무사항이 포함된 GPL 2.0, GPL 3.0, LGPL 2.0, LGPL 3.0 AGPL 3.0 등과 같은 강한 OSS 라이선스와 MPL 1.0, EPL 1.0 등과 같이 OSS가 포함된 파일이나 모듈만 공개하는 약한 OSS 라이선스, 그리고 MIT 1.0, Apache 2.0, BSD 등과 같이 개발 SW를 전혀 공개하지 않아도 되는 OSS 라이선스가 있음을 사전에 파악하여 기술 개발 과정에서 어떤 OSS를 사용할 것인지 고려할 필요가 있다.

나. OSS에 포함된 특허권 유효 여부

특허가 포함된 OSS를 사용하는 경우 해당 OSS와 포함된 특허를 일반적으로 무상 사용할 수 있다고 GPL3.0, LGPL 3.0, AGPL 3.0, Apache 2.0, 그리고 MPL 2.0 및 EPL과 같은 주요 OSS라이선스는 명시하고 있다. 다만, 제 3자의 특허가 해당 OSS에 포함된 경우 특허권을 주장할 수 있기에 OSS 사용 전 해당 OSS에서의 제 3자 특허 포함 여부와 특허권 유무상 여부의 확인이 필요하다. 그 외 OSS를 이용해 개발한 SW의 제 3자 배포 시 개발 소스 코드 공개 의무사항이 전혀 없는 라이선스 경우 개발 기술에 특허를 포함시켜 특허 유효 항목을 추가해 특허권을 주장할 수도 있으니 이 또한 해당 OSS를 사용하기에 앞서 반드시 라이선스 내용에 대한 확인이 필요하다.

다. 개발 SW 소스 코드에 특허 포함 여부

OSS를 사용한 개발 기술 소스 코드에 특허가 포함될 경우 GPL 3.0, LGPL 3.0, AGPL 3.0, MPL 계열, EPL 계열, 그리고 Apache 2.0등에서는 특허권을 주장할 수 없기에 가능한 특허 등 지적 재산권 내용을 포함하지 않도록 주의해야 한다. 만약 GPL 계열 OSS를 이용해 SW를 개발하고 개발한 SW 소스 코드에 특허를 포함해 공개하여 제 3자가 해당 SW를 사용할 때 이에 대한 특허권을 주장하거나 특허 소송을 제기하면 오히려 GPL OSS 라이선스는 종료되며 그 외 특허 소송 등에 따른 비용도 청구될 수 있음을 고려해야 한다. 따라서 OSS를 이용해 SW를 개발하는 경우 개발 SW 소스 코드 내용에 가능한 특허 등 지적 재산권이 포함되지 않도록 하는 것이 좋다.

라. 2개 이상의 서로 다른 OSS 사용가능 여부

GPL 2.0 및 Apache 2.0과 같이 2개 이상의 서로 다른 OSS를 사용하여 SW를 개발하는 경우 각 라이선스 간의 의무사항 충돌로 양립할 수 없는 경우가 발생할 수 있으니 OSS를 이용한 SW 개발에서 2개 이상의 서로 다른 OSS 사용은 가능한 피하는 것이 좋다. 따라서 2개

이상의 OSS를 이용하여 SW를 개발하는 경우 사용한 두 개 이상의 OSS 라이선스 의무사항 준수에 따른 상호 충돌 등 OSS 라이선스 양립성 문제가 발생하지 않도록 해당 OSS 라이선스 의무사항을 사전에 반드시 확인할 필요가 있다.

마. OSS 라이선스 정보 사전확인 여부

OSS를 사용하여 SW를 개발하는 경우 사용에 앞서 해당 라이선스 정보의 확인이 필요하다. OSS 라이선스 정보 확인은 보통 공개 SW 프로젝트 사이트나 소스코드가 배포되는 사이트에서 확인 가능하다. 그리고 다운받아 사용하는 OSS의 경우, 다운 시 포함된 파일 중 LICEENSE, README, COPYING, LEGAL, 그리고 NOTICE 등의 파일 내용을 통해 해당 OSS의 라이선스 정보 확인이 가능하다. 그 외에 다운 받은 OSS 소스코드의 헤더 주석문에 고지된 라이선스 문구로 확인할 수 있으며 일부 공개 SW의 경우 라이선스 확인이 어려운 경우가 있으나 이 경우는 웹 사이트(구글, github, sourceforge 등)에서 코드 검색을 통해 추가로 확인할 수 있다.

바. OSS 라이선스 고지방법 확인 여부

OSS 라이선스 항목에 소스코드 반환 의무가 있을 경우는 소스코드의 제 3자 배포 시 OSS 수정 내용과 저작권 정보 그리고 해당 라이선스 사본을 함께 고지 및 제공해야 한다. 또한, OSS를 이용해 SW를 개발하는 개발자는 개발 SW의 제 3자 배포 시 사용한 OSS 라이선스 정보를 삭제하지 않도록 주의해야 한다. OSS를 이용한 개발 소스코드의 공개에 따른 라이선스 고지 및 제공 방법은 해당 소스코드나 제품에 포함하여 배포 및 고지하는 방법, 웹 사이트에 소스코드를 업로드 해놓고 사용 OSS의 입수 경로와 수정 내용, 저작권 정보와 라이선스 사본 등에 대한 고지문을 제공하는 방법, 그 외 소스코드 요청 시 전자우편이나 CD 등을 이용하여 보내는 방법 등이 있다. OSS 라이선스 사본은 SPDX(Software Package Data Exchange)를 통해 표준화된 라이선스 정보 형식을 구할 수 있고, 그 외 한국저작권위원회의 OLIS(Open source SW License Information System)을 통해 국문 형태의 라이선스 정보를 도움 받을 수 있다.

사. OSS 보안 취약점 및 품질 수준검토 여부

현재 수많은 OSS가 공개되고 있으며 기술 및 제품 개발이나 SW 개발 프로젝트 수행에서 누구나 공개된 OSS 활용이 가능하다. 그러나 공개된 OSS 사용에 따른 품질 및 보안 취약점, 그리고 기타 어떤 문제 발생에 대해서도 OSS 공개자에게는 책임 의무가 없는 것이 현실이다. 따라서 OSS를 이용해 SW를 개발하는 경우 해당 OSS의 라이선스에 대한 확인도 필요하지만, 해당 OSS의 보안 취약점 및 품질 수준에 대한 사전 점검도 고려해야 하며 이와 같은 사전 점검이 전체 기술 개발 기간이나 프로젝트 수행 기간, 그리고 OSS 활용에 따른 비용 측면에서 유리하다. OSS의 보안 취약점 및 품질 수준 점검은 CC 인증 받은 상용 정적분석 툴 이용하여 확인할 수 있으며 굳이 상용 점검 툴을 이용하지 않는 경우는 일부 보안 취약점과 품질을 점검할 수 있는 수준으로 컴파일러에서 기본적으로 제공하는 기능을 통해 확인할 수 있고, 그 외 무료로 활용할 수 있는 CppCheck, PMD, SonarQube 등과 같은 오픈소스 기반 점검 툴을 개발 환경에 설치하여 활용하는 방법도 고려할 수 있다.

Ⅲ. OSS 라이선스 검증 방법과 환경

1. OSS 라이선스 검증 방법

OSS는 저작권이 있으며 OSS 사용과 배포 등에 따라 다양한 라이선스 의무사항을 포함하고 있다. 따라서 OSS 사용에 따른 라이선스 인식 부족과 의무사항 미유

지로 인해 원치 않는 법적 분쟁 발생이나 조직 이미지 손실 및 개발 지연 등의 문제가 야기될 수 있다. 그러므로 이런 문제를 사전에 예방할 수 있도록 OSS 사용에 따른 라이선스 검증은 반드시 필요하다. OSS 라이선스 검증은 크게 사람에 의한 검증, 기계에 의한 검증, 그리고 사람과 기계에 의한 병행 검증을 고려할 수 있다.

가. 사람(개발자/사용자)에 의한 검증

OSS를 이용해 기술과 제품을 개발하는 경우 먼저 사용하려는 해당 OSS 라이선스 의무사항을 확인해야 한다. 라이선스 의무사항 확인을 통해 개발자는 OSS 사용 여부를 최종 결정하고 해당 OSS를 사용하면 된다. 이때 해당 OSS 라이선스 내용에 대한 해석을 개발자 혼자 판단하기 어려운 경우는 법무팀 협조를 받아 진행할 수 있다. 인터넷(구글 등) 소스코드 검색을 통해 찾은 OSS의 경우 개발자 및 사용자는 OSS 사용에 앞서 해당 OSS 소스코드에 대한 추가적인 추적을 통해 OSS에 대한 최초 라이선스 확인이 필요하며 또한 OSS 사이트를 통해 확보한 OSS를 사용할 때 개발자는 해당 OSS 라이선스를 사전에 확인하고 그 내용을 숙지하고 있어야 한다. 이런 방법을 사람에 의해 수동[7]으로 수행하는 OSS 라이선스 검증 방식이라 할 수 있다. 이렇게 인터넷 검색 혹은 OSS 사이트를 통해 확보한 OSS와 사전 확인 및 검증한 해당 OSS 라이선스 경우는 이후 SW 개발 프로젝트 수행에서 OSS를 이용한 기술 및 제품 개발 진행에 따른 추가적인 라이선스 검증을 진행하지 않을 수 있다. 따라서 OSS 사용에 앞서 개발자가 해당 OSS 라이선스의 사전 확인을 한 경우가 OSS 라이선스 검증을 위한 방법중 한 경우라 할 수 있다.

나. 툴을 이용한 검증

툴을 이용한 OSS 라이선스 검증이란 개발자 및 사용자가 OSS를 이용하여 SW 개발할 때, 사용한 OSS 소스코드에 대한 라이선스 검증을 자동화 툴을 이용하여 수행하는 방법을 의미한다. 툴을 이용한 OSS 라이선스 검증은 해당 OSS 소스코드를 스캔 한 후 툴의 DB에 저장된 소스코드와 비교하여 검증하는 방식, 해당 OSS 소스코드 내 주석문 및 문자열 검색을 통해 비교 및 검증하는 방식, 그리고 해당 OSS 소스 코드에 대한 파일 사이즈를 비교하여 Checksum 방식으로 검증하는 방식 등이 있다.

- 문자열 검색 및 비교를 통한 검증: 해당 소스코드 내 주석문 및 문자열 검색을 통해 비교 및 검증하는 방식을 의미하는 것으로 대표적인 문자열 비교 OSS 라이선스 검증 툴은 GPL 2.0 오픈소스 기반 FOSSology가 있으며 그 외 AGPL 3.0 오픈소스 기반 Ninka와 TripleCheck가 있다.
- CheckSum 방식을 이용한 검증: OSS를 이용해 개발한 소스코드 파일 크기를 CheckSum 방식으로 비교해서 라이선스를 검증하는 방식을 의미한다. 아웃소싱을 통해 들어오는 개발 코드에서 사용한 OSS 소스코드에 대한 검증과 공급망 관리 차원에서 이를 적용하는 예도 있다.
- 코드 스캔을 이용한 검증: OSS 라이선스의 보편적인 기계적 검증으로 가장 많이 사용되고 있다. OSS를 이용해 개발한 SW 소스코드를 스캔하여 검증 툴의 DB로 갖고 있는 OSS 소스코드와 비교해서 일치하거나 유사한 패턴의 코드 존재 여부를 확인해서 검증하는 방식이다. 대표적인 상용 툴로 Black Duck(현재는 Synopsys)의 Protex가 있으며 그 외 FlexNet Code Insight 사의 Palamida가 있고, 기타 한국 저작권 위원회에서 개발한 오픈소스 기반 Code eye가 있다.

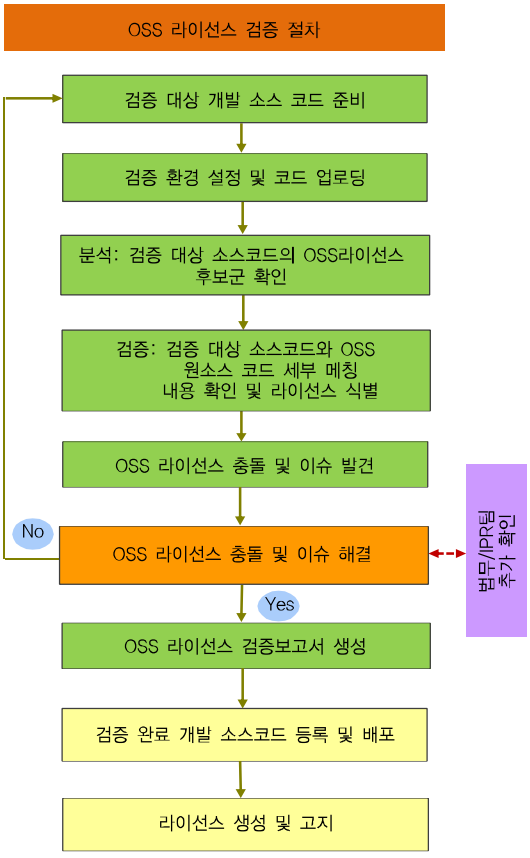
다. 툴과 사람에 의한 검증

OSS를 이용해 개발한 소스코드 크기가 크고 OSS 라이선스 검증을 위한 시간적 여유가 없는 경우 툴을 이용

해 OSS 라이선스 검증하는 것이 쉽고 빠를 수 있으나, 툴에서는 기본적으로 OSS 소스코드와의 유사 및 일치 패턴 만을 찾아 줄 뿐이다. 이 경우 툴에서 찾아낸 유사 및 일치 패턴의 OSS 소스코드는 한 개 이상 검출될 수 있다. 이렇게 여러 개의 유사 및 일치 패턴 소스코드를 찾아 낸 경우 SW 개발에서 실제 사용한 OSS 라이선스는 어떤 것인지 판단하기가 쉽지 않다. 더구나 오픈소스 특성상 시간이 지나면서 여러 개발자에 의해 수정 및 배포가 반복되면서 다수의 새로운 오픈 소스로 재 배포될 수 있는데 이런 경우 툴에서 찾아낸 여러 개의 OSS에서 실제 개발에서 사용한 OSS 라이선스를 찾아 결정하는 것이 쉽지 않다. 따라서 이런 경우 1차 툴에서 찾아낸 여러 개의 OSS 라이선스 중 SW 개발에서 실제 사용한 OSS가 어떤 것인지 최종 결정은 사람(개발자/사용자)이 해야 한다. 이렇게 1차 툴에서 유사 및 일치한 OSS 소스코드를 찾아내고 찾아낸 여러 개의 OSS에서 실제 개발에서 사용한 OSS에 대해 사람이 최종 결정하는 OSS 라이선스 검증 방식을 툴과 사람에 의한 검증이라 할 수 있다. 개발자는 검증 툴을 통해 찾아낸 소스 코드 패턴이 일치한 여러 개의 OSS 라이선스 중 실제 개발에서 사용한 OSS 라이선스가 어떤 것인지를 최종 결정하는 데는 소스코드 내 주석이나 라이선스 정보, 기타 소스코드에 대한 인터넷 검색을 통해 추가적인 정보 확인을 통해 진행할 수 있다. 그러나 오픈소스 라이선스에 대한 이해가 익숙하지 않는 개발자나 사용자 혹은 해당 SW를 직접 개발하지 않은 사람이면 개발 SW에서 사용한 OSS 라이선스에 대한 최종 판단을 하기가 쉽지 않을 수 있다. 따라서 이 경우는 OSS 검증 툴 사용에 익숙한 사람이나 라이선스 이해를 도울 수 있는 전문가를 통해 추가적인 지원을 받아 개발 SW에서 사용한 OSS 라이선스가 어떤 것인지를 최종 결정하도록 하는 것이 바람직하다. 즉, OSS를 이용해 SW 개발할 때 사용 한 OSS에 대한 라이선스를 툴과 사람에 의해 병행할 경우 1차 OSS 분석 툴을 이용하여 라이선스 검증을 진행한 후 SW를 실제 개발한 개발자나 사용자가 툴 및 라이선스 해석 전문가의 도움을 받아 최종적으로 결정하는 것이 정확한 검증 결과를 가져올 수 있다.

2. OSS 라이선스 검증 절차

프로젝트 수행 시작 단계에서 OSS를 이용하여 기술이나 제품 개발을 결정한 경우 먼저 기술이나 제품 개발 과정에서 사용할 OSS를 찾고, 이 OSS의 라이선스를 사용 전 반드시 확인할 필요가 있다. 또한, 기술이나 제품 개발 과정에서 사용한 OSS의 라이선스를 사전에 확인하지 못하고 OSS를 이용하여 기술 개발을 진행한 경우 OSS 라이선스 분석 툴을 이용하여 (그림 2)와 같은 절차를 통해 개발 소스 코드에 대한 라이선스 검증을 진행할 필요가 있다. (그림 2)는 OSS를 이용해 개발한 소스



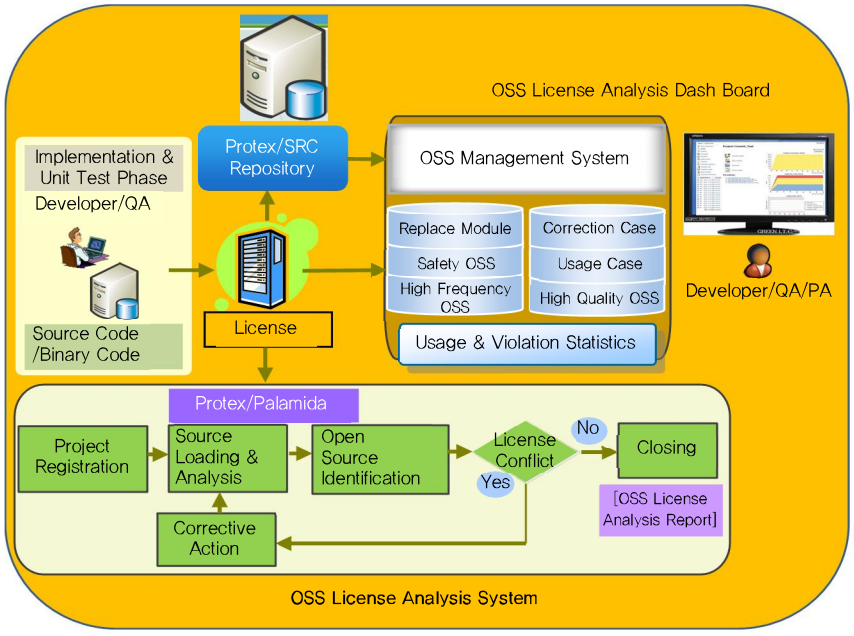
(그림 2) OSS 라이선스 검증 절차

코드에 대해 자동화 툴을 이용하여 OSS 라이선스를 분석 및 검증하는 과정으로, 먼저 개발한 소스 코드를 자동화 툴에 업로드 하고 스캔 한 후 자동화 툴 DB 내 구축된 OSS 코드와 비교하여 일치하는 OSS 코드를 찾아낸 다음 OSS 소스 코드 확인을 통해 해당 OSS 라이선스를 찾아 낸다. 이 과정에서 두 개 이상의 OSS가 SW 개발에 실제 사용한 OSS와 일치하는 경우 툴의 DB에서 찾아낸 OSS 소스 코드분석을 통해 실제 SW 개발에 사용한 OSS 라이선스를 찾아내야 한다. 또한 툴을 통해 찾아낸 OSS 라이선스가 개발 기술 이전이나 외부 배포에 따른 개발 소스코드 공개 의무 상황이 발생한 경우는 개발 소스 코드 공개 의무가 없는 다른 OSS로 대체하거나 해당 OSS부분에 대한 추가적인 개발을 통해 발생한 OSS 라이선스 충돌 및 이슈를 해결해야 한다. 아울러 발생한 OSS 라이선스 이슈에 대한 법률적 해석이나 자문이 필요한 경우 별도의 법률 전문가의 도움을 통해 개발 기술에서 사용한 OSS 라이선스 문제 및 이슈를 명백히 정리하고 종결할 필요가 있다.

3. OSS 라이선스 검증 환경

(그림 3)은 앞서 기술한 OSS 라이선스 검증 방법과 절차를 지원할 수 있는 룰기반 OSS 검증 환경 예이다. OSS 거버넌스를 수행하는 대부분의 조직에서는 OSS 라이선스 검증을 위해 기본적으로 (그림 3)과 같은 검증 환경 구축 방향으로 진행한다[8].

OSS 라이선스 검증 환경 구축을 위해서는 검증 서비스를 진행할 인력과 조직 그리고 OSS 검증 툴로 구성된다. 그 외 OSS 검증 환경 구축에는 OSS 검증 결과를 공유하고 다른 기술 개발에서 재활용할 수 있는 라이브러리 환경 구축도 고려할 수 있다. 따라서 OSS 거버넌스 환경을 갖는 조직에서는 대부분 비슷한 OSS 라이선스 검증 환경을 구축하여 운영하고, 그 외 자체적으로 OSS 라이선스 검증 환경을 구축하는 대신 OSS 검증 환경을 구축.운영하는 외부 기관을 통해 OSS를 이용해 개발한



(그림 3) OSS 라이선스 검증 환경

[출처] Reprinted from J.-H. Park, "Code Release Model for Technology Transfer on R&D Project of SW Development," PICMET, Honolulu, HI, USA, Aug. 19-23, 2018.

소스 코드에 대한 OSS 라이선스 검증 서비스를 지원 받을 수 있다.

4. OSS 라이선스 분석 및 검증 툴[3]-[5]

OSS를 사용할 때는 반드시 사용할 OSS 라이선스를 사전에 확인할 필요가 있다. SW 개발에 사용한 OSS가 외부 배포에 따라 공개 의무가 있는 라이선스 경우 OSS 사용하여 만든 기술이나 제품 그리고 서비스 연관된 관련 SW 소스코드를 모두 공개해야 하기 때문이다. 또한, 제품이나 기술 개발 일정이 촉박하여 OSS 라이선스 사전 확인 없이 OSS를 사용하는 경우는 개발한 기술이나 제품을 배포하기에 앞서 기술 개발에 사용한 OSS에 대한 라이선스 검증 과정이 필요하다. OSS 라이선스 검증은 개발 SW 소스코드의 크기에 따라 개발자 스스로 확인하는 방법과 OSS 라이선스 검증 툴을 이용하여 분석 및 검증하는 방법 등이 있다. 〈표 2〉는 OSS 라이선스 검증을 위해 현재 사용 가능한 상용 OSS 라이선스 분석 툴이다.

국내에서는 OSS 라이선스 자동 분석 및 검증 툴로 Protex를 가장 많이 사용하는 편이다. ETRI를 포함해 NIPA, LG, KT, 그리고 CJ 등에서 OSS 라이선스 분석 및 검증 툴로 사용하고 있는 Protex는 Black Duck 회사에서 개발해 판매 및 서비스해 오다가 현재는 Synopsys 사에서 인수하여 판매 및 서비스를 하고 있으며 국내의 경우만 Black Duck Software Korea가 별도 기술 지원을 하는 형태이다. 또한, 대부분의 OSS 라이선스 분석 및 검증 툴을 소스 코드를 스캔하여 툴 DB로 갖고 있는 OSS 소스 코드와 비교하여 라이선스를 분석 및 검증하는 방식을 가지며 바이너리 및 실행 파일을 비교 및 분석하는 툴도 일부 있다. 그 외 한국저작권위원회에서는 오픈소스 기반 OSS 라이선스 분석 및 검증 툴인 Code Eye를 개발하여, OSS 검증 및 컨설팅 서비스를 무료로 지원하고 있으며 NIPA 역시 공개 SW 역량 플라자 사업을 통해 OSS 라이선스 분석 및 검증 환경을 구축 및 운영해 오고 있으며 기업이나 기관에 OSS 라이선스 검증 및 컨설팅 서비스를 무료로 지원하고 있다.

〈표 3〉은 OSS 라이선스 분석 및 검증을 위해 사용 가능한 오픈소스 기반 OSS 라이선스 툴이다. 오픈소스 기

〈표 2〉 OSS 분석 및 검증 도구 I (상용)

도구명	제조사/라이선스	분석 방법	특징	비고
Protex	Black Duck --〉 Synopsys (BDSK*)	소스 코드 비교 분석/바이너리(기능) 비교 분석	Code/Binary Scanning	https://www.blackducksoftware.com/; https://www.synopsys.com/ [ETRI, NIPA*, 국방기술품질원, LG, 삼성, KT, 네이버 등 사용 중]
Code Eye	한국저작권위원회	소스 코드 비교 분석	Code Scanning	https://www.olis.or.kr/;[무료 검증 서비스 지원]
Protecode	Synopsys	소스 코드 비교 분석	Code Scanning	http://www.protecode.com/
WhiteSource	WhiteSource	소스 코드 비교 분석	Code Scanning	https://www.whitesourcesoftware.com/
Palamida	FlexNet Code Insight	소스 코드 비교 분석	Code Scanning	http://www.palamida.com/
FOSSID	FOSSID	소스 코드 비교 분석	Code Scanning	http://fossid.com/
OpenLogic	RogueWave	소스 코드 비교 분석	Code Scanning	http://www.roguewave.com/
BAT	Apache-2.0 --〉 Synopsys	바이너리 분석	Binary Scanning	http://www.binaryanalysis.org/
Codenomicon AppCHeck	Synopsys	바이너리 분석	Binary Scanning	http://www.codenomicon.com/; https://www.synopsys.com/

* NIPA[공개SW역량프라자], https://www.oss.kr/;[무료 검증 서비스 지원]
* BDSK: Black Duck Software Korea *BSD2 (BSD version 2.0)

[출처] Reprinted from 박정현 외, "OSS 거버넌스 동향과 라이선스 분석 툴," 주간기술동향, 2018. 10. 3, pp. 14-25

〈표 3〉 OSS 분석 및 검증 도구 Ⅱ (오픈소스)

도구명	제조사/라이선스	분석 방법	특징	비고
FOSSology	GPL-2.0	소스코드 파일내 라이선스 고지 문구 (문자열)/주석문 분석	문자열 [탐색] 분석 도구	https://www.fossology.org/
ScanCode & AboutCode	Apache-2.0	소스코드 파일내 라이선스 고지 문구 (문자열)/주석문 분석	문자열 [탐색] 분석 도구	https://github.com/nextB/scancode-toolkit
NinKa	AGPL-3.0	소스코드 파일내 라이선스 고지 문구 (문자열)/주석문 분석	문자열 [탐색] 분석 도구	http://ninka.turingmachine.org/
TripleCheck	AGPL-3.0	소스코드 파일내 라이선스 고지 문구 (문자열)/주석문 분석	문자열 [탐색] 분석 도구	http://triplecheck.net/download.html

[출처] Reprinted from 박정현 외, "OSS 거버넌스 동향과 라이선스 분석 툴," 주간기술동향, 2018. 10. 3, pp. 14-25

반 OSS 라이선스 분석 및 검증 툴은 FOSSology가 대표적이며 검증 방식은 소스코드 내 라이선스 주석이나 문자열을 검색 및 비교를 통해 분석하는 툴이다.

Ⅳ. 결론

IoT, 빅데이터, 로봇, AI는 물론 ERP, 게임, 보안, 클라우드, 가상화 등 다양한 분야에서 OSS의 활용은 계속 증대하고 있다. 무엇보다도 빠른 기술 변화 속도에 대한 대응과 개발 비용 절감 부분이 OSS 활용 목적의 가장 큰 이유이기도 하지만 기술 종속성 탈피와 OSS의 품질 수준도 뒤따르기 때문에 기술이나 제품 개발에서 더 많은 OSS 사용과 활성화가 증대되는 것이다. 대표적인 OSS 기반 제품으로 OS(Operating System) 분야에서는 Redhet과 CentOS를 들 수 있고, WEB/WAS 분야에서는 Apache, Nginx, Tomcat, Jboss를 들 수 있으며, DB 분야에서는 PostgreSQL/PAS, MariaDB/MySQL가 있으며, 그리고 솔루션 분야에서도 APOne과 ALM(Application Lifecycle Management)을 들 수 있듯이 요즘 많은 분야와 기술 및 제품 개발에서 OSS 활용이 증가하고 있는 추세이다. 또한 앞으로 기업과 공공기관에서 시스템 개발과 구축, 기술 및 제품 개발에서 OSS 활용 증가 추세는 지속될 것으로 기대한다. 이에 본고에서는 OSS 활용 증대에 따라 필요한 OSS 활용 절차와 고려사항, OSS 활용에 따른 OSS 라이선스 검증 방법과 분석 절차, 그리고 OSS 라이선스 분석 및 검증 환경을 기술하였다. 이는 OSS 활용과 라이선스 분석 및 검증 환경 구축, 그리고 OSS 거버넌스 수립 방향에 참고될 것으로 기대한다.

약어 정리

AGPL3	Affero General Public License Version 3.0
Apache2	Apache Version 2.0
BDSK	Black Duck Software Korea
BSD2	BSD Version 2.0
EPL1	Eclipse Public License Version 1.0
GPL2/GPL3	GNU General Public License Version 2.0/3.0
LGPL2/LGPL3	GNU Library or Lesser General Public License Version 2.0/3.0
MPL1	Mozilla Public License Version 1.0
OLIS	Open source SW License Information System
OS	Operating System
OSS	Open Source Software
SPDX	Software Package Data Exchange

참고문헌

[1] SK Telecom, "The Background & Future Direction of SK Telecom OSS Governance," Black Duck Korea Open Source Conference, 2015.

[2] 정보통신산업진흥원, "제5회 공개SW 거버넌스 아카데미," 2018.6.

[3] 정보통신산업진흥원, “제6회 공개SW 거버넌스 아카데미,” 2018.7.
[4] 정보통신산업진흥원, “공개SW라이선스 가이드,” 정보통신산업진흥원, 2017.1.
[5]한국저작권위원회, “오픈소스소프트웨어 라이선스 가이드 3.0,” 한국저작권위원회, 2016.11.
[6] 김병선, “OSS 라이선스 이해 및 관리,” Black Duck Korea, ETRI Seminar, 2017.8.
[7] TTA, “공개SW 성숙도 및 적용성 평가 표준”, 2017.12.
[8] J.-H. Park, “Code Release Model for Technology Transfer on R&D Project of SW Development,” PICMET, Honolulu, HI, USA, Aug. 19-23, 2018.
[9] 박정현 외, “OSS 거버넌스 동향과 라이선스 분석 툴,” 주간기술동향, 2018. 10. 3, pp. 14-25.