社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

Debianパッケージに対する依存関係を含むSPDXファイルの自動生成ツール

田邉　傑士†　　眞鍋　雄貴‡　　神田　哲也†　　井上　克郎*

† 大阪大学大学院情報科学研究科　〒565–0871 大阪府吹田市山田丘 1 番 5 号
‡ 福知山公立大学情報学部　〒620–0886 京都府福知山市字堀 3370
* 南山大学理工学部ソフトウェア工学科　〒466–8673 愛知県名古屋市昭和区山里町 18
E-mail: †{tk-tanab,t-kanda}@ist.osaka-u.ac.jp, ‡manabe-yuki@fukuchiyama.ac.jp, *inoue599@nanzan-u.ac.jp

あらまし　近年，サプライチェーン上のリスクを管理するため SBoM（Software Bill of Materials）の導入が進んでいる．SPDX（Software Package Data Exchange）は主要な SBoM の仕様の 1 つであり，利用者にライセンスを遵守させることを目的の一つとして作成された．他方，ライセンスを遵守するには依存関係を理解しておくことが重要である．しかし現状，依存関係を含めた SPDX ファイルを自動生成するツールは見受けられない．そこで本研究では Debian パッケージを対象に依存関係の記述を含めた SPDX ファイルを自動生成するツールを開発した．その結果，依存関係の記述を含んだ上で SPDX の要件を正しく満たす SPDX ファイルの生成に成功した．本論文ではその手法と生成結果について示す．

キーワード　SPDX，SBoM，OSS，依存関係，ソフトウェアライセンス

SPDX file generation tool for Debian packages including dependency relations

Taketo TANABE†, Yuki MANABE‡, Tetuya KANDA†, and Katsuro INOUE*

† Faculty of Engineering, First University　1–2–3 Yamada, Minato-ku, Tokyo, 105–0123 Japan
‡ Faculty of Informatics, The University of Fukuchiyama　3370 Azahori, Fukuchiyama, Kyoto, 620–0886 Japan
* Faculty of Science and Technology, Nanzan University　18 Yamazato-cho, Showa-ku, Nagoya, 466-8673 Japan
E-mail: †{tk-tanab,t-kanda}@ist.osaka-u.ac.jp, ‡manabe-yuki@fukuchiyama.ac.jp, *inoue599@nanzan-u.ac.jp

Abstract　In recent years, the Software Bill of Materials (SBoM) has been increasingly adopted to manage risks in the supply chain, and the Software Package Data Exchange (SPDX) is one of the main SBoM specifications, created to ensure that users comply with licenses. On the other hand, it is important to understand the dependencies in order to comply with the license. Currently, however, there are no tools that automatically generate SPDX files including dependencies. Therefore, we developed a tool to automatically generate SPDX files including dependency descriptions for Debian packages. As a result, we succeeded in generating SPDX files that include dependency descriptions and satisfy the SPDX requirements. This paper describes the method and results of the generation.

Key words　SPDX, SBoM, OSS, Dependencies, Software License

1. まえがき

近年ソフトウェアアプリケーションの機能複雑化に伴い，ソフトウェアの部品化はソフトウェア開発において避けられない傾向となっており，サードパーティ製のライブラリの利用はますます増加している．Contrast Security 社によると、今日のソフトウェアの大部分（79 %）にサードパーティ製のライブラリが使用されている [1]．サードパーティ製のソフトウェア部品やライブラリを利用することは開発にかかる費用と時間を大きく削減し，ある程度の安全性を担保してくれる一方で，依存関係がいくつかの問題を引き起こす潜在的なリスクをソフトウェアに持ち込むことがある．

それらの問題の 1 つがソフトウェアライセンス（以降単にライセンス）である．ソフトウェア部品を再利用する中で誤ったライセンスを使用すると，法的紛争に発展するケースもある [2] [3]．例えば，GPL ライセンスのソフトウェアに依存しながら MIT のライセンスで配布されているライブラリがあった場合，そのライブラリを利用したソフトウェアを出荷している

商用ベンダーにも法的な問題を引き起こす可能性がある．

このような問題に対処するべく多くの組織がソフトウェアリソースを記述するために Software Package Data Exchange（SPDX）のフォーマットを採用する傾向も出てきている [4]．SPDX とは、ソフトウェアパッケージに関連するコンポーネント、ライセンス、および著作権を伝達するために標準化されたフォーマットである [5]．

そのような傾向に追随するように様々な SPDX に関するツールが生まれてきている [6]．しかしながら，現状ではソフトウェアパッケージから SPDX ファイルを自動生成するツールの中にパッケージ間の依存関係を含む SPDX ファイルを生成するようなツールは見受けられない．ライセンス問題に取り組む上では依存関係の理解も重要な要素の 1 つである．そこで本研究ではまず Debian のパッケージを対象としてパッケージ間の依存関係を含む SPDX ファイルを自動生成するツールを開発した．

今回作成したツールでは Debian の binary パッケージを入力とし，ライセンスや著作権などの情報に加えて，そのパッケージが依存するパッケージとの関係を SPDX ファイルとして自動で生成する．また，特定のファイルを選択して分析にかけることで，Debian パッケージの入力を受け付ける主要な既存ツールと比較し，ライセンスや著作権表示の項目については，より詳細な情報を出力するようになっている．

この論文は次のように構成している．はじめに 2. 章で背景として SPDX の上位概念となる SBoM と SPDX について述べる．次に，3. 章で今回のツールを作成するにあたって参照した，既存の主要な SPDX の生成ツールについて紹介する．4. 章では，作成したツールによって生成される SPDX の仕様と処理の手順，実際の処理結果を示す．そして 5. 章では仕様や処理内容を決めるにあたっての議論について述べる．最後に 6. 章で論文を締めくくり，まとめと今後の研究の方向性について概説する．

2. 背　　景

本論文で説明するツールが生成する SPDX とその上位概念となる SBoM が置かれる立場は 2021 年に大きく変化し，SBoM の認知度は向上した．それでもなお，一般的に認知されているとまでは言えない．そこで本章では SBoM と SPDX をそれらに関する近年の動向とともに説明する．

2.1 SBoM

SBoM とは「Software Bill of Materials」の略称であり，ソフトウェアのコンポーネントとその依存関係、およびライセンスデータを一意に識別する形式的で機械可読なメタデータのことを指す．

SBoM を利用することによってソフトウェアの利用者はソフトウェアの運用する上で重要な情報を管理しやすくなり，ライセンスや脆弱性といった問題に対処しやすくなるとして，その活用と普及が期待されている．特に，SolarWinds 社の Orion platform に対する攻撃が明らかになって以降，ソフトウェアのサプライチェーンを管理しようという動きは高まりを見せている [7] [8]．SBoM にとって大きな転機となったのは 2021 年 5 月にアメリカのバイデン政権が署名した「Executive Order on Improving the Nation's Cybersecurity」に関する大統領令であり，その「Sec. 4. Enhancing Software Supply Chain Security.」の項では SBoM の提供をソフトウェアの配布者に要求している [9]．また，2021 年 7 月には先の大統領令を受けて NTIA（アメリカ合衆国国家電気通信情報管理庁）が SBoM の最小要素を定めるなど SBoM の概念は整備されつつある [10]．

2.2 SPDX

SPDX とは「Software Package Data Exchange」の略称であり，LinuxFoundation が主体となって作成している「来歴、ライセンス、セキュリティ、およびその他の関連情報を含む、ソフトウェアの部品表情報を伝達するための標準化されたフォーマット」である．SPDX は SBoM の形式の 1 つである [10]．

パッケージのメタデータを記述する形式を標準化することによって組織の垣根を超えてメタデータをやり取りすることができ，組織独自の記法でメタデータを作成・交換するよりも経費や労力を削減できる．すでにいくつかの標準化された SBoM の形式が存在しているが SPDX はその中で最も国際的に認められた形式の 1 つであり，2021 年 9 月には ISO/IEC JTC1 標準として認められている [11]．

SPDX 作成の目的の 1 つはソフトウェアの利用者にライセンスを遵守させることである．SPDX ライセンスリストを作り，ライセンスの表記を整備していることからも示唆される．このことについては NTIA も同様の見解を示している [10] [12]．一方でライセンスを遵守する上で重要となるパッケージの依存関係に関する記述は必須項目ではなく，依存関係を自動で生成するツールは，公式サイトが記載する Open Source Tools[1]の中には存在しなかった．

SPDX は Tag:Value 形式や RDF 形式，json 形式など様々な形式を取ることが可能となっており，どの形式でも伝えられる情報に差はない．The Software Package Data Exchange® (SPDX®) Specification Version 2.2.2 には Tag:Value 形式と RDF 形式の例のみが記載されているため，この 2 つの型式が基本と考えられる．

SPDX には多くのフィールドがあるが，全てのフィールドを入力すると情報が多く大変なため，手作業での作成を念頭に置いて使いやすさを重視した SPDX Lite[2]という SPDX のサブセットにあたる形式が存在する．

3. 既存ツール

本論文で説明するツールの作成にあたってはいくつかの既存ツールを参照し，利用した．本章ではツールによるファイルの解析にあたって API を利用した FOSSology と，SPDX を生成する主要なオープンソースツールとして FOSSology とともに Open Source Tools や経済産業省の「サイバー・フィジカル・セキュリティ確保に向けたソフトウェア管理手法等検討タスクフォースの検討の方向性」に記載されている scancode-toolkit について紹介する．

（注1）：https://spdx.dev/tools-community/
（注2）：https://spdx.github.io/spdx-spec/SPDX-Lite/

3.1 FOSSology

FOSSology とは、ライセンスや copyright を検出するシステムおよびツールキットである。Web アプリケーションや CLI ツールとしてパッケージやファイルのライセンスや copyright などのメタデータを解析し，SPDX 形式を含む様々な形で解析結果に対するレポートを出力する．また，Web UI[3]を提供しているため，手軽に SPDX 生成ツールとして利用することができる．

対応しているパッケージの形式は（iso, tar, rpm, jar, zip, bz2, msi, cab, etc）と記載されており幅広い．しかし，著者らが試用した際は Debian パッケージの解析はできなかった．

3.2 scancode-toolkit

scancode-toolkit とはコード内の依存関係、ライセンス、およびその他の関連情報を検出するオープンソースのツールである．

パッケージをインストールして利用する CLI ツールであるため，利用するためには scancode-toolkit のインストールが必要となるが，Linux や MacOS，Windows など主要な OS に対応しており導入は容易である．

公式サイト[4]によれば様々なパッケージ対応していると書かれており，Debian パッケージにも対応していると書かれているが，著者らが試用した際はパッケージを展開しなければ解析することができなかった．

同じライセンスを重複して検出することがあることに加え，ファイル単体での解析方法が不明であったため，本ツールにおいては FOSSology を利用した．

4. 実装内容

本研究では 3.1 節にて紹介した FOSSology を利用して SPDX ファイルを自動生成するツールを作成した．生成する SPDX ファイルの仕様は SPDX のサブセットである SPDX Lite をベースにしている．本章では生成する SPDX ファイルの仕様の詳細と，Debian パッケージの入力を受けてから SPDX ファイルを生成するまでの手順，そして SPDX ファイルの生成結果について説明する．

4.1 生成する SPDX ファイルの仕様

本節では生成する SPDX ファイルの仕様の概要と SPDX Lite と比較してどのような仕様変更を行なっているか，そしてその理由について述べる．

SPDX は非常に広範囲なパッケージの情報を表記することができる（ここでは必須項目かどうかについては議論しない）．しかしながらその全てを記述することは簡単では無い上に，そのような SPDX ファイルでは可読性も落ちてしまう．加えて手作業での解析を前提としたようなフィールドも存在するため，自動生成に適さない場合もある．

そこで本ツールにおいては読みやすさと使いやすさに重点を置いた SPDX のサブセットである SPDX Lite をベースとして SPDX ファイルの仕様を作成した．本ツールが生成する SPDX ファイルの仕様は SPDX Lite を本ツールの用途に合わせて一部を変更したものとなっている．

本ツールが生成する SPDX ファイルが持つ SPDX のフィールドとその簡単な説明を表 1 に示す．SPDX は様々な形式を取ることが可能だが，本ツールでは人にとっての読みやすさを考慮して Tag:Value 形式のみをサポートしている．

本ツールが生成する SPDX ファイルの仕様では大きく分けて以下の 5 種類の情報を出力する．ただし現状では，依存先のパッケージに SPDX ID を付与するだけで，パッケージの解析を推移的には行わない．

Docmnet Information　SPDX ドキュメント自体に関する情報
Creator Information　SPDX ドキュメントの作成に関する情報
Package Information　パッケージに関する情報
License Information　ライセンスリストにないライセンスが検出された場合に，そのライセンスの詳細に関する情報
Dependency Packages Information　依存関係にあるパッケージの情報

ここからは本ツールが生成する SPDX ファイルの仕様における SPDX Lite の仕様[5]からの変更点について説明する．

まず，License list version フィールドを追加した．このフィールドはどのバージョンのライセンスリストを利用しているかを示すものである．SPDX では各ライセンスに固有の識別名を付けており，それによってライセンスの表記を冗長な文章から大きく短縮している．また，名前に対応するライセンスのテキストを明確にしたことは曖昧な部分も多かったライセンス表記の整備にも貢献している．この識別名とライセンスの内容のセットのリストがライセンスリストである．ライセンスリストが更新され，識別名とライセンスの対応関係が変わった際にも対応できるようにするため，このフィールドは追加すべきと考えた．

次に Relationship フィールドを追加した．このフィールドは SPDX ID を付与した SPDX の要素間の関係を記述するものである．記述できる関係としては包含関係や依存関係など様々なものがある[6]．本ツールが生成する SPDX ファイルの仕様においては SPDX ドキュメントがどのパッケージについて記述したものか示すためとパッケージ間の依存関係を記述するために使用している．このフィールドは本ツールの目的であるパッケージ間の依存関係を記述する上で不可欠であるため追加した．

そして License Information にあった License Comment フィールドを削除した．このフィールドはライセンスリストにないライセンスが見つかった際にそのライセンスに対する深い分析やライセンスの文章以外の詳細情報を記載するものである．SPDX Lite は手作業での生成を前提としたものであるため，ライセンスの情報を捕捉するフィールドとして含まれていたと考えられる．しかし，ツールによる分析ではそこまでを行うことは難しく，FOSSology の解析結果からも提供されない．さらにこのフィールドは SPDX の必須要素でもないため本ツールが生成する SPDX ファイルの仕様からは削除した．

（注3）：https://FOSSology.osuosl.org/repo/
（注4）：https://scancode-toolkit.readthedocs.io/en/stable/getting-started/home.html
（注5）：https://spdx.github.io/spdx-spec/SPDX-Lite/#g3-table-of-spdx-lite-fields
（注6）：https://spdx.github.io/spdx-spec/relationships-between-SPDX-elements/#111-relationship-field

SPDX のフィールド名	簡単な説明
SPDX Version	SPDX 仕様のバージョン
Data License	SPDX ドキュメント自体のライセンス
Document Information	
SPDX Identifier	SPDX ドキュメント自体の ID
Document Name	SPDX ドキュメントの名前
SPDX Document Namespace	SPDX ドキュメント固有の名前空間
Creation Information	
Creator	SPDX ドキュメントの作成者 （人名・ツール名）
Created	SPDX ドキュメントの作成日
LicenseListVersion	準拠したライセンスリスト のバージョン
Package Information	
Package Name	パッケージの名前
Package SPDX Identifier	パッケージにつける ID
Package Version	パッケージのバージョン
Package File Name	パッケージのファイル名
Package Download Location	パッケージのダウンロード場所 （git など）
Files Analyzed	パッケージのファイルを解析したか
Package Home Page	パッケージのホームページ
Concluded License	宣言されたライセンスから 結論づけられるライセンス
Declared License	宣言されたライセンス
Comments on License	ライセンスに対するコメント （解析ツールなど）
Copyright Text	copyright
Package Comment	パッケージへのコメント
Relationship	SPDX 要素間の関係
License Information	
License Identifier	ライセンスリストに無い ライセンスの ID
Extracted Text	ライセンスの内容
License Name	ライセンス名
Dependency Packages Information	
Package Name	パッケージの名前
Package SPDX Identifier	パッケージにつける ID
Package Download Location	パッケージのダウンロード場所
Concluded License	宣言されたライセンスから 結論づけられるライセンス
Declared License	宣言されたライセンス
Copyright Text	copyright
Files Analyzed	パッケージのファイルを解析したか

表 1　生成する SPDX が持つフィールドとその説明
（太字は SPDX-Lite から付け加えたフィールド・情報）

4.2 処理内容

Debian パッケージの入力から SPDX ファイルの生成までの処理の手順を図 1 に示す．

（1） 入力された Debian パッケージファイルを展開する

（2） その中から control ファイルと copyright ファイルを検出する

（3） control ファイルから SPDX に必要なデータを抽出する

（4） copyright ファイルを FOSSology を使って解析し，ライセンスと copyright の情報を検出する

（5） control ファイルから抽出したデータと，FOSSology の解析結果を合わせて SPDX を生成する

この copyright ファイルと control ファイルは Debian パッケージにおける必須ファイルであるため必ず存在すると仮定した．もし見つからなかった場合は対応するフィールドは NOASSERTION として処理する．

ここからは control ファイルの情報の扱いについて詳細に説明する．control ファイルのフィールドを SPDX のどのフィールドに対応させるか，またその control ファイルのフィールドが必須であるかどうかを表 2 に示す．

まず，Package フィールドと Version フィールド，Homepage フィールドはそれぞれパッケージの名前，パッケージのバージョン，パッケージの Web サイトの URL を表すフィールドであり，対応する SPDX のフィールド Package Name，Package Version，Package Home Page に対応させた．

次に，パッケージ間の依存関係にあたる control ファイルのフィールドと，それに対応させる SPDX ファイルの Relationship フィールドでの依存関係の表現について説明する．対応関係は以下の 3 パターンである．RUNTIME_DEPENDENCY_OF などの依存関係の表現は SPDX で定義されたものを用いている．

- Depends はパッケージが主要な機能を提供する上で必要となるパッケージ群を指定するフィールドである．そのため，実行時に必要な依存関係として SPDX ファイルでは RUNTIME_DEPENDENCY_OF を使って関係を表す．
- Recommends は特別な理由がなければインストールすることを推奨するパッケージ群を指定するフィールドであり，Suggests は存在すればパッケージの有用性が増すパッケージ群を指定するフィールドである．そのため，選択的な依存関係として SPDX ファイルでは OPTIONAL_DEPENDENCY_OF を使って関係を表す．実際には Recommends の方が強力な依存関係であるが SPDX で定義されている依存関係の表し方においてはその差異を表現できない．
- Pre-Depends は制約付きの依存関係であり，パッケージのインストール前にインストールされていなければパッケージが機能しないパッケージ群を指定するフィールドである．そのため，ビルドの時点で必要なパッケージ群であると解釈し，SPDX ファイルでは BUILD_DEPENDENCY_OF を使って関係を表す．

これらの関係に加えて，control ファイルでは競合してしまうパッケージ群などの関係を示している場合があるが，現状の SPDX では定義された表し方が存在しないことや表したい依存関係とは異なる概念を含むため，本ツールでは処理しない．また，control ファイルが表すことのできるパッケージ間の関係ではバージョンを指定できるが SPDX の記法には無いため省略する．さらに「|」という複数のパッケージのどれらかがあれば良いという記法もあるが今回のツールが動く状況下ではどちらが実際にインストールされているか判断できず，また SPDX に存在しない記法でもあるため，その全てのパッケージを依存関係

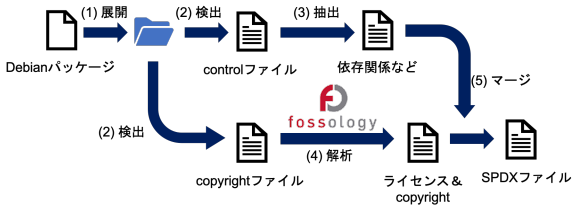

図 1　Debian パッケージから SPDX ファイルを生成する処理手順

contol ファイルのフィールド	Required	対応させる SPDX ファイルのフィールド
Package	Yes	Package Name
Version	Yes	Package Version
Homepage	No	Package Home Page
Depends	No	Relationship (RUNTIME_DEPENDENCY_OF)
Recommends	No	Relationship (OPTIONAL_DEPENDENCY_OF)
Suggests	No	Relationship (OPTIONAL_DEPENDENCY_OF)
Pre-Depends	No	Relationship (BUILD_DEPENDENCY_OF)
Description	Yes	Package Comment

表 2　control ファイルのフィールドと SPDX ファイルのフィールドとの対応関係

として併記することとした．

4.3　生 成 結 果

作成したツールを用いて実際に SPDX ファイルを生成した．対象としたのは python3.9 の ubuntu20.04 用の Debian パッケージである[7]．対象とした Debian パッケージの control ファイルの内容を図 2 に示す．また，その結果ツールが生成した SPDX ファイルの一部を図 3 と図 4 に示す．図 4 では License Information の一部を省略している．これらの図から表 2 の通り，Package，Version，Depends，Suggests，Description のフィールドを SPDX ファイルのフィールドに対応させられていることが示された．この SPDX ファイルを公式の SPDX Online Tool の Validate 機能[8]で検証したところ「Success!」と表示され，SPDX の要件を正しく満たしていることが確認できた．また，Debian Popularity Contest が公開している使用率が高い Debian パッケージ Top10[9]（perl-base，libc6，debianutils，dpkg，libacl1，debconf，tar，adduser，libstdc++6，coreutils）を対象にツールを実行したところ，いずれも SPDX の要件を正しく満たす SPDX ファイルが生成された．ただ，libstdc++6 においては copyright ファイルが検出できず，ライセンスと copyright の解析ができなかった．

5.　実装についての考察

本ツールが生成する SPDX ファイルの情報については資源や

```
Package: python3.9
Version: 3.9.7-2build1
Architecture: amd64
Maintainer: Ubuntu Developers <ubuntu-devel-
discuss@lists.ubuntu.com>
Original-Maintainer: Matthias Klose <doko@debian.org>
Installed-Size: 554
Depends: python3.9-minimal (= 3.9.7-2build1), libpython3.9-stdlib
(= 3.9.7-2build1), media-types | mime-support
Suggests: python3.9-venv, python3.9-doc, binutils
Breaks: python3-all (<< 3.6.5~rc1-1), python3-dev (<< 3.6.5~rc1-1),
python3-venv (<< 3.6.5-2)
Section: python
Priority: optional
Multi-Arch: allowed
Description: Interactive high-level object-oriented language
(version 3.9)
 Python is a high-level, interactive, object-oriented language. Its
3.9 version
 includes an extensive class library with lots of goodies for
 network programming, system administration, sounds and graphics.
```

図 2　python3.9 パッケージの control ファイル

```
SPDXVersion: SPDX-2.1
DataLicense: CC0-1.0

##-------------------------
## Document Information
##-------------------------

DocumentNamespace: http://spdx.org/spdxdocs/
python3.9_3.9.7-2build1-55975825-90aa-465f-afcd-d2026e906ecb
DocumentName: python3.9_3.9.7-2build1
SPDXID: SPDXRef-DOCUMENT

##-------------------------
## Creation Information
##-------------------------

Creator: Tool: SPDX Lite + dependencies
Creator: Tool: spdx2
Creator: Person: Taketo
Created: 2022-05-26T03:07:22Z
LicenseListVersion: 2.6

##-------------------------
## Package Information
##-------------------------

PackageName: python3.9
PackageVersion: 3.9.7-2build1
PackageFileName: python3.9_3.9.7-2build1_amd64.deb
SPDXID: SPDXRef-python3.9
PackageDownloadLocation: NOASSERTION
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: Python-2.0
PackageLicenseDeclared: BSD-2-Clause
PackageLicenseDeclared: Zlib
PackageLicenseDeclared: OpenSSL
```

図 3　生成した SPDX ファイルの一部

技術的な観点から代理の情報を出力しているフィールドが存在する．この章ではそれらのフィールドの現在の出力内容とその是非について考える．

5.1　SPDX Document Namespace フィールド

このフィールドには本来 SPDX ファイルの内容を参照できる URL が入る．しかし，本ツールではそのような SPDX ファイルを配布する Web サイトをホストしていないため SPDX の定義に基づいた固有の識別子を与えている．しかし，このままの仕様ではこのパッケージに依存する新たなパッケージの SPDX ファイルを生成する際など，他の SPDX ファイルからこのファイル内の SPDX 要素を参照する場合，外部参照先となる URL が存在しないため，SPDX ファイルの閲覧者がこのパッケージの情報を見ることができないという問題がおきる．従って将来的には SPDX ドキュメントを置くための Web サイトをホストする必要がある．

5.2　Concluded License フィールド

このフィールドには本来パッケージ内部で宣言されたライセ

（注7）：http://archive.ubuntu.com/ubuntu/pool/universe/p/python3.9/python3.9_3.9.5-3ubuntu0~20.04.1_amd64.deb
（注8）：https://tools.spdx.org/app/validate/
（注9）：https://popcon.debian.org/by_vote ※Accessed on 06/26/2022

```
FilesAnalyzed: false
PackageComment: <text> Interactive high-level object-
oriented language (version 3.9)
 Python is a high-level, interactive, object-oriented
language. Its 3.9 version
 includes an extensive class library with lots of goodies
for
 network programming, system administration, sounds and
graphics.
</text>
Relationship: SPDXRef-DOCUMENT DESCRIBES SPDXRef-python3.9
Relationship: SPDXRef-python3.9-minimal
RUNTIME_DEPENDENCY_OF SPDXRef-python3.9
Relationship: SPDXRef-libpython3.9-stdlib
RUNTIME_DEPENDENCY_OF SPDXRef-python3.9
Relationship: SPDXRef-media-types RUNTIME_DEPENDENCY_OF
SPDXRef-python3.9
Relationship: SPDXRef-python3.9-venv RUNTIME_DEPENDENCY_OF
SPDXRef-python3.9
Relationship: SPDXRef-python3.9-doc OPTIONAL_DEPENDENCY_OF
SPDXRef-python3.9
Relationship: SPDXRef-binutils OPTIONAL_DEPENDENCY_OF
SPDXRef-python3.9

##-------------------------
## License Information
##-------------------------

LicenseID: LicenseRef-Dual-license
LicenseName: Dual-license
...

##-------------------------
## Dependency Packages Information
##-------------------------

PackageName: python3.9-minimal
SPDXID: SPDXRef-python3.9-minimal
PackageDownloadLocation: NOASSERTION
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
FilesAnalyzed: false

PackageName: libpython3.9-stdlib
SPDXID: SPDXRef-libpython3.9-stdlib
PackageDownloadLocation: NOASSERTION
PackageLicenseConcluded: NOASSERTION
PackageLicenseDeclared: NOASSERTION
PackageCopyrightText: NOASSERTION
FilesAnalyzed: false

PackageName: media-types
```

図 4　生成した SPDX ファイルの一部

ンスが 1 つの場合はそのライセンスが入り，パッケージ内部で宣言されたライセンスが複数存在する場合はそれらのライセンスから互換性を考慮してパッケージ全体に適用されると考えられるライセンスが入る．しかし，現在互換性を確認する明確な基準は存在しない．そのため本ツールでは Declared License フィールドでライセンスが複数存在する場合には Concluded License フィールドは NOASSERTION として処理することとした．これにより SPDX ファイルの閲覧者にとっては自分でライセンスを読み解く手間が生じる事になるが，ライセンスを誤認してパッケージを使用してしまうリスクを避けることが最も重要であるため，完全なライセンスの互換性解決ツールが出ない限りはこの手法をとるべきだと考える．

6. まとめと今後の課題

本研究では依存関係を含めた SPDX の自動生成ツールを作成した．本ツールでは Debian パッケージを解析するにあたって実際にパッケージを展開し control ファイルや copyright ファイルを分析することでライセンスや copyright，依存関係などにまつわる SPDX ファイルの生成に必要な情報を入手している．

このツールを利用すれば Debian パッケージを入力としてパッケージに関する依存関係を Relationship フィールドに示した SPDX ファイルを得ることができる．このツールで得られる SPDX ファイルは SPDX の簡易なサブセットである SPDX Lite をもとに，パッケージの基本情報と依存関係に重点を置いた簡潔なものであるが，SPDX の要件を正しく満たし，SPDX Online Tool の Validate にも通るものである．

今後の課題としては依存関係が構築された環境下にあるパッケージを推移的に解析し，依存先にあるパッケージに対しても SPDX ファイルを作成することで，実際に動作環境にあるパッケージ群の依存関係を示す SPDX を生成するツールの開発に取り組んでいきたい．

謝　　辞

本研究は JSPS 科研費 JP18H04094, JP19K20239, JP21K02862 の助成を受けたものです．

文　　献

[1] J. Williams and A. Dabirsiaghi: “The unfortunate reality of insecure libraries”, https://cdn2.hubspot.net/hub/203759/file-1100864196-pdf/docs/contrast_-_insecure_libraries_2014.pdf (2012).

[2] R. M. Azzi: “Cpr: how jacobsen v. katzer resuscitated the open source movement”, University of Illinois Law Review, p. 1271 (2010).

[3] F. S. F. Europe: “Fsfe compliance workshop discovers gpl violation by fantec, welte wins in court - fsfe”, https://fsfe.org/news/2013/news-20130626-01.en.html (2013).

[4] F. Mancinelli, J. Boender, R. di Cosmo, J. Vouillon, B. Durak, X. Leroy and R. Treinen: “Managing the complexity of large free and open source package-based software distributions”, 21st IEEE/ACM International Conference on Automated Software Engineering (ASE’06), pp. 199–208 (2006).

[5] Linux Foundation and its Contributors: “A common software package data exchange format”, 2.0 edition (2015).

[6] L. Foundation: “Tools - software package data exchange (spdx)”, https://spdx.dev/resources/tools/.

[7] C. D. Andrew Archer, Doug Bienstock and G. Edwards: “Highly evasive attacker leverages solarwinds supply chain to compromise multiple global victims with sunburst backdoor | mandiant”, https://www.mandiant.com/resources/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor (2020). (Accessed on 06/26/2022).

[8] 経済産業省 商務情報政策局：“最近の産業サイバーセキュリティに関する動向について”, https://www.meti.go.jp/shingikai/mono_info_service/sangyo_cyber/wg_seido/wg_uchu_sangyo/pdf/003_03_00.pdf (2021).

[9] U. S. E. O. of the President[Joe Biden]: “Executive order 14028: Improving the nation’s cybersecurity | the white house”, https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/ (2021).

[10] N. Telecommunications and I. Administration: “The minimum elements for a software bill of materials (sbom)”, https://www.ntia.doc.gov/files/ntia/publications/sbom_minimum_elements_report.pdf (2021).

[11] I. O. for Standardization: “Iso - iso/iec 5962:2021 - information technology — spdx specification v2.2.1”, https://www.iso.org/standard/81870.html (2021).

[12] 株式会社エヌ・ティ・ティ・データ経営研究所：“令和元年度サイバー・フィジカル・セキュリティ対策促進事業（ソフトウェアを安全に利活用するための基盤構築に向けた調査）”, https://www.meti.go.jp/meti_lib/report/2019FY/000537.pdf (2020).