SPDX Generation via Yocto and the LiD Scanner

Mark Charlebois (Director of Engineering), Craig Northway (Director of Engineering)
Qualcomm Technologies, Inc.
February 2017
Agenda

- Primer: License scanning, Yocto Project™, SPDX™, FOSSology, Dronecode™
- Current Limitations of Yocto code scanning
- New Yocto srcmap layer
  - srcmap.bbclass overview
- License Identifier (LiD)
  - Background & Motivation – FOSSology
  - Approach & Evaluation
  - Example Output
- Status and Future Work
- How can you help?
Related LF Collaborative Projects

**SPDX, FOSSology, Yocto, Dronecode**

- **SPDX**
  - Standard format for documenting license information for files and packages

- **FOSSology**
  - Originally from HP, used for code scanning

- **Yocto Project**
  - LF Collaborative Project, includes core components form OpenEmbedded project
  - Provides SPDX support using FOSSology

- **OpenEmbedded**
  - Uses bitbake, derived from Gentoo portage

- **Dronecode**
  - Project for UAVs that will use Yocto for SW releases
Existing SPDX bbclass

- Adds `do_spdx` task after `do_patch`
  - Only analyzes patched source directory
- Runs FOSSology during a build
  - Causes license analysis to have the overhead of a full build
- No package dependency information captured
- Could not get it to work with current FOSSology version
- Does not use latest SPDX encoding lib ([https://github.com/spdx/tools-python.git](https://github.com/spdx/tools-python.git))
Unpack, patch, and configure are all done in ARCHIVER_WORKDIR
Current Limitations of Yocto code scanning

- **spdx.bbclass**
  - Scans only the `$S` dir after patching
  - Only runs in the context of a full build
  - Different results for every patch, `bbappend`, etc
  - Only package level granularity

- **archive.bbclass**
  - Provides way to store original src, patches and recipe, without full build
  - Designed for src distribution, not code scanning (no dependency info)

- **FOSSology integration**
  - Seemingly unmaintained
  - Jethro branch did not work with available FOSSology version
  - FOSSology was not simple to setup
New Yocto srcmap layer

Objectives

- Store license analysis of upstream packages in common repository
- Packages (from recipe, inc, bbappend) are only small deltas of patches
  - Provides high degree of re-use
- Minimal copies of upstream code
- No full build required
- Separates indexing of code locations and package dependencies from scanning
  - Enables parallel License scanning
- Can be used with any code scanner
- Can produce SPDX output with package dependency info
Overview of srcmap bbclass

- Generate full package hierarchy with locations of source files
- LiD scan for all source files for original sources, patched files, and package specific files
- Duplicate packages not re-scanned
  - Independent of Yocto build env and can be analyzed in parallel
- Postprocessing and LiD use pypy for speed
Output of srcmap
Background & Motivation: FOSSology

FOSSology Nomos

- Regular expression based snippet matching tool
- Accurate in detecting common OS license types (over 80%)
  - Limitation on coverage (only 2/3 of verbatim SPDX licenses were covered at the time of evaluation)
  - Performed better in our real world evaluation data set (94%)
- Not intended for full license text matching.
  - Only snippets are matched.
  - Occasionally, no strings are matched, but suggests the presence of license.
- Challenges:
  - Updating or adding new OS licenses
  - Handling corner cases (dealing with comments or other unexpected sequence of characters)
  - Computationally not cheap – depending on the input and the regular expression rules -> hence, only snippets.
Background & Motivation: FOSSology

FOSSology Monk

- A sequence-of-words matching tool
  - Comments and other non-defined words can be handled through pre-defined number of jumps and ignores

- Designed for full license matching

- Challenges:
  - Low coverage in identifying SPDX licenses

Coverage of the FOSSology tool for SPDX licenses (Dec. 2015)
LiD Goals

- Identify OS licenses and OS license text in source code
- Aid in license compliance due diligence
- Generate SPDX file level information to accompany product distribution
LiD (License Identifier)

- Scans source code and identifies the license and license text region using known license templates.

- LiD Approach:
  - Built using the bag-of-words model used in natural language processing
  - Training set built from SPDX license list and ability to add custom templates to training set
  - Computes a distribution of unigrams, bigrams and trigrams for detecting the license type
  - When detecting license text regions, LiD employs edit distance metrics to find the optimal start and end positions of the identified license text

- LiD vs. FOSSology (Nomos, Monk)
  - Performs equal or better than nomos agent in our evaluation (see backup slides).
  - Ease of setup and adding new licenses
  - Parameter tuning and scoring
LiD Example Output

Number of matches: 1
Match: GPL-2.0-based_on_1.0
(lines = [0:16], orig score = 0.708295, region score = 0.106578)

Copyright (C) 1990 Free Software Foundation, Inc.

This file is part of GNU CC.

GNU CC is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2, or (at your option) any later version.

GNU CC is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with GNU CC; see the file COPYING. If not, write to the Free Software Foundation, 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA. */
For an example package in our recipes DoSOCSv2 found 5 licenses using Nomos agent:

DoSOCSv2 was integrated into Yocto by Fujitsu, but the patches never made it to the mainline

Example SPDX output from DoSOCSv2:

LicenseID: LicenseRef-MIT-style
LicenseName: MIT-style
ExtractedText: <text>Permission to use, copy, modify, distribute, and sell this software
#  and its documentation for any purpose is hereby granted without
#  fee, provided that the above copyright notice appear in all copies</text>
LicenseComment: <text>found by nomos</text>
Yocto – Sample package

Using LiD

LiD found 8 licenses: “curl”, “MIT”, “GPL-3.0”, “HPND”, “FSPAP”, “X11”, “FSFUL”, “FSFULLR”.

Example SPDX output generated using LiD:

LicenseID: LicenseRef-8
LicenseName: MIT
LicenseComment: <text>Found by LiD with score: 0.48763573088431555</text>
ExtractedText: <text>
Copyright (c) 2005, 2006, Oracle and/or its affiliates. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice (including the next paragraph) shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. </text>
Status and Future Work

- Initial LiD Release: [https://www.codeaurora.org/qualcomm-ostg-lid](https://www.codeaurora.org/qualcomm-ostg-lid)
- Initial prototype of srcmap layer: [https://github.com/mcharleb/meta-srcmap](https://github.com/mcharleb/meta-srcmap)
- Dockerfiles: [https://github.com/mcharleb/container-lid](https://github.com/mcharleb/container-lid)
- srcmap LiD integration and Dockerfile not yet public
- Re-architect srcmap on top of archiver.bbclass
- Optimal and parallel file hashing and code scanning
- Utilize Yocto License info
  - COPY_LIC_MANIFEST
  - COPY_LIC_DIRS
Status and Future Work

Continued

- Create commons of scanned upstream src
  - Can GREATLY reduce scan times

- How to handle/share manual review/changes to automated license data?
  - Tooling for viewing and reviewing 100Mbs of SPDX data?

- LiD:
  - Performance improvements when testing multiple licenses candidates within large files
  - General LiD accuracy improvements
    - Improve ability to match on short licenses
  - Binary files
    - Extract strings for matching?
  - Integration into other tools (FOSSology, DoSOCsv2, etc...)
Backup Slides
LiD vs FOSSology

Coverage

How many SPDX licenses can it detect?

- 100% by LiD
- 70% by Nomos
- 29% by Monk
LiD vs FOSSology

Accuracy

License Type Identification
- LiD performs equal to or better than Nomos in accuracy.
  - LiD & Nomos in our evaluation dataset (94% accuracy).
    - Data set only contains popular SPDX license types.
    - Qualcomm created evaluation dataset from OSS files (see backup slides)

License Text Region Identification
- LiD finds the whole license text
- Nomos finds snippets
- Monk finds the full text, but has low coverage
  - 29% of SPDX license list (at the time of evaluation)
  - Only found MIT and NCSA in our evaluation dataset
LiD Vs FOSSology

Flexibility

- Ease of setup
- Adding new licenses
  - LiD: Easy to add new licenses, just add templates, automated updating to official SPDX license list.
  - Nomos: Difficult to add new license files – need to create a new regex rule
- Parameter tuning
  - LiD: Score and other parameters allow LiD to be integrated into different applications
  - Nomos/Monk: Parameter tuning is either not available or intuitive
- Score
  - LiD: Review priority may be set.
  - Nomos and Monk do not produce scores for a match.
Evaluation Dataset

- **OS category**
  - 223 total files (exe, conf files)
  - 135 labeled as OS
  - 20 OS license types
  - 5 non-SPDX license types (ANTLR, LibTomCrypt, Public Domain, Perl, RSA Data Security).

- **Proprietary category**
  - 381 files total
  - 5 files labeled as OS license text
  - 3 OS license types (1 unique one – ISC)
LiD vs FOSSology

Recall Accuracy on Presence of License

Does a file contain open source license text or not?

<table>
<thead>
<tr>
<th>Scanner</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nomos</td>
<td>94.07%</td>
</tr>
<tr>
<td>Monk</td>
<td>22.22%</td>
</tr>
<tr>
<td>Hybrid</td>
<td>94.07%</td>
</tr>
<tr>
<td>License Identifier*</td>
<td>94.28%</td>
</tr>
</tbody>
</table>

Recall

% of the files that contain open source license text that are correctly identified by the scanner as such.
LiD vs FOSSology

Recall Accuracy on License Type

Does a file contain open source license text or not, and if it does, what license(s) does it contain?

<table>
<thead>
<tr>
<th>Scanner</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nomos</td>
<td>64.65%</td>
</tr>
<tr>
<td>Monk</td>
<td>20.86%</td>
</tr>
<tr>
<td>Hybrid</td>
<td>82.73%</td>
</tr>
<tr>
<td>License Identifier*</td>
<td>94.28%</td>
</tr>
</tbody>
</table>

Recall

% of the files that are correctly identified as containing open source license text and for which the correct license is identified.
Overview of srcmap bbclass

Adds new tasks
- do_srcmap_unpack, do_srcmap_patch and do_srcmapall

Analyzes all SRC_URI elements
- Downloads are separate, dependent packages
- Patches and other config are part of package SPDX record
- Recipes with no SRC_URI usually set ${S} to some previously unpacked src
  - Create a dependency on the package that provides the unpacked src
- Generate full package hierarchy with locations of source files

Analyze license info
- LiD scan for all source files for original sources, patched files, and package specific files
- Duplicate packages not re-scanned
- Independent of Yocto build env and can be analyzed in parallel
Acknowledgements

- Yocto Project™, SPDX™, Dronecode™ are trademarks of the Linux Foundation in the United States and other countries.