

Thesis
Presentation

Ferdaous
Boughanmi
Supervisor :

Giuliano Antoniol
and Yann-Gaël
Guéhéneuc

Introduction

Context

Problematic

Thesis

Related work

System
meta-model for
license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of
License Statements
and SC

Study Definition

Study Setup

Analysis Methods

not all caps

SOURCE CODE AND LICENSE STATEMENT CO-EVOLUTION

~~Thesis Presentation~~

Ferdaous Boughanmi

~~Supervisor : Giuliano Antoniol and Yann-Gaël Guéhéneuc~~

SOCCKER Lab. and Ptidej Team – DGIGL, École Polytechnique de
Montréal, Québec, Canada

5 December 2012

Introduction

Context

Introduction

Context

Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

Definition ~~License - License Statement - Notice File~~

- ▶ Software ~~L~~icense: governs the legal use and redistribution of a system and its components.
- ▶ License Statement - Notice File: license information is included in each source code file as a textual, it includes copyright information: the names of contributors to the source code file and the copyright owner~~s~~? or as a notice file for the whole system or for each component.

Introduction

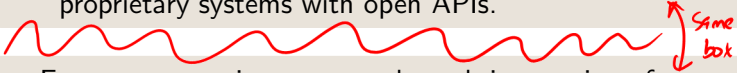
Context

Why do ~~Heterogeneously~~ ^Heterogeneously-licensed ~~Systems~~ ^Systems exist?

- ▶ Need to ~~decrease~~ ^{reduce} the cost of software development and ~~more rapid~~ ^{Speed up} product development.
- ▶ Availability of Open Source Software (OSS) and of proprietary systems with open APIs.

⇒ Encourage creating systems through integration of pre-existing components instead of writing the whole system by themselves.

⇒ This practice leads to systems composed of heterogeneously-licensed components.



Introduction

Problematic

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

Why licenses could be a Constraint?

- ▶ large number of licenses, i.e., more than 70 OSS licenses exist today.

⇒ Make it hard to understand the legal constraints of a complete software.

⇒ Increases the probability of violating one or more licenses.

Introduction

Problematic

~~The kind of reuse could even add additional problems~~

A derivative work is “a work based upon one or more preexisting works in which a work may be recast, transformed, or adapted”.

⇒ Poses more constraints in the case of reciprocal license, i.e GPL.

⇒ For example, when we connect to a GPL-licensed components by instantiating a class, this is considered to be derivative work, which requires the final work to be licensed under the GPL.

one by

Introduction

Problematic

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

Evolutionary Problems

~~The~~ license can evolve like any other software artifact.

- ▶ Can either be changed pervasively throughout a software system (e.g., ~~the~~ switch GPLv2 to GPLv3) or only locally (e.g., contributor name added to one file).
- ▶ Can be coarse-grained (switch to a different license), fine-grained (copyright year updated) or anything in between (clause added or removed).

Introduction

Problematic

Evolutionary problems: Evolution introduces an additional risk of license terms violation.

If one component changes its license, then it might no longer be possible to use it because of incompatibility of licensing with other components.

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

Introduction

Problematic

\usepackage[english]{babel}

~~Evolutionary problems:~~ Example

“Java Classpath exception”: the Java JDK was distributed under the Common Development and Distribution License. Sun then decided to change the license of the JDK to GPLv2.

⇒ ~~A~~ ~~problem~~: any system that runs under the JVM dynamically links to the runtime library that is part of the JVM. Hence, this system is considered to be derivative work of the JVM ~~and~~ and should be licensed under the GPLv2.

⇒ Sun added the Classpath exception to the GPL2 to resolve this issue ~~X~~: linking to the provided library is not considered a derivative work.

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the Meta-Model

Concret Part of the Meta-Model

Architecture Part of the Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

Introduction

Problematic

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

*merge with
previous slide ?*

⇒ Developers should be aware of license changes and their possible effects.

Introduction

Problematic

License evolution study

To avoid impact of license evolution we must:

- ▶ Look at changes to the license statements.
- ▶ We must also analyse who changes those statements, since regular developers likely are not sufficiently trained to deal with licenses.
- ▶ In addition, detecting various licenses and their interaction.

⇒ It is laborious task to perform it manually.

⇒ There is need for license evolution management techniques to assist developers to organise their software licenses in a better way.

Introduction

Thesis

Thesis

License statements are **changing frequently**, but **do not necessarily coevolve** with source code and managed by a **minority of developers** that are probably experts.

describes the

needed

~~We will follow two steps to validate our thesis:~~

Validation

- Step1 - System Meta-model for License Analysis:
Our meta-model indicates which data ~~is~~ *needed* related to license evolution, and needs to be analysed. Our meta-model might be also the support to develop a tool for license evolution management.

move to p. 12/42 and put the two step one after the other

Introduction

Context

Problematic

Thesis

Related work

System

meta-model for license analysis

Abstract Part of the Meta-Model

Concret Part of the Meta-Model

Architecture Part of the Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

Introduction

Thesis

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

- Step2 - Co-evolution of License Statements and Source Code:

RQ1 Do licenses co-evolve with source code at the system level?

RQ2 What types of license changes are performed?

RQ3 Who performs license changes?

Our results could be used for future work to develop better licensing tools and techniques.

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

Meta-model and Software License Analysis

- ▶ German et Hassan (2009):
 - Established a meta-model limited to licenses modeling.
 - Identified 12 patterns commonly used to solve license incompatibilities.
- ▶ German et al. (2010):
 - ✗ Mined inconsistencies in Fedora-12 OS ~~and~~ identified the licenses and dependencies of all files using RPM package description. ✗ They found many cases in which the license of a package changed, and this created problems, e.g., the package still declared the old license, making the package use potentially incompatible.

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

Meta-model and Software License Analysis

- ▶ Alspaugh et al. (2009) /:
 - Derive a meta-model for licenses from the meta-model of German and Hassan (2009).
 - Identified 12 patterns commonly used to solve license incompatibilities.
- ▶ Tuunanen et al. (2009) X J:
 - Identified dependencies using compiling information from GCC, ar (an archive tool), and ld (a linker) J-
 - License identification using templates and regular expressions.

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

License Change Analysis

- ▶ Hindle et al. (2008):
 - Identified license statement changes as one of the reasons of large commits.
- ▶ Di Penta et al. (2010):
 - Changes occurring to the copyright years depend on the amount of changes made by developers during the years.
- ▶ Manabe et al. (2010):
 - Projects sometimes choose radically different licenses.
 - The usage of different licenses in the kernel files of operating systems is similar to each other.

System *M* meta-model for *L* license analysis *A*

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

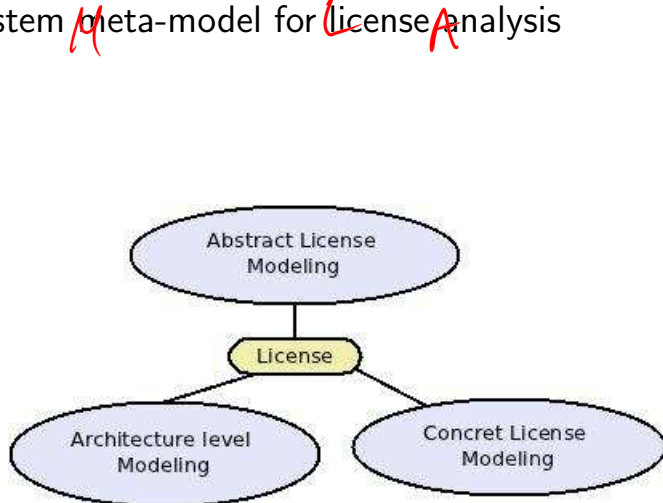
Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion



System ~~Meta~~-model for ~~License~~ analysis

Abstract Part of the Meta-Model

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

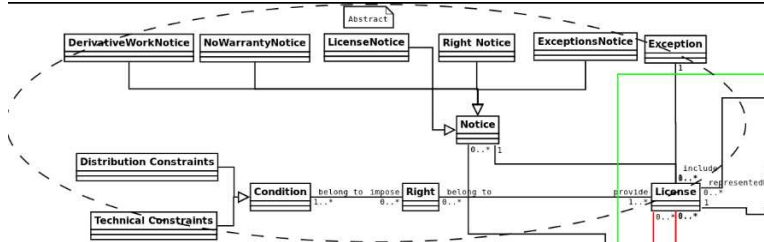
Study Setup

Analysis Methods

Results of the study

Conclusion

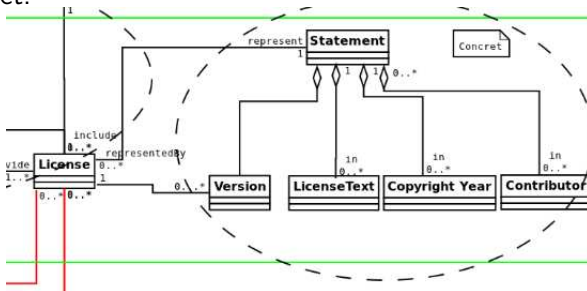
Abstract:



System Meta-model for License analysis

Concret Part of the Meta-Model

Concret:



Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

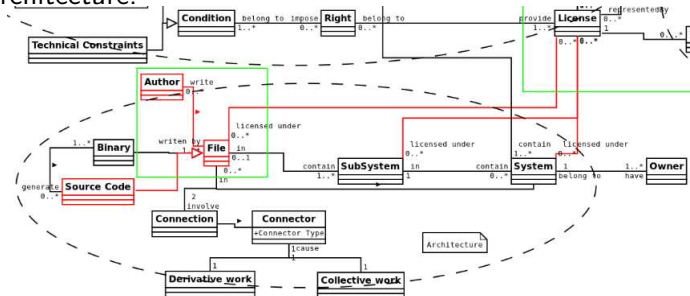
Analysis Methods

Results of the study

Conclusion

Architecture Part of the Meta-Model

Architecture Part of the Meta-Model



Co-evolution of License Statements and SC

Study Definition

Introduction

Context

Problematic

Thesis

Related work

System

meta-model for
license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

?

remove
blank

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

Co-evolution of License Statements and SC

Study Definition

Goal is to perform an exploratory analysis of the co-evolution of license statements and source code.

Purpose is to better understand when developers change license statements, who performs such changes. Such an understanding could help improve license change management.

Quality focus is the consistency of license changes.

Perspective is of both researchers and practitioners who are interested in understanding license statement change activities in software projects.

Context are the CVS/SVN repositories of seven OSS: JFreeChart, Jitsi, PHP, Rhino, Tomcat, XalanJ, and XercesJ.

Introduction

Context

Problematic

Thesis

Related work

System
meta-model for
license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of
License Statements
and SC

Study Definition

Study Setup

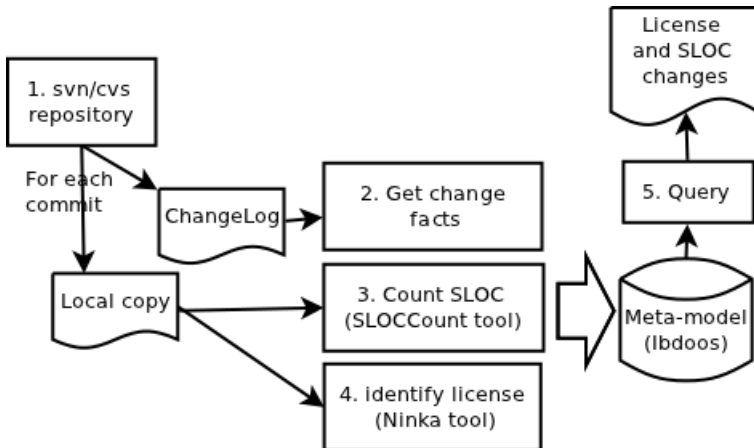
Analysis Methods

Results of the
study

Conclusion

Co-evolution of License Statements and SC

Study Setup



Co-evolution of License Statements and SC

Analysis Methods

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup

Analysis Methods

Results of the study

Conclusion

RQ1: Do licenses co-evolve with source code at the system level?

1. We compute the number of license statement changes performed in different periods of time discretised on a 15-day basis by addition of the number of change in each interval.
2. We compute the difference in SLOC between successive versions in each object system discretised on a 15-day basis by addition of the number of change in each interval.

Co-evolution of License Statements and SC Analysis Methods

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup

Analysis Methods

Results of the study

Conclusion

?

remove
blank

Co-evolution of License Statements and SC

Analysis Methods

RQ1: Do licenses co-evolve with source code at the system level?

On this data, we perform~~X~~:

1. Quantitative study: We compute the cross-correlation between two time series, i.e., the time series describing the number of all license statement changes and the time series describing the evolution of SLOC for all the files in a system.
2. Qualitative study: We plot the three time series, i.e., (1) the number of license statement changes performed in different periods excluding the initial addition of a license, (2) including all license changes, and (3) the number of added/removed lines of code. We analyse these curves to assess whether there is a relation between license changes and the evolution of SLOC.

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup

Analysis Methods

Results of the study

Conclusion

Co-evolution of License Statements and SC

Analysis Methods

Introduction

Context

Problematic

Thesis

Related work

System

meta-model for
license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition


Study Setup

Analysis Methods

Results of the study

Conclusion

RQ2: What types of license changes are performed?

1. We analyze  Ninka's output to distinguish different types of changes.
2. Using a histogram, we get information about how changes are distributed different types of changes.
3. We compute the cross-correlation for each type of license statement change between two time series, i.e, the number of license statement changes discretised on a 15-days basis and the evolution of SLOC.

⇒ The cross-correlation results of RQ2 are more refined than the ones of RQ1, because we are considering each type of license statement changes separately instead of aggregating all types of changes together.

Co-evolution of License Statements and SC

Analysis Methods

RQ3: Who performs license changes?

1. We compute the number of commits performed by each developer.
2. We identify the top seven committers that changed license statements.
3. We ranked the committers using their total number of performed SLOC changes to measure their activities.

⇒ The cross-correlation results of RQ2 are more refined than the ones of RQ1, because we are considering each type of license statement changes separately instead of aggregating all types of changes together.

already said ⇒
not the matching
conclusion ?

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup

Analysis Methods

Results of the study

Conclusion

Results of the study

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

Study Setup

Analysis Methods

Results of the study

Conclusion

RQ1: Do licenses co-evolve with source code at the system level?

Quantitative Study?

1. We cannot observe systematic large-scale license changes accompanying large restructurings of the system, for example PHP cross-correlation values vary between ~~-\$5%~~ and ~~+\$5%~~.
2. Except some case like Tomcat where cross-correlation reaches 80%.

Results of the study

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

RQ1: Do licenses co-evolve with source code at the system level?

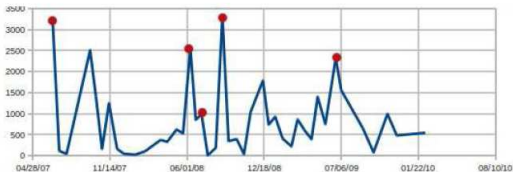
Qualitative ~~Study~~: ~~Because~~ the cross-correlations value are different from zero and reach up to 80% in some cases. We performed our qualitative study on three systems out of the seven analysed systems, i.e., JFreeChart, PHP, and XercesJ.

Results of the study

Evolution of SLOC and license statement changes over time in JFreeChart:



(b) License changes including the introduction of licenses to newly created files.



(f) SLOC evolution.

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

Results of the study

Evolution of the SLOC and license changes over time in PHP:

Introduction

Context

Problematic

Thesis

Related work

System

meta-model for
license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

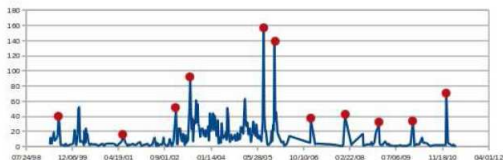
Study Definition

Study Setup

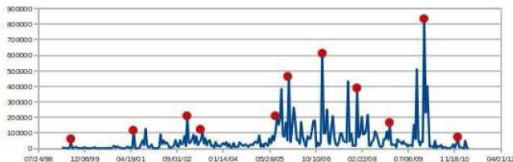
Analysis Methods

Results of the study

Conclusion



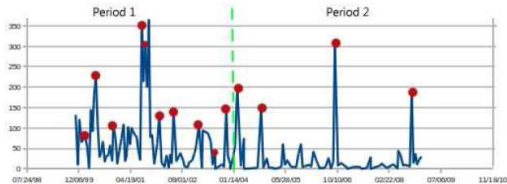
(b) Evolution of the number of license changes including the introduction of license statement to newly created files.



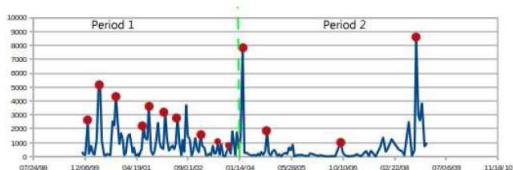
(c) SLOC evolution.

Results of the Study

Evolution of the SLOC and license statement changes over time in XercesJ:



(b) Evolution of the number of license changes including the introduction of license statement to newly created files.



(c) SLOC evolution.

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

RQ2: What types of license changes are performed?

The qualitative study of RQ1 allowed us to identify the most popular types of license statement changes:

1. Addition of contributors.
2. Updating the version of the license.
3. Change of the license type.
4. Miscellaneous changes.

Results of the study

Introduction

Context

Problematic

Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model

Concret Part of the
Meta-Model

Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition

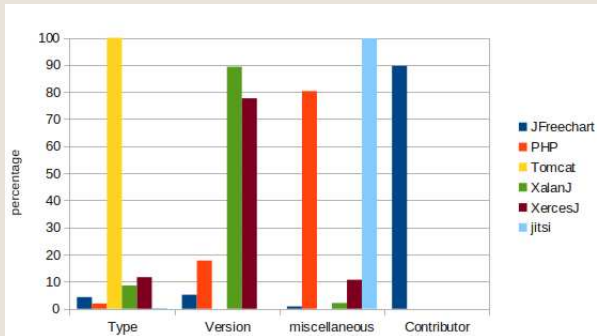
Study Setup

Analysis Methods

Results of the study

Conclusion

RQ2: What types of license changes are performed?



Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

RQ2: What types of license changes are performed?

We find that license type and version changes co-occur more often with SLOC changes than other license change types do. The popularity of these change types is not uniform across all projects.

Ferdaous
Boughanmi
Supervisor :
Giuliano Antoniol
and Yann-Gaël
Guéhéneuc

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

?

Remove
blank

Results of the study

RQ3: Who performs license changes?

XercesJ	
ID	# of license statement changes
mrglavas	1536 (49%)
lehors	275 (9%)
elena	247 (8%)
no author	188 (6%)
andyc	178 (6%)
sandygao	178 (6%)
arkin	110 (4%)
Total top 7	2,712
Total license statement changes	3,116
% license statement changes top 7	87%

XercesJ	
ID	# of changes
mrglavas	4070 (29.62%)
elena	2253 (16.39%)
no author	1841 (13.40%)
lehors	1583 (11.52%)
neilg	1234 (8.98%)
jeffreyr	503 (3.66%)
andyc	425 (3.09%)
Total top 7	11609
% changes top 7	86.68%

License statement changes are limited to a minority of committers: most active committers, and they are also the project members with a leading role.

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC



Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

Conclusion

Context

License changes could have negative impacts.   License evolution is worth studying to help in automatic license change tracking.

Thesis

License statements are changing frequently, but do not necessarily coevolve with source code and managed by a minority of developers that are probably experts.

Approach

- Step1 System Meta-model for License Analysis
- Step2 Co-evolution of License Statements and Source Code

Conclusion

next slide
and put the
3 Rqs on 1
slide

[step1] System Meta-model for License Analysis

Our meta-model is general meta-model that we used in our study for license evolution and also could be used in other studies related to licenses. Our meta-model could be extended to be more fine-grained if there is need.

[step2] RQ1: Do licenses co-evolve with source code at the system level?

We find that license statements are changing frequently and continuously, but not necessarily together with source code. License statement changes occur either when a substantial contribution is made (addition of contributors) or whenever the legal team advises so (update of license version or type).

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

Conclusion

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

[step2] RQ2: What types of license changes are performed?

Different kinds of license statement changes can evolve differently. We identified three main types of license changes: license type change, license version change, and contributor change. We find that license type and version changes co-occur more often with SLOC changes than other license change types do.

Conclusion

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

[step2] RQ3: Who performs license changes?

License statement changes are limited to a minority of specialised committers, We observe that the most active committers (in the CVS or SVN repository) performing license statement changes are also the project members with a leading role.

Questions?

Introduction

Context
Problematic
Thesis

Related work

System meta-model for license analysis

Abstract Part of the
Meta-Model
Concret Part of the
Meta-Model
Architecture Part of the
Meta-Model

Co-evolution of License Statements and SC

Study Definition
Study Setup
Analysis Methods

Results of the study

Conclusion

Questions?

stay on previous
slide