Compatibility of Terms in Open Source Software Licenses

Elena Petrou
University of Cyprus
P.O. Box 20537
Cyprus
epetro10@cs.ucy.ac.cy

Georgia M. Kapitsaki
University of Cyprus
P.O. Box 20537
Cyprus
gkapi@cs.ucy.ac.cy

ABSTRACT

Nowadays, we see more and more systems to be composed using information from various open source software systems that may carry different open source licenses. As a result compatibility issues arise, as conflicts need to be avoided. Also the variety of open source licenses is making dealing with such incompatibilities difficult for organizations. In this extended abstract, we describe the vision for a system that can help in deducting conclusions for software licenses based on the terms the licenses contain.

KEYWORDS

Open source software, licenses

1 INTRODUCTION

An open source software (OSS) license contains the permission and restrictions to reproduce, modify and redistribute a software [1]. An open source license is a software license that follows Open Source Definition and may be approved by organizations, such as the Open Source Initiative (OSI). Developers of big systems usually use open source software in many parts of their system. But not all the source code files used are under the same license. Developers must understand those software's licenses and check whether those licenses have a conflict with the license under which the developed software is distributed

2 VISION FOR TERM COMPARITIBLITY

To check the compatibility between the licenses and to find the possible inconsistencies we have used an initial modeling of the OSS licenses from a previous work [2]. We have adapted the model by studying the texts of several licenses.

License Terms refer to the conditions of use specified in the license text in natural language. Based on the terms we can express the relationship between the terms to declare, for instance, that two terms are contradictory (e.g., *MaySublicense* and *NotSublicense*) or declaring the implication of one definition to another. We can then use these relationships to check compatibilities between licenses. After studying the licenses' texts and the terms that have been produced based on the aforementioned model, we can export important relationships between them. These relationships can be declared in deontic logic and can be subsequently used in tools, such as the Spindle reasoner system [3], where the compatibility check will be done.

An example for the terms' relationships in deontic logic are the following:

- MaySublicense->-NotSublicense: The existence of MaySublicense implies the non-existence of the term NotSublicense
- MustMarkModifications->MayMakeModification: The existence of MustMarkModifications implies the existence of the term MayMakeModifications.

An initial set of rules has been defined. We have used these rules in Spindle in order to proceed with the compatibility verification process. Spindle is a defeasible logic reasoner capable of inferencing defeasible theories with hundredth of thousands rules. The result can be used in order to make sure that if a developer is using specific licenses in her organization, the licenses she may choose to adopt in a new system will not cause any violations. This will be feasible if our approach is integrated in a relevant recommender system [ref]. An example of these relations (between licenses to be used and licenses already used) is depicted in a graph in Fig. 1.

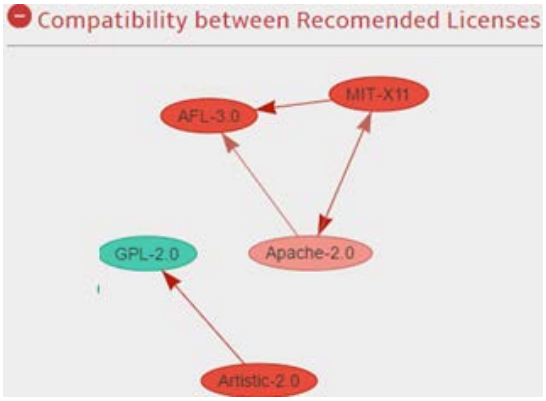

Figure 1: Compatibilities between licenses based on term rules.

REFERENCES

[1] Fitzgerald, B. "The transformation of open source software." Mis Quarterly (2006): 587-598.
[2] Kapitsaki, G. M., and Charalambous G. 2016. "Find your Open Source License Now!." In Software Engineering Conference (APSEC), 2016 23rd Asia-Pacific, pp. 1-8. IEEE, 2016.
[3] Lam, H.-P., and Governatori, G. "The making of SPINdle." International Workshop on Rules and Rule Markup Languages for the Semantic Web. Springer Berlin Heidelberg, 2009.