# Automatic License Compatibility Checking

Giray Havur[1,2], Simon Steyskal[1,2], Oleksandra Panasiuk[3], Anna Fensel[3], Victor Mireles[4], Tassilo Pellegrini[5], Thomas Thurner[4], Axel Polleres[1], and Sabrina Kirrane[1]

[1] Vienna University of Economics and Business, Austria
[2] Siemens AG Österreich, Austria
[3] STI Innsbruck, University of Innsbruck, Austria
[4] The Semantic Web Company, Austria
[5] St. Pölten University of Applied Sciences, Austria

**Abstract.** In this paper, we introduce the Data Licenses Clearance Center system, which not only provides a library of machine readable licenses but also allows users to compose their own license. A demonstrator can be found at https://www.dalicc.net.

## 1 Introduction

Licensing in general and rights clearance in particular are complex topics that require a high level of domain expertise and legal expertise. Primary challenges include the high transaction costs associated with the manual clearance of licensing terms and conditions; the need for sufficient expertise to detect compatibility conflicts between licenses; and the ability to resolve such conflicts. An alternative approach could be to model licenses in a manner that supports automatic license compatibility checking. Among the most prominent Rights Expression Language (REL) vocabularies used to represent licenses are the Creative Commons Rights Expression Language (ccREL)[6], the Open Digital Rights Language (ODRL)[7], and an ODRL profile called RightsML[8]. When it comes to reasoning over license representations, an early proposal for a generic logic for reasoning is provided by Pucella and Weissman [7], but it has not been implemented with existing RELs like ODRL or MPEG-21 nor has it been evaluated in practice. García and Gil [2] propose an ontology to describe copyright issues in closed datasets for rights clearance purposes. Hosking et al.[6] present a rule-based engine, built on top of the Carneades Framework [3], to reason over various sets of licenses, while additionally suggesting potential licenses by which to safely share derived outputs. Instead of applying deductive reasoning they used a non-monotonic formalism suitable for modeling situations in which contradictory statements are being processed. Villata and Gandon [8] and Governatori et al. [4] describe the formalization of a license composition tool for derivative works. They extend their research by introducing semantics based on a deontic logic [5] for the comparison

[6] https://www.w3.org/Submission/ccREL/
[7] https://www.w3.org/TR/odrl-model/
[8] https://iptc.org/standards/rightsml/

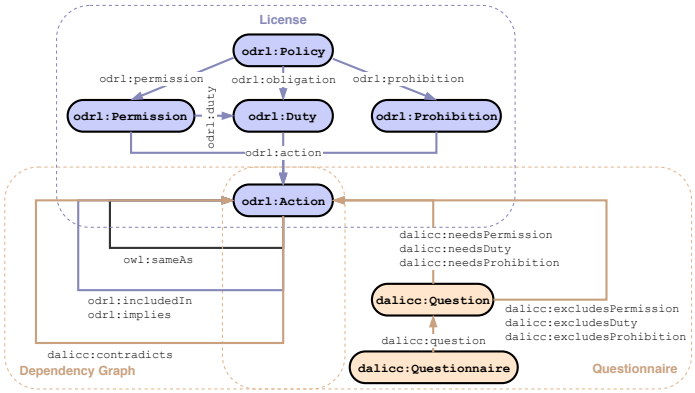


**Fig. 1.** Interaction between the constituent parts of the framework

of the permissions, prohibitions and duties stated in a given license. The limitation of existing work is the fact that compatibility can just be checked against a handful of selected permissions, obligations and prohibitions and not against a selection of licenses. In this paper, we present the Data Licenses Clearance Center (DALICC) system[9], which focuses on extending existing vocabularies to enable modeling and reasoning over several well-known license texts. With this demo we aim to demonstrate: (i) our machine ODRL representation for a number of well-known license families (CC, Apache, BSD, MIT, GPL); and (ii) the DALICC system that can be used to both generated custom licenses and check automatically check license compatibility.

## 2 Modelling Licenses using ODRL

In DALICC licenses are modelled using ODRL, which was recently released as a W3C recommendation. The model is further extended with a dependency graph, which is necessary for checking license consistency, and a model that underpins a dynamic questionnaire that enables users of the DALICC system to search for licenses. Figure 1 depicts the central role of `odrl:Action` in integrating the licenses, dependency graph and questionnaire. For the modelling, we selected 14 commonly used licenses from 5 license families (CC, Apache, MIT, BSD, GPL), which can be applied to various data assets, such as creative works, software and datasets. The ODRL information model is particularly suitable for modeling licenses in the form of *policies* that express *permissions*, *prohibitions* and *duties* related to the usage of *assets*. ODRL also defines a vocabulary of general terms (e.g., `odrl:reproduce`, `odrl:distribute`, `odrl:modify`) that can be further extended with terms from other vocabularies such as CC REL[10]. However, during our analysis we identified the need for additional terms (e.g., `dalicc:perpetual` as a validity type, `dalicc:worldwide` as a

[9] https://www.dalicc.net/
[10] https://creativecommons.org/ns#

jurisdictional property, `dalicc:chargeLicenseFee` as permission and prohibition actions, and `dalicc:modificationNotice` as a duty action), which we modelled using the DALICC vocabulary. DALICC utilizes a dependency graph for representing the semantic relationship between defined actions. The core function of the dependency graph is to encode expert knowledge about the implicit and explicit semantic dependencies between actions. The corresponding dependency graph represents the semantics of an action in another action (e.g., `odrl:sell odrl:includedIn odrl:commercialize`), implications derived from a specific action (e.g., `cc:Attribution odrl:implies cc:Notice`), equalities (e.g., `odrl:copy owl:sameAs odrl:reproduce`), and contradictions between specific actions (e.g., `cc:ShareAlike dalicc:contradicts dalicc:addStatement`). Additionally, the DALICC questionnaires are encoded using RDF, enabling multilingual interfaces and rapid refactoring using the RDF editing capabilities of PoolParty Semantic Suite. To this end, we have created four controlled vocabularies, one each for: (i) questions, (ii) question types, (iii) interaction between the UI and the License Search, and (iv) interaction between the UI and the Composer. Each question is an instance of `dalicc:Question` class, a subclass of `skos:Concept`, with three attributes that define their appearance and behaviour in the UI: `skos:prefLabel`, `skos:definition`, and `rdf:type`, all of which are adopted by the DALICC system.

## 3 Reasoning

To reason over licenses we use Answer Set Programming (ASP)[1], a declarative (logic-programming-style) paradigm for solving combinatorial search problems by defining and evaluating rule sets. Licenses are represented in ASP as a set of rules of the form `rule(L,C,I,`$\alpha$`,T)` where `L`, `C`, `I`, $\alpha$, and `T` correspond to *license name*, *category of rule*, *assignee*, *action*, and *asset*, respectively.

Policies are derived from the RDF graphs of the licenses. Herein, a rule that permits or prohibits the execution of an action on certain assets does not only affect other rules that govern the execution of the same action on the same asset(s) but also those permitting or prohibiting related actions on the same asset(s). In this sense, CLINGO is an alternative to extensive materialization, which in this case is essential for search, and also enables listing sets of compatible statements. This is necessary for effective computation of conflicts between licences, in particular for identifying the conflicting and non-conflicting parts of a license.

## 4 The DALICC System

The DALICC framework consists of the three main functional components, namely: *license library*, *license search*, and *license composer*, as shown in Figure 2. The DALICC system, which is an implementation of the DALICC framework is the result of coupling a Virtuoso[11] triplestore, a Drupal[12] based web application, the PoolParty Semantic Suite[13], and a Clingo Answer Set Programming

[11] https://virtuoso.openlinksw.com/
[12] https://www.drupal.org
[13] https://www.poolparty.biz/

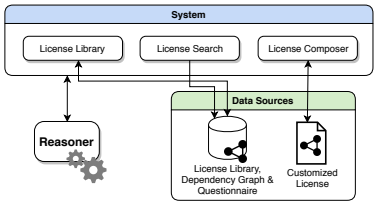


**Fig. 2.** The DALICC Framework

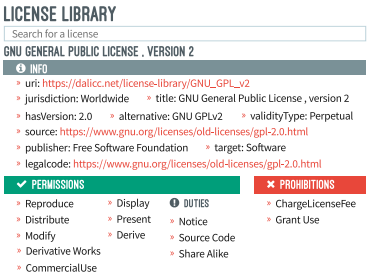


**Fig. 3.** The License library UI

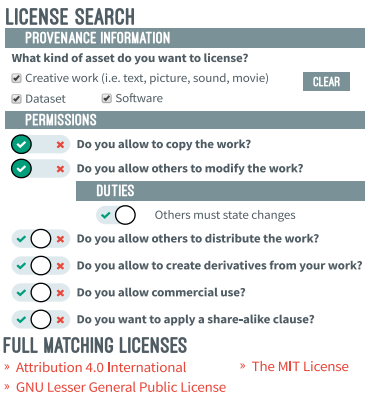


**Fig. 4.** The License search UI

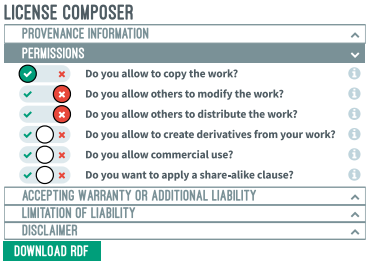


**Fig. 5.** The License composer UI

(ASP) reasoner[14]. This web application provides the user with three different workflows that cover the functionality of the framework: (i) displaying the licenses in the license library; (ii) searching for a license that meets the user's needs; and (iii) composing customized licenses from scratch. The license library is a repository that contains machine-readable and human-readable representations of the licenses. Licenses properties are queried using SPARQL and presented to the user in an easily digestible manner, as seen in Figure 3. In the case of license search (cf. Figure 4), the user fills in a dynamic questionnaire which is used to find the most suitable license based on their individual needs via communicating with the reasoner. When the form is submitted, the underlying JavaScript triggers a SPARQL query that retrieves the actions of type `odrl:action` and other relations with respect to the answer. Afterwards, this information is sent to the reasoner so that the reasoner returns the licenses that are consistent with the given input. The license composer (cf. Figure 5) is a tool that allows customized licenses to be easily created from a set of questions which are mapped to ODRL, ccREL and DALICC vocabularies. In order to ensure the validity of the machine readable licenses and the corresponding license compatibility assessment, both

[14] https://potassco.org/clingo/

the development and the testing of the platform's components have been carried out in close collaboration with legal experts within the DALICC consortium.

## 5 Conclusion

In this paper, we discussed how licenses can be modeled using various Rights Expression Languages and how reasoning can be applied to detect licensing conflicts. The DALICC demonstrator is a viable proof of concept and illustrates the practical applicability of semantic technologies for legal purposes. We are planning to mature the system and extend its functional scope from license management towards policy management.

## References


1. G. Brewka, T. Eiter, and M. Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12), 2011.
2. R. García and R. Gil. Copyright licenses reasoning using an owl-dl ontology. *Law, Ontologies and the Semantic Web: Channelling the Legal Information Flood*, 188, 2009.
3. T. F. Gordon, H. Prakken, and D. Walton. The Carneades model of argument and burden of proof. *Artificial Intelligence*, 171(10-15):875–896, July 2007. ISSN 00043702. doi: 10.1016/j.artint.2007.04.010. URL `http://linkinghub.elsevier.com/retrieve/pii/S0004370207000677`.
4. G. Governatori, H.-P. Lam, A. Rotolo, S. Villata, G. A. Atemezing, and F. L. Gandon. Live: a tool for checking licenses compatibility between vocabularies and data. In *International Semantic Web Conference*, 2014.
5. G. Guido, L. Ho-Pun, R. Antonino, V. Serena, and G. Fabien. Heuristics for Licenses Composition. *Frontiers in Artificial Intelligence and Applications*, 2013.
6. R. Hosking, M. Gahegan, and G. Dobbie. An escience tool for understanding copyright in data driven sciences. In *10th IEEE International Conference on e-Science, eScience 2014, Sao Paulo, Brazil, October 20-24, 2014*, pages 145–152, 2014. doi: 10.1109/eScience.2014.37. URL `https://doi.org/10.1109/eScience.2014.37`.
7. R. Pucella and V. Weissman. A Logic for Reasoning About Digital Rights. In *Proceedings of the 15th IEEE Workshop on Computer Security Foundations*, CSFW '02, pages 282–294, Washington, DC, USA, 2002. IEEE Computer Society. ISBN 0-7695-1689-0. URL `http://dl.acm.org/citation.cfm?id=794201.795182`.
8. S. Villata and F. Gandon. Licenses compatibility and composition in the web of data. In *Third International Workshop on Consuming Linked Data (COLD2012)*, 2012.