The Journal of Open Source Software

OnToCode: Template-based code-generation from ontologies

Philipp Matthias Schäfer[1]

1 DLR Institute of Data Science, Mälzerstraße 3, 07745 Jena, Germany

DOI: 10.21105/joss.01513

Software

- Review
- Repository
- Archive

Submitted: 08 May 2019
Published: 10 August 2019

Summary

In support of research into software-based methods for spacecraft development and manufacturing, ontologies have been (e.g. (Hennig, Viehl, & Benedikt Kämpgen, 2016)) and are being developed. These ontologies serve as domain models in model-driven engineering processes for the development of software tool prototypes.

Since modern software applications usually have components written in multiple programming languages (e.g. a front end in JavaScript and a back end in Python) with derivatives of the domain model (platform-specific models as well as additional code) in each component, it is attractive to generate those derivatives from a single (platform-independent) domain model.

Internet based research and a review of the relevant literature (Strmečki, Magdalenić, & Radosević, 2018) (Syriani, Luhunu, & Sahraoui, 2018) showed that there was no tool that helps generating code in arbitrary languages from ontologies. OnToCode satisfies this need.

OnToCode is a Python package that allows a developer to generate code given a set of ontologies and templates. Based on Owlready2 (Lamy, 2017), OnToCode provides a fixed API for loading and querying ontologies. In addition, it offers an extensible API for processing query results (result processors) and for template instantiation (templates). OnToCode comes with a set of result processors for common tasks as well as support for Jinja2 based templates.

Since OnToCode is a Python package, it can be easily integrated into the build process of Python projects. Integration into build systems outside the Python ecosystem are not part of the package. The source code for OnToCode has been archived on Zenodo with the linked DOI: (Schäfer, 2019)

Currently, spacecraft systems engineers have to rely on data collected from human readable data sheets and individual correspondence with the manufacturer. As part of a research project on the merits of automatic transfer of spacecraft part data ((Schäfer et al., 2018), (Peters, Fischer, Schäfer, Opasjumruskit, & Gerndt, 2019)) from suppliers to integrators, we develop an ontology ((Opasjumruskit & Schäfer, 2019)) that formally captures all spacecraft part concepts relevant during the design phase of a spacecraft. To use this formal specification as a basis for all models in the implementation of our software prototypes that form the basis of systems engineering support method's evaluation, we developed OnToCode that generates code in multiple languages from a single main model, our ontology.

Acknowledgements

I acknowledge contributions in the form of code and documentation review from Kobkaew Opasjumruskit, Diana Peters, and Laura Thiele.

References

Hennig, C., Viehl, A., & Benedikt Kämpgen, H. E. (2016). Ontology-based design of space systems. In *The Semantic Web – ISWC 2016* (pp. 308–324). doi:10.1007/978-3-319-46547-0_29

Lamy, J.-B. (2017). Owlready: Ontology-oriented programming in python with automatic classifcation and high level constructs for biomedical ontologies. *Artificial Intelligence in Medicine*, *80*, 11–28. doi:10.1016/j.artmed.2017.07.002

Opasjumruskit, K., & Schäfer, P. M. (2019, March). Spacecraft parts ontology. Zenodo. doi:10.5281/zenodo.2616374

Peters, D., Fischer, P. M., Schäfer, P. M., Opasjumruskit, K., & Gerndt, A. (2019). Digital availability of supplier information for collaborative engineering of spacecraft. In (p. to appear).

Schäfer, P. M. (2019, February). OnToCode: Template-based code-generation from ontologies. Zenodo. doi:10.5281/zenodo.2652065

Schäfer, P. M., Fischer, P. M., Brehm, N., Erfurht, C., Gerndt, A., Opasjumruskit, K., & Peters, D. (2018). Toward a digital platform for spacecraft manufacturing. In *8th International workshop on Systems & Concurrent Engineering for Space Applications Conference (SECESA)*.

Strmečki, D., Magdalenić, I., & Radosević, D. (2018). A systematic literature review on the application of ontologies in automatic programming. *International Journal of Software Engineering and Knowledge Engineering*, *28*(5), 11–28. doi:10.1142/S0218194018300014

Syriani, E., Luhunu, L., & Sahraoui, H. (2018). Systematic mapping study of template-based code generation. *Computer Languages, Systems & Structures*, *52*, 43–62. doi:10.1016/j.cl.2017.11.003