SCOOP-UI: SHACL Shape Extraction in Just a Click!

Xuemin Duan[1], David Chaves-Fraga[2,1], and Anastasia Dimou[1]

[1] KU Leuven – Flanders Make@KULeuven – Leuven.AI, Belgium
{xuemin.duan,anastasia.dimou}@kuleuven.be
[2] Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, Spain
david.chaves@usc.es

Abstract. The proliferation of knowledge graph validation using the Shapes Constraint Language (SHACL) has catalyzed significant efforts toward automating the extraction of SHACL shapes. These shapes may be derived from RDF graphs, or the various components which are involved in their creation, e.g., ontologies, raw data schemas, and mapping rules. In SCOOP, we integrate shapes extracted from these components, however, no system exists that enables the users to streamline the extraction and integration processes. In this work, we present SCOOP-UI, a web application built on top of SCOOP to provide an editor for the users to handle the different component's resources to extract and integrate their SHACL shapes. This work enables users to get directly involved in the translation of various resources into SHACL shapes, as well as in the integration of these shapes to produce a unified representation.
License: Apache-2.0
Demo URL: https://demos.citius.usc.es/scoop/
Source Code: https://github.com/dtai-kg/SCOOP-UI
Video: https://demos.citius.usc.es/scoop/#doc

Keywords: SHACL · Shape integration · Web application

1 Introduction

Automatic extraction of shapes e.g., in the Shape Constraint Language[1] (SHACL), to validate RDF graphs is crucial to guarantee consistency in data transformation. Extracting SHACL shapes leverage either the RDF graph directly [5] or associated sources such as ontologies [1], mapping rules [2], and raw data schemas [4] utilized in constructing the RDF graph. The former approach entails constraint extraction via direct analysis of the RDF graph, whereas the latter involves the translation of pre-existing constraints into SHACL constraints.

While the majority of current research concentrates on extracting shapes from an individual source [3], e.g., from RDF graphs, ontologies, mapping rules, and raw data schemas, no attention has been given to the potential coexistence of

[1] https://www.w3.org/TR/shacl/

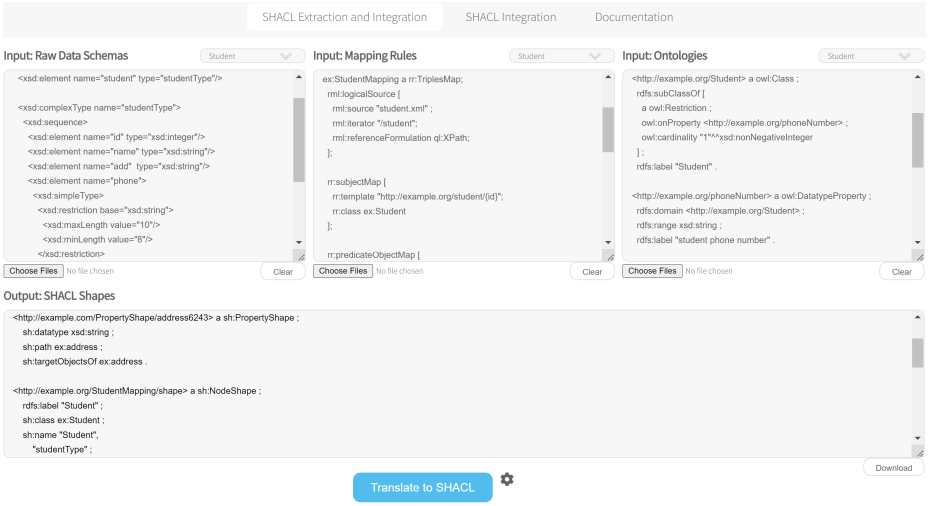

Fig. 1: SHACL shape extraction and integration via SCOOP-UI

diverse sources related to RDF graph creation. SCOOP[2] [3], a shape integration framework, addressed this research gap. By leveraging SCOOP, SHACL shapes extracted from the data schema and ontology used to create an RDF graph according to a set of mapping rules, can be integrated into a unified SHACL shape, facilitating comprehensive validation of RDF graphs.

This work presents the SCOOP-UI[3], an open-source web-based application built on top of SCOOP[4], aiming to facilitate the automatic extraction and integration of SHACL shapes. SCOOP-UI supports (i) the extraction of a shapes graph from a single file, such as XML Schema (XSD) via its XSD2SHACL component, RDF Mapping Language (RML) via its RML2SHACL component, and Web Ontology Language (OWL) via its OWL2SHACL component; (ii) the extraction and integration of a unified shapes graph from multiple files (integrate either shapes extracted or existing). Last, it offers various predefined integration strategies; and also allows user-defined integration source priority.

2 SCOOP-UI Features

This section presents the functions and configurations supported by SCOOP-UI.

Core functions SCOOP-UI may have a single file or multiple files as the input which can be raw data schemas, ontologies, mapping rules, or SHACL shapes.

Single file input. When a single file is given as input, e.g., a raw data schema, an ontology, or a set of mapping rules, the application automatically triggers the corresponding shapes extraction component to directly generate the

[2] Paper accepted by ESWC 2024 Resource Track
[3] `https://demos.citius.usc.es/scoop/`
[4] `https://github.com/dtai-kg/SCOOP`

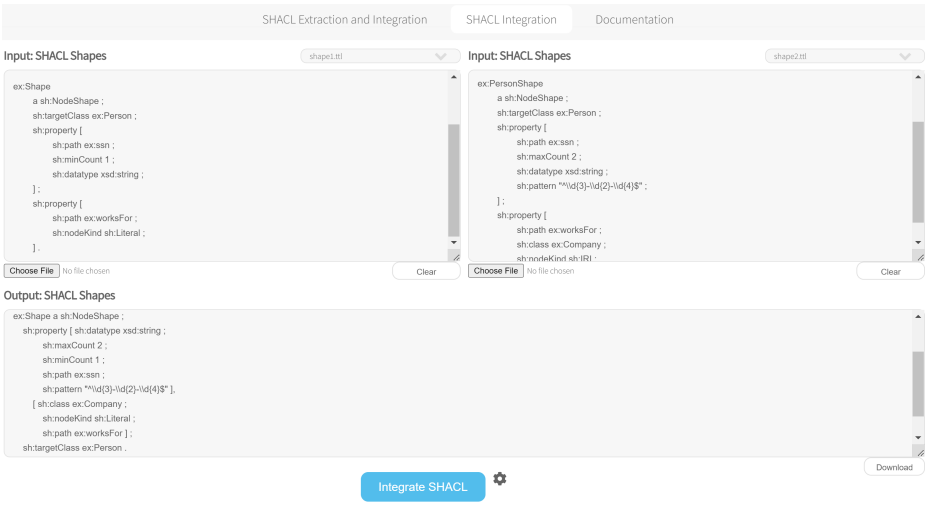

Fig. 2: SHACL shape integration via SCOOP-UI

corresponding SHACL shapes. As the layout illustrated in Figure 1, if only the Input: Mapping Rules box is filled while leaving other text boxes empty, a set of SHACL shapes will be extracted via the RML2SHACL component.

Multiple files inputs. When multiple files from a single source or multiple sources are given as inputs, the application invokes the corresponding shape extraction components to extract the shapes (e.g., the XSD2SHACL component when files from the XSD source are given or the RML2SHACL component if some mapping rules from the XSD source are given). Subsequently, the shape integration module in SCOOP integrates the extracted shapes from various files into a unified shapes graph. As depicted in Figure 1, upon giving multiple files through populating the text area or uploading files via the upload button within Raw Data Schemas, Mapping Rules, and Ontologies, the application automatically performs the extraction and integration, then presents a unified shapes graph in the SHACL shapes text area.

Multiple SHACL shapes inputs. While SCOOP is designed to accommodate various input sources to extract and then integrate shapes, this application isolates the integration module of SCOOP to enable the support for integrating multiple files from SHACL shapes. As illustrated in Figure 2, the application also caters to users who seek to integrate multiple existing SHACL shapes graphs without the need for extraction.

Configurations SCOOP-UI empowers users to select from three integration strategies: SCOOP-**All**, SCOOP-**Prior**, and SCOOP-**Prior-R**. The adoption of different strategies influences the resolution of inconsistent constraints. The **All** aims to integrate all constraints from extracted or existing shapes, utilizing logical constraint components (e.g., sh:or) to resolve potential inconsistencies. The **Prior** integrates constraints from lower-priority sources that are not inconsistent with constraints from higher-priority sources, based on user-defined priorities.

The **Prior-R** is designed to filter out redundant shapes from the ontology source relative to higher-priority sources. If the SCOOP-Prior or SCOOP-Prior-R strategy is chosen, users have the flexibility to customize the priority of sources.

3 Architecture and Demonstration

SCOOP-UI[3] is comprised of a front-end user interface, back-end service, and API, facilitating an end-to-end workflow. The front-end interface is developed using HTML, CSS, and JavaScript. It encompasses text areas and upload buttons for inputs, clear buttons to clear given inputs, configuration options for shape integration, an output text area for displaying results, and a download button to download the generated shapes graph. The back-end service is constructed using the FastAPI framework[5], which receives requests from the front-end interface and invokes the corresponding SCOOP modules. SCOOP currently incorporates XSD2SHACL [4] for handling XSD files, RML2SHACL [2] for RML files, and Astrea [1] for OWL files.

The SCOOP-UI offers test examples about students and real-world use cases from railway infrastructure register[6] (RINF) for each input source, conveniently accessible through direct clicks, as depicted in Figures 1 and 2. In the demo, we demonstrate the extraction of SHACL shapes from the single file of RML, XSD, or OWL, and illustrate the impact of different configurations on the extracted constraints when multiple files are given. We showcase the direct integration process when multiple SHACL shapes are provided as inputs.

4 Conclusion

Through SCOOP-UI, users can seamlessly extract and integrate a unified shapes graph directly from various sources without the need to familiarize themselves with multiple shape extraction tools. Presently, our application supports input from OWL, RML, XSD, and existing SHACL shapes. Its versatility allows for potential expansion to accommodate additional sources in the future, contingent upon the emergence of new shape extraction tools targeting other sources.

Acknowledgments

Xuemin Duan and Anastasia Dimou are partially supported by Flanders Make, the research centre for the manufacturing industry, and the Flanders innovation and entrepreneurship (VLAIO) through the KG3D project. David Chaves-Fraga is funded by the Galician Ministry of Education, University and Professional Training and the European Regional Development Fund (ERDF/FEDER program) through grants ED431C2018/29 and ED431G2019/04.

[5] `https://fastapi.tiangolo.com/`
[6] `https://www.rinf-ch.ch/documentation?lang=EN`

References

1. Cimmino, A., Fernández-Izquierdo, A., García-Castro, R.: Astrea: Automatic Generation of SHACL Shapes from Ontologies. In: ESWC. Springer (2020). https://doi.org/10.1007/978-3-030-49461-2_29
2. Delva, T., Smedt, B.D., Oo, S.M., Assche, D.V., Lieber, S., Dimou, A.: RML2shacl: RDF generation taking shape. In: Proceedings of the 11th on Knowledge Capture Conference. ACM (2021). https://doi.org/10.1145/3460210.3493562
3. Duan, X., Chaves-Fraga, D., Derom, O., Dimou, A.: SCOOP all the Constraints' Flavours for your Knowledge Graph. In: Proceedings of the 21th Extended Semantic Web Conference (ESWC) (2024)
4. Duan, X., Chaves-Fraga, D., Dimou, A.: XSD2SHACL: Capturing RDF Constraints from XML Schema. In: Proceedings of the 12th Knowledge Capture Conference 2023. K-CAP '23, ACM (2023). https://doi.org/10.1145/3587259.3627565
5. Rabbani, K., Lissandrini, M., Hose, K.: SHACTOR: Improving the Quality of Large-Scale Knowledge Graphs with Validating Shapes. In: SIGMOD-Companion '23. ACM (2023). https://doi.org/10.1145/3555041.3589723