Ademar Crotti Junior, Christophe Debruyne, and Declan O'Sullivan. Juma: An editor that uses a block metaphor to facilitate the creation and editing of R2RML mappings. In ESWC (Satellite Events), volume 10577 of Lecture Notes in Computer Science, pages 87–92. Springer, 2017

Juma: an Editor that Uses a Block Metaphor to Facilitate the Creation and Editing of R2RML Mappings

Ademar Crotti Junior, Christophe Debruyne, Declan O'Sullivan

ADAPT Centre, Trinity College Dublin, Dublin 2, Ireland
{crottija,debruync,declan.osullivan}@scss.tcd.ie

Abstract. R2RML is the W3C standard mapping language used to define customized mappings from relational databases into RDF. One issue that hampers its adoption is the effort needed in the creation of such mappings, as they are stored as RDF documents. To address this problem, several tools that represent mappings as graphs have been proposed in the literature. In this paper, we describe a visual representation based on a block metaphor for creating and editing such mappings that is fully compliant with the R2RML specification. Preliminary findings from users using the tool indicate that the visual representation was helpful in the creation of R2RML mappings with good usability results. In future work, we intend to conduct more experiments focusing on different types of users and to abstract the visual representation from the R2RML mapping language so that it supports the serialization of other uplift mapping languages.

Keywords. R2RML; Visual Representation; Data Mapping.

1 Introduction

A significant part of the Linked Data web is achieved by converting non-RDF resources into RDF. This conversion process is typically called *uplift*. For relational databases, one can rely on the W3C Recommendation R2RML[1] for creating mappings from relational databases into RDF datasets. Though useful, some problems with its adoption can be observed. Firstly, R2RML mappings are stored as RDF. We argue that writing any RDF graph by hand can be troublesome and prone to errors. Secondly, the R2RML mapping language has a steep learning curve, where the creation of mappings is time consuming, and syntactically heavy in various cases [9]. Initiatives have emerged to address these problems and make the technology more accessible such as visual *graph* representations for engaging with mappings [6, 7] and others that will be discussed in Section 2.

In this paper, we describe a visual representation for mappings called Juma. Our representation is based on the block (or jigsaw) metaphor that has become popular with visual programming languages – where it is called the block paradigm – such as Scratch[2]. This metaphor allows one to focus on the logic instead of the language's

[1] `https://www.w3.org/TR/r2rml/`
[2] `https://scratch.mit.edu/`, last accessed March 2017

syntax and it has been successfully used in introducing programming to non-experts. We have applied this representation for the R2RML mapping language, being fully compliant with its specification. The main contributions of this research to date are: 1) a visual representation for mappings based on a block metaphor; 2) a tool that uses this representation for creating and managing R2RML mappings and 3) an initial user evaluation of the approach.

The remainder of this paper is structured as follows: Section 2 reviews the related work. Section 3 presents our visual representation. Section 4 presents an initial user evaluation and Section 5 concludes the paper.

2 Related Work

In this section, we discuss the different mapping tools developed and the mapping representations used for R2RML mappings.

Karma [6] is a web-based application where the data is loaded before it can be mapped into RDF. The ontologies used during the mapping process are represented in a tree structure and the data as a table. A graph visualization of the mapping is available while the user creates them. The creation of mappings using Karma can be troublesome because of the data centric approach, where every input is shown in a different table. This makes the interlinking between tables unnecessarily complex. Ontop-Pro[3] [11] is a Protégé plugin that uses a proprietary mapping language internally to create mappings. Lembo et al. [7] also uses a graph representation for R2RML mappings. The tool is fully compliant with R2RML's specification and offers support for syntactic and semantic checking of mappings. Even though the tool offers a graph visualization of mappings, its creation and/or editing is done through text, which may make the mapping process prone to errors. RMLeditor [5] has support for R2RML and RML [4] mapping languages. The RMLeditor also uses a graph representation for the mapping. The input data and RDF output are shown as tables. MapOn [10] is yet another graph representation tool for R2RML mappings. It also provides a graph visualization for ontologies and databases. SQuaRE [1] is a tool that provides a visual environment for the creation of R2RML mappings. This tool also uses a graph visual representation for mappings, similar to RMLeditor and MapOn. Ontologies and RDF vocabularies that will be used in the mapping process are shown as a tree.

These tools offer different abstractions for the creation of R2RML mappings. In [2], the authors proposed the use of the block metaphor for SPARQL. To the best of our knowledge, no other tool has yet used the block metaphor for representing R2RML mappings. In section 3, we discuss how we use the metaphor in our design.

3 Juma: A Visual Representation for Mappings

In this section, we introduce a method called **J**igsaw p**u**zzles for representing **ma**ppings, Juma, applied to the R2RML mapping language. As outlined in Section 1, there are a number of issues with how R2RML mappings are created. For example, creating mappings with a text editor – even with support for RDF – is a time consuming process, being syntactically heavy even for simple mappings in various cases. Moreover,

[3] `http://ontop.inf.unibz.it`

it has a steep learning curve if one is not acquainted with the R2RML vocabulary and RDF, amongst others. To facilitate the creation and maintenance of mappings, and to leverage the uptake of R2RML to a wider set of stakeholders, we have developed a tool[4] that applies the Juma method to the R2RML mapping language. For the development of the visual representation, we derived some requirements:

1. users should be able to create a mapping without being preoccupied with R2RML's vocabulary or with a particular RDF serialization format's syntax;
2. the visual representation should guide users in creating and editing valid R2RML mappings, according to the standard specification;
3. common patterns, which are completed by a user for a given context, should be available.

Our implementation (screenshot shown in Fig. 1) uses Google's Blockly API[5]. In our tool, each block has been designed to represent an R2RML statement that automatically generates a correspondent R2RML construct, as specified in requirement 1. For requirement 2, when a user drags a block near other blocks, if that is valid according to the R2RML specification, then the visual representation highlights this. For requirement 3, we have defined a menu option called **templates**. This option shows one complete triple map and one complete predicate object map. The other menu options provide one with all other possibilities within the R2RML mapping language. We have defined the menu using a tree structure, which gives users a hint of how the blocks connect to each other.

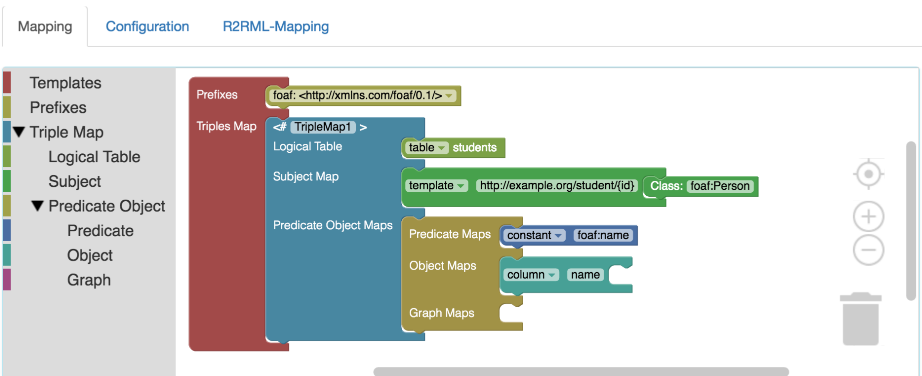

Fig. 1. Visual representation of an R2RML mapping

4 User study

We have initially evaluated two aspects of the tool: the accuracy of the mappings created and the usability of the tool using a standard usability test.

This initial user study was built on top of the Northwind[6] database for MySQL and

---

[4] `https://www.scss.tcd.ie/~crottija/juma/`
[5] `https://developers.google.com/blockly/`
[6] `https://github.com/dalers/mywind`

it involved 15 participants split into 3 groups of 5. The first group had no knowledge of Semantic Web technologies. The second group was familiar with Semantic Web technologies, such as RDF and OWL, but not R2RML. The last group was familiar with the R2RML mapping language. Participants were asked to create one R2RML mapping. The task involved the use of different R2RML constructs, such as *parent triples maps* and others, in order to explore the visual representation. A sample RDF output was shown to participants. In addition, they could run the mapping and compare the output from the tool to the sample provided. To enable this in the experiment, we integrated an R2RML processor [3] into the tool.

The accuracy of the mappings created was calculated by counting the number of correct triples in the respective RDF output. Additionally, any help needed during the execution of the task was recorded. The high accuracy of the mappings (95.5%) indicates that the visual representation was helpful to participants. In relation to the help needed, the most common need found was how to interlink triples maps with the use of the *parent triples map* construct. Participants were able to create the R2RML construct using the visual representation but they had difficulties defining the parent and child values for the join condition, which required knowledge on SQL joins.

We also evaluated the usability of the tool using the Post-Study System Usability Questionnaire (PSSUQ). PSSUQ evaluates a system in 4 aspects using a questionnaire with a Likert scale from 1 (strongly agree) to 7 (strongly disagree) (see [8]). The PSSUQ scores by participant can be seen in **Fig. 2**. Participants 1, 11 and 13 had a score of 1 for interface quality.

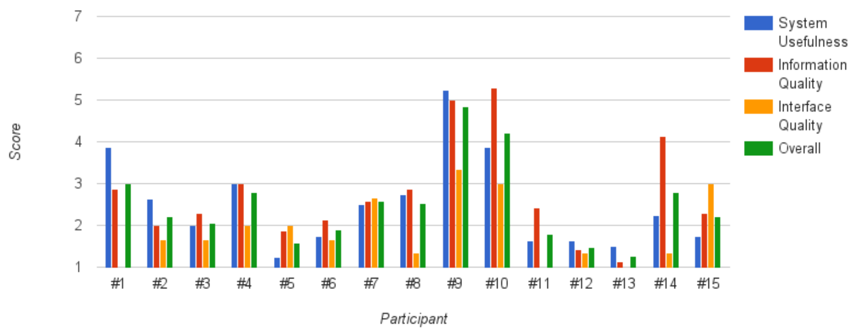

Fig. 2. PSSUQ questionnaire responses by participant

The scores indicate that the visual representation in the tool has good usability results for most participants, where these are less than half the scale (3.5).

5 Conclusions and Future Work

In this paper, we have presented a visual representation applied to the R2RML mappings language that is fully compliant with its specification. Our representation is based on the block metaphor, where each R2RML construct is represented as a block. An initial user study indicated that that the visual representation was beneficial in the creation of R2RML mappings and shows promising usability results.

Future work includes more extensive experimentation including using the representation in managing uplift mappings. We also intend to implement the loading of existing mappings into the tool, incorporation of function representation in mappings and to abstract our implementation from R2RML to support other uplift mapping languages. One limitation of our approach is the reuse of resources, which are supported by R2RML mappings written using Turtle syntax.

Acknowledgements. This paper was supported by CNPQ, National Counsel of Technological and Scientific Development – Brazil and by the Science Foundation Ireland (Grant 13/RC/2106) as part of the ADAPT Centre for Digital Content Technology (http://www.adaptcentre.ie/) at Trinity College Dublin.

References

1. Blinkiewicz, M., Bak, J.: Square: A Visual Support for OBDA Approach. In: Proceedings of the Second International Workshop on Visualization and Interaction for Ontologies and Linked Data (VOILA@ISWC 2016).
2. Bottoni, P., Ceriani, M.: SPARQL playground: A Block Programming Tool to Experiment with SPARQL. In: Proceedings of the International Workshop on Visualizations and User Interfaces for Ontologies and Linked Data (VOILA@ISWC 2015).
3. Debruyne, C., O'Sullivan, D.: R2RML-F: Towards Sharing and Executing Domain Logic in R2RML Mappings. In Proceedings of the Workshop on Linked Data on the Web, LDOW 2016, co-located with the 25th International World Wide Web Conference (WWW 2016).
4. Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., Van de Walle, R.: RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In: Workshop on Linked Data on the Web (LDOW 2014).
5. Heyvaert, P., Dimou, A., Herregodts, A.L., Verborgh, R., Schuurman, D., Mannens, E., Van de Walle, R.: RMLEditor: A Graph-based Mapping Editor for Linked Data Mappings. In: The Semantic Web - Latest Advances and New Domains (ESWC 2016).
6. Knoblock, C., Szekely, P., Ambite, J.L., Goel, A., Gupta, S., Lerman, K., Muslea, M., Taheriyan, M., Mallick, P.: Semi-Automatically Mapping Structured Sources into the Semantic Web. In: Proceedings of the 9th Extended Semantic Web Conference (ESWC 2012).
7. Lembo, D., Rosati, R., Ruzzi, M., Savo, D.F., Tocci, E.: Visualization and Management of Mappings in Ontology-Based Data Access (progress report). In: Informal Proceedings of the 27th International Workshop on Description Logics (2014).
8. Lewis, J. R.: Psychometric Evaluation of the Post-Study System Usability Questionnaire: The PSSUQ. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Vol. 36, No. 16, pp. 1259-1260). SAGE Publications (1992).
9. Pinkel, C., Binnig, C., Haase, P., Martin, C., Sengupta, K., Trame, J.: How to Best Find a Partner? An Evaluation of Editing Approaches to Construct R2RML Mappings. In: European Semantic Web Conference (ESWC 2014).
10. Siciliaa, Á., Nemirovskib, G., Nolleb, A.: Map-on: A Web-Based Editor for Visual Ontology Mapping. Semantic Web Journal, (Preprint):1–12.
11. Rodrıguez-Muro, M., Hardi, J., Calvanese, D.: Quest: Efficient SPARQL-to-SQL for RDF and OWL. In: 11th International Semantic Web Conference on Posters and Demonstrations (ISWC 2012).