Christophe Debruyne, Jonathan Riggio, Emma Gustafson, Declan O'Sullivan, Mathieu Vinken, Tamara Vanhaecke, and Olga De Troyer. In 14th IEEE International Conference on Semantic Computing, ICSC 2019, San Diego, CA, USA, February 3 - February 4, 2020. Pages 315-320. IEEE Computer Society, 2020

Facilitating Data Curation: a Solution Developed in the Toxicology Domain

Christophe Debruyne*‡ Jonathan Riggio*, Emma Gustafson†, Declan O'Sullivan‡,
Mathieu Vinken†, Tamara Vanhaecke†, and Olga De Troyer*
*WISE Lab, Vrije Universiteit Brussels, Brussels, Belgium
†Research Group of In Vitro Toxicology and Dermato-Cosmetology (IVTD), Vrije Universiteit Brussel, Brussels, Belgium
‡ADAPT Centre, Trinity College Dublin, Dublin, Ireland

***Abstract*—Toxicology aims to understand the adverse effects of chemical compounds or physical agents on living organisms. For chemicals, much information regarding safety testing of cosmetic ingredients is now scattered in a plethora of safety evaluation reports. Toxicologists in our university intend to collect this information into a single repository. Their current approach uses spreadsheets, does not scale well, and makes data curation and querying cumbersome. Semantic technologies (e.g., RDF, OWL, and Linked Data principles) would be more appropriate for this purpose. However, this technology is not very accessible for toxicologists without extensive training. In this paper, we report on a tool that supports subject matter experts in the construction of an RDF–based knowledge base for the toxicology domain. The tool is using the jigsaw metaphor for guiding the subject matter experts. We demonstrate that the jigsaw metaphor is a viable option for generating RDF. Future work includes investigating appropriate methods and tools for knowledge evolution and data analysis.**

***Index Terms*—Data Curation, Jigsaw Metaphor, Toxicology**

I. Introduction

The goal of toxicology is to understand the adverse effects of chemical compounds or physical agents on living organisms. The In Vitro Toxicology and Dermato-Cosmetology research group within the VUB[1] aims to collect safety testing data of cosmetic ingredients, available in a plethora of publicly available safety evaluation reports issued by the Scientific Committee on Consumer Safety (SCCS)[2], with the goal of creating a "computational database" of safety testing data for future purposes, e.g., screening, use for experiments.

Challenges that these subject matter experts–from here on now called "experts"–faced were *data curation* and *data analysis*. In terms of data curation, their current practices consisted of analyzing these safety evaluation reports and keeping track of findings in a big spreadsheet. Fig. 1 illustrates how one kept track of the various studies mentioned in a publication by filling in the relevant column. While fine when corpora are small (in terms of number) and the information sought is limited (in terms of columns), this approach does not scale well for more ambitious projects.

In terms of data analysis, they were limited by the functionality provided by spreadsheets. Even though these tables can be called a "computational database"[3], they lack the structure and functionality for more advanced information retrieval and analysis. Even simple queries that aim to compare the outcome in specific tests between compounds that possess similar chemical characteristics are difficult to answer. This is complicated even more when those tables are stored in multiple workspaces or files. The use of spreadsheets also leads to issues in terms of semantics; the semantics of columns are not explicit, and different experts could enter values in a heterogeneous manner. Note that the use of a relational database would solve some of the issues but not this last one.

Therefore, the problem we tackle is: "How can we facilitate subject matter experts in the domain of toxicology in the creation of a *knowledge base* for the available safety evaluation reports, which would facilitate data curation and data analysis?" Note that we chose to replace the term "computational database" with the term "knowledge base". The adoption of Semantic Web technologies (such as RDF and Linked Data principles) not only solves some of the aforementioned scalability problems, but they also provide an opportunity to enrich the data with external resources turning it into a knowledge base.

However, Semantic Web technology such as RDF is not very accessible for subject matter experts who are not-ICT literate. Studies have shown that subject-matter experts face challenges in curating, linking, and using Linked Data. McKenna et al., for instance, conducted a survey to identify the challenges faced by subject matter experts in the library domain [1]. We know from these studies that the creation and management of graphs in plain RDF is a challenge for these experts. For this reason, we chose to adopt in our solution a metaphor, i.e., the block or jigsaw metaphor–which became popular with programming languages such as Scratch–and which has been proven successful for the creation of Linked Data mappings [2] and the formulation of SPARQL queries [3].

This paper reports on the platform we are currently developing to support subject matter experts in data curation. We elaborate on the various components ranging from the ontology to knowledge organization, and on how the jigsaw metaphor is

[1]Vrije Universiteit Brussel
[2]https://ec.europa.eu/health/scientific_committees/consumer_safety_en

[3]"Computational database" is a term that experts in this domain use for any technology and representation allowing for some data manipulation and analysis, ranging from spreadsheets to more complex (domain-specific) database technologies.

used to accommodate non-ICT literate. The remainder of the paper is organized as follows: Section 2 provides information on why the jigsaw metaphor is adopted; Section 3 provides details on the platform with a focus on ontologies, graphs and knowledge organization; Section 4 discusses the prototype we have developed; Section 5 provides a discussion. In Section 6, we review related work. Section 7 concludes the paper.

II. The Jigsaw Metaphor

Metaphors are described in [4] as the use of familiar concrete objects to help structure our thoughts and comprehension of more abstract concepts. In user interface design, an interface metaphor is thus drawing upon the knowledge of familiar concepts to facilitate learning and using a system. Erickson proposed a method, presented in [4], for designing or adopting appropriate interface metaphors. The method consists of the following steps, which we will immediately apply to our problem:

1) Functional definition. Because the purpose of a metaphor is to help the users to understand how the system works, the first step is to define this purpose. Our focus is on data curation. Therefore, the purpose of the metaphor is supporting experts in creating and managing knowledge graphs.
2) Identify users' problems. In this step, we must identify why understanding how the system works and how they can use it for their purpose may prove to be challenging for the users. In our case, the experts need to provide data according to a particular structure to create the knowledge graph. While the RDF data model allows one to create complex graphs, the experts–who are not necessarily trained in RDF–perceives this as difficult and experience troubles in constructing valid and correct graphs.
3) Metaphor generation. In this step, suitable metaphors should be identified, or new ones should be created. In our case, based on the identified user's problems, we can conclude that experts need to be guided in creating valid graphs. This could be done by means of templates, but since not all values need to be provided in all cases, these templates must account for properties that are not known or irrelevant. Furthermore, since pieces of information need to be placed in the right location, we deemed the adoption of a jigsaw metaphor justified for our purpose.

To illustrate how this metaphor is used, we provide an example in Fig. 2. An expert provides a label and the URL of a (safety evaluation) report and is then led to the jigsaw environment, containing an empty report-piece acting as a "root". A safety evaluation report can refer to many studies, which can be of different types. Each type of study has its jigsaw piece that can be snapped inside a report-piece. These study-pieces are accessible from the category "Studies" in the toolbox on the left. To capture the content of a study, various so-called attribute groups are accessible from "Components" in the toolbox, which depends on the current selection (see Section 4 for more details). The example in Fig. 2 shows, on the left, the jigsaw representation of a report for Vitamin A referring an In-Vitro Skin Absorption (Non-OECD) study for which some values have been filled in. On the right, one can see the RDF that is generated based on the specification given by the jigsaw pieces. Note that this is currently done for debugging purposes, as we do not expect our expert to inspect the generated RDF.

The jigsaw pieces are implemented by means of "blocks" in Google Blockly[4].

We recognize that the actual use of the knowledge base (i.e., for data analysis) is also challenging. However, this will be investigated in the future.

III. Knowledge Organization

A. Ontology and Knowledge Base

The previous section ended with an illustration of how the jigsaw metaphor is used for guiding data entry. In this section, we elaborate on the knowledge organization. The two main components are the ontology and the knowledge base:

1) The ontology keeps track of the safety evaluation reports (from now on called reports, the studies they mention, and, more importantly, the attributes of studies (such as the method of analysis and the age of the test animals, for instance –represented by OWL properties) that the experts want to catalog. Appropriate definitions of these attributes allow experts to understand what needs to be entered. For each "simple" attribute, we create a property in our ontology and declare the range of these properties (e.g., literals, resources, or XSD datatypes). We use the range declaration to generate code that will validate input, e.g., only accepting strict positive integers when an attribute expects values that are `xsd:nonNegativeInteger`.
2) The knowledge base is a triplestore containing the triples for each safety evaluation report. The triplestore will become the so-called "computational database" that the experts seek. A SPARQL endpoint allows one to to explore this information.

The ontology is implemented using OWL 2 [5]. The ontology has been published according to best practices and guidelines in the Semantic Web community, albeit behind a firewall. Documentation was generated using WIDOCO [5]. The former generates documentation for the OWL ontology using its axioms and annotations, and the latter visualizes the ontology. It furthermore generates multiple serializations of the ontology.

While the ontology is "static" (i.e., published as files), we have a triplestore that contains the data of the safety evaluation reports. We store the information on the safety evaluation reports in separate graphs (one per report). We generate a URI for each report and relate it with its publication using a `dcterms:source` predicate.

The URIs for our reports follow a certain pattern that facilitates support for Linked Data principles. For example,

[4]https://developers.google.com/blockly/

[5]http://www.w3.org/TR/2012/REC-owl2-overview-20121211/

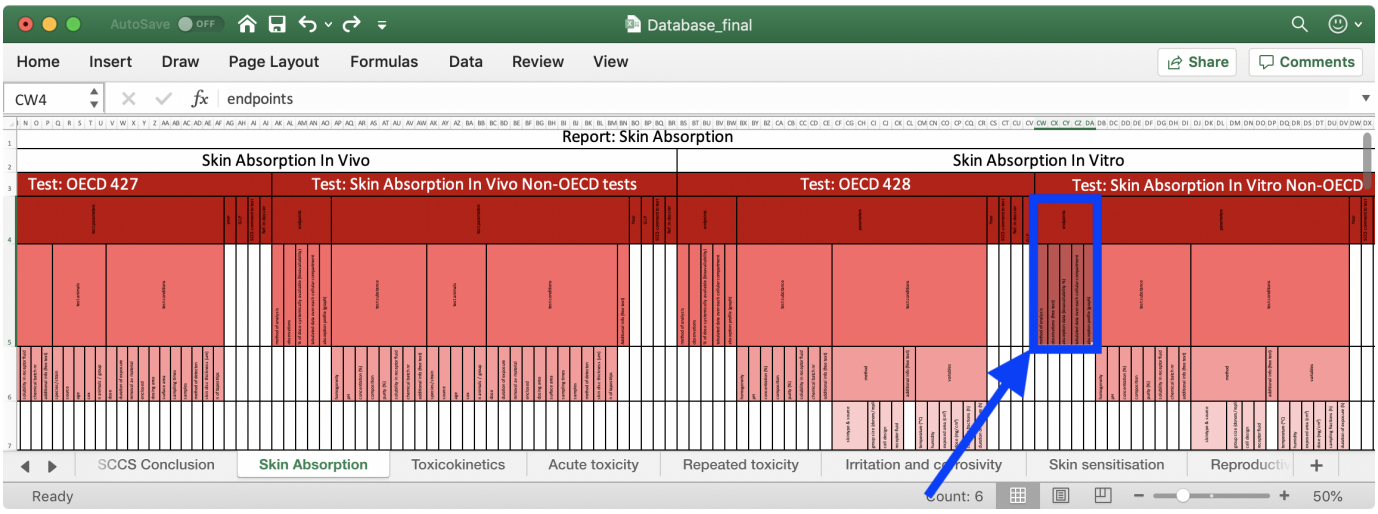

Fig. 1: Example of a spreadsheet developed by subject matter experts for keeping track of studies mentioned by SCCS safety evaluation reports. This sheet, for reports on skin absorption, groups separates in vivo tests from the in vitro tests. In each group, we have two tests: those according to OECD directives and those that are not. In this paper, we will focus on the endpoints of Non-OECD in vitro skin absorption tests.

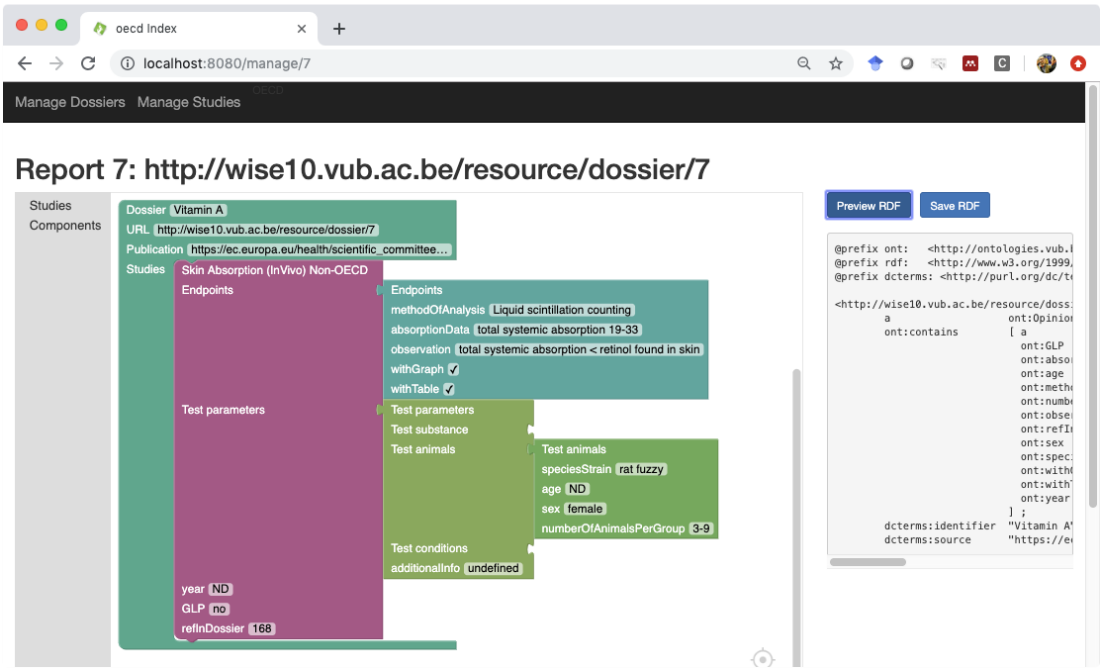

Fig. 2: Using the jigsaw metaphor for providing information on studies mentioned in a report. Notice the corresponding RDF statements on the right.

in order to obtain an HTML page describing a report, we use the URI of that report to be redirected to the HTML page.

B. Representing the structure of reports and studies

To structure the content of a study, our experts use columns and groups of columns in their spreadsheets (see Fig. 1). The nesting of columns can be arbitrary and the grouping depends on the study, meaning that groups may contain different columns or groups of columns in different studies. To support this practice, we introduced the notion of *attributes* and *attribute groups* in a separate graph. While those groups could be modeled in the ontology, it would have added an additional layer of complexity, which would have an impact on scalability and reusability. For this reason, we have chosen to keep the ontology as simple as possible and limit the number of axioms. However, as we need to know those attributes and groups of attribute to know how the jigsaw pieces should be grouped and rendered, we store these attributes and attribute groups, as well as their order in a graph (one graph per study).

It is justified to keep this separated from the ontology because they are only a representational structuring mechanism. To this end, we introduced a couple of predicates. The predicates are (using the prefix ns)[6]:

1) `ns:attributeGroup` relating instances of `ns:Study` or `ns:AttributeGroup` to instances of `ns:AttributeGroup`;
2) `ns:attribute` relating instances of `ns:Study` and `ns:AttributeGroup` to instances of `ns:Attribute`;
3) `ns:predicate` relates an `ns:Attribute` to a predicate from the ontology;
4) `ns:order` for ordering attributes and attribute groups within a study or attribute group; and `ns:color` for indicating the color of a study or attribute group–requested by experts after an initial version of the prototype.

The use of these predicates is exemplified in Listing 1. The resource `:SkinAbsorptionInVitroNonOECD`, which is stated to be a test in the ontology, has two attribute groups ("endpoints" and "parameters") and four attributes (year, reference in dossier, SCCS comments to text, and GLP). Those attribute groups and attributes happen to be in that order. While attributes only provide information on their predicate (an OWL property) and their order, attribute groups also provide information of their attributes and attribute groups. The attribute "endpoints" contains attributes, for instance, and the attribute group "parameters" contains an attribute group "test substance".

This graph containing the "presentation layer" is used as follows. For each type of study and attribute group X in this graph, we create the jigsaw piece based on the attributes and attribute groups of X, and their order. We also ensure that pieces fit properly by ensuring that the "parent" piece's checks correspond with the "child" piece's output.

Depending on the range of a predicate, there is some support for data validation; `xsd:boolean` is mapped to a "field_checkbox" (rendered as a checkbox in the jigsaw piece), `xsd:string` and `rdfs:Literal` to "field_input", for instance. Blockly does not provide fields for integers, floats, and doubles. Instead, we generate a field of the type "field_number" and additional conditions for each of the XSD datatypes; for example a precision of 1 for `xsd:integer`, and a precision of 1 and a minimum value of 0 for `xsd:nonNegativeInteger`.

To illustrate the generation of the jigsaw pieces, the attribute group "Endpoints" given in Fig. 3 (right) is generated from the snippet in Fig. 3 (left) (only showing the attribute "method of analysis"). The output of this block matches the "check" clause of its parent block SkinAbsorptionInVitroNonOECD's endpoint argument. The predicate methodOfAnalysis's range is `rdfs:Literal` and hence only strings are allowed. Fig. 3 also illustrates how attributes are linked to predicates in the ontology via "name".

[6]http://ontologies.vub.be/oecd# This URI is currently only accessible from within the VUB's firewall. The ontology and knowledge base are not to become publicly available.

```
ns:SkinAbsorptionInVitroNonOECD
  ns:attributeGroup [
    rdfs:label "endpoints" ;
    ns:order "A" ;
    ns:attribute [
      ns:predicate ns:methodOfAnalysis ;
      ns:order "A" ] ;
    ns:attribute [
      ns:predicate ns:observation ;
      ns:order "B" ] ;
    # Rest omitted for brevity
  ] ;
  ns:attributeGroup [
    rdfs:label "parameters" ;
    ns:order "B" ;
    ns:attributeGroup [
      rdfs:label "test substance" ;
      ns:order "A" ;
      ns:attribute [
        ns:predicate ns:homogeneity ;
        ns:order "A" ] ;
      ns:attribute [
        ns:predicate ns:pH ;
        ns:order "B" ] ;
      # Rest omitted for brevity
    ] ;
    # Rest omitted for brevity
  ] ;
  ns:attribute [
    ns:predicate ns:year ;
    ns:order "C" ] ;
  ns:attribute [
    ns:predicate  ns:Ref_in_dossier ;
    ns:order "D" ] ;
  ns:attribute [
    ns:predicate ns:SCCS_comment_to_test ;
    ns:order "E" ] ;
  ns:attribute [
    ns:predicate ns:GLP ;
    ns:order "F" ]
  .
```

Listing 1: A snippet of the representational structure of a study.

To summarize, we have three different graphs in our platform. These are depicted visually in Fig. 4. The graphs are: 1) The ontology, stored as a file and accessible via a URL. 2) The named graphs containing the structure of reports and tests (one graph for each). Notice how the URIs of these named graphs correspond with the URIs of reports and tests in the ontology. These graphs are used to generate the jigsaw pieces. 3) The named graphs for the reports assembled by the subject matter experts. The URIs for these named graphs are created by the system.

IV. Working Prototype

Currently, the prototype is built as an Apache Tapestry application built on top of Apache Jena, which provides the triplestore and SPARQL endpoint. Google Blockly was used to implement the jigsaw metaphor.

The only block that is not rendered dynamically is the top-level block "Dossier", which requires a name, a URL of the document, and refers to studies. The system consults the ontology to retrieve a list of studies. For each study, SPARQL is used to retrieve information about the study in terms of

```
Blockly.Blocks['...#SkinAbsorptionInVitroNonOECD-Endpoints'] = {
  init: function() {
    this.jsonInit({
      "type": "...#SkinAbsorptionInVitroNonOECD-Endpoints",
      "message0": 'Endpoints',
      "message1": 'methodOfAnalysis %1',
      "args1": [{
        "type": "field_input",
        "name": "http://ontologies.vub.be/oecd#methodOfAnalysis" }],
      // OMMITTED FOR BREVITY
      "output": "...#SkinAbsorptionInVitroNonOECD-Endpoints",
      "colour": 202 });    }    };
```

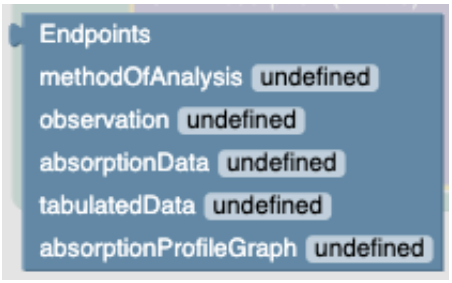

Fig. 3: We have a snippet of a jigsaw definition on the left. The various blocks are identified by strings: URIs for studies, and a concatenation of a study's URI and headings for attribute groups. For brevity, we had to omit a part of the "paths" (3 dots). On the right, we have a block generated from that snippet.

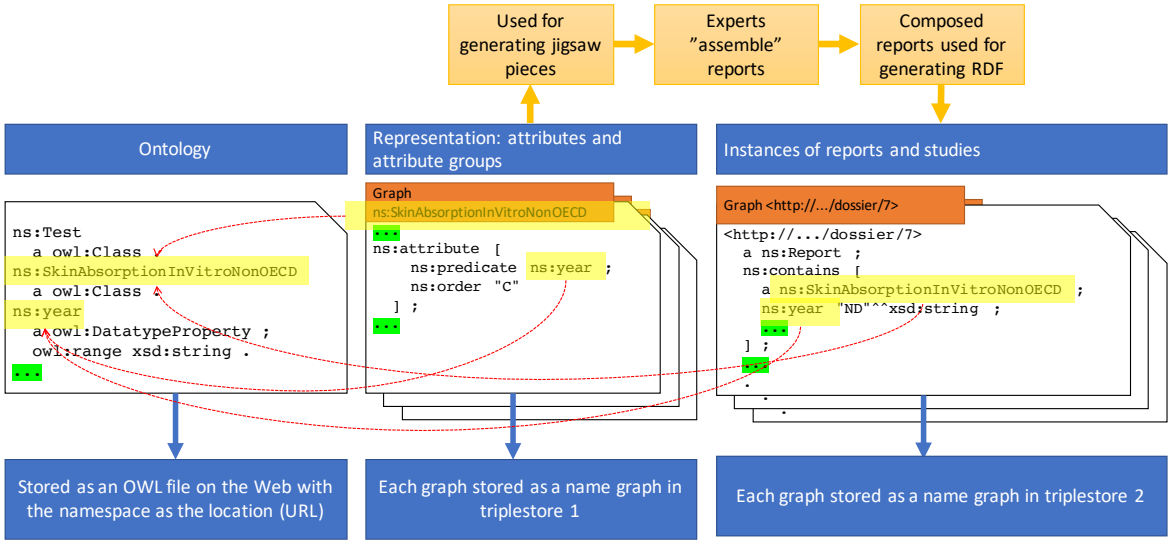

Fig. 4: Visual representation of the different graphs in our platform and their use: the ontology accessible via a URL, the graphs containing the structure for each report and test, and the graphs containing data for each report.

its constituents and that information is used to generate the various blocks. All blocks concerning studies are available from the category "Studies". In order to guide the experts, the toolbox's interface changes depending on the context, because different studies have different attributes and attribute groups. When the piece of a study or an attribute group is selected, then the category "Components" will show the attribute groups that can be connected to that piece.

With Blockly, one has to build "generators" for generating the desired output from the blocks. This would have required generating a generator for each block. We instead created an XSLT file to transform the XML representation of the blocks into RDF. This was possible since the various blocks embed the URIs of the classes and properties of our ontology. The RDF produced with the XSLT is then used to populate the knowledge base.

V. Discussion

1) On the Jigsaw Metaphor: Erickson proposed several aspects to evaluate the usefulness of a metaphor [4]: the amount of structure provided by the metaphor, the applicability of the metaphor, the ease of representing the metaphor (i.e. representability), the suitability to an audience, and the metaphor's extensibility. We argue that the amount of structure provided by the metaphor is good. Different bits of information can be combined, and users can only piece bits of information together that fit. This also allows us to argue that the metaphor is applicable and relevant for the problem at hand, i.e. experts are now challenged with the freedom that RDF provides. Furthermore, the jigsaw metaphor is easy to represent in a visual way. The adoption of Google Blockly comes with auditory cues (clicks when pieces fit) and support for collapsing and expanding pieces to maintain oversight. These contribute to the representability of the metaphor. In terms of suitability to the target audience, puzzles are familiar to most people, which mean that there will be no problem to understand the metaphor. A known limitation of this metaphor is its tree-like structure, which is fine for the fairly "flat" data entry currently requested by experts. When the need for graph-like data entry would arise, the metaphor may need to be revised.

2) On Reusing the Metaphor: In this paper, we adopted the jigsaw metaphor for data entry. Future work will focus on

data analysis. To this end, the adoption of jigsaw pieces for SPARQL queries, as proposed by [3], might be worthwhile considering as experts will have already become acquainted with the jigsaw metaphor.

3) On Genericity: We recognize that our approach is likely suitable for domains outside toxicology. We, however, want to avoid to introduce our approach as a generic solution prior to any evaluation. Synthesizing generic principles for knowledge elicitation with metaphors is part of future work.

VI. Related Work

The jigsaw metaphor became popular with programming languages such as Scratch. The adoption of the jigsaw metaphor in the Semantic Web community has been used, with success, for the creation of Linked Data mappings [2] and the formulation of SPARQL queries [3]. Experiments already indicated that users with similar backgrounds achieved higher performance and had a lower perceived mental workload when creating Linked Data mappings [6]. Recently, [7] reported on a block-based approach for instantiating a recipe ontology and an evaluation with an experiment involving 14 participants. Their method consists of designing blocks and mapping their contents to RDF according to their ontology.

This indicates that this metaphor has already been adopted within the community for different purposes. Similar to [7], we adopt this metaphor for the creation of RDF. While the advantage of [7] is the control over the blocks they design for their ontology, the disadvantage is that the approach will not scale well as the ontology evolves, as is the case in our setup. We overcome this problem by rendering the generation of blocks in a more generic way by introducing an additional layer from which blocks are derived.

In terms of data curation, the community has looked into templates. KawaWiki [8], for instance, creates templates based on RDFS ontologies. In DaCura [9], the authors propose a framework in which so-called data architects design schema's, which are then used to generate interfaces for the data curators. Where KawaWiki integrated their template engine in a wiki, DaCura's proposal is a bit more elaborate in terms of user roles and schema evolution. While the forms generated by templates are hypothetically accessible for people who are non-ICT literate, we believe such an approach would have not scaled well in our case, as a report could provide details on an arbitrary number of studies.

VII. Conclusions and Future Work

In order to facilitate subject matter experts in the creation of knowledge graphs for toxicology, we proposed an approach based on the jigsaw metaphor. This metaphor has proven to work for creating SPARQL queries and R2RML Mappings, but its use for data curation was not yet tried. The prototype we present in this paper consists of the following components: the ontology, a graph capturing the way experts structure available information from studies, and the actual knowledge base containing all data. The current version of the tool has been trialed with a subject matter expert whose feedback is used to improve the prototype.

Next to providing support for the actual data analysis, we are aware that we also need to investigate appropriate methods and techniques for knowledge base evolution–managing or propagating changes when properties change, for instance. A combination of the representational structure and the ontology can be used to assess the impact of a change and identify which parts of the knowledge base need revision.

Finally, except for the references to the SCCS safety evaluation reports and the adoption of a couple of vocabularies, the data in the knowledge base is not yet linked to other Linked Data datasets. Choosing datasets and enriching the knowledge base with such links depends on the data analysis use cases, which are still needed to be identified.

Acknowledgment: This work was partially supported by the funds of Cosmetics Europe. Christophe Debruyne is supported by the ADAPT Centre for Digital Content Technology, which is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

References

[1] L. McKenna, C. Debruyne, and D. O'Sullivan, "Understanding the position of information professionals with regards to linked data: A survey of libraries, archives and museums," in *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, JCDL 2018, Fort Worth, TX, USA, June 03-07, 2018*, J. Chen, M. A. Gonçalves, J. M. Allen, E. A. Fox, M. Kan, and V. Petras, Eds. ACM, 2018, pp. 7–16. [Online]. Available: https://doi.org/10.1145/3197026.3197041

[2] A. C. Junior, C. Debruyne, and D. O'Sullivan, "Juma uplift: Using a block metaphor for representing uplift mappings," in *12th IEEE International Conference on Semantic Computing, ICSC 2018, Laguna Hills, CA, USA, January 31 - February 2, 2018*. IEEE Computer Society, 2018, pp. 211–218. [Online]. Available: https://doi.org/10.1109/ICSC.2018.00037

[3] P. Bottoni and M. Ceriani, "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 co-located with 14th International Semantic Web Conference (ISWC 2015), Bethlehem, Pennsylvania, USA, October 11, 2015.*, ser. CEUR Workshop Proceedings, V. Ivanova, P. Lambrix, S. Lohmann, and C. Pesquita, Eds., vol. 1456. CEUR-WS.org, 2015, p. 103. [Online]. Available: http://ceur-ws.org/Vol-1456/paper12.pdf

[4] T. D. Erickson, "Working with interface metaphors," in *Readings in Human–Computer Interaction*. Elsevier, 1995, pp. 147–151.

[5] D. Garijo, "Widoco: A wizard for documenting ontologies," *The Semantic Web – ISWC 2017*, pp. 94–102, 2017.

[6] A. C. Junior, C. Debruyne, L. Longo, and D. O'Sullivan, "On the mental workload assessment of uplift mapping representations in linked data," in *Human Mental Workload: Models and Applications - Second International Symposium, H-WORKLOAD 2018, Dublin, Ireland, 20-21 September, 2018, Revised Selected Papers*, ser. Communications in Computer and Information Science, L. Longo and M. C. Leva, Eds., vol. 1012. Springer, 2018, pp. 160–179.

[7] Övünç Öztürk and T. Özacar, "A case study for block-based linked data generation: Recipes as jigsaw puzzles," *Journal of Information Science*, 2019. [Online]. Available: https://doi.org/10.1177/0165551519849518

[8] K. Kawamoto, Y. Kitamura, and Y. A. Tijerino, "Kawawiki: A semantic wiki based on RDF templates," in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology - Workshops, Hong Kong, China, 18-22 December 2006*. IEEE Computer Society, 2006, pp. 425–432. [Online]. Available: https://doi.org/10.1109/WI-IATW.2006.85

[9] K. C. Feeney, D. O'Sullivan, W. Tai, and R. Brennan, "Improving curated web-data quality with structured harvesting and assessment," *Int. J. Semantic Web Inf. Syst.*, vol. 10, no. 2, pp. 35–62, 2014. [Online]. Available: https://doi.org/10.4018/ijswis.2014040103