# A Jigsaw-Based End-User Tool for the Development of Ontology-Based Knowledge Bases*

Audrey Sanctorum[1][0000−0002−4872−4687], Jonathan Riggio[1][0000−0001−7277−5356], Sara Sepehri[2][0000−0003−1023−7987], Emma Arnesdotter[2][0000−0002−5891−5479], Tamara Vanhaecke[2][0000−0002−6685−7299], and Olga De Troyer[1][0000−0002−8457−7143]

[1] WISE Lab, Vrije Universiteit Brussel, Brussels, Belgium
[2] Research Group of *In Vitro* Toxicology and Dermato-Cosmetology (IVTD), Vrije Universiteit Brussel, Brussels, Belgium
{Audrey.Sanctorum,jonathan.riggio,sara.sepehri,emma.arnesdotter, tamara.vanhaecke,Olga.DeTroyer}@vub.be

**Abstract.** Knowledge bases are used to store and centralize information on certain topics in a domain. Using a well-structured and machine-readable format is a prerequisite for any AI-based processing or reasoning. The use of semantic technologies (e.g., RDF, OWL) has the advantage that it allows to define the semantics of the information and supports advanced querying. However, using such technologies is a challenging task for subject matter experts from a domain such as life science who are, in general, not trained for this. This means that they need to rely on semantic technology experts to create their knowledge bases. However, these experts are usually IT-experts and they are, in turn, not trained in the subject matter, while knowledge of the domain is essential for the construction of a high-quality knowledge base. In this paper, we present an end-user development (EUD) tool that supports subject matter experts in the construction of ontology–based knowledge bases. The tool is using the jigsaw metaphor for hiding the technicalities of the semantic technology, as well as to guide the users in the process of creating a knowledge base. The approach and the tool is demonstrated for building a knowledge base in the toxicology domain. The tool has been evaluated by means of a preliminary user study with nine subject matter experts from this domain. All participants state that with a little practice they could become productive with our tool and actually use it to represent and manage their knowledge. The results of the evaluation resulted in valuable suggestions for improving the tool and highlighted the importance of well adapting the terminology to the target audience.

**Keywords:** Knowledge Representation · Domain Ontology Creation · Knowledge Base · End-User Tool · Jigsaw Metaphor.

* Financially supported by Vrije Universiteit Brussel and Cosmetics Europe and the European Chemical Industry Council (CEFIC).

## 1 Introduction

In many domains of life-science, there is a huge amount of data and information (publicly) available and research in these domains is not possible without considering previously collected knowledge. However, the information is often in the form of documents and reports with various formats and different levels of details, which requires manual processing of the data to unlock the information. This makes it hard and time-consuming to aggregate knowledge from these documents and reuse it in research. Therefore, and in order to allow intelligent data processing, there is an increased interest in the field of knowledge representation & reasoning.

In this paper, we will use the term *knowledge base* to refer to the storage of information on certain topics in a domain using a well-structured and machine-readable format and that enables AI-based processing or reasoning. Knowledge base creation can either be done manually or automatically. Although work exist in the context of automated knowledge base construction [1, 10, 12, 28], this remains a true challenge, and the works are usually very domain specific or cover only a certain aspect of the knowledge base creation, e.g., the automatic identification of concepts with their hierarchy or populating a knowledge base from unstructured data. In this article we focus on the manual creation of the knowledge base which is still a widespread practice.

In some cases, subject matter experts use spreadsheets to collect and maintain the data (e.g., [37, 20]). However, spreadsheets do not scale well, are not able to deal with variations in the data, and they do not support advanced querying and analysis [3]. The use of semantic technologies (e.g., RDF [13], OWL [2]) is more appropriate for the purpose of creating knowledge bases, as it allows to structure the information in richer and more flexible ways, supports reasoning and more advanced querying mechanisms. However, using such technologies to create a knowledge base is a challenging task for subject matter experts who are, in general, not trained for this. Note that when using semantic technologies for creating a knowledge base, usually the concept of an ontology is used, which is a formal representation of knowledge pertaining to a particular domain [29]. Current tools for creating ontologies (e.g. Protégé [30], OntoEdit [36]; see [35] for an overview) are rather technical. This means that subject matter experts need to rely on semantic technology experts, also called ontology engineers, to create the ontology/knowledge base. However, these are usually IT-experts and they are, in turn, not trained in the subject matter, while the manual construction of ontologies requires extensive knowledge of the domain. This lack of knowledge about each other's domain results in a vast knowledge gap between the two groups of experts. In addition, each group of experts uses its own vocabulary and has its own concerns. Bridging this gap is a difficult and laborious process.

In the literature, different authors have studied and analyzed the gaps and barriers in interdisciplinary research [9, 27, 34] and proposed various approaches for bridging the gap. For instance, in [27] the use of human translators or intermediaries who are trained in both disciplines, is suggested to solve the communication problem between collaborators from different disciplines. However, in the case of the development of an ontology-based knowledge base, this only moves the problem of mastering two completely different disciplines to the intermediaries. Kertcher [27] also mentions the use of technology to bridge collaboration barriers. This is the approach we want to follow. If we could provide the subject matter experts with tools that are easier to use than current semantic technology tools and that hide the technicalities, such tools could facilitate the creation of knowledge bases without being dependent on ontology engineers.

In this paper, we present an end-user development (EUD) tool for the creation of domain-specific ontology-based knowledge bases. The developed tool is an improvement and extension of an earlier version, discussed in [16]. While the previous version only allowed to compose and fill the knowledge base by means of predefined domain concepts, the new version also provides the possibility to define these concepts, that is, it also allows the creation of a domain ontology defining the required domain concepts. The solution we provide for the creation of an ontology-based knowledge base exploits the jigsaw metaphor [19] by making use of Blockly blocks[3]. These jigsaw-based blocks are used for two purposes. Firstly, blocks are used to define domain concepts. Secondly, the (automatically) generated blocks for these domain concepts are assembled to create new blocks that define the structure of the knowledge base, and the knowledge base is populated by filling in the fields in those blocks.

The paper is organized as follows: In Section 2, we explain the approach and principles used for an ontology-based knowledge base. Section 3 first presents the overall approach, followed by an explanation of the tool developed, i.e. DIY-KR-KIT (Do It Yourself Knowledge Representation Kit), as well as its implementation. Section 4 presents the results of a first user study. Related work is discussed in Section 5. The paper ends with conclusions and future work (Section 6).

## 2 Ontology-Based Knowledge Bases

A knowledge base can be compared to a database that organizes data according to a certain data schema, also called data model. In this way, the database is an instantiation of the data model. In the same way, a knowledge base can be considered as an instantiation of an ontology [11]. While a data model allows to define the structure of data in a domain, an ontology is much more powerful. Typically, an ontology describes concepts in a domain and their properties, as well as relationships between the concepts and domain rules that apply to them. Note that in some applications, the instances of the concepts and relationships (i.e., the real data) are also considered as part of the ontology, removing the strict separation between model and data. We follow the approach proposed by Chasseray et al. in [11], where the distinction between model and data is kept: a knowledge base is composed of a *domain ontology* and an *instantiated ontology*. The domain ontology is used to specify the organizational structure of the knowledge base, and as the name indicates, the instantiated ontology is

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

an instantiation of the domain ontology containing the actual instances (data). Chasseray et al. combine ontologies with the OMG's Model-Driven Engineering (MDE) approach that defines four modelling levels: data level, model level, meta-model level, and meta-metamodel level. Following this MDE approach, the authors consider the domain ontology as an instantiation of an *upper ontology*, which defines the concepts and relationships needed to define domain ontologies and corresponds to the meta-model level from the MDE domain. This structure is shown in Fig. 1. Such an upper ontology contains general modeling concepts such as Concept, Relation, and Instance.

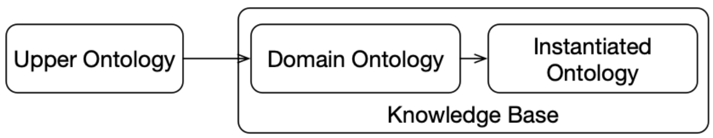



Fig. 1: Knowledge base structure with respect to MDE (adapted from [11])

We illustrate the structure of such an ontology-based knowledge base with a use case from the domain of toxicology. This use case will be used throughout the paper. In toxicology, Safety Evaluation Opinions issued by the Scientific Committee on Consumer Safety (SCCS), provide collections of information on safety testing of cosmetic ingredients. These Safety Evaluation Opinions are text documents[4]. In order to create a knowledge base containing all the information from these documents, and following the structure described above, the first step is to define a domain ontology describing the type of information that experts want to capture from these opinions, e.g., *test species used* and *test reliability*, and how this information should be structured. Next, this ontology needs to be populated by the actual information from the opinions, resulting into the instantiated ontology. The upper ontology should allow to define the domain ontology for the safety evaluation opinions, i.e., the concepts and relationships needed for capturing the information contained in the opinions.

## 3 Creating Ontology-Based Knowledge Bases with DIY-KR-KIT

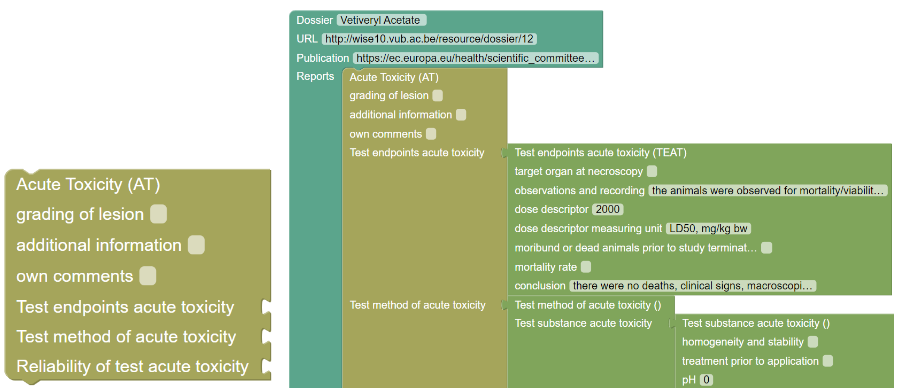
In order to create an EUD tool that could help subject matter experts in building an ontology-based knowledge base, we decided to use the jigsaw metaphor [19], a metaphor that became popular with the programming language Scratch [32]. This metaphor is already used successfully in a number of domains (see Section 5). We start by explaining how the jigsaw metaphor is used to create a domain-specific ontology-based knowledge base (Section 3.1), next we discuss the tool and its implementation (Section 3.2).

[4] Example Safety Evaluation Opinion: https://ec.europa.eu/health/scientific_committees/consumer_safety/docs/sccs_o_199.pdf

### 3.1 Using the Jigsaw Metaphor for Knowledge Base Construction

We first illustrate the principle of using the jigsaw metaphor for creating a knowledge base for the toxicological use case provided in the previous section. The concepts that typically appear in the Safety Evaluation Opinions are represented as jigsaw blocks (puzzle pieces), which contain placeholders for property values and connection points for composing concepts, see Fig. 2a for an example block. This example block represents the domain concept *Acute Toxicity*. Its main domain-specific property is *grading of lesion*. The two other properties, *additional information*, and *own comments* will be used to capture additional information provided in an opinion and comments that the subject matter experts wants to add. Its composing concepts are *Test endpoints acute toxicity*, *Test method of acute toxicity* and *Reliability of test acute toxicity*.

When a subject matter expert wants to store the information of an opinion into the knowledge base, (s)he composes a so-called *dossier* (representing the opinion) by connecting the relevant puzzle blocks and filling in the value fields in the blocks (see Fig. 2b for a (partial) example dossier). The jigsaw blocks can only be composed in a restricted way and validation for data fields is provided. The names of the blocks and fields correspond with the terminology used in the Safety Evaluation Opinions and the subject matter experts can fill the knowledge base while scanning the opinions. Based on the puzzle composition and its values, RDF is generated forming (a part of) the instantiated ontology.




(a) Example jigsaw block for the domain concept Acute Toxicity

(b) Example jigsaw block for the dossier Vetiveryl Acetate

Fig. 2: Example blocks

However, not all Safety Evaluation Opinions contain the same type of data. This means that it would be necessary to extend or change the domain ontology whenever new opinions are considered which contain information about domain concepts not yet covered in the domain ontology, or when the structure of an existing domain concept (i.e., its properties and sub concepts) does not fit anymore with the one used in the new opinion. In addition, these adaptations to the domain ontology should be translated to the jigsaw blocks and their composition rules. In this way, the subject matter experts would still be dependent on the IT-experts each time changes or new jigsaw blocks are needed. Extending or adapting the domain ontology and adapting the jigsaw block tool accordingly is a time-consuming procedure.

A first solution to solve this issue is trying to anticipate all needed domain concepts and create all possible puzzle blocks beforehand. However, as the domain of toxicology is very broad, creating such a domain ontology would not only be a very large and time-consuming undertaking (and without any guarantee that the ontology is complete), it would also result in a very extensive domain ontology, which would probably not be manageable and usable by the subject matter experts. Although, the existence of such a domain ontology (even for a part of the domain) would be very useful and efforts to realize this are already undertaken (e.g., the Adverse Outcome Pathway Ontology[5], [24, 22, 7]), this should be a collaborative effort of different stakeholders from the domain.

We therefore opted for a different solution. We want to allow the subject matter experts to specify, adapt and/or extend their domain ontology by themselves, using the same principle as how we allow them to create, adapt and extend the instantiated ontology, i.e., using the jigsaw metaphor. By doing so, IT-experts are not required to create the domain ontology, adapt or extend it. The tool remains the same, however depending on the objective, i.e., creating the instantiated ontology, or creating the domain ontology, different jigsaw blocks are used. For creating the instantiated ontology, the jigsaw blocks are based on domain concepts (defined in the domain ontology), while for creating the domain ontology, the jigsaw blocks are based on general modeling concepts (defined in the upper ontology). This process is illustrated in Fig. 3. First, the subject matter experts will use general jigsaw blocks to create the domain ontology (Step 1 in Fig. 3), i.e., to define the concepts and their relationships of the specific domain. For each defined domain concept, a jigsaw block will be generated. These jigsaw blocks can then, in turn, be used by the subject matter experts to compose and fill the knowledge base (Step 2 in Fig. 3).

### 3.2 DIY-KR-KIT Tool

As mentioned before, the tool is an improvement and extension of an earlier version [16]. The previous version only allowed to compose and fill the knowledge base by means of predefined domain concepts. This new version also provides the possibility to define domain concepts, i.e., to create the domain ontology. While the previous version was implemented using the Apache Tapestry framework, the new version is using Spring Boot. Our tool is a web-based application, built on top of Apache Jena, which provides the triplestore and SPARQL endpoint. The jigsaw metaphor is implemented via the Google Blockly JavaScript library.

The current DIY-KR-KIT tool is composed of two main parts, the *domain concept management* part, supporting step 1 of the knowledge base construction

[5] https://github.com/DataSciBurgoon/aop-ontology

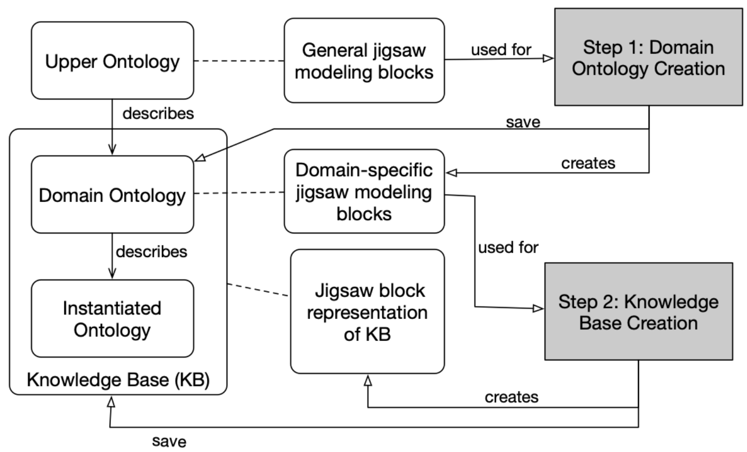



Fig. 3: Knowledge base construction process

process, and the *dossier management* part, supporting step 2 of the knowledge base construction process. On the tool's home page the user can either choose to create (or modify) a dossier or add (or modify) a domain concept. Adding such an item is done by giving a name and optionally a link (url) referring to extra information about the item. For example, in case of a dossier, the url is a link to the corresponding Safety Evaluation Opinion. Once the item is created, its name will appear in the dossiers or domain concepts menu, respectively. Note that the current tool allows to perform step 1 and step 2 at the same time, allowing for more flexibility: domain concepts can be added or modified when needed. Further, the current tool has been customised to already provide the basis structure needed for defining a knowledge base for the Safety Evaluation Opinions. This means that the domain concepts *Dossier* and *Report* are predefined and shown as such in the user interface. This is done to give more guidance to the subject matter experts; in this way the knowledge base will always consist of dossiers, which are composed of reports. However, the subject matter expert still needs to define the different blocks needed to compose a report by defining the necessary domain concepts (if not yet available in the existing list of domain concepts).

The Manage Domain Concept page allows users to specify or adapt a domain-specific concept. Fig. 4 shows the page for the domain concept *Test conditions acute toxicity*. On the left hand-side of the page, the top-level blue block "Domain Concept" allows to specify the structure of the domain concept. Pre-defined properties are: the name of the concept, an abbreviation, and a description, which the user can fill in. More properties can be added to a domain concept block by dragging and dropping the empty *Property* block from the *Custom Properties* tab in the menu at the left. Properties have a name, a value type and an optional default value. In the example, the first property of the domain concept is given the name "type of study" with value type "text" and default value "in vivo". Note that in the left sidebar menu we also have the *Default Properties* tab, which provides recurring property blocks, such as the "year"

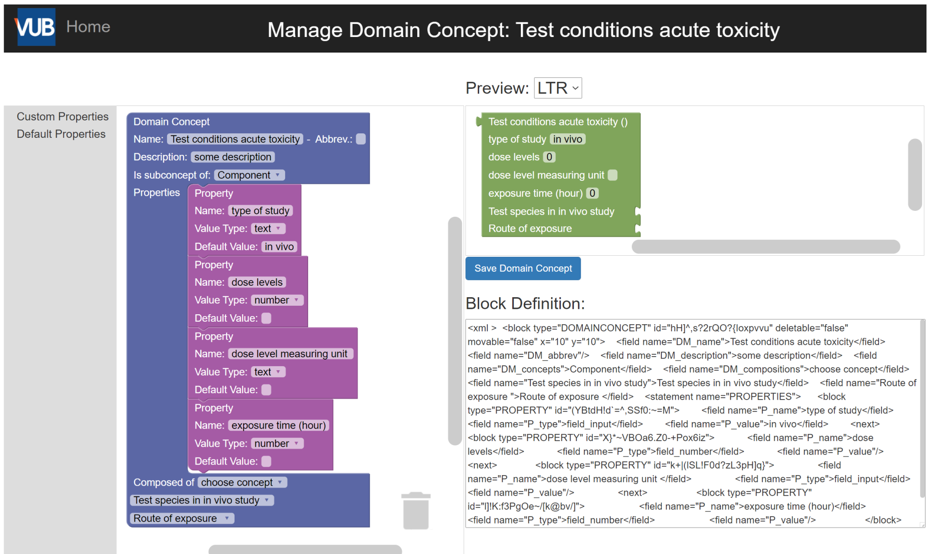



Fig. 4: Screenshot of the domain concept modification page

property block. By using these default properties blocks, users can save time. A domain concept can be composed of other domain concepts. In Fig. 4, we see that the *Test conditions acute toxicity* concept is further composed of the *Test species in in vivo study* and the *Route of exposure* concept. This is shown in the blue Domain Concept block under the "Composed of" field. The "Composed of" dropdown allows to add other composing concepts.

On the right in Fig. 4, a preview of the generated jigsaw block for the domain concept defined at the left is given. In this case the block contains two puzzle connectors on the right side, one for each composing concept given. Any changes made on the left side is reflected in real time in the preview on the right.

Blockly defines blocks in XML format as shown in the bottom right of the figure (currently only shown for debugging purposes). However, when a domain concept is saved (using the "Save Domain Concept" button), this XML representation is transformed into RDF using an XSLT file so that it can be integrated into the domain ontology.

## 4 User Study

The tool has been evaluated with a preliminary user study. The purpose of this first user study was to investigate the usability of the tool for the target audience (subject matter experts), validate the use of the jigsaw metaphor, as well as the terminology and principles used. This user study did not evaluate whether the subject matters experts would be able to identify domain concepts without the help from ontology engineers. This will be evaluated in a next user study.

The study was done in the context of the example use case, i.e. the creation of a knowledge base for Safety Evaluation Opinions. The participants were 9 (7 female) subject matter experts, researchers and lab technicians of the IVTD (*In Vitro* Toxicology and Dermato-Cosmetology) team of our university. They were familiar with the terminology used in the Safety Evaluation Opinions, but they had no expertise in knowledge modeling or ontologies.

Each participant received a time slot to perform tasks related to the creation of the domain ontology, creating a dossier for an opinion, and entering data from this opinion. In the instruction document, it was explicitly mentioned that the goal of the user study was to evaluate the ease with which the tasks could be done with the provided tool by people without IT background. They were also informed that they would be asked to fill in an online questionnaire about the tool. The evaluation was done on an individual basis and at any point during the study they could contact the first author (via email or via a video call)[6] for more information or clarifications. The participants were asked to record their session and were encouraged to speak aloud while interacting with the tool.

The participant could start by watching a YouTube video that explained the tool using similar tasks but based on a different opinion than the opinion used for the actual tasks. The video could also be consulted during the tasks.

The participants had to perform two tasks. In the first task they had to create three new domain concept blocks for an existing domain ontology. Because it was not the purpose to evaluate their capability to identify domain concepts from an opinion, we already identified the required domain concepts and properties, and provided them in the form of a text hierarchy: the domain concepts and properties were given as a bullet list; indented bullet lists were used for subconcepts. The value type for each property was given. In the second task, they had to use the available blocks (some created by performing the first task, some were already existing blocks) to create a dossier. Its structure was given in the form of a text hierarchy following the same conventions as in the first task. Participants were also asked to fill in values for properties, which were given and highlighted in yellow in the text hierarchy.

After finishing the tasks, the participants filled in the Post-Study System Usability Questionnaire (PSSUQ)[7], which consists of 16 questions with a 7 point likert scale (1 (strongly agree) to 7 (strongly disagree)) and an NA option. Participants could also leave comments. We also asked for their age and gender.

**Results.** The means per question (Q1 to Q16) and per category are summarised in Fig 5. While the "Overall" score gives the average score of all 16 questions, PSSUQ also groups the questions into three categories, namely system usefulness (SYSUSE), information quality (INFOQUAL), and interface quality (INTERQUAL). Note that, the lower the score the better the result.

[6] Due to the COVID-19 restrictions it was not possible to be physically present while the participant was performing the tasks.

[7] https://uiuxtrend.com/pssuq-post-study-system-usability-questionnaire

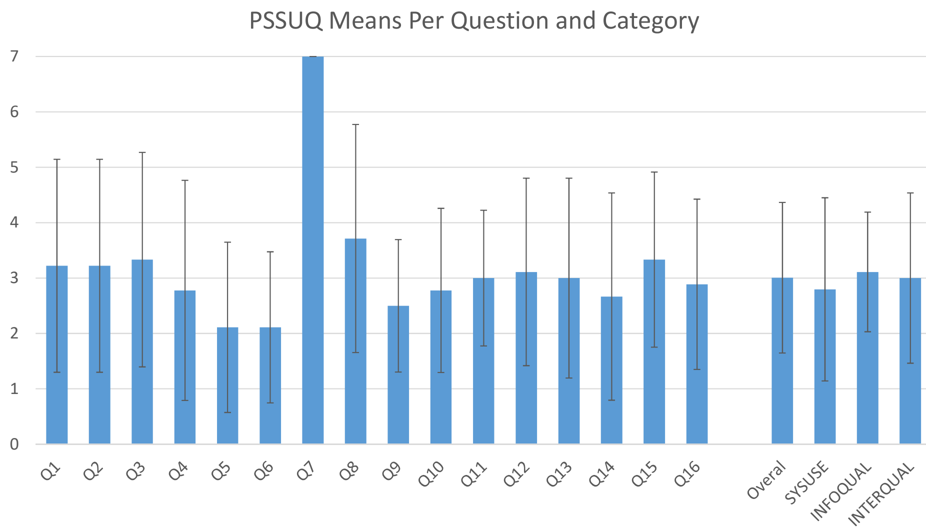



Fig. 5: Results of the post-study system usability questionnaire

All questions, except Q7, have a score better than the neutral score (4). Q7, related to whether the tool gave any error messages to fix problems, has been answered by only two participants who both rated it as "strongly disagree". These two participants encountered a bug which had to be fixed in order for them to continue with the user study, possibly explaining their low rating for this question. The questions with the best score are Q5 and Q6 related to how easy it was to learn the tool and quickly become productive using it. The score of Q8, scored by 7 of the 9 participants, is quite close to the neutral score (3.71). This question is related to Q7 and asked participants whether they could easily recover from a mistake using the system. This score was expected because our tool is at the prototyping stage and we did not yet focus on error handling and correction. Q3 and Q15 have the next highest score (3.33). The first one asked participants whether they were able to complete the tasks and scenarios quickly using the system. All participants spent approximately an hour on performing the tasks and watching the demo video. Compared to our own time when testing the scenario, i.e., 30 min, we believe that the participants who were using our tool for the first time still had a good time performance. However, the score could be an indication that some optimization in entering information could be necessary. Q15 asked whether the tool has all expected functionality. One participant mentioned that (s)he would like to be able to change the order of properties and sub-concepts which is not yet possible, as all properties are listed above all sub-concepts. This suggestion will be considered in the next version.

Note that certain questions, Q1 to Q4 and Q8, have a quite high standard deviation of $\pm 2$. The first questions are about whether the participant felt comfortable using the system. For this, the opinions were quite mixed: 3 participants responded rather negatively, while one person had a fairly neutral score and the remaining 5 participants were positive. Excluding Q7, the questions with the lowest standard deviation (1.2) are Q9 and Q11 asking participants whether the received information was clear and effective in helping them completing the tasks. This means that the participants agree on the fact that the explanation and demo video they received were useful to perform the tasks.

When comparing our results to the the norm defined by Sauro and Lewis [33] based on 21 studies and 210 participants, we see (Table 1) that the tool scored well on the system usefulness. The information quality is also well rated, but the interface quality should be improved: two participants had a really difficult time using the system and rated the interface as unpleasant to use.

| | **Sauro and Lewis [33]** | **Preliminary study** |
|---|---|---|
| **Overall** | 2.82 | 3.01 |
| **SYSUSE** | 2.80 | 2.80 |
| **INFOQUAL** | 3.02 | 3.11 |
| **INTERQUAL** | 2.49 | 3.00 |

Table 1: Preliminary study results compared to Sauro and Lewis' norm [33]

We also analysed the participants' interactions with our prototype:

- We noticed some confusion when moving blocks. When blocks are stacked on top of each other, Blockly considers this as a group of blocks that moves together. This means that whenever a user moves a block, all blocks underneath this block moves as well. While this was explained in our demo video, participants did not understand how to change the order of a block in the stack of blocks they created, causing some frustration.
- Certain participants had difficulties with converting the provided text hierarchy into a knowledge structure. They, for example, did not know when to create a property inside a concept (block) and when to create a new concept (block). This led to concepts with a single property, which did not make sense for the given case. A possible explanation could be that the text hierarchy did not provide enough information to decide when to model something as a property and when to model it as a domain concept. Given that this text hierarchy has been introduced uniquely for this study (to summarize information from an opinion), this problem might be gone when users use the opinion directly. However, another possible explanation could be that the difference between 'domain concept' and 'property' was not completely clear to these participants, which could be solved by providing a dedicated tutorial with criteria for deciding between modeling concepts and examples.
- A tutorial could also deal with the remarks received about the terminology used (i.e. reports, components, properties, ...), which some participants found a bit confusing at first. Although, we carefully avoid to use software specific terms, like sub-concepts and associations, we observed that using the right terminology is extremely important as it impacts the ease of learning. One participant mentioned: "I do however think that second use and thereafter would be much easier, since the system is not too complex to use once you get used to the terminology of things." Also the use of tool-tips giving

additional information when hovering a modeling concept, could help. However, in addition, the terminology will be carefully revised in collaboration with the participants. It is possible that some terms used for structuring the knowledge, such a component, have a different meaning in their domain.
- Further, we observed the need for additional small adjustments to the user interface, such as adding an undo button, making the horizontal scroll-bars more visible and providing more space for composing the blocks.

Given that this was a first evaluation mainly intended to evaluate the principles used and to collect feedback to improve the tool, we are satisfied with the results. All participants state that with a little practice they could become productive with our tool and actually use it to represent and manage their knowledge. The use of the jigsaw metaphor was not criticized or questioned at all. None of the participants had any issue in understanding the metaphor and use it for representing knowledge, indicating that this was a good choice. However, we see room for improvement: improving the ease of use (e.g., the undo button); adding some extra functionality (e.g., changing the order of properties) and optimizing the input process; and improvements to further ease the learning, for example adding tool-tips, an online tutorial, and an improved terminology.

## 5 Related Work

In [8], it is proposed to derive ontologies from conceptual maps (Cmaps). Conceptual maps allow to express concepts and their relationships in the form of concept-relation-concept. Concepts are represented as boxes or circles and the relationships as lines, all labeled. The paper presents a set of heuristic rules to map a conceptual map into an OWL ontology. A distinction is made between classification relations, composition relations, bidirectional relations and other relations. A first implementation of the translation system was made using Prolog, but no evaluation with end users was reported. It remains an open question whether conceptual maps are easier to use for subject matter experts than for instance a graphical representation of RDF.

Some papers (e.g., [4]) propose to transform UML class diagam into OWL. However, we are not convinced that UML class diagrams are suitable for people without computer science background, as the ease to learn and understand them has already been questioned for computer analysts [17].

Another direction is the use of a natural language interface. For instance, GINO (Guided Input Natural language Ontology editor) [5] is using a natural language approach. However, to avoid the limitations of full natural language interfaces, the authors are using a guided and controlled language akin to English. To add a new construct to the ontology, the user should start by typing "there is" or "there exists" after which a popup shows possible constructs, such as "class". After having selected the appropriate construct, the user is prompted to give a label and ends the sentence with a full stop. Next the sentence, e.g., "there is a class Lake.", is translated into OWL triples and loaded into the ontology. Properties are also defined in this way where the datatype or object property, domain and range are specified by means of pop ups. GINO has been evaluated for usability by six users without experience in ontology building and with no computer science background. The participants performed a small task consisting of creating one class, one subclass, one datatype property, one object property, adding one instance with values for two properties, and changing the value of a property. An average SUS score of 70,83 was obtained. However, it must be noted that the task was very small. To define a large ontology in this way may be very time consuming as the interface is quite verbose. Other works that follow a similar approach are CLOnE (Controlled Language for Ontology Editing) [18] which allows multiple classes to be expressed in a single sentence and Rabbit [21] which language is also somewhat richer than GINO's.

The jigsaw metaphor has been used in the Semantic Web community for the creation of Linked Data mappings [26] and the formulation of SPARQL queries [6]. Junior et al. [25] report on an experiment that indicates that users achieved higher performance and had a lower perceived mental workload when creating Linked Data mappings using the jigsaw metaphor.

Recently, Öztürk and Özacar [31] propose a block-based approach, based on Blockly, for instantiating a recipe ontology. Their approach is similar to the second step in ours as they use the blocks to populate the ontology. The blocks are predefined, which has the disadvantage that the blocks need to be adapted when the ontology evolves, and for each new ontology, blocks need to be programmed. We overcome this problem by rendering the generation of blocks from the meta level in which the end user defines the necessary blocks. The system proposed in [31] was evaluated for usability with 14 participants (students), however more than half of them had a background in ontology engineering.

Next to the use of the jigsaw metaphor for the programming language Scratch, intended for children between the ages of 8 and 16, the metaphor has also been used to support end-user development in other domains, e.g., for the development of IoT [23]; for the development of mobile applications [15]; and for debugging IF-THEN rules in an IoT context [14].

## 6 Conclusion and Future Work

In order to support subject matter experts in the creation of ontology-based knowledge bases, we proposed an EUD tool based on the jigsaw metaphor, a metaphor that became popular with the programming language Scratch. The purpose of applying this metaphor is to hide the technicalities and terminology of the semantic technologies used for creating ontology-based knowledge bases. The tool allows subject matter experts to create their own domain ontology, meaning that they can define the concepts and relationships used in their domain and needed to formally represent the available knowledge. In this way, the tool reduces the need to completely rely on an ontology engineer for creating the domain ontology. Next, the tool also allows the subject matter experts to actually set up the knowledge base and fill it with data. The approach, and the tool, is demonstrated and evaluated for building a knowledge base in the toxicology domain. The evaluation was a first preliminary evaluation done with nine subject matter experts from this domain, with the goal to use their feedback to improve the tool. All participants stated that with a little practice they could become productive with our tool and actually use it to represent and manage their knowledge. An important conclusion based on the received feedback is the need to pay due attention to the terminology used in the tool. The use of terminology that is not familiar to the subject matter experts or that has a different meaning in the subject domain might impact how comfortable users feel using the system.

As future work, the tool will first be improved and re-evaluated for overall usability. Next, a user study will be set up to investigate whether subject matter experts are able, after some training, to identify domain concepts, needed for structuring the knowledge base, without the help of an ontology engineer. Other future work includes the investigation for appropriate EUD methods and tools for knowledge base evolution, i.e., managing and propagating changes in the domain ontology to the knowledge base, for visualization of knowledge, for formulating advanced queries, and to provide reasoning capabilities.

## References

1. Alobaidi, M., Malik, K.M., Hussain, M.: Automated Ontology Generation Framework Powered by Linked Biomedical Ontologies for Disease-Drug Domain. Computer Methods and Programs in Biomedicine **165**, 117–128 (2018). https://doi.org/10.1016/j.cmpb.2018.08.010
2. Antoniou, G., Van Harmelen, F.: Web Ontology Language: Owl. In: Handbook on Ontologies, pp. 67–92. Springer (2004)
3. Baškarada, S.: How spreadsheet applications affect information quality. Journal of Computer Information Systems **51**(3), 77–84 (2011)
4. Belghiat, A., Bourahla, M.: An Approach based AToM3 for the Generation of OWL Ontologies from UML Diagrams. International Journal of Computer Applications **41**(3), 41–48 (2012)
5. Bernstein, A., Kaufmann, E.: GINO - A Guided Input Natural Language Ontology Editor. In: Proceedings of ISWC 2006, International Conference on the Semantic Web. Athens, USA (November 2006). https://doi.org/10.1007/11926078_11
6. Bottoni, P., Ceriani, M.: SPARQL Playground: A Block Programming Tool to Experiment with SPARQL. In: Proceedings of ISWC 2015, International Workshop on Visualizations and User Interfaces for Ontologies and Linked Data. Bethlehem, USA (October 2015)
7. Boyles, R.R., Thessen, A.E., Waldrop, A., Haendel, M.A.: Ontology-Based Data Integration for Advancing Toxicological Knowledge. Current Opinion in Toxicology **16**, 67–74 (2019). https://doi.org/10.1016/j.cotox.2019.05.005
8. Brilhante, V.V.B.B., Macedo, G.T., Macedo, S.F.: Heuristic Transformation of Well-Constructed Conceptual Maps into OWL Preliminary Domain Ontologies. In: Proceedings of IBERAMIA-SBIA-SBRN 2006, Workshop on Ontologies and their Applications. Ribeirao Preto, Brazil (October 2006)
9. Bruhn, J.G.: Beyond discipline: Creating a culture for interdisciplinary research. Integrative Physiological and Behavioral Science **30**(4), 331–341 (1995). https://doi.org/10.1007/BF02691605

10. Cahyani, D.E., Wasito, I.: Automatic Ontology Construction Using Text Corpora and Ontology Design Patterns (ODPs) in Alzheimer's Disease. Jurnal Ilmu Komputer dan Informasi **10**(2), 59–66 (2017). https://doi.org/10.21609/jiki.v10i2.374
11. Chasseray, Y., Barthe-Delanoë, A.M., Négny, S., Le Lann, J.M.: A Generic Metamodel for Data Extraction and Generic Ontology Population. Journal of Information Science (2021). https://doi.org/10.1177/0165551521989641
12. Cimiano, P., Völker, J.: A Framework for Ontology Learning and Data-Driven Change Discovery. In: Proceedings of NLDB 2005, International Conference on Applications of Natural Language to Information Systems. Alicante, Spain (June 2005). https://doi.org/10.1007/11428817_21
13. Consortium, W.W.W., et al.: Resource Description Framework (RDF)¡ http://www. w3. org. RDF/[Last accessed 2008-06-04]
14. Corno, F., Russis, L.D., Roffarello, A.M.: My IoT Puzzle: Debugging IF-THEN Rules Through the Jigsaw Metaphor. In: Proceedings of IS-EUD, International Symposium on End-User Development. Hatfield, UK (July 2019). https://doi.org/10.1007/978-3-030-24781-2_2
15. Danado, J., Paternò, F.: Puzzle: A mobile application development environment using a jigsaw metaphor. Journal of Visual Languages and Computing **25**(4) (2014). https://doi.org/10.1016/j.jvlc.2014.03.005
16. Debruyne, C., Riggio, J., Gustafson, E., O'Sullivan, D., Vinken, M., Vanhaecke, T., De Troyer, O.: Facilitating Data Curation: a Solution Developed in the Toxicology Domain. In: Proceedings of ICSC 2020, International Conference on Semantic Computing. pp. 315–320 (January 2020)
17. Dobing, B., Parsons, J.: How uml is used. Communications of the ACM **49**(5), 109–113 (2006)
18. Funk, A., Tablan, V., Bontcheva, K., Cunningham, H., Davis, B., Handschuh, S.: CLOnE: Controlled Language for Ontology Editing. In: Proceedings of ISWC 2007, International Semantic Web Conference (November 2007). https://doi.org/10.1007/978-3-540-76298-0_11
19. Gozzi, R.: The Jigsaw Puzzle as a Metaphor for Knowledge. ETC: A Review of General Semantics **53**(4), 447–451 (1996)
20. Gustafson, E., Debruyne, C., De Troyer, O., Rogiers, V., Vinken, M., Vanhaecke, T.: Screening of Repeated Dose Toxicity Data in Safety Evaluation Reports of Cosmetic Ingredients Issued by the Scientific Committee on Consumer Safety between 2009 and 2019. Archives of Toxicology **94**(11), 3723–3735 (2020)
21. Hart, G., Johnson, M., Dolbear, C.: Rabbit: Developing a Control Natural Language for Authoring Ontologies. In: Proceedings of ESWC 2008, European Semantic Web Conference. Tenerife, Spain (June 2008). https://doi.org/10.1007/978-3-540-68234-9_27
22. Hessel, E.V., Staal, Y.C., Piersma, A.H.: Design and Validation of an Ontology-Driven Animal-Free Testing Strategy for Developmental Neurotoxicity Testing. Toxicology and Applied Pharmacology **354**, 136–152 (2018). https://doi.org/10.1016/j.taap.2018.03.013
23. Humble, J., Crabtree, A., Hemmings, T., Åkesson, K., Koleva, B., Rodden, T., Hansson, P.: "Playing with the Bits" User-Configuration of Ubiquitous Domestic Environments. In: Proceedings of UbiComp 2003, International Conference on Ubiquitous Computing. Seattle, USA (October 2003). https://doi.org/10.1007/978-3-540-39653-6_20
24. Ives, C., Campia, I., Wang, R.L., Wittwehr, C., Edwards, S.: Creating a Structured Adverse Outcome Pathway Knowledgebase via Ontology-

Based Annotations. Applied In Vitro Toxicology **3**(4), 298–311 (2017). https://doi.org/10.1089/aivt.2017.0017
25. Junior, A.C., Debruyne, C., Longo, L., O'Sullivan, D.: On the Mental Workload Assessment of Uplift Mapping Representations in Linked Data. In: Proceedings of H-WORKLOAD 2018, International Conference on Human Mental Workload: Models and Applications. Amsterdam, The Netherlands (September 2018). https://doi.org/10.1007/978-3-030-14273-5_10
26. Junior, A.C., Debruyne, C., O'Sullivan, D.: Juma Uplift: Using a Block Metaphor for Representing Uplift Mappings (January - February 2018). https://doi.org/10.1109/ICSC.2018.00037
27. Kertcher, Z.: Gaps and Bridges in Interdisciplinary Knowledge Integration. In: e-Research Collaboration, pp. 49–64 (2010). https://doi.org/10.1007/978-3-642-12257-6_4
28. Maynard, D., Funk, A., Peters, W.: Using Lexico-Syntactic Ontology Design Patterns for Ontology Creation and Population. In: Proceedings of ISWC 2009, Workshop on Ontology Patterns. Washington, USA (October 2009)
29. Noy, N.F., McGuinness, D.L.: Ontology Development 101: A Guide to Creating Your First Ontology. Tech. rep., Standford Knowldge Systems, AI Lab (2001), https://corais.org/sites/default/files/ontology_development_101_aguide_to_creating_your_first_ontology.pdf
30. Noy, N.F., Crubézy, M., Fergerson, R.W., Knublauch, H., Tu, S.W., Vendetti, J., Musen, M.A.: Protégé-2000: An Open-Source Ontology-Development and Knowledge-Acquisition Environment: AMIA 2003 Open Source Expo. In: Proceedings of AMIA 2003, Annual Symposium on American Medical Informatics. Washington, USA (November 2003)
31. Öztürk, Ö., Özacar, T.: A Case Study for Block-based Linked Data Generation: Recipes as Jigsaw Puzzles. Journal of Information Science **46**(3) (2020). https://doi.org/10.1177/0165551519849518
32. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J.S., Silverman, B., Kafai, Y.B.: Scratch: programming for all. Communications of the ACM **52**(11), 60–67 (2009). https://doi.org/10.1145/1592761.1592779
33. Sauro, J., Lewis, J.R.: Quantifying the User Experience: Practical Statistics for User Research. Morgan Kaufmann, 2nd edn. (2016)
34. Siedlok, F., Hibbert, P.: The Organization of Interdisciplinary Research: Modes, Drivers and Barriers. International Journal of Manangement Reviews **16**(3), 194–210 (2014)
35. Slimani, T.: Ontology Development: A Comparing Study on Tools, Languages and Formalisms. Indian Journal of Science and Technology **8**(24), 1–12 (2015). https://doi.org/10.17485/ijst/2015/v8i1/54249
36. Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., Wenke, D.: OntoEdit: Collaborative Ontology Development for the Semantic Web. In: Proceedings of ISWC 2002, International Semantic Web Conference. Sardinia, Italy (June 2002). https://doi.org/10.1007/3-540-48005-6_18
37. Vinken, M., Pauwels, M., Ates, G., Vivier, M., Vanhaecke, T., Rogiers, V.: Screening of Repeated Dose Toxicity Data Present in SCC (NF) P/SCCS Safety Evaluations of Cosmetic Ingredients. Archives of Toxicology **86**(3), 405–412 (2012)