Semantic Web 1 (0) 1–4
IOS Press

Cluedo4KG: clues for Practising SW technologies

Camille Pradel [a], William Charles [b] and Nathalie Hernandez [b,*]
[a] *Chapsvision, Toulouse, France*
E-mail: cpradel@chapsvision.com
[b] *IRIT, University of Toulouse, France*
E-mails: william.charles@irit.fr, nathalie.hernandez@irit.fr

Abstract. This paper introduces SPARQLuedo and OWLuedo, two open-source educational resources designed for hands-on learning of Semantic Web technologies: SPARQL and OWL. Inspired by the board game Cluedo, these resources challenge learners to act as investigators solving a murder. SPARQLuedo guides users in formulating SPARQL queries to interrogate a dedicated RDF knowledge graph and uncover details of the crime, including the victim, the murderer, the location, and the murder weapon. OWLuedo, on the other hand, prompts learners to extend an existing ontology to model the crime scene in greater depth and leverage an OWL reasoner to identify the culprit. These resources, intended to complement lectures, aim to make learning Semantic Web technologies more engaging and interactive. Positive feedback from students who have used SPARQLuedo and OWLuedo demonstrates the effectiveness of this playful approach for acquiring practical skills in SPARQL and OWL.

Keywords: SPARQL, OWL, learning material, educational ressources, knowledge graph

1. Introduction

Offering hands-on that make it easy to master the languages and technologies of the Semantic Web is necessary to support its use and development. Most of the educational resources available today are either books ([3], [4] [1]), online courses[1], MOOCs[2], or exercises that practice these technologies on toy examples (such as the construction of the famous pizza ontology[3], [13] or guided querying of public knowledge graphs[4]).

In this article, we propose to take advantage of our fifteen years' experience in teaching these technologies to different audiences (engineers and doctoral students, IT specialists and non-specialists) to share resources putting these technologies into practice in a fun way, both engaging and interactive. These open-source educational resources have been designed to complement lectures or previously cited resources.

Based on the famous board game Cluedo, we have designed two open-source tutorials aimed at practising the use of SPARQL and OWL. They are proposed to be used once the learner has discovered these two languages via lectures. For the two tutorials, the learner has to carry out an investigation using clues to find out who committed

*Corresponding author. E-mail: nathalie.hernandez@irit.fr.
[1] https://www.classcentral.com/subject/semantic-web
[2] https://open.hpi.de/courses/semanticweb, https://open.hpi.de/courses/knowledgegraphs2023/overview, https://www.fun-mooc.fr/fr/cours/introduction-to-a-web-of-linked-data/
[3] https://protegewiki.stanford.edu/wiki/Protege4Pizzas10Minutes
[4] https://www.emse.fr/~zimmermann/Teaching/SemWeb/

a murder in a house. The goal is to help learners master the various features of SPARQL 1.1 and OWL 2 by motivating them by solving an investigation. They are designed to be integrated into face-to-face teaching sessions, while providing maximum guidance to learners via the web application we offer. The various stages to be followed are deliberately designed to arouse the curiosity of the learner, who can ask the teacher for help or additional information. As part of the Cluedo4KG[5] Web application that we propose, these two tutorials focus on the two aspects that we believe are essential to give knowledge graphs their full potential: (i) mastering access, interrogation and reuse of the quantity of data now available on the Linked Data Web, and (ii) mastering the essential notions linked to knowledge representation to provide a high level of expressiveness and reusability in knowledge graphs.

Section 1 introduces SPARQLuedo, the tutorial we set up to solve a murder, and practise the main features of the SPARQL query language. Section 2 presents OWLuedo, a tutorial with a similar concrete goal, that puts into practice the design of ontologies with OWL. In each section we describe the skills targeted, the resources we have designed, the implementation of the tutorial and the feedback we have had from the students who have followed it.

2. Related Works

Several works have tackled the issue of teaching semantic web technologies. Works such as [5, 7, 8] provide a broad overview of semantic web technologies aimed at beginners. MOOCs, such as ”Knowledge Engineering with Semantic Web Technologies” proposed by OpenHPI[6], are another entry point for students to acquire theoretical knowledge. [6, 9, 13] detail end-to-end courses, showcasing the lectures and practicals that are proposed for student. [13], specifically, is very broad-scoped in its coverage of semantic web technologies. In particular, the project it features provides a great coverage of the issues of ontology engineering. However, to the best of our knowledge, the related resources are not available. Tools to facilitate practicals have all also been proposed, such as [11]. These resources, in particular those bringing theoretical knowledge, can be used as a complement to the proposed resources.

3. SPARQLuedo

The goal of SPARQLuedo is to get learners to produce SPARQL queries while trying to solve an investigation linked to a murder. Based on the principle of Cluedo, clues are given at each stage so that the learner can, by querying a dedicated RDF knowledge graph, discover the victim, the murderer, the location, the weapon of a crime and the accomplice of the murderer. The tutorial is designed to put into practice the use of the main features defined in SPARQL 1.1 Query Language[7].

3.1. Targeted skills

Many knowledge graphs are now available on the web. Knowledge graphs are recognised as an effective way of sharing data and making it reusable. Exploiting the potential represented by these graphs means knowing how to manipulate and query them.

3.1.1. Querying skills

The main part of the skills to be acquired corresponds to the writing of SPARQL queries based on adapted features.

We target the following:

- **SPARQL.S1.1** Understanding the usefulness of querying a knowledge graph
- **SPARQL.S1.2** Writing a simple SPARQL query (SELECT and WHERE clauses)

[5]https://w3id.org/cluedo4KG/
[6]https://open.hpi.de/courses/semanticweb2015
[7]http://www.w3.org/TR/sparql11-query/

- **SPARQL.S1.3** Mastering Basic Graph Patterns (Syntax for IRI, Matching RDF Literals, Restricting value ...)
- **SPARQL.S1.4** Mastering complex patterns (group graph pattern, optional graph patterns, alternative graph pattern, negation, variable assignment, aggregate ...)
- **SPARQL.S1.5** Writing complex queries (function, subqueries...)
- **SPARQL.S1.6** Writing Federated queries

3.1.2. Kwnoledge representation skills

We believe that to promote the quality and reusability of knowledge graphs, they should be based on descriptions whose vocabulary is defined in one or more ontologies. For this tutorial we have built an ontology (the Cluedo4KG ontology [8]) and a knowledge graph relying on this ontology for describing the crime scene.

Targeted skills therefore also relate to understanding the vocabulary used for expressing the triples of the knowledge graph and using this vocabulary to formulate a SPARQL query.

For the first part of the tutorial, an extract of the Cluedo4KG ontology relevant to the query is provided in the interface. In the second part, an additional ontology to be used in the queries is indicated (the DBpedia ontology[9]) and the learner must find out for himself the elements of the ontology to be used. Finally, the learner must find the ontologies used in a given knowledge graph (Wikidata [10]) and the elements to be used in the query.

The targeted skills related to knowledge representation are:

- **SPARQL.S2.1** Understanding a given extract of an ontology
- **SPARQL.S2.2** Understanding an ontology from its documentation
- **SPARQL.S2.3** Finding and understanding the ontologies used in a given knowledge graph

3.1.3. Technical skills

Implementing SPARQL queries also involves mastering technical skills such as:

- **SPARQL.S3.1** Understanding a set of given RDF triples in turtle
- **SPARQL.S3.2** Understanding the results of a SPARQL query
- **SPARQL.S3.3** Understanding what is a SPARQL endpoint and using it

3.2. Description of the resources

We have defined several resources to implement the tutorial:

- the cluedo4KG ontology: https://w3id.org/cluedo4KG/onto
- the cluedo4KG knowledge graph accessible through a SPARQL endoint: https://w3id.org/cluedo4KG/KG
- a Web application containing the information needed to solve the investigation and allowing SPARQL queries to be formulated directly: https://w3id.org/cluedo4KG.

These resources are also available on gitlab[11]. Our goal is to make them either directly usable as such via a browser (a solution that makes their use the most intuitive) or installable locally in the learner's environment.

The tutorial requires the use of a SPARQL server. To facilitate its configuration, the Web application, that we provide, uses our hosted SPARQL server. The version on gitlab is configurable to use a server installed locally in the learner's environment.

3.2.1. Cluedo4KG ontology

The ontology we have built to describe the crime scene is relatively simple. It defines 4 `Classes`, 7 `ObjectProperties` and 1 `DataProperty`. The classes correspond to the types of entities in the scene: `House`, `Room`, `Person` and `Object` that can be used as a weapon. The `ObjectProperties` are defined to describe the house in which each room is located, the organisation of the rooms (adjoining rooms) and the room in which a person or object is located. The `InverseProperties` are also defined to leave the learner free to

[8] https://w3id.org/cluedo4KG/onto
[9] https://www.dbpedia.org/resources/ontology/
[10] https://www.wikidata.org/
[11] https://gitlab.irit.fr/cluedo4kg/cluedo4kgResource

express the query patterns when using them. The `DataProperty` is for expressing if a `Person` is alive. French and English labels are stated for each of the ontology's entities. We have deliberately chosen not to reuse ontologies in our ontology so that the vocabulary used for writing queries is as basic as possible. Part 2 of the SPARQLuedo tutorial does, however, include the appropriation of vocabularies used in DBPedia and Wikidata. While we believe it is important to introduce widely-used vocabularies, we also feel that it makes educational sense to discover complex vocabularies once the basics of SPARQL have been put into practice.

The ontology is accessible via a permanent W3id IRI and its available documentation is online.

3.2.2. *Cluedo4KG knowledge graph*

The knowledge graph has been designed so that it can both describe a murder scene with the Cluedo4KG ontology and be queried by considering the different skills identified in section 3.1. It is composed of 41 `Individuals` with a total of 106 `object property` and 16 `data property assertions`. A label is associated to each Individual, in French and in English when relevant. We have deliberately chosen as labels for `Indivuals` typed as `Person` names involving a pun. As the tutorial is meant to be fun, when the labels associated with these resources are returned, the learner is in most cases amused and motivated to discover the label of the other resources.

The use of a reasoner on the SPARQL server (or to saturate the knowledge graph before importing it), is necessary to generate all the triples (some of which are generated from `inverse properties` or the symmetry of the adjoining room object property, the `domain` and `range` of the properties etc).

3.2.3. *Tutorial walk-through*

The tutorial must be run from the Web application. Figure 1 shows a screenshot of the interface. It is in 2 parts. The first being the main part leads up to the murder elucidation and puts into practice the main features of SPARQL. Table 1 details in which part the various targeted skills are addressed. The second part is optional but allows to practice the advanced features of SPARQL (federated queries and subqueries). Its concrete goal is to find the murderer's accomplice.

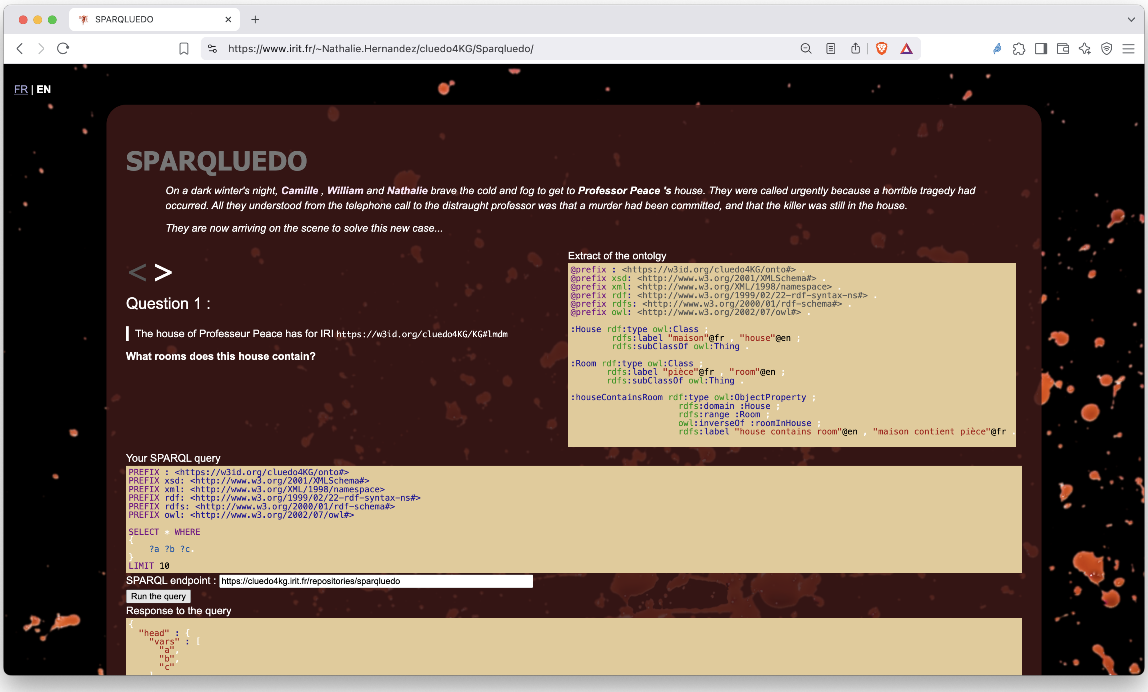

Fig. 1. Screenshot of the Web application

Table 1
Summary of the skills tackled by each question of each step of the SPARQLuedo tutorial

Skill	S1.1	S1.2	S1.3	S1.4	S1.5	S1.6	S2.1	S2.2	S2.3	S3.1	S3.2	S3.3
Part 1	All	Q1-Q6	Q1-Q6 Q14	Q7-18	Q10-Q13		All			All	All	
Part 2	All	Q21,Q24	Q22,Q25	Q21,Q25		All		Q20-Q22	Q23-Q26	All	All	All

In the first part, for each step, a clue, a question and an extract from the ontology to be used in the SPARQL query is displayed. This part is composed of 18 steps. In the area provided for the learners to enter their SPARQL query, a basic query template is given. In it we have deliberately included the basic prefixes as well as the prefix **:** referring to the cluedo4KG ontology. From the very first query, the learner must take into account the fact that this given prefix cannot be used directly to refer to the IRIs of the KG indivudals which allows him to take into account the management of IRIs in SPARQL. The learner must use the full URI of the house where the murder took place or define a new prefix. The IRI fragments of the ontology entities and the knowledge graph instances are strings defined from the English labels of the resources to give the learner a little extra help (which would not have been the case if we had used opaque IRIs). To perfect the understanding of IRIs and the added value of labels, the fragment of the IRI of the murder victim deliberately does not contain the characters of his name so as to encourage the learner to retrieve the label associated with the resource in the pattern of the query. The results of the queries are directly displayed in the application so that the learner can decide to move on to the next stage if the query answers the question asked. For a new step, the query used in the previous one is kept in the dedicated area, as for most steps the aim is to increase it.

In the second part, the same principle is followed for the 8 different steps. Clues and questions are given. The interesting point about this part is that the queries to be written must integrate information present in other knowledge graphs: DBpedia and Wikidata. Our aim is to demonstrate the potential of Linked Open Data and to show concretely how information from several knowledge graphs can be exploited in a single query. We chose these 2 knowledge graphs as they are widely used on the LOD and because each of them follows differently structured vocabularies or ontologies. We expect that, based on the skills acquired in part 1, the learner will acquire autonomy in searching for and understanding these vocabularies. For querying DBpedia (steps 19 to 22), we indicate that the query patterns to be formulated must use the vocabulary of the DBO ontology[12]. However, for querying Wikidata (steps 23 to 26), we leave it to the learner to discover the relevant vocabulary. For the steps in this part 2, the learner is asked to first formulate their query in the web interface associated with the endpoints of these two knowledge graphs before integrating the query into the application via a federated query. This has a twofold advantage. The learner can benefit from the functionalities of these web interfaces, which make it easier to write queries for these graphs. It also avoids saturating our SPARQL server with inappropriate federated queries. We have deliberately not provided any query pattern so that the learner can discover the specification of federated queries in SPARQL or return to the course material.

3.3. Evaluation

3.3.1. Specification coverage

The main aim of this tutorial is to learn how to write queries in SPARQL 1.1. All the features presented in the specification are covered by the content of SPARQLuedo, apart from the use of property paths in query patterns, named graphs and Construct form queries.

3.3.2. Practical information and Student feedback

This tutorial has been used every year for the past 12 years to train Masters-level students in semantic web technologies. It has been used for classes of students from engineering schools or university masters in computer science who had previously had a lecture on the Semantic Web and SPARQL. We have also used them for one-off

[12] https://www.dbpedia.org/resources/ontology/

Table 2
Feedback from students on SPARQLuedo during 2024-25 academic year

	bad (0)	very dissatisfied (1)	dissatisfied (2)	satisfied (3)	good (4)	excellent (5)	Total
overall assessment	0	1	5	17	34	45	102
fun aspect	0	0	0	3	42	57	102
clarity of instructions	0	0	2	20	38	42	102
acquisition of skills	0	2	5	15	36	43	102

training courses for doctoral students (in computer science and language science), as well as for training courses for in-house engineers looking to upgrade their skills. The first part is generally completed in 2 hours 30 minutes, the second in 1 hour. Based on this experience, we have updated the hints given to ensure that students receive the best possible guidance. For example, we indicate what type of queries are expected (an ASK query is expected for question 7, using a subquery from the previous query in question 10, recalling the DBpedia endpoint in question 20, etc.) so as to encourage the learner to acquire all the targeted skills.

The main guidance we provide is for the first question, which requires familiarisation with the principles of IRIs, prefixes and query patterns. We also have questions on the use of aggregates and group by (questions 9 and 10) and on the use of regex function to manipulate string labels (questions 12 and 22) and the use of bind (question 26). When learners discover how to query DBpedia and Wikidata in part 2, they are often surprised by the complexity of the vocabulary used in these knowledge graphs. Guided by the desire to solve the crime, they quickly become motivated to look at these resources carefully and formulate the queries requested. Most of them use the tools provided on the web querying interfaces of the two endpoints, even though they often have questions about the vocabulary to be used for Wikidata.

Table 2 shows the responses to the questionnaire we sent out to 134 students (from university and engineering school) for the 2024-25 academic year. 102 replied. 94% of them were satisfied by the tutorial and consider that they have acquired skills. Everyone appreciated the fun aspect of the tutorial.

4. OWLuedo

Intended to be carried out after SPARQLuedo, OWLuedo builds on the understanding of the ontology used for the SPARQL tutorial. OWLuedo is designed as a tutorial whose goal is to extend the initial ontology to both increase its expressivity with regard to the complexity of the described murder situation, as well as using the reasonning possibilities of an OWL reasoner to automatically solve the murder once enough knowledge has been described.

4.1. Targeted skills

OWLuedo focuses on the use of OWL to model knowledge, as well as the use of the well-known Protégé[13] editor to achieve it. It also includes an introduction to the use of reasoners and mapping languages to produce RDF datasets based on other file formats. This section will detail in depth the skills that are targeted by the OWLuedo tutorial.

4.1.1. Semantic Web and Knowledge representation skills

The semantic web relies on providing a machine-readable representation of knowledge. As such, ontologies play a key part in it as they provide a way to both standardize representation of facts, and provide an explicit computer-readable semantic for the represented knowledge. Regarding ontology design, the tutorial targets the following skills:

– **OWL.S1.1** Building class and property hierarchies based on a natural language description of the domain characteristics.

[13]https://protege.stanford.edu/

- **OWL.S1.2** Instanciating an ontology by adding individuals and relations between individuals thus creating a knowledge graph.
- **OWL.S1.3** Improving the explicit semantic of the ontology by adding advanced representations of classes and properties (equivalent classes, property chains, etc.).

In addition to ontology design, this tutorial addresses several other critical aspects of the semantic web. More specifically, it puts forward the issue of the coexistence of multiple IRIs describing the same entity in a knowledge graph, the need for mapping languages to automatically and reproducibly convert files into RDF datasets, as well as the main advantages and drawbacks of the Open World Assumption. In practice what is targeted is as follows :

- **OWL.S1.4** Understand the issue of having multiple IRIs describing the same real-world entity.
- **OWL.S1.5** Familiarize with the concept of mapping to produce RDF resources.
- **OWL.S1.6** Understand the Open World Assumption and its consequences.

4.1.2. Technical Skills

In addition to the understanding of how creating ontology and a knowledge graph based on this ontology tie into the many challenges of the semantic web, this tutorial aims to acquire some technical skills, notably regarding the use of software and languages. Specifically, this tutorial tackles how to:

- **OWL.S2.1** Use Protégé to create or edit an OWL ontology.
- **OWL.S2.2** Use the Manchester syntax[14] to write complex OWL axioms.
- **OWL.S2.3** Use an editor for a RDF-generation language (alternatively RMLEditor[15] or SPARQL-Generate[16]) to create a mapping from a .csv file to an RDF dataset.
- **OWL.S2.4** Read and write RDF triples using the Turtle [17] serialization.

4.2. Description of the resource

4.2.1. Provided materials

The tutorial materials are multiple. Regarding the instructions, both the web interface [18] and a PDF file[19] are provided and will guide the learner along the tutorial. In the downloadable materials [20], a turtle file, `OWLuedo.ttl` is included, and should serve as a base for building the ontology. This file is essentially the same as the ontology used for SPARQLuedo, where the object properties characteristics were removed to enable the learners to add them by themselves during the tutorial. In addition, files to add individuals are provided:

- In the best case scenario, the learners should use the RMLEditor to create a mapping and instanciate a graph basing on *.csv* files. Three files are provided: `Item.csv`, `Room.csv` and `Person.csv`, which respectively contain informations about instances of item, room and person.
- As a redundancy due to RMLEditor and SPARQL-Generate's playground being web interfaces that might be down at times, we provided an alternate solution to create the knowledge graph. This solution does not however leverage any mapping technology and simply consists in a bash script that will allow the learner to edit a Turtle file containing the description of the individuals, by replacing the IRIs by those of the entities previously created along the tutorial. This solution revolves around three files: the aforementioned bash script `replaceIRIs.sh`, `entityNames.txt` in which one should replace the indicative entity names by those used in their ontology (non-prefixed names, assuming that all the entities use the same namespace) and `OWLuedo_InstancesEdit.ttl` which contains the RDF descriptions of instances that will be edited to match the devised ontology.

---

[14] https://www.w3.org/TR/owl2-manchester-syntax/
[15] https://rml.io/tools/rmleditor/
[16] https://ci.mines-stetienne.fr/sparql-generate/playground.html
[17] https://www.w3.org/TR/turtle/
[18] https://w3id.org/cluedo4KG/Owluedo
[19] https://www.irit.fr/~Nathalie.Hernandez/cluedo4KG/Owluedo/materiels/OWLuedo_EN.pdf
[20] https://www.irit.fr/~Nathalie.Hernandez/cluedo4KG/Owluedo/materiels/OWLuedo.zip

Table 3

Number of entities by category (**Categories in bold are an estimation based on a possible solution**)

Type	Number of Entities
Place	18
Person	17
Item	29
Individual Total	64
Classes	32
Object Properties	32
Data Properties	3

Table 4

Key figures of the RDF datasets used/produced along the Tutorial

Number of Triples	Initial	Instance file	Example of Solution
Object Property Assertion	0	250	250
Data Property Assertion	0	6	6
Class Assertion	0	64	64
OWL Axiomatization Triples	36	64	504
Annotations	2	17	27
Total	38	401	851

4.2.2. Dataset Key figures

The knowledge graph produced during this tutorial is of moderate size, with a final graph counting several hundreds of triples. Though such an amount does not represent a significant amount with regard to computing, it is significant enough that it might take time for a human to solve the murder by himself, thus highlighting the usefulness of the reasoner, while sticking to an amount that remains manageable for a tutorial. Table 3 describes the number of entities handled during the OWLuedo tutorial. On the other hand, table 4 provides the detail of the triples that are to be created during the tutorial. In both tables, the figures given are calculated on the basis of a possible solution, another solution will likely have a different (though similar) amount of entities and triples.

4.2.3. Tutorial walk-through

The tutorial can be carried out by relying on the web interface which provides step by step instructions. Alternatively, the PDF instructions file is a scarcer way to carry out the tutorial, as it lacks the detailed instructions included in the web interface. As such, it is more suitable for a class with a teacher providing insights as to how to use the various exploited tools.
The tutorial is composed of four steps, that are to be carried out sequentially. Table 5 details in which step the various targeted skills are addressed. Each step is described bellow.

Lightweight ontology design: In the first step, the goal is to create a minimal knowledge modeling, that barely uses the expressivity of OWL. Most of what is represented during this first step could have been done using only RDFS. The learner opens the base ontology file and starts to edit it. As a first sub-step, the class hierarchy will be expanded by adding both super and sub-classes to the existing concepts. The sequence of instructions is designed so that the class hierarchy might have to be reorganised at some point due to additional information, typically to add an intermediary superclass to previously created classes. During this sub-step, instructions will also guide the learner to create alternate labels for an existing class, rather than creating a new equivalent class. The second sub-step is the extension of the existing property hierarchy. Both `ObjectProperties` and `DatatypeProperties` will be added, and it is up to the learner to determine which property should be which. During this sub-step, the domain and range characteristics of the properties will be specified. The final sub-step will consist in creating some individuals and adding some triples linking them using the previously created properties and classes. A case of two individuals being actually the same is included to show how it can be achieved.

Heavyweight ontology design: This second step will consist in improving the semantics of the created classes and properties. The instructions of this step are ordered so that they get more and more difficult to carry out.

Specifically, the first instructions focus on the charasteristics of properties (Functional, Symmetric, etc.), which can easily be added using Protégé. Then come some class disjointness and disjoint unions. Once again, those instructions are rather easy to carry out using the interface of Protégé. Afterwards, learners are required to write some `EquivalentClass` axioms using the Manchester Syntax, which requires to get some grasp of the syntax. The most difficult instructions, which use datatype restrictions come later. Finally, the learner is led to write some property chains. This being done, the instructions lead the learners to add some new facts about individuals, which are meant to be rather ambiguous. While, we previously stated that Room and Floor were disjoint classes, we ask the learner to represent the following facts:

- The greenhouse is a floor.
- The greenhouse contains the room greenhouse.

More often than not, the learners are confused, and represent both greenhouses as the same entity, which leads to an inconsistency when launching/synchronising the reasoner. This is by design, and is meant to highlight the confusions that may come from the ambiguity of natural language, even when creating individuals.

Knowledge graph creation using a mapping language: The individual facts that are described in the first two steps are minimal, as it would be quite daunting and uninteresting to have the learner manually create dozens of entities and hundreds of triples. Thus, in this step, the point is to show the learners a way to quickly populate their ontology and create the knowledge graph using provided resources. The tutorial uses this opportunity to showcase the use of mappings to create RDF triples from other file formats. Several languages for RDF generation have been proposed over the years. [12] proposes an overview of these various languages. Out of the options presented, RML [2] is the only one to propose an online visual editor, the RMLEditor, which is a web interface that allows to generate mappings basing on files such as `.csv` or `.json`. The use of a visual editor aims to avoid having the learners discover and understand a new language, which would greatly impact the time required to complete the practical. Alternatively, SPARQL-based solutions can be used, as the learners having practiced the SPARQLuedo should have a good grasp of SPARQL. The SPARQL-based solution identified in [12] appear to be equivalent in what they allow to do. Once again, the choice boils down to the availability of an online editor, which is only proposed by SPARQL-Generate[10]. Thus, the tutorial guides learners to use alternatively the interfaces for RML or SPARQL-Generate. Using the interface, the learner will create a mapping for each of the provided `.csv` files. Once it is done, they will check the produced result and, when satisfied, will export it as a Turtle file. During this step, an emphasis is put on the importance of checking that the IRIs produced for the entities that were created during the first two steps are the same when those entities are referenced in the mapped dataset. In particular, the subclass of item to which an instance of item pertains is listed in one of the columns of `Item.csv`. This is interesting for two reasons:

- If the mapping is not made carefully, the name of the classes might be erroneous which might prevent the proper execution of the reasoner in the next step.
- It requires the learners to consider the class as any entity, and the `rdf:type` property as any property, which reminds the learners that everything is essentially a triple in RDF, upon which OWL builds to produce its semantics.

Finally, the instructions guide the learner to manually edit the loaded Turtle file and add some triples by typing them, to show how what they do in Protégé can be achieved in any text editor. The content of the produced Turtle is subsequently added to the main ontology file. Alternately, this step can be skipped, for example if the mapping functionality of both editors happen to be down. To populate the ontology, the learners can alternately use the bash script `replaceIRIs.sh`, after having edited `entityNames.txt` to replace the indicative entity names by those used in their ontology (non-prefixed names, assuming the all entities use the same namespace), which will edit `OWLuedo_InstancesEdit.ttl` and produce a Turtle file that will have the proper IRIs and whose content can be directly added to the ontology file.

Reasoning under the Open World Assumption: Finally, the goal is to solve the murder case by launching the reasoner. What the learners have written might seem to them as sufficient to conclude as of who is the murderer. From what the learners wrote, it might be clear to them that there is only be one item that might be the murder weapon (the only described firearm). In this last step, the goal is to show to the learners how the open world

Skill	OWL.S1.1	OWL.S1.2	OWLS1.3	OWL.S1.4	OWL.S1.5	OWL.S1.6	OWL.S2.1	OWL.S2.2	OWL.S2.3	OWL.S2.4
Lightweight ontology design	✓	✓					✓			
Heavyweight ontology design		✓	✓				✓	✓		
KG creation using a mapping language				✓	✓				✓	✓
Reasoning under **the Open World Assumption**			✓			✓	✓	✓		

Table 5

Summary of the skills tackled by each step of the tutorial

assumption impacts the way reasoning is carried out. The goal is then to add other axioms, which are specific to this murder case to more or less "close the world". For example, they are guided to declare that the `Firearm` class is actually an enumeration of only one element. The instructions guide the learners to add three axioms of the world, which, if everything was properly done, should allow them to identify the murderer.

4.3. *Evaluation*

Once again, we evaluate this tutorial over two axes: first, we discuss the coverage of our instructions as compared with the whole expressivity of the addressed languages. Second, we discuss how the tutorial was perceived and carried out by students.

4.3.1. *Specification coverage*

The most prominent goal of this tutorial is to learn to use OWL to create ontologies. Table 6 summarizes the coverage of the OWL specification. It is to be noted that the tutorial does not encompass anything regarding the ontology description-related operators. The category used in the table are based on those proposed in OWL 2's primer[21]. Overall, the tutorial leads to using roughly two thirds of the specification. More specifically, it does not detail the use of the following:

- **Basic Modeling**: the tutorial omits the use of negative property assertions, both `NegativeDataPropertyAssertion` and `NegativeObjectPropertyAssertion`.
- **Advanced Class Relashionships**: there is no use in the tutorial for the `ObjectComplementOf` property. Neither does it require `ObjectAllValuesFrom` as `ObjectSomeValuesFrom` is better suited for what is described. Finally, `ObjectHasSelf` is not used.
- **Advanced Use of Properties**: The tutorial does not make use of all the characteristics of properties. `IrreflexiveObjectProperties` and `InverseFunctionalObjectProperty` are never used. Neither does it make use of the disjointness of properties (`DisjointObjectProperty`) or the key system (`HasKey`).
- **Advanced Use of Datatypes**: The tutorial is not very thorough regarding the use of complex datatypes, and ignores most ways to define them (`DataIntersectionOf`, `DataOneOf`, `DataComplementOf`, `DataUnionOf`).

Still, the tutorial gives a broad overview of what OWL allows to achieve. Out of the unused parts of the OWL vocabulary, it appears that the negative property assertions and the use of keys are what stands out as a potential improvement. Indeed, the other unused OWL operators are rather similar in their use to the ones that are used

[21] https://www.w3.org/TR/owl2-primer/

Table 6
Tutorial coverage of OWL expressivity

Category	Covered vocabulary	Defined vocabulary
Basic Modeling	13	15
Advanced Class Relationships	8	11
Advanced Use of Properties	7	11
Advanced Use of Datatypes	4	8
Total	32	45

in the tutorial. The use of negative property assertions, on the other hand, is not necessarily intuitive, and could help students understand better the open world assumption. As for the keys, they appear as a critical feature to disambiguate entities in the semantic web. The tutorial could thus gain from their integration.
Regarding the use of RML and Turtle, what is primarily amiss is the use of Blank Nodes. The tutorial showcases the use of both datatypes and individuals, the use of some OWL operators as part of triples, as well as how to write multiple triples regarding the same entity in Turtle. However, the use of blank nodes is never made, and could be an improvement.

4.3.2. Practical use and student feedbacks

A version of this tutorial without the RML mapping step has been used for two years with classes of roughly 60 M2 students in computer science, who already took a class explaining the concepts of the semantic web. The tutorial was divided into two 1h45 sessions, for a total of 3h30. For these sessions, the PDF tutorial was used, along with onsite displays of the use of the software provided by the teacher at various points to help the students. From a teacher point of view, the following points should be highlighted:

- Some students (roughly 40%) are very intuitive, and manage to use Protégé without much guidance. They easily grasp the meaning of the instructions and go through the tutorial with ease. Usually, those students manage mostly to complete the first two steps during the first session, and are done before the end of the second.
- Most students (roughly 50%) are able to go through the tutorial with a little help, and complete it in the allotted time.
- Finally, roughly 10% of students have a hard time grasping the concepts, and it is necessary to go more in depth into how to use the software and syntaxes. Those students usually take the whole of the two sessions to complete the first two steps.

The provided figures are based on estimates from teaching experience, and might not be accurate. Overall, the part of the tutorial that appears to be the most confusing for students is the use of the Manchester Syntax. Indeed, while Protégé's interface is mostly intuitive and allows to write OWL axioms in a graphical manner, writing Manchester Syntax axiom requires to understand said syntax, which proves to be a challenge as its official documentation[22] is rather difficult to read and lacks examples. It is often necessary to provide them with syntactic examples resembling what they will want to write, which we do in the web version of the tutorial. Examples are not included in the PDF version, as it is intended for a classroom use, where a teacher can provide said examples.

Feedback from students were collected through the same questionnaire as SPARQLuedo. Table 7 shows the responses to the questionnaire for the 2024-25 academic year. The overall feedbacks are very positive, with students providing a Good or Excellent appreciation on all criterions. There is room for improvement however, with some students indicating they are dissatisfied. The clarity of instructions appears to be the most prominent point of imprevement, with 9 students being unsatisfied with it. Indeed, complementary explanations are often required, particularly in the last part of the tutorial. The acquisition of skills dissatisfaction is harder to tackle.

[22]https://www.w3.org/TR/owl2-manchester-syntax/

Table 7
Feedback from students on OWLuedo during 2024-25 academic year

	bad (0)	very dissatisfied (1)	dissatisfied (2)	satisfied (3)	good (4)	excellent (5)	Total
overall assessment	0	4	5	14	36	43	102
fun aspect	0	0	3	5	39	55	102
clarity of instructions	0	4	5	15	37	41	102
acquisition of skills	0	2	5	14	41	40	102

5. Conclusion

In this article we present new tutorials designed to put into practice the use of SPARQL and OWL to master the creation and the querying of knowledge graphs in a fun way. These resources are integrated into the Cluedo4KG application and can be used to complement lectures or resources presenting these languages. The resources used for the application are available under an open licence and published in accordance with semantic web best practice. They can be used as they are or adapted and integrated into any course. The feedback we get from using these tutorials with our students shows that they are very interesting and motivating. We are currently extending cluedo4KG to include SHACLuedo, a tutorial dedicated to SHACL using the same principles.

References

[1] D. Allemang and J. Hendler, *Semantic web for the working ontologist: effective modeling in RDFS and OWL*, Elsevier, 2011.
[2] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens and R. Van de Walle, RML: A generic language for integrated RDF mappings of heterogeneous data., *Ldow* **1184** (2014).
[3] J. Domingue, D. Fensel and J.A. Hendler, *Handbook of semantic web technologies*, Springer Science & Business Media, 2011.
[4] B. DuCharme, *Learning SPARQL: querying and updating with SPARQL 1.1*, " O'Reilly Media, Inc.", 2013.
[5] J.E.L. Gayo, E. Prud'Hommeaux, I. Boneva and D. Kontokostas, *Validating RDF data*, Morgan & Claypool Publishers, 2017.
[6] J.M. Gomez-Berbis, R. Colomo-Palacios, B. Ruiz-Mezcua and A. Garcia-Crespo, Teaching the semantic web: can semantics do the trick?, *International Journal of Teaching and Case Studies* **1**(4) (2008), 267–274.
[7] A. Hogan and A. Hogan, *Web of data*, Springer, 2020.
[8] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G.D. Melo, C. Gutierrez, S. Kirrane, J.E.L. Gayo, R. Navigli, S. Neumaier et al., Knowledge graphs, *ACM Computing Surveys (Csur)* **54**(4) (2021), 1–37.
[9] V. Klimov, A. Chernyshov, A. Balandina and A. Kostkina, Problems of teaching students to use the featured technologies in the area of semantic web, in: *AIP Conference Proceedings*, Vol. 1797, AIP Publishing, 2017.
[10] M. Lefrançois, A. Zimmermann and N. Bakerally, A SPARQL extension for generating RDF from heterogeneous formats, in: *The Semantic Web: 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28–June 1, 2017, Proceedings, Part I 14*, Springer, 2017, pp. 35–50.
[11] L. Pieschel, S. Welten, L.C. Gleim and S. Decker, Teaching Semantic Web Technologies through Interactive Jupyter Notebooks., in: *SEMANTiCS (Posters & Demos)*, 2021.
[12] D. Van Assche, T. Delva, G. Haesendonck, P. Heyvaert, B. De Meester and A. Dimou, Declarative RDF graph generation from heterogeneous (semi-) structured data: A systematic literature review, *Journal of Web Semantics* **75** (2023), 100753.
[13] N.P. Zlatareva, A Hands-on Project for Teaching Semantic Web Technologies in an Undergraduate AI Course, *Journal of Machine Intelligence and Data Science (JMIDS)* **2** (2021).