An Ontology for Representing Curriculum and Learning Material

Antrea Christou[a], Chris Davis Jaldi[a], Joseph Zalewski[b], Hande Küçük McGinty[b], Pascal Hitzler[b], Cogan Shimizu[a]

[a]*Wright State University, USA*
[b]*Kansas State University, USA*

Abstract

Educational, learning, and training materials have become extremely commonplace across the Internet. Yet, they frequently remain disconnected from each other, fall into platform silos, and so on. One way to overcome this is to provide a mechanism to integrate the material and provide cross-links across topics. In this paper, we present the Curriculum KG Ontology, which we use as a framework for the dense interlinking of educational materials, by first starting with organizational and broad pedagogical principles. We provide a materialized graph for the Prototype Open Knowledge Network use-case, and validate it using competency questions sourced from domain experts and educators.

Keywords: Education, Ontology, Knowledge Graph, OWL

1. Introduction

The Internet and various platforms, from social media (e.g., Medium [24]) to educational sites (e.g., Coursera [9]), have enabled a vast proliferation of educational materials across nearly any conceivable domain. However, this has resulted in various learning materials being scattered across different sites or platforms; becoming difficult to find or access; often redundant (or worse, contradictory), silo-ed away from other complementary material, or may be created once for a narrow use case and never used again. Moreover, educational materials span various formats, and thus multimodality remains a challenge.

Our goal is to provide a mechanism by which educational materials (across various forms and formats, e.g., multimedia) can be densely interlinked, so as to provide seamless progression across learning pathways, irrespective of the source materials' location.

In the end, this is a classic scenario for a data integration task. Of course, Knowledge Graphs (KGs) are well-suited for this task [19, 18, 26]. Specifically, we envision a fully materialized RDF [29] graph equipped with Web Ontology Language (OWL) [17] ontology as a schema, which describes the metadata, content, and links between educational, learning, and training (ELT) material and, more broadly, curricula.

As such, this ontology is one first step towards reasonable personalized instruction, where the needs of the learner can be accounted for, alongside contextualization of each piece of ELT material. Indeed, by using formally structured organization (i.e., the ontology) for the metadata of the ELT material inventories, coupled with its interconnections, we make this an effective retrieval augmented generation (RAG) [23] target for agentic Large Language Models (LLMs) [38, 36]. By emphasizing the agentic nature of these systems (e.g., using SPARQL [30] to retrieve relevant information, which is in turn used to access or otherwise locate specific relevant documents), we furthermore facilitate an *explainable* process, which is paramount in educational scenarios. Thus, in this paper, we lay the foundation for a class of agentic LLM applications, which we would call neurosymbolic pedagogical agents [21]. Specifically and concretely, we cover in this paper our additions to the state of the art:

- the Curriculum KG Ontology, specified in OWL 2 DL [17];
- its materialization, serialized in RDF [29]; and
- its validation and a worked example, via competency questions.

Next, Section 2 describes the ontology for the Cur-

Email addresses: `christou.2@wright.edu` (Antrea Christou), `jaldi.2@wright.edu` (Chris Davis Jaldi), `jzalewski@ksu.edu` (Joseph Zalewski), `hande@ksu.edu` (Hande Küçük McGinty), `hitzler@ksu.edu` (Pascal Hitzler), `cogan.shimizu@wright.edu` (Cogan Shimizu)

riculum KG, with its materialization and evaluation expanded towards Section 3. This is followed by a brief discussion of related work in Section 4, accompanied by a brief critical analysis on how our work advances the state of the art. Finally, in Section 5, we conclude with next steps.

2. The Curriculum KG Ontology

In this section, we discuss the design methodology, overarching use-case, and formalization of the Curriculum KG Ontology. The formalization and accompanying documentation can be found online [10]. The ontology and KG (Section 3) are licensed under a dual license for complete coverage: the ontology, schema diagrams, and the KG itself are under the CC-BY-SA-4.0 license, whereas the materialization scripts are licensed under the Apache 2.0. We note that the materials themselves retain their own licenses (and, indeed, they are modeled – and indicated – as such in KG).

2.1. Use-case scenario

The Prototype Open Knowledge Network (Proto-OKN) [27] is a collaborative program across six U.S. Federal agencies, including the National Science Foundation (NSF[1]), the National Aeronautics and Space Administration (NASA[2]), the National Institutes of Health (NIH[3]), the National Institute of Justice (NIJ[4]), the National Oceanic and Atmospheric Administration (NOAA[5]), and the U.S. Geological Survey (USGS[6]). The purpose is to build *open knowledge network*, defined as "publicly accessible, interconnected sets of data repositories and associated KGs that enable data-driven, artificial intelligence-based solutions for a broad set of societal and economic challenges." The program is organized into 15 projects that solve various domain-specific problems, two projects that provide the common infrastructure that integrates them, and one project that seeks to provide the educational, learning, and training interface to accessing, using, and leveraging the Proto-OKN.

The Curriculum Knowledge Graph (CurrKG) and its ontology directly inform this final project: The Education Gateway for the Proto-OKN (EduGate), and subsequently provides the overarching use-case scenario for the design decisions and competency

questions. For example, the CurrKG could be used to answer such questions as:

- What is the OKN and how can it be used to address climate change?
- What technologies do I need to understand and contribute to the OKN Project X?
- What are all the materials in the CurrKG that explain SPARQL?
- What are all the topics that have the most associated media resources in CurrKG?

While the first two questions are specific to the Proto-OKN, it is the latter two which are of particular interest for broader impacts, including educational initiatives. We provide more detailed questions – and their respective SPARQL queries – in Section 3.2.

In the end, the primary objective, through the design, implementation, and deployment of the CurrKG ontology, will be to provide a model that supports the meaningful discovery of ELT materials and their contexts. Specifically, EduGate – and through the CurrKG and its ontology – seeks to model audience-specific curriculum, which we call a Persona. As such, we see this directly modeled in our ontology, and it becomes a guiding principle for our ontology design decisions.

2.2. Design Methodology

Individualized instruction generally requires significant flexibility. As such, we have chosen the Modular Ontology Design methodology [31].

Modular Ontology Modeling (MOMo) builds complex ontologies out of small, manageable subunits, which we call Modules. A module typically has one core concept and several ancillary concepts and properties conceptually related to its core concept, but may not be directly related to other modules. This segmentation facilitates understanding, future modifications to ontology (as each modification can often be confined to one module, making re-use and adaptation simpler [34], and can help with future (knowledge) alignment activities [4, 33]. During the execution of MOMo for the CurrKG ontology, we created three modules: Persona, Learning Path, and Module.[7]

Often modules are designed by *instantiating*[8] an *ontology design pattern* (ODP) [15], a small generic

---

[7]We accidentally overload this term, but here we mean the concept of *learning* modules – as largely self-contained sets of educational material, and they should not to be confused with ontology modules. These are also called *learning units* in some literature [3].

[8]This is fully known as *template-based instantiation*, from [14].

ontology that reflects a modeling best practice. ODPs are also used in ontology design methodologies besides MOMo [7], and there are several online repositories [1] and libraries [32] of ODPs. Our Persona module is based on the AgentRole pattern, and our Learning Path on the Sequence pattern, from the MODL library [32].

In accordance with the final steps of the MOMo process, we annotated the ontology using the Ontology Pattern Language (OPLa) [16], which makes the division of classes and properties into modules explicit.

2.3. Overview of Concepts and Relations

This section is a non-exhaustive description of the CurrKG ontology; for brevity, we have not specified lists of subclasses, all data properties, or the contents of controlled vocabularies. For complete documentation, see [10]. Our ontology is formalized in the OWL 2 [17, 25], specifically in OWL 2 DL. Below, selected axioms are shown in each respective block of text, expressed using DL notation.

In Figure 1, we provide a **schema diagram** of our ontology. A schema diagram is an intuitive, graphical depiction of how concepts are related in the ontology. For each *head_node-edge-tail_node* in the diagram, these can be minimally construed as representing axioms of the form

$$\text{Head} \sqsubseteq \geq 0 \text{ edge.Tail}$$

which we call a structural tautology [11]. Essentially, it acts as an axiom with no side effects, but helps humans intuit the purpose of a role. The visual syntax is defined as follows. Gold boxes represent classes, purple boxes represent concepts that act as controlled vocabularies[9], yellow ellipses represent data types, closed arrows indicate object properties, open arrows indicate subsumption in the direction of the arrow, and grouped boxes are subclasses.

This framework's key concept is the LearningPath, a dynamic series of LearningSteps developed to accommodate different Personas. These Personas, so far consisting of developers, enthusiasts, executives, and contributors, direct the learning paths' content and structure. Anyone interested identifies a Persona that fits their role and professional experience, and each Persona shapes the architecture of their respective LearningPath.

The FirstLearningStep is the first step of a learning path, which is composed of a number of steps that are connected to one another. The learning process flows continuously since each step is connected by the hasNextLearningStep and hasPreviousLearningStep properties. These steps are all connected to particular Modules that provide organized educational resources. Instead of existing independently, these modules are arranged within the Curriculum, the top entity representing the comprehensive educational framework that offers a structured educational experience.

Multiple important metadata characteristics define each module. These consist of a title, the level of difficulty, and the coversTopic property, which links the module to specific curriculum subjects. The topics themselves are organized semantically, allowing for an elaborate topic structure through the use of hierarchical relationships like broaderThan and narrowerThan. For example, the term "Hydrolysis" is recognized as a more narrow problem within the wider field of "Chemistry." This semantic structure enables the efficient navigation of educational materials.

A Module is a formal unit; it is not identical to the documents, videos, teaching activities, etc. that are used when a student takes the module. These things are to be classified as Media, and a media object can be linked to a Module by the references property.

The Event class is for any educational event that is not directly related to any curriculum, such as an academic conference or workshop. An Event can be a part of a larger event, expressed with the hasSubEvent property. For example, a paper presentation may be part of a conference.

ParticipantRole. Several aspects of the ontology are dedicated to modeling how *people* interact with the curricula and modules: namely, authorship and as a learner. This involvement – or participation – is modeled according to the AgentRole ODP [32].[10] In this modeling paradigm, an agent (and usually in our case, that agent is person) is not simply classified as a student, teacher, or so on, but rather, we record that the person *played the role of* a student or teacher at a certain time, or in connection with a certain event. That is, we disconnect what an agent does (or performs) from what that agent is (or who they are).

---

[9]By this, we mean a class with defined individuals (and no others).

[10]We note that the AgentRole formulation exists in many forms, but we have sourced in particular these patterns from the MODL pattern library.

Of particular importance for us is the contextualization of that ParticipantRole. Traditionally, the contextualization for a role is the spatiotemporal extent (i.e., an anchoring in space and time). However, for us, we are interested at a more abstract level: "Who is engaging with specific learning material?" and "Who authored a certain piece of material." While the date of authorship is important for understanding the usefulness of material (i.e., perhaps in relation to the state of the art), these details are more aptly modeled as part of the media in question.

Thus, we focus on providesParticipantRole. In our ontology, this is obfuscated however, essentially by focusing on the nature of the contextualization: determination of a suitable learning path and authorship. We note that an individual of type ParticipantRole is not something generic, like 'teacher'. Instead, it represents a specific instance of teaching carried out by one person in one context. For generic roles, we chose to represent them in our ontology as *subclasses*: Persona and Author – and their subclasses, in turn!

Curriculum. A Curriculum is a course of study, by which we mean some set of educational material that has been curated for a certain purpose. In higher education, curricula might be described by a syllabus, which includes necessary metadata for engaging with the material: sources (e.g., textbook(s), topics, and a schedule. A prepared course may have learning outcomes and objectives. For now, we largely focus on the topics and the order in which they should be ingested. For the former, these are modeled as Modules and, for the latter, they are modeled as LearningPaths. Modules are organized as bags, which is sufficient for answering questions such as "What topics does this curriculum cover?" Axiomatically, we dictate that the Curriculum should be titled, and that (we argue) a curriculum is pointless – and not much of a curriculum at all – if there are no modules (and thus topics) that it addresses. While this is a minimal specification, it's important as it acts as a top-level organizational mechanism.

$$\text{Curriculum} \sqsubseteq \exists\text{hasTitle.xsd:string} \quad (1)$$

$$\text{Curriculum} \sqsubseteq \exists\text{hasModule.Module} \quad (2)$$

1. Every Curriculum has a title represented as a string.
2. Every Curriculum has at least one Module.

Person & Authorship. A Person can assume different roles, such as an Author, which would be responsible for generating content, or taking on a Persona, which represents an abstracted perspective or set of requirements for an individual. Instead of being only labels, these roles serve to distinguish the kind of contributions or actions that are linked to the person. One may ask, for instance, "Who developed this module?" or "For whom is this series of topics intended?" Axiomatically, each Author must be a Person, and we consider that each individual's learning path is a reflection of their personal goals.

$$\exists\text{assumesAuthorship.Author} \sqsubseteq \text{Person} \quad (3)$$

$$\exists\text{assumesPersona.Persona} \sqsubseteq \text{Person} \quad (4)$$

$$\text{Author} \sqsubseteq \exists\text{hasName.xsd:string} \quad (5)$$

3. There exists a Person who assumes the role that is an Author.
4. There exists a Person who assumes a role that is a Persona.
5. Every author has some name and that is represented as a string.

LearningPath. A LearningPath is a series of learning steps an individual has to go through in order to follow a certain Curriculum to satisfy a personal educational goal. Just like attending a University class, a certain goal is set for the students that attend, requiring them to go through a set of material in order (path) and therefore accomplish that said goal. It would satisfy the question of "What Modules do I need to go through to get familiar with Knowledge Graphs in a superficial manner and in what order?" That path contains certain Modules in sequence, noting that the given Persona will have to complete each one in order to complete their learning objective.

We abbreviate hasLearningStep as hLS.

$$\text{LearningPath} \sqsubseteq \exists\text{scopedBy.Curriculum} \quad (6)$$

$$\text{LearningPath} \sqsubseteq \exists\text{hLS.LearningStep} \quad (7)$$

$$\text{LearningPath} \sqsubseteq \exists\text{determines}^-.\text{Curriculum} \quad (8)$$

6. Every Learning Path is scoped by a Curriculum.
7. Every Learning Path has at least one Learning Step.
8. Every Learning Path is determined by at least one Curriculum.

LearningStep. A LearningStep is an individual step in a LearningPath. This sequence is used to order the Modules in a certain Curriculum, as determined by the relevant Persona. LearningSteps are ordered to provide a mechanism for curation (i.e., to enable an instructor to specify the exact approach), as well as to otherwise encode prerequisite. In some sense, but which we do not state

explicitly, a Module plays the role of a Learning-Step, meaning that it is a standalone piece of material, and the context is the order of the Learning-Path. The LearningPath is based off the *Semantic Trajectory* [32] pattern and, as such, includes a FirstLearningStep and LastLearningStep. This concept in particular enables answering simple CQs such as, "As a developer who wants to work with knowledge graphs in a hands-on way, what module should I go through after 'What is an Ontology?'"

In the axiomatization below, we abbreviate has-NextLearningStep as hNLS (similarly for "previous"), LS for LearningStep (and prepend F or L for first or last, respectively).

$$\top \sqsubseteq \leq 1 \text{ hNLS}.\top \qquad (9)$$

$$\top \sqsubseteq \leq 1 \text{ hPLS}.\top \qquad (10)$$

$$\text{hPLS.LearningStep} \sqsubseteq \neg\text{FLS} \qquad (11)$$

$$\text{hNLS.LearningStep} \sqsubseteq \neg\text{LLS} \qquad (12)$$

$$\text{LearningStep} \sqsubseteq =1 \text{ refersTo.Module} \qquad (13)$$

9. Every learning step has exactly one or no next learning step.
10. Every learning step has exactly one or no previous learning step.
11. If a learning step has a previous learning step, then it is not the first learning step in a path.
12. If a learning step has a next learning step, then it is not the last learning step in a path.
13. Every learning step refers to exactly one module.

Module. A Module covers one or more Topics, has a title to identify it, is assigned a Level to indicate its difficulty, belongs to a specific Category for organization, and may reference relevant Media to support learning. A Module represents an educational material that covers a specific Topic. For example, an article titled "100+ Data Science Projects in Python for Beginners" would represent a Module, and the topic of it would be categorized as "Coding". It can be categorized into levels, determining a "difficulty" measure an audience member can use to satisfy a certain objective while also being prompted to a Media reference such as a book or a video.

$$\text{Module} \sqsubseteq \exists\text{coversTopic.Topic} \qquad (14)$$

$$\text{Module} \sqsubseteq \exists\text{hasTitle.xsd:string} \qquad (15)$$

$$\text{Module} \sqsubseteq \exists\text{hasLevel.Level} \qquad (16)$$

$$\text{Module} \sqsubseteq \exists\text{belongsTo.Category} \qquad (17)$$

$$\text{Module} \sqsubseteq \exists\text{references.Media} \qquad (18)$$

14. Every Module always covers a Topic.

15. Every Module always has a title as a string.
16. Every Module always has a level.
17. Every Module always belongs to a Category.
18. Every Module always references some Media.

Category. A Category is a subject-matter grouping that organizes relevant modules in a curriculum, helping to meaningfully and conveniently structure the learning content. At least one category must be covered in each module, which gives a foundation for the types of topics covered. We have initially identified four categories; they include Foundation, which provide fundamental ideas and principles; Survey, which covers broad descriptions of a topic or field; Methodology, which describe tools and procedures; and Standard, which offer formal specifications and recommendations. We specifically differentiate from Topic for a more connotative approach. In this case, a Category is a Topic in the broadest sense. Of course, a Topic might cover what a "standard" is, but in this case, we are more interested in the fact that the Topic covering, e.g., OWL 2, is about a Standard. Axiomatically, these are very simple. We do not make any claims to disjointness, and merely require that any particular Category be populated by at least one Module.

$$\text{Category} \sqsubseteq \exists\text{hasModule}^-.\text{Module} \qquad (19)$$

19. Every category has some module.

Event. An Event, within the scope of our ontology, is a structured learning activity that can take several forms, including workshops, tutorials, and presentations, each of which is intended to provide a specific learning modality. Sub-events are related activities that can be included or are part of Events to allow for a more granular organization of sessions and content. Events can offer access to or otherwise provide Media (e.g, podcasts, articles, videos, and transcripts) that supplement in-person interactions. Events are frequently tied to the modules they cover. "What type of media is being provided by this Event?" is a question that can be answered using the above framework. From [35], we note that the subeventOf property is effectively a subproperty of po-feature, meaning that we also have meronymous transitivity.

$$\text{Event} \sqsubseteq \geq 0 \text{ hasSubEvent.Event} \qquad (20)$$

$$\text{Event} \sqsubseteq \exists\text{provides.Media} \qquad (21)$$

20. Every event has zero or more sub-events.
21. Every event provides some Media.

Persona. We established a set of Personas to guide progress and evaluate ease of use, covering a range of different kinds of user types, including Developers, Instructors, Analysts, Executives, and Graduate Students. By following a customized learning path, each persona allows us to make sure the ontology supports a wide range of technical backgrounds and goals. Respectively, we developed custom Learning Paths for them, including Modules and materials needed for them to go through in order to achieve their separate educational goals. We have specified these strict cardinalities, per below, as each Persona is very granular (i.e., finely and exactly tailored).

$$\text{Persona} \sqsubseteq =1\text{hasProfession.Profession} \quad (22)$$
$$\text{Persona} \sqsubseteq \exists\text{hasType.PersonaType} \quad (23)$$
$$\text{Persona} \sqsubseteq =1\text{ determines.LearningPath} \quad (24)$$
$$\text{Persona} \sqsubseteq \exists\text{assumesPersona}^-.\text{Person} \quad (25)$$

22. Every persona has exactly one profession.
23. Every persona has a type and that type is a persona type.
24. Every persona determines exactly one learning path.
25. Every persona is assumed by at least one person.

Topic. A Topic helps to clarify the main idea covered in a module and assists recognize its purpose. Every topic has a title, and they all act as important content organizers. Because the subjects are arranged in a hierarchy, learners can examine how ideas relate to one another at various levels of complexity. A topic may have both more broad and more narrow related topics. The query, "What are the more advanced topics I should explore after learning about RDF?" is an instance that this model can properly answer.

$$\text{Topic} \sqsubseteq =1\text{asStringxsd:String} \quad (26)$$

26. Every topic is represented by exactly one string value.

Controlled Vocabularies. The terms PersonaType, Level, Audience, and Language are all controlled vocabularies. This means that the class consists of exactly only the individuals specified. This makes use of the **Explicit Typing** ODP [32], and allows for quick and easy modifications to the ontology without perturbing the overall subsumption hierarchy.

3. The Curriculum Knowledge Graph

In this section, we describe the construction and materialization process of the Curriculum Knowledge Graph (CurrKG) by showcasing how structured curriculum data, which is stored in tables (e.g., CSV files), is turned into a KG using RDF triples. We also present an evaluation using real-world use cases to demonstrate its validity. We utilize competency questions (CQs) to validate the CurrKG and thus demonstrate how the graph effectively represents the required information.

3.1. Materialization

The materialization process consists of translating structured tabular curriculum data into triples conforming to the ontology. While ontology itself provides the definition for conceptual entity models like Module, Media, and Persona (discussed in Section 2), the instantiation of these concepts into an RDF graph is achieved using a flexible, code-driven pipeline.

The data used for materialization is drawn from a variety of open and curated sources. It majorly includes The Knowledge Graphs Conference and Community's Open Knowledge Graph Curriculum (Open KGC)[11], data from National Student Data Corps (NSDC)[12] – particularly the Ontology Flash Card Series and other consolidated curriculum-related datasets, as well as some internally curated data from our Education Gateway (EduGate) project[13]. This includes our detailed persona information and learning path structures. All of these datasets collectively are rich and diverse enough to generate instances for key concepts and relationships defined in our ontology.

In abstract, the construction pipeline starts with data ingestion, specifically by inputting a UTF-8 formatted CSV file where each row contains curriculum data entry. These entries are iteratively processed, enabling efficient row-wise iteration and data manipulation. The column headers in the CSV serve as signals for the types of entities and properties to expect in the Python script. We then proceed with the initialization of an RDF graph, using the RDFLib library [28], with a set of our predefined namespaces for consistent URI and vocabulary, either by loading an existing graph or by creating one from scratch as we proceed from ingestion to triplification. Triplification consists of constructing

---

[11]https://github.com/KGConf/
open-kg-curriculum
[12]https://nebigdatahub.org/nsdc/
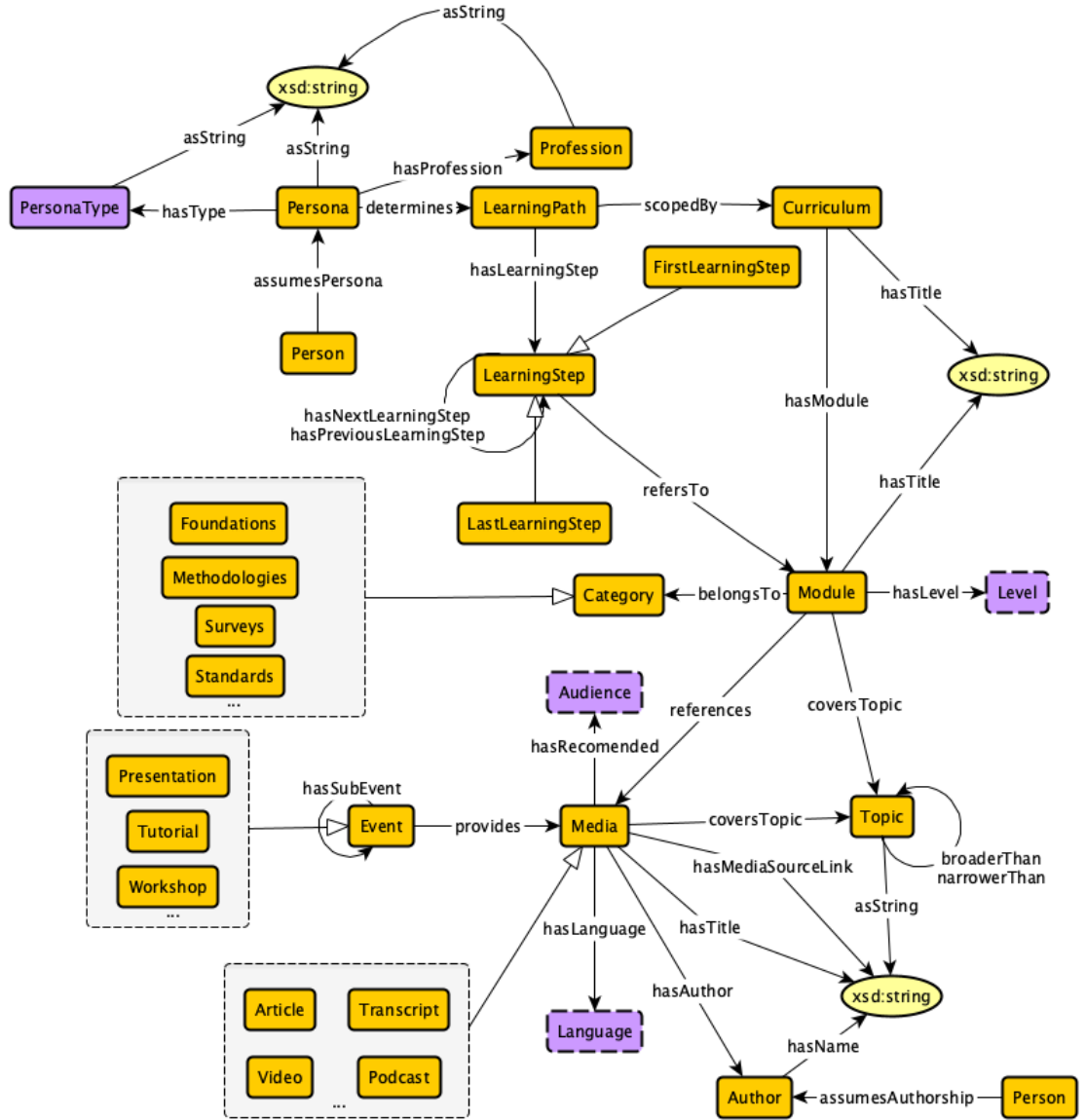[13]https://edugate.cs.wright.edu/

Figure 1: The schema diagram for the overall Curriculum KG Ontology. The yellow boxes indicate the classes, the purple ones represent the controlled vocabulary and the yellow ellipses indicate the datatypes tied to those classes. Open arrows indicate SubclassOf relations.

RDF triples based on the presence and completeness of specific fields in each row of the input data, i.e., whenever a field is missing or empty, the script skips triple creation for that field but continues processing the remainder of the row, preserving semantic structure where available. For example, if both 'Module Title' and 'Curriculum' are present in a row, all related triples

```
(Curriculum_X, hasTitle, Curriculum_Y)
(Module_Y, hasTitle, Module Title)
(Curriculum_X, hasModule, Module_Y)
```

are generated. When minting a URI of an entity, we use its identifier in the URI construction by first sanitizing it – the process of removing all special characters and whitespace replaced by underscores – and also tagging some available metadata. Finally, we serialize the newly constructed RDF graph into a format such as Turtle (.ttl) [6], which serves as a materialized KG instance of the CurrKG ontology.

Our materialization pipeline and script are adaptable in nature and are not hard-coded to a specific data format or schema. It dynamically responds to the input data structure, handling missing values gracefully and allowing for partial population of entities by instantiating all available individuals of their corresponding ontology classes, even when the data is incomplete. For instance, for classes such as Module, Media, and LearningPath, it creates schema-specific semantic links between individuals, such as hasModule or references and correctly types individuals entities using ontology-defined properties, such as hasTitle or asString as appropriate (see Figure 1 for details). As long as new data follows the expected column headers that correspond to known field mappings, the script can generate all possible RDF triples and materialize a KG instance without modification. This allows CurrKG to be adapted to new datasets, use cases, or domains with minimal engineering effort. The working script and data are available in the CurrKG repository [10].

3.2. Evaluation

We evaluate the ontology and KG materialized from it via a diverse set of competency questions (CQs) that cover real-world needs and assess various characteristics of the ontology's quality, including completeness and expressiveness. CQs are natural language queries representing the kinds of questions a KG should be able to answer. They serve as a practical benchmark for assessing the graph or ontology and its alignment with the intended domain [22].

PREFIX	URI
edu-r:	<https://edugate.cs.wright.edu/lod/resource/>
edu-ont:	<https://edugate.cs.wright.edu/lod/ontology/>
rdf:	<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
rdfs:	<http://www.w3.org/2000/01/rdf-schema#>

Table 1: Set of Prefixes used in CurrKG

We then expressed these questions as SPARQL queries [8], where we focused on efficiency – but do not make any guarantees on optimality – to be executed over the graph using Apache Jena Fuseki Triplestore [12]. For instance, below are some of the example competency question-query sets used for evaluation; their respective prefixes are listed in Table 1.

CQ1: Which persona is associated with which learning path, and what are the learning steps within that path?

```
SELECT ?persona ?personaName
?learningPath ?learningStep
?learningStepName ?prevStep
?prevStepName ?nextStep
?nextStepName

WHERE {
 ?persona a edu-ont:Persona ;
  edu-ont:determines
            ?learningPath ;
  edu-ont:asString
            ?personaName.
 ?learningPath
      edu-ont:hasLearningSteps
            ?learningStep .
 ?learningStep
      edu-ont:asString
            ?learningStepName.
 OPTIONAL {
  ?learningStep
    edu-ont:hasPreviousLearningStep
            ?prevStep .
  ?prevStep
    edu-ont:asString
            ?prevStepName.
 }

 OPTIONAL {
  ?learningStep
    edu-ont:hasNextLearningStep
            ?nextStep .
  ?nextStep edu-ont:asString
            ?nextStepName.
 }

}
```

This query retrieves all the available Persona, the Learning Path it determines, its corresponding Learning Steps, along with its respective previous and next learning steps if they exist.

CQ2: Which authors have contributed to multiple media resources?

```
SELECT ?author ?authorName
(COUNT(?media) AS ?count)
```

```
WHERE {
 ?media rdf:type edu-ont:Media .
 ?media edu-ont:hasAuthor
            ?author .
 ?author edu-ont:hasName
            ?authorName .
}

GROUP BY ?author ?authorName
HAVING (COUNT(?media) > 1)
ORDER BY DESC(?count)
```

This query retrieves all the available Authors in our KG who have authored more than one Media resource.

CQ3: What topics have the most associated media resources?

```
SELECT ?topic ?topicName
(COUNT(?media) AS ?mediaCount)

WHERE {
 ?media rdf:type edu-ont:Media .
 ?media edu-ont:coversTopic
            ?topic .
 ?topic edu-ont:asString
            ?topicName .
}

GROUP BY ?topic ?topicName
ORDER BY DESC(?mediaCount)
LIMIT 10
```

This query retrieves the top 10 Topics available in our KG are most frequently covered across all available Media.

CQ4: How many modules belong to each category?

```
SELECT ?categoryName
(COUNT(?module) AS ?moduleCount)

WHERE {
 ?module a edu-ont:Module ;
    edu-ont:belongsToCategory
            ?category .
 ?category edu-ont:asString
            ?categoryName .
}

GROUP BY ?categoryName
ORDER BY DESC(?moduleCount)
```

This query retrieves the count of all available Modules and the Category they belong to.

CQ5: What are the top 10 most referenced media resources?

```
SELECT ?media ?mediaTitle
(COUNT(?referencingEntity) AS
            ?referenceCount)

WHERE {
 ?referencingEntity
    edu-ont:references ?media .
 ?media edu-ont:hasTitle
            ?mediaTitle.
}

GROUP BY ?media ?mediaTitle
ORDER BY DESC(?referenceCount)
LIMIT 10
```

This query retrieves the top 10 Media that have been referenced

the most.

Rest of the CQs and their respective SPARQL queries used for evaluation are available in the git repository[14]. This approach of executing the SPARQL queries to answer the CQs over materialized CurrKG helped confirm that the ontology sufficiently captured the semantics, hierarchies, and properties of the educational material, and all the entities and relationships were correctly instantiated in materialization [37, 5].

We did observe some limitations when the data was incomplete or when key properties (e.g., Event information) were missing, but the ontology and materialization pipeline's flexibility still allowed for meaningful results to be returned whenever available. This CQ-based evaluation demonstrates and validates the effectiveness of CurrKG's ontology design and its practical implementation in materialization workflow.

4. Related Work

In this section, important applicable work that offers basic principles of the use of KGs to the organization and customization of educational materials is addressed. We mention these materials because they support our objective to improve learning through an ontology-based approach. To develop a more adaptable, scalable framework that is customized to each learner's distinct profile, we expand on the approaches used in such studies by examining how they handle entity linking, curriculum representation, and customized learning paths.

A systematic literature review of knowledge graph construction and application in education

KG construction techniques and their educational applications are carefully reviewed in the systematic review of the literature by Abu-Saliha and Alotaibi [2]. Using a structured evaluation approach, their work creates a collection of knowledge about KG development, integration, and application in educational environments. Through a review of various approaches to entity linking, knowledge representation, and ontology design, their study identifies important frameworks that improve the organization of educational content.

This review provides helpful details for our ontology-driven KG building, especially when it comes to identifying a structured schema for learning materials and curriculum representation. Their

---

focus on automated knowledge extraction techniques and semantic connections is important to note because it guides our strategy to enhance scalability and flexible learning paths in our KG architecture. In utilizing their research, we improve individualized educational environments by refining our ontology to be more responsive to educational concepts and enable more effective knowledge access.

Towards a Teaching Knowledge Graph for Knowledge Graphs Education

This work describes a structured framework created especially to enable KG education by integrating key educational components like skills, subjects, courses, instructors, and resources [20]. Their approach is a useful tool for structuring and organizing educational material since it makes use of a higher-order ontology with semantic constraints to guarantee consistency and accessibility. This format makes it possible for learners to access the content in a logical, sequential way, helping to improve understanding and navigation.In contrast to Ilkou's work, which focuses on creating a knowledge graph for learning, our method specifically addresses the difficulties associated with scattered, multimodal instructional resources on many platforms. In order to facilitate smooth progress across educational paths tailored to individual needs, we developed personas and the Curriculum KG ontology, which is linked together and incorporates resources from multiple sources.

Our work herein was developed in parallel with Ilkou et al. In fact, we were able to cross-pollinate in some cases, leveraging shared experiences. In particular, we build on this approach by modeling individualized learning paths with respect to the personas directly within the KG, capturing unique goals, prior knowledge, and skills to enable truly personalized learning experiences. This enables users with different backgrounds, skill levels, and learning goals to receive tailored educational resources. It improves flexibility and customization, transforming our KG into a dynamic and customized learning assistant in addition to a structured knowledge source.

Knowledge Graph-Based Teacher Support for Learning Material Authoring

Grévisse et al. [13] describe a system in Knowledge Graph-Based Teacher Support for Learning Material Authoring that uses KGs to assist teachers create educational resources. Their approach,

SoLeMiO, uses accessible KGs to create a structured semantic representation while integrating semantic technologies to extract key ideas from lesson plans. In addition to providing semi-automatic classifying to improve reusability across digital learning contexts, this allows the system to identify relevant resources from digital libraries and MOOC platforms. Using current KGs, their strategy mainly aims to enhance educators' content organization and discovery.

To customize educational paths, our work builds on this base but takes a different approach by integrating unique learner personas. Although Grévisse et al. focus on improving the creation process for educators, we broaden the application of KGs to tailor learning outcomes according to the specific needs of the user. To ensure that each individual receives material that is in line with their background, objectives, and past knowledge, we model numerous learner profiles within the KG and offer personalized suggestions and flexible learning paths. This distinction makes KGs more dynamic and learner-centric, allowing us to progress beyond content organization topic of personalized learning.

Computer Science Curricula

Developed by the Accocication for Computing Machinery (ACM), the IEEE Computer Society (IEEE-CS) and the Association for Advancement of Artificial Intelligence (AAAI), it presents an elaborate model and taxonomy for teaching computer science at the undergraduate level. It organizes the domain into clearly defined Knowledge Areas (KAs), each of which consists of Knowledge Units (KUs) and related Learning Outcomes (LOs) that are arranged according to the levels of intellectual difficulty. Institutions can adapt programs to local needs while keeping them consistent with international standards thanks to this structured approach, which makes it easier to create a modular and competency-based curriculum. By establishing the connections between these taxonomic features and coordinating them with personalities, learning paths, and instructional materials, our Curriculum KG Ontology directly expands upon this framework. We offer an intuitive foundation for dynamically navigating educational content by treating Knowledge Areas and Learning Outcomes as linked graph elements [3].

5. Conclusion

Ultimately, we have defined an ontology for educational materials and materialized a KG with some available content, using the ontology as a schema.

By utilizing the concept of *Personas*, we are able to provide customizable learning paths for each member of a specific audience that wishes to get involved with knowledge of this nature. This enables scenarios where individual may learn at their own pace, with respect to their own goals and objectives.

Carefully arranged ontologies like this one allows for further developments and additions with respect to classes, properties, and data. Modularity and reusability are key features that add on to the dynamic nature of the work. Hence, future steps include:

- incorporating the ontology to the features of interactive tools such as the Interactive Knowledge (InK) Browser[39]. Serving as an interactive educational platform for the audience members (Personas) to learn according to their personal goals.
- Continue to expand the KG with more Personas and Modules that arise throughout our efforts.
- Utilizing the organized taxonomies provided in the CS 2023 report as an example for integrating assessment features straight into the Curriculum KG, with special attention to the definition of KAs, KUs, and LOs. Through the compatibility of our ontology with the hierarchical model and complexity-based learning objectives of CS 2023, we hope to facilitate individualized evaluation of learning and automated progress tracking. Workflows for competency-based evaluations will be made possible as a result, assessing comprehension according to semantic approach for ideal results.

References

[1] Association for ontology design patterns. `https://odpa.github.io/`. Accessed: 10 May 2025.

[2] B. Abu-Salih and S. Alotaibi. A systematic literature review of knowledge graph construction and application in education. *Heliyon*, 10(3), 2024.

[3] ACM, IEEE-CS, and AAAI. Computer science curricula 2023: Curriculum guidelines for undergraduate degree programs in computer science, 2023. Accessed April 12, 2025.

[4] R. Amini, S. S. Norouzi, P. Hitzler, and R. Amini. Towards complex ontology alignment using large language models. In S. Tiwari, B. Villazón-Terrazas, F. Ortiz-Rodríguez, and S. Sahri, editors, *Knowledge Graphs and Semantic Web - 6th International Conference, KGSWC 2024, Paris, France, December 11-13, 2024, Proceedings*, volume 15459 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2024.

[5] N. Y. Ayadi, S. Bernard, R. Bossy, M. Courtin, B. G. H. Happi, P. Larmande, F. Michel, C. Nedellec, C. Roussey, and C. Faron. A unified approach to publish semantic annotations of agricultural documents as knowledge graphs. *Smart Agricultural Technology*, 8:100484, 2024.

[6] D. Beckett, T. Berners-Lee, E. Prud'hommeaux, and G. Carothers, feb 2014. W3C recommendation.

[7] E. Blomqvist, K. Hammar, and V. Presutti. Engineering Ontologies with Patterns – The eXtreme Design Methodology. In P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, and V. Presutti, editors, *Ontology Engineering with Ontology Design Patterns – Foundations and Applications*, volume 25 of *Studies on the Semantic Web*, pages 23–50. IOS Press, 2016.

[8] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, and G. XIao. Ontop: Answering SPARQL queries over relational databases. *Semantic Web*, 2017. To appear, available from http://www.semantic-web-journal.net.

[9] Coursera. Coursera — Degrees, Certificates, & Free Online Courses. `https://www.coursera.org/`. Accessed: 2025-04-10.

[10] Curriculum knowledge graph repository. https://github.com/kastle-lab/curriculum-kg.

[11] A. Eberhart, C. Shimizu, S. Chowdhury, M. K. Sarker, and P. Hitzler. Expressibility of owl axioms with patterns. In *The Semantic Web: 18th International Conference, ESWC 2021, Virtual Event, June 6–10, 2021, Proceedings*, page 230–245, Berlin, Heidelberg, 2021. Springer-Verlag.

[12] A. S. Foundation. `https://jena.apache.org/documentation/fuseki2/`. accessed: 10 May 2025.

[13] C. Grévisse, R. Manrique, O. Marino, and S. Rothkugel. *Knowledge Graph-Based Teacher Support for Learning Material Authoring*, pages 177–191. Springer, 10 2018.

[14] K. Hammar and V. Presutti. *Template-based content ODP instantiation*, pages 1–13. IOS Press, 12 2017.

[15] P. Hitzler, A. Gangemi, K. Janowicz, A. Krisnadhi, and V. Presutti, editors. *Ontology Engineering with Ontology Design Patterns – Foundations and Applications*, volume 25 of *Studies on the Semantic Web*. IOS Press, 2016.

[16] P. Hitzler, A. Gangemi, K. Janowicz, A. A. Krisnadhi, and V. Presutti. Towards a simple but useful ontology design pattern representation language. In E. Blomqvist, Ó. Corcho, M. Horridge, D. Carral, and R. Hoekstra, editors, *Proceedings of the 8th Workshop on Ontology Design and Patterns (WOP 2017) co-located with the 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21, 2017.*, volume 2043 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.

[17] P. Hitzler, M. Krötzsch, B. Parsia, P. Patel-Schneider, and S. Rudolph, editors. *OWL 2 Web Ontology Language: Primer (Second Edition)*. W3C Recommendation, 11 December 2012. Available at http://www.w3.org/TR/owl2-primer/.

[18] P. Hitzler, M. Krötzsch, and S. Rudolph. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 2010.

[19] A. Hogan, E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, and A. Zimmermann. Knowledge graphs. *ACM Comput. Surv.*, 54(4), July 2021.

[20] E. Ilkou, H. Abu-Rasheed, D. Chaves-Fraga, and E. Engel-brecht. Teaching knowledge graph for knowledge graphs education. Under review in the Semantic Web Journal, 2024.

[21] C. D. Jaldi, E. Ilkou, N. Schroeder, and C. Shimizu. Education in the era of neurosymbolic ai. *Journal of Web Semantics*, 85:100857, 2025.

[22] C. M. Keet and Z. C. Khan. On the roles of competency questions in ontology engineering. In M. Alam, M. Rospocher, M. van Erp, L. Hollink, and G. A. Gesese, editors, *Knowledge Engineering and Knowledge Management*, pages 123–132, Cham, 2025. Springer Nature Switzerland.

[23] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[24] Medium. Medium — Read and Write Stories. `https://medium.com/`. Accessed: 2025-04-10.

[25] B. Motik, P. F. Patel-Schneider, and B. Cuenca Grau, editors. *OWL 2 Web Ontology Language: Direct Semantics*. W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-direct-semantics/.

[26] N. Noy, Y. Gao, A. Jain, A. Narayanan, A. Patterson, and J. Taylor. Industry-scale knowledge graphs: lessons and challenges. *Commun. ACM*, 62(8):36–43, July 2019.

[27] Proto-OKN. Proto-OKN — NSF Proto Open Knowledge Networks. `https://www.proto-okn.net/`. Accessed: 2025-04-10.

[28] RDFLib Team. RDFLib — Python library for working with RDF. `https://rdflib.readthedocs.io/`. Accessed: 2025-04-11.

[29] G. Schreiber, Y. Raimond, F. Manola, E. Miller, and B. McBride, jun 2014.

[30] A. Seaborne and E. Prud'hommeaux. SPARQL query language for RDF. 2008. http://www.w3.org/TR/rdf-sparql-query/.

[31] C. Shimizu, K. Hammar, and P. Hitzler. Modular ontology modeling. *Semantic Web*, 14(3):459–489, 2023.

[32] C. Shimizu, Q. Hirt, and P. Hitzler. Modl: A modular ontology design library. In *WOP@ISWC*, 2019.

[33] C. Shimizu and P. Hitzler. Accelerating knowledge graph and ontology engineering with large language models. *J. Web Semant.*, 85:100862, 2025.

[34] C. Shimizu, P. Hitzler, Q. Hirt, D. Rehberger, S. G. Estrecha, C. Foley, A. M. Sheill, W. Hawthorne, J. Mixter, E. Watrall, R. Carty, and D. Tarr. The enslaved ontology: Peoples of the historic slave trade. *Journal of Web Semantics*, 63:100567, 2020.

[35] C. Shimizu, P. Hitzler, and C. Paul. Ontology design patterns for winston's taxonomy of part-whole relations. In *Emerging Topics in Semantic Technologies*, pages 119–129. IOS Press, 2018.

[36] A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei. Agentic retrieval-augmented generation: A survey on agentic rag. *arXiv preprint arXiv:2501.09136*, 2025.

[37] R. Tahsin, Y. Li, M. S. Abolhasani, and F. Ameri. Generation of semantic knowledge graphs from maintenance work orders data. *J. Maintenance Engineering*, 11(2):45–60, 2024.

[38] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[39] J. Zalewski, L. Zhou, C. Shimizu, and P. Hitzler. Ink browser - the interactive knowledge browser. In O. Seneviratne, C. Pesquita, J. Sequeda, and L. Etcheverry, editors, *Proceedings of the ISWC 2021 Posters, Demos and Industry Tracks: From Novel Ideas to Industrial Practice co-located with 20th International Semantic Web Conference (ISWC 2021), Virtual Conference, October 24-28, 2021*, volume 2980 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2021.