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 [19]) to educational sites (e.g., Coursera [7]), have enabled a vast proliferation of educational materials across nearly any conceivable domain. However, this has resulted in the various learning material being scattered across different sites or platforms; difficult to find or access; 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 becomes a new challenge.

Our goal is to provide a mechanism by which educational materials (across these 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 are well-suited for this task [14, 13, 21]. Specifically, we envision a fully materialized RDF [24] graph equipped with an OWL [12] 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, will make this an effective retrieval augmented generation (RAG) [18] target for Agentic LLMs []. By emphasizing the agentic nature of these systems (e.g., using SPARQL [25] to retrieve 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 [16]. Specifically and concretely, we cover in this paper our additions to the state of the art:

- the Curriculum KG Ontology, specified in OWL 2 [12];
- its materialization, serialized in RDF [24]; and
- its validation, via competency questions.

The next section describes the ontology for the Curriculum KG. In Section 3, we present our validation of the ontology, primarily through the materialized graph and evaluating against specific competency questions. This is followed by a brief discussion of related work in Section 4. Finally, in Section 5, we

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)

conclude.

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 [8]. The ontology and KG (Section 3) are licensed under the license.

Make sure that this is true.

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) [22] 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*, which is "publicly accessible, interconnected sets of data repositories and associated knowledge graphs 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 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 OKN Project X?
- What are all 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 educational initiatives. We provide more detailed questions – and their answers! – in Section **??**.

In the end, the primary objective, through the design, implementation, and deployment of the CurrKG Ontology, will be 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 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 [26].

Modular Ontology Modeling (MOMo) builds up complex ontologies out of small, manageable ontologies called 'modules.' A module typically has one core concept, and several ancillary concepts and properties conceptually related to its core concept, but not directly to other modules. This allows for greater ease of understanding, and if there is a need to modify the ontology, each modification can often be confined to one module, making re-use and adaptation simpler. The three modules in our ontology have core concepts Persona, Learning Path, and Module (the concept of learning modules, not to be confused with ontology modules).

Often modules are designed by *instantiating* an *ontology design pattern* (ODP) [10], a small generic ontology that reflects a modeling best practice. ODPs are also used in ontology design methodologies besides MOMo [5], and there are several online repositories [1] and libraries [27] of ODPs. Our Persona module is based on the AgentRole pattern, and our Learning Path on the Sequence pattern, from the MODL library [27].

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, obvious data properties, and the contents of controlled vocabularies. For the complete documentation, see [8].

[1] https://www.nsf.gov/
[2] https://www.nasa.gov/
[3] https://www.nih.gov/
[4] https://nij.ojp.gov/
[5] https://www.noaa.gov/
[6] https://www.usgs.gov/

This framework's key concept is the LearningPath, a dynamic series of *LearningSteps* developed to accommodate different categories of *Personas*. These Personas, consisting of developers, enthusiasts, executives, and contributors, act as categories that direct the learning paths' content and structure. Everyone that is interested, chooses a Persona that fits their role and professional experience, and each Persona shapes the architecture of their respective *LearningPath*. The ideal learning trajectory is thus determined by the chosen Persona, guaranteeing that the content is appropriate and suited to the learner's needs.

The *FirstLearningStep* is the first step of a learning path, which is composed up 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 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 with 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. A significant chunk of the ontology is dedicated to data about *people* involved in the making of curricula/modules. The involvement of people is modeled according to the *agent-role* ontology design pattern [11]. In this modeling paradigm, an agent/person is not simply classified as a 'student', a 'teacher', etc., but rather, we record that the person *played the role of* a student, teacher, etc. at a certain time, or in connection with a certain event.

Thus we keep track of 3 types of data objects: persons, ParticipantRoles, and the contexts in which persons assume roles. In the CurrKG ontology, these contexts can be Events, Media, or Modules, any of which can be linked to a ParticipantRole by the providesParticipantRole property.

A ParticipantRole instance is not just a generic role like 'teacher'. It represents a specific instance of teaching carried out by one person in one context. Generic roles are represented in our ontology as *subclasses*, not instances, of ParticipantRole.

Curriculum. A Curriculum defines the scope of a Learning Path by offering a plan of action with a title and one or more Modules to direct the evolution of a particular Persona's learning experience.

$$\mathsf{Curriculum} \sqsubseteq \exists \mathsf{hasTitle.xsd{:}string} \tag{1}$$
$$\mathsf{Curriculum} \sqsubseteq \exists \mathsf{hasModule.Module} \tag{2}$$

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

Person. A Person is a someone who takes on different roles, such as Author, who is in charge of generating material, or Persona, who represents a particular perspective or kind of user who actively participates in a framework as an agent carrying out particular tasks within those roles.

$$\mathsf{Person} \sqsubseteq \exists \mathsf{assumesRole.Author} \tag{3}$$
$$\mathsf{Person} \sqsubseteq \exists \mathsf{assumesRole.Persona} \tag{4}$$

3. There exists a Person assumes a role that is an Author.
4. There exists a Person assumes a role that is a Persona.

LearningPath. A LearningPath is a series of learning steps defined within a Curriculum, and the Curriculum contains at least one Module, which is a unit of learning content.

$$\mathsf{LearningPath} \sqsubseteq \exists \mathsf{scopedBy.Curriculum} \tag{5}$$
$$\mathsf{Curriculum} \sqsubseteq \exists \mathsf{hasModule.Module} \tag{6}$$

5. Every Learning Path is scoped by a Curriculum.
6. Every Curriculum contains at least 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.

$$\mathsf{Module} \sqsubseteq \exists \mathsf{coversTopic.Topic} \tag{7}$$
$$\mathsf{Module} \sqsubseteq \exists \mathsf{hasTitle.xsd{:}string} \tag{8}$$
$$\mathsf{Module} \sqsubseteq \exists \mathsf{hasLevel.Level} \tag{9}$$
$$\mathsf{Module} \sqsubseteq \exists \mathsf{belongsTo.Category} \tag{10}$$
$$\mathsf{Module} \sqsubseteq \exists \mathsf{references.Media} \tag{11}$$

7. Every Module covers a Topic.
8. Every Module has a title as a string.
9. Every Module has a level.
10. Every Module belongs to a Category.
11. Every Module references some Media.

Author. An Author is a Role assumed by a Person, responsible for creating content for Media and is identified by a unique Name.

$$\mathsf{Author} \sqsubseteq \exists \mathsf{hasName.xsd{:}string} \tag{12}$$

12. Every author has some name and that is represented as a string.

Category. A Category has different Subclasses, i.e. Foundations, Surveys, Methodologies, Standards. Modules belong to the categories respectively.

$$\mathsf{Category} \sqsubseteq \exists \mathsf{hasModule.Module} \tag{13}$$

13. Every category has some module.

Event. An Event can include different types of activities, such as presentations, tutorials, and workshops. The Event can also include smaller, related events (called sub-events), and it can offer Media materials.

$$\mathsf{Event} \sqsubseteq \geq 0\ \mathsf{hasSubEvent.Event} \tag{14}$$
$$\mathsf{Event} \sqsubseteq \exists \mathsf{provides.Media} \tag{15}$$

14. Every event has zero or more sub-events.
15. Every event provides some Media.

LearningStep. We abbreviate $\mathsf{hasNextLearningStep}$ as hNLS, and similarly for "previous."

A Learning Step is referring to a specific Module within the Curriculum that a given Persona will follow in order to complete their goals. It has a starting point and an ending point, following an order and pointing to next and previous steps.

$$\mathsf{LearningStep} \sqsubseteq =1\ \mathsf{hNLS.LearningStep} \tag{16}$$
$$\mathsf{LearningStep} \sqsubseteq =1\ \mathsf{hPLS.LearningStep} \tag{17}$$
$$\mathsf{LastLearningStep} \sqsubseteq =1\ \mathsf{refersTo.Module} \tag{18}$$
$$\mathsf{FirstLearningStep} \sqsubseteq =1\ \mathsf{refersTo.Module} \tag{19}$$

16. Every learning step has exactly one next learning step.
17. Every learning step has exactly one previous learning step.
18. Every last learning step refers to exactly one module.
19. Every first learning step refers to exactly one module.

Persona. We established a set of personas to guide progress and evaluate ease of use, covering a range of different kinds of users, 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 fo through in order to achieve their separate educational goals.

$$\mathsf{Persona} \sqsubseteq = 1 \mathsf{hasProfession.Profession} \tag{20}$$
$$\mathsf{Persona} \sqsubseteq \forall \mathsf{hasType.Profession} \tag{21}$$
$$\mathsf{Persona} \sqsubseteq \exists \mathsf{hasType.PersonaType} \tag{22}$$
$$\mathsf{Persona} \sqsubseteq = 1\ \mathsf{determines.LearningPath} \tag{23}$$

20. Every persona has exactly one profession.
21. Every persona has a type and thst type is a profession.
22. Every persona has a type and that type is a persona type.
23. Every persona determines exactly one learning path.

Topic. A Topic acts as a concept that is covered by a Module that has a title. Topics are organized hierarchically connecting to both broader and narrower topics.

$$\mathsf{Topic} \sqsubseteq = 1 \mathsf{asStringxsd{:}String} \tag{24}$$
$$\mathsf{Topic} \sqsubseteq \exists \mathsf{broaderThan.Topic} \tag{25}$$
$$\mathsf{Topic} \sqsubseteq \exists \mathsf{narrowerThan.Topic} \tag{26}$$
$$\tag{27}$$

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

25. Every topic is broader than some topic.
26. Every topic is narrower than some topic.

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 [27], and allows for quick and easy modifications to the ontology without perturbing the overall subsumption hierarchy.

Our ontology is formalized in the Web Ontology Language (OWL 2) [12, 20], specifically in OWL 2 DL, with the axioms as shown in each respective block of text, expressed using description logic notation.

In Figure 1, we provide a **schema diagram** of our ontology, which shows all the classes discussed above except those in the Level and Audience modules.

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)[7], data from National Student Data Corps (NSDC)[8] – 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[9]. 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[23], 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 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 resolving 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`) [4], which serves as a materialized knowledge graph 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

[7] https://github.com/KGConf/open-kg-curriculum
[8] https://nebigdatahub.org/nsdc/
[9] https://edugate.cs.wright.edu/

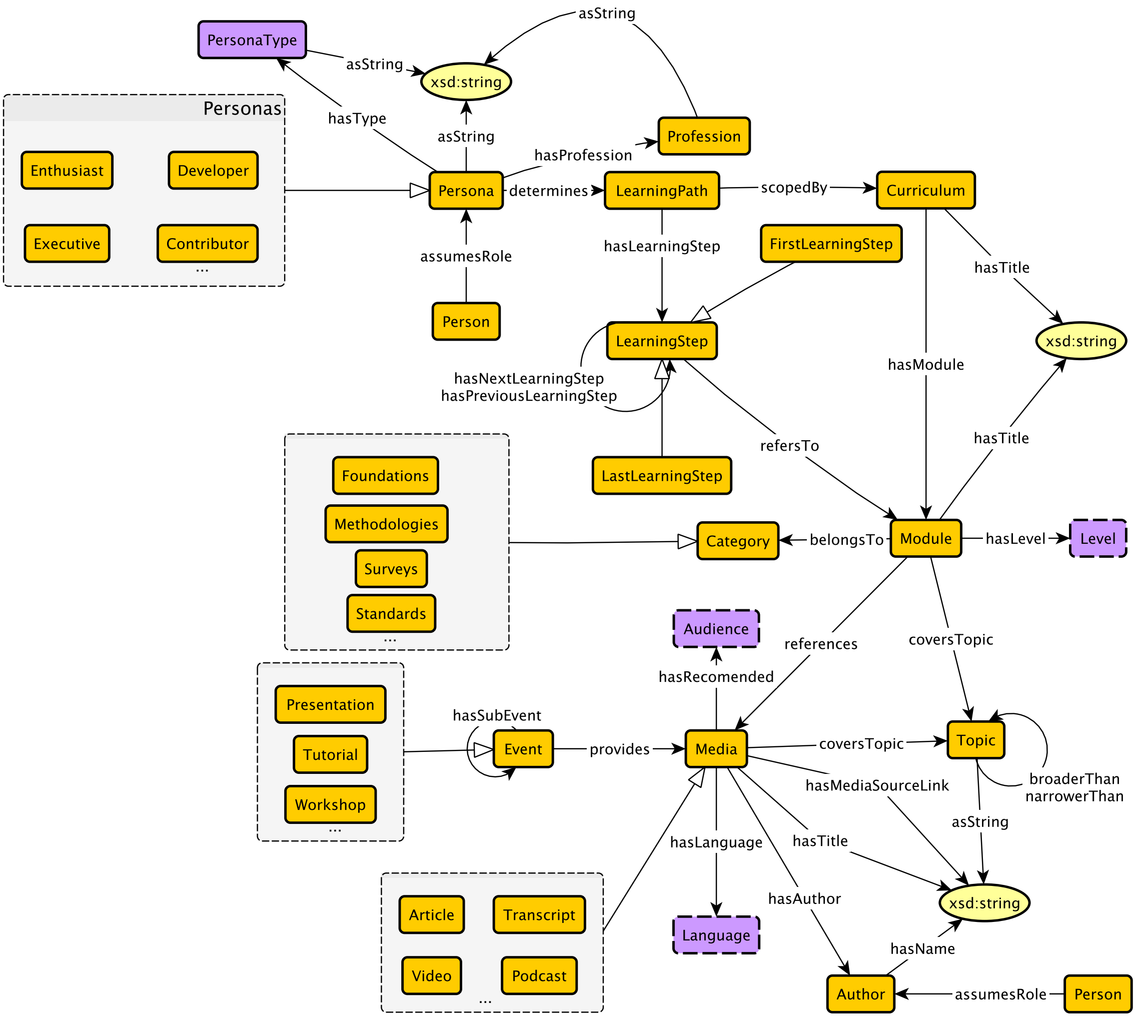

Figure 1: The schema diagram for the overall Curriculum KG Ontology.

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

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 knowledge graph 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 [8].

3.2. Evaluation

We evaluate the ontology and knowledge graph 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 knowledge graph should be able to answer. They serve as a practical benchmark for assessing the graph or ontology and its alignment with the intended domain [17].

We then expressed these questions as optimal SPARQL queries[6] to be executed over the graph using Apache Jena Fuseki Triplestore[9]. 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 optimized SPARQL queries used for evaluation are available in the git repository[10]. This approach of executing the SPARQL to answer the CQs over materialized CurrKG helped confirm that the ontology sufficiently captured the semantics of the educational data, and all the entities and relationships were correctly instantiated in materialization.

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 Knowledge Graphs (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 customizable learning paths.

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

Knowledge graph construction techniques (KG) 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.

Teaching Knowledge Graph for Knowledge Graphs Education

A structured framework created especially to enable Knowledge Graph education by integrating key educational components like skills, subjects, courses, instructors, and resources, is described in the study by Ilkou et al. [15]. 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. Our approach goes beyond their work, which focuses on developing a single KG for learning, taking into account the different needs from various personas.

Our own work 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 follow the same trend by modeling individualized learning paths by also encoding individual learner profiles in the KG. This enables users with different backgrounds, skill levels, and learning goals receive tailored educational resources. It

[10]clean up repo and post direct link or repharse to say it lives in the repo

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. describe a system in Knowledge Graph-Based Teacher Support for Learning Material Authoring that uses knowledge graphs (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 knowledge graphs more dynamic and learner-centric, allowing us to progress beyond content organization toopic of personalized learning.

Computer Science Curricula

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, by defining a structured ontology, educational materials were successfully organized in a query-able KG. 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 can allow an individual to learn at their own pace, with respect to their own goals an objectives.

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

- Incorporation of the ontology to the features of the Interactive Knowledge (InK) Browser[28]. Serving as a platform for the audience members to learn.
- 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 Knowledge Areas, Knowledge Units, and Learning Outcomes. 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.

Acknowledgment. The author(s) acknowledge support from the National Science Foundation under award #2333532 "Proto-OKN Theme 3: An Education Gateway for the Proto-OKN (EduGate). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of National Science Foundation or the U.S. Department of Education.

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] I. ACM and AAAI. Computer science curricula 2023: Curriculum guidelines for undergraduate degree programs in computer science, 2023. Accessed April 12, 2025.

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

[5] 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.

[6] 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.

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

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

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

[10] 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.

[11] P. Hitzler and A. Krisnadhi. On the roles of logical axiomatizations for ontologies. 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 73–80. IOS Press, 2016.

[12] 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/.

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

[14] 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.

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

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

[17] 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.

[18] 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.

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

[20] 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/.

[21] 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.

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

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

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

[25] A. Seaborne and E. Prudhommeaux. SPARQL query language for RDF. 2008. http://www.w3.org/TR/rdf-sparql-query/.

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

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

[28] 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.