

The Course Recommender Ontology

Final Project Report - Ontology Engineering, Fall 2020

Team: Sola Shirai, Owen Xie, Kelly Fellenzer, Jacob Schomstein

Table of Contents

The Course Recommender Ontology	1
Abstract	1
Introduction	2
Use Case	2
Technical Approach	3
Ontology Structure	3
People	4
Courses	5
Requirements and Plan of Study	6
Proposed Recommender System Design	8
Evaluations	8
Competency Question 1	8
Competency Question 2	9
Competency Question 3	10
Discussion	11
Key Features	11
Value of Semantics	12
Limitations	13
Website	13
Related Work	14
Future Work	14
Conclusion	15
Acknowledgements	15
References	15

Abstract

University students can benefit from carefully planning out which courses they will take through their academic career. Developing such a plan of study can be difficult because students need to consider factors such as how each course applies to their graduation requirements, whether courses have schedule conflicts, whether the courses are interesting to the student, and how course prerequisites can affect their plan. In this work, we present the Course Recommendation Ontology, which aims to represent relevant information about courses, schedules, graduation requirements, and students to enable a recommender system. Our ontology can be used to provide rule-based

recommendations that adhere to requirements and provide explanations to students about why certain courses are recommended to them.

Introduction

Choosing which courses to sign up for each semester is a universal experience for college students. This process of selecting courses can require consideration of a number of factors. First and foremost, students must ensure that they select appropriate courses to satisfy the graduation requirements necessary for their intended degree. This often involves developing a plan of study that organizes the student's intended course selections for several semesters or years in advance. While planning out how to choose courses that fulfil graduation requirements, students also must consider potential requirements that exist for individual courses (such as prerequisite or corequisite courses). Additionally, a student must have a level of flexibility in their plan of study because schedule conflicts or course capacity may prevent them from registering for some courses. Finally, it would behoove the student to construct their plan of study such that they can participate in courses that cover topics they are most interested in - which in turn requires the student to read through various listings of courses and inspect their contents.

Together, the aforementioned considerations make it challenging for students to effectively devise their plan of study. Information regarding graduation requirements are often unclear and typically vary based on the particular major of the student. Information about courses, including their requirements and topic areas, must be acquired by tediously navigating through course catalogs. Students may rely on various academic advisors to help select courses, but the expertise of such advisors tends to be dispersed. For example, a computer science professor may offer expert advice on which computer science courses would most appeal to a particular student, but they may be less knowledgeable in all the details for graduation requirements. Conversely, a faculty advisor that oversees graduation requirements may be an expert in whether a plan of study can satisfy graduation requirements, but they might lack knowledge about specific topics and areas of study.

To empower academic advisors and students to more effectively select which courses to enroll in, we present the Course Recommender Ontology (CRO). The CRO describes the domain of university courses, including relevant information surrounding graduation requirements and students, and offers a means to effectively query the knowledge needed to develop a student's plan of study. While we developed the CRO to specifically capture information about Rensselaer Polytechnic Institute (RPI) and its students, the underlying models are sufficiently flexible for the CRO to be applied to other universities.

Use Case

The goal of the CRO is to generate a set of recommended courses for an RPI student to register in for their upcoming semester. The key stakeholders in this use case are the students, academic advisors, and other faculty members who provide students with course registration guidance at RPI. In this use case, we limit the scope to (1) only consider undergraduate RPI students enrolled in their senior year, (2) only provide recommendations for a single semester, and (3) only capture

graduation requirements for a Computer Science major. Information about courses and various requirements were curated from RPI's Course Catalog [9].

To produce course recommendations, the CRO represents semantics pertaining to hard and soft constraints. Hard constraints refer to conditions that must be met in the resulting set of courses that are recommended to the student. The hard constraints that are modeled in the CRO are graduation requirements, required prerequisite courses, course cross-listings (i.e., alternate names for the same course), course availability, and course schedules (i.e., hard constraints on schedule conflicts). Information about which courses a student has already completed are also captured by the CRO because course history will influence constraints on graduation requirements and prerequisites.

On the other hand, soft constraints refer to aspects of user preference that are helpful to rank recommended courses. These semantics include topic areas covered by a course, topic areas that a student is interested in, relations among topic areas, and recommended prerequisite courses. A student's completed courses also can be used to infer topic areas that the student might like. Additionally, course level can also be used to facilitate preferences for taking "easier" or "harder" courses. Courses with lower levels are typically considered easier or more introductory, and choosing whether to recommend "easier" or "harder" courses may be influenced by the student's preferences or how close they are to graduation.

Our use case focuses on describing several usage scenarios involving student profiles, their goals, and their motivations to receive course recommendations. Using the requirements that could be extracted from these scenarios, we further developed the key concepts and relationships necessary in our ontology.

Further details about the use case, including detailed requirements and usage scenarios, can be viewed in our [use case document](#).

Technical Approach

Ontology Structure

The technical approach to modelling the domain can be summarized in three main components: people, primarily relating to stakeholders of students and advisors; courses, including distinct offerings and sections; and graduation requirements, which capture different layers and interactions among requirements needed for a student to obtain their degree. A concept that is common to both people and graduation requirements is the student's plan of study. For organizational purposes, we present details on the plan of study together with graduation requirements.

We first present the high-level conceptual model of the CRO below. In the following sections, we will break down the components that make up the high-level model into more manageable pieces. The conceptual model can also be obtained from [our website](#).

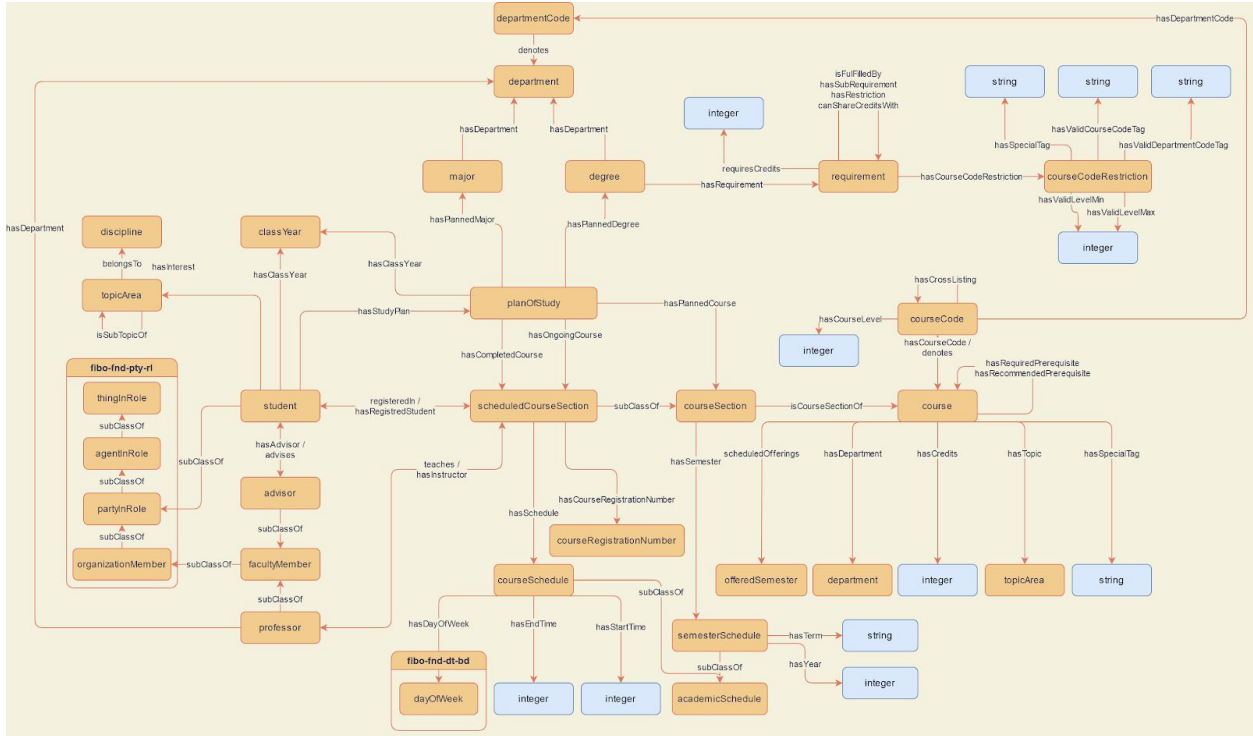


Figure 1: High-level conceptual model of the CRO

The combination and interaction of the classes within the components essentially compose the larger CRO, allowing us to clearly describe paths to graduation and the context around that path. It also allows the CRO to represent the knowledge necessary to provide recommendations through a flexible approach. The following sections will further explore the deeper details that allowed for ontology to be successful.

People

The types of people that we model in the CRO are students and faculty members. We capture the most information about students, because their academic progress is the main motivator of our use case. Students have a class year, which relates to when they entered the university and when they will graduate, as well as some interests. Interests are modeled as topic areas, which can iteratively be sub-topics of other topic areas and belong to a discipline. For example, a student might be interested in the topic area of “Ontologies”, which might in turn be a subtopic of “Artificial Intelligence” or “Knowledge Representation”. All of these topics then belong to the discipline of “Computer Science”. In this way, we capture a hierarchical structure of topic areas to enable a flexible means of capturing student interest to enable better recommendations.

The conceptual model for classes related to People is shown in Figure 2. Note that students also have a connection to their plan of study, which is described in a later section.

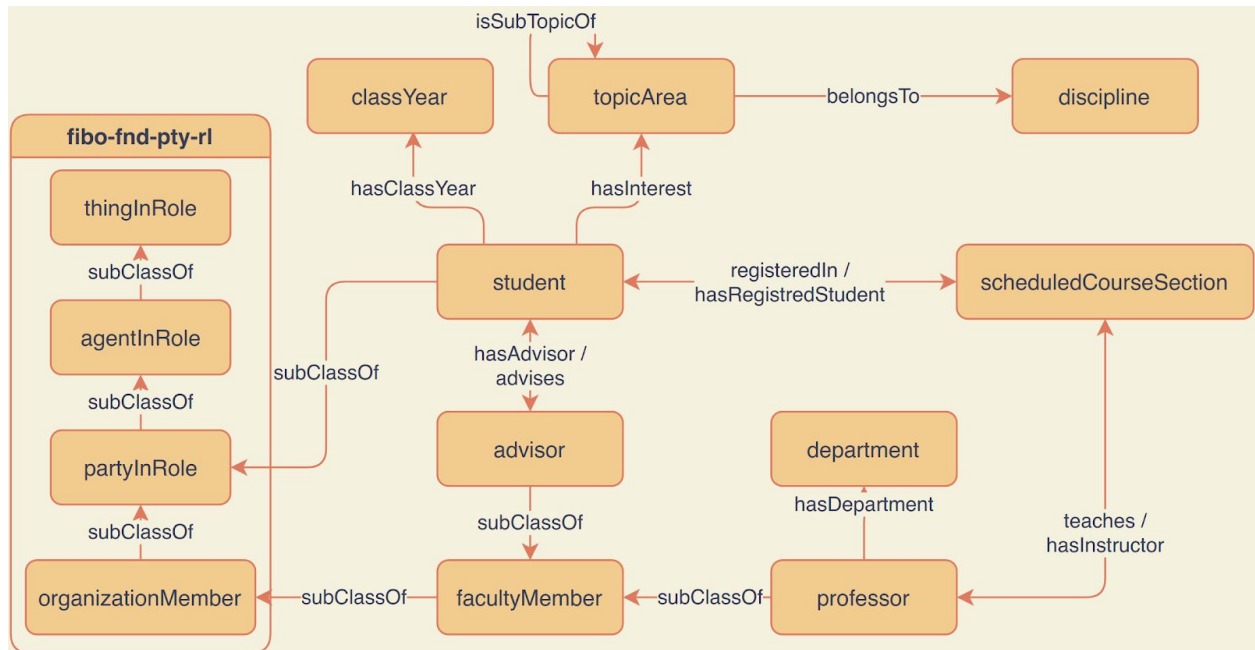


Figure 2: The “People” conceptual model of the CRO

For faculty members, we decided that two specific types were relevant to our use case; academic advisors, and instructors. Advisors capture the general categorization of faculty members who interact with students to provide advice, including advice about what courses to take or how to pursue certain degrees. Instructors are used by instances of scheduled course sections to specify the faculty member who is tasked with teaching the course.

Courses

The basic properties of a course are its department, its credits, the topic areas it covers, its prerequisites, and its course code. Prerequisites can either be “required” or “recommended”, in which the former case enforcing a hard constraint on a student’s ability to register for a particular course. Course codes correspond to strings like “CSCI-4340”, which break down into the course level (e.g., “4340”) and department code (e.g., “CSCI”). Course codes also can be cross-listed (e.g., “CSCI-4340” and “CSCI-6340” are cross-listed), which means that the same course can be offered by different departments or at different levels while containing the same content.

The conceptual model of classes and relationships surrounding courses can be seen in Figure 3. We can see how the course section class points to a course that it corresponds to. We use such techniques to keep distinctions between course sections - which are meant to be closer to real-world instances of classes - versus persistent courses that are offered regularly at RPI.

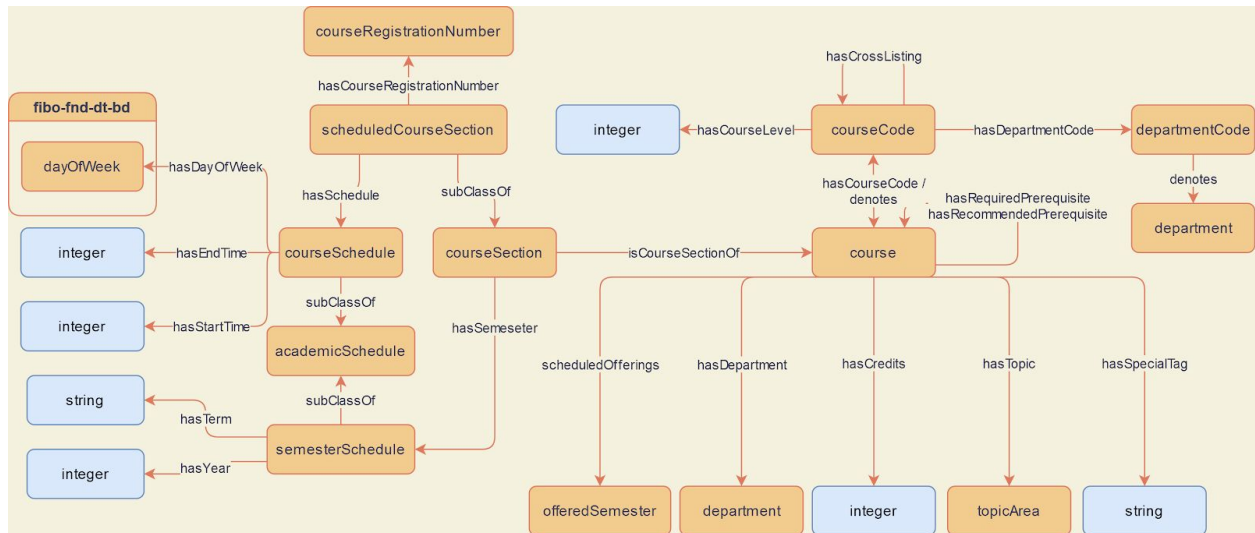


Figure 3: The “Course” conceptual model of the CRO

A major modelling decision that we made for the CRO is the distinction of courses and course sections. In this case, a “Course” is the persistent idea of a class that is offered at RPI. While small details may change over time, the general framework and topics of the course should remain the same. In addition, courses do not change often with regards to the overall prerequisite directed graph. A “Course Section” is a semester-specific instance of a class. Within this concept, there are further distinctions between a “Scheduled Course Section” and a “Course Section”. A scheduled instance is one that has been given an explicit registration number, instructor, meeting location, and meeting schedule by RPI’s administration. The scheduled course section of a course is, therefore, a real-world instance of a course that students can enroll in.

This structure allows us to separately represent real-world courses that the student can actually enroll in versus courses that will probably be offered in the future at a specific semester. Being able to represent some notion of future courses greatly enhances our ability to provide recommendations for future plans beyond the upcoming semester. Additionally, this allows us to treat a “Course” and “Course Section” as data, which improves the reusability of the ontology for other universities and avoids complex modelling patterns, such as punning.

Requirements and Plan of Study

The conceptual model shown in Figure 4 displays classes and relationships about graduation requirements and the student’s plan of study. While this model contains fewer classes than the models related to courses, it is capable of representing much more complicated relationships (especially for requirements).

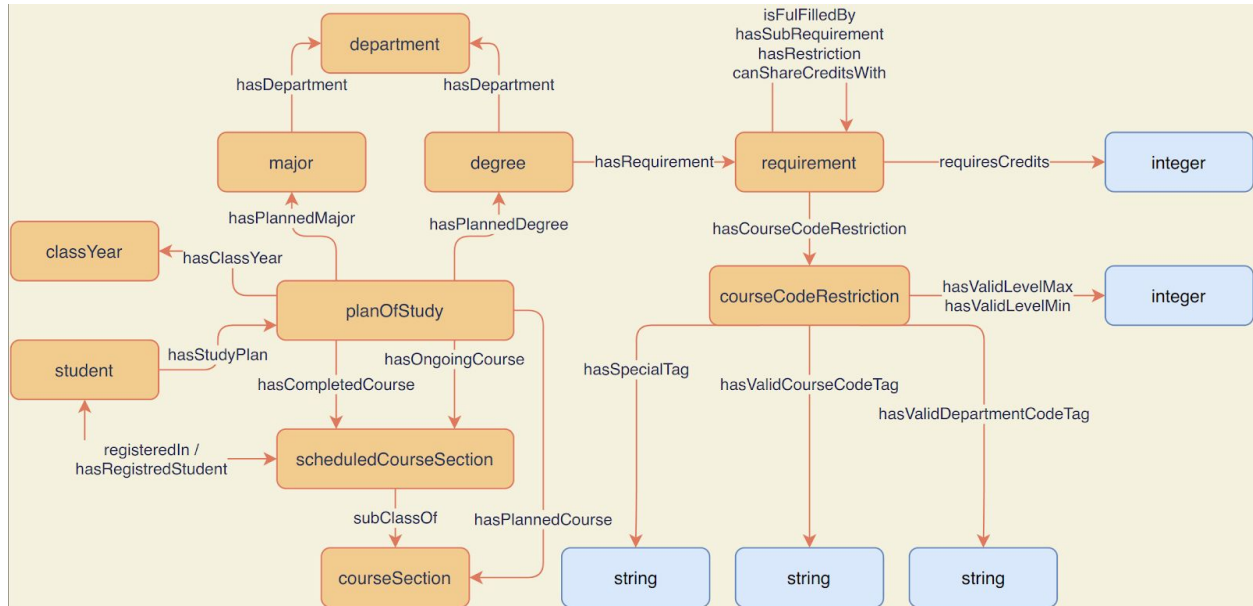


Figure 4 : The “Requirements” conceptual model of the CRO

For a student to obtain some degree, they must complete that degree’s requirements. Requirements consist of some number of credits needed to fulfill the requirement, some restriction on the courses that can be used to fulfill it, and relationships to other requirements. Relationships to other requirements allows us to break down the various graduation requirements in a modular fashion.

First, a requirement can be considered fulfilled if any of the requirements connected by a “isFulfilledBy” relationship are fulfilled. Next, a requirement is only considered fulfilled if *all* of the other requirements connected by the “hasSubRequirement” relationship are also fulfilled. This essential allows composition of requirements through “AND” and “OR’ logical operators. Additionally, sometimes credits from a given course can be re-used to fulfill requirements that are not directly related through the aforementioned relations. We enable this through the “canShareCreditsWith” relationship. Finally, we use the “hasRestriction” relationship to represent certain conditions that must *not* be met by the student using the same structure as regular requirements. These four relationships allow for a very flexible but powerful way to model the graduation requirements for RPI students.

Also found in this conceptual model is the plan of study. Students have a few attributes that represent them: their interests, class year, and the courses they take. For their courses, students have a role “hasStudyPlan” that points towards an instance of the plan of study class. The plan of study then has multiple links under the roles “hasCompletedCourse”, “hasPlannedCourse”, and “hasOngoingCourse”, which dictate which courses’ sections a student has taken or plans to take. The relevant object, a “Course Section”, has the necessary information which semester the course occurs on, which enables us to divide the student’s plan of study into different semesters. This information is important in enabling more personalized recommendations to be produced.

For more details on the actual ontology's contents, the ontology can be obtained from [our github repository](#). Further details and prior versions of the ontology can also be accessed through [our website](#).

Proposed Recommender System Design

The CRO allows us to represent all of the information that is relevant to solving our use case and provide students with course recommendations. However, the ontology alone is not capable of carrying out the process of actually producing recommendations. To do this, we would need to develop a recommender system that utilizes the CRO to help carry out its functionalities. While such an application was beyond the scope of this project, we do have some ideas for how the prototype of such an application may be done.

In such an application, the CRO would be used to inform the structure for how to capture data about courses and graduation details. The recommender system could leverage this well-structured data to more effectively query and obtain information needed to provide good recommendations, such as the remaining requirements and interests of the student. Relationships between classes, like topic areas, could be used to develop metrics of similarity to produce good recommendations for classes that are most relevant to a student's interests.

Additionally, the CRO could provide additional benefits to this recommender system by enabling reasoning and inference to expand its knowledge base. For example, a classic problem with recommender systems is the "cold-start" problem, when a system has no prior information about a user to base its initial recommendations off of. We can use semantics from the CRO to help fill in some missing information (e.g., by inferring user preferences) to help handle this situation more effectively.

Evaluations

For the scope of this project, we evaluate our ontology using several competency questions from our use case. We perform this assessment by demonstrating the ability of our ontology to construct a SPARQL query and provide a sufficient answer to each competency question. Note that this evaluation only serves to assess the ability to answer such questions through manual inputs, and it does not cover any automatic question answering or demonstrate the capabilities of a recommendation system that might utilize the CRO. Information about potential evaluations that may be carried out for applications of the CRO can be found in the Future Work section.

Our use case contains 5 competency questions, of which 3 are marked for active development and 2 are marked for secondary development. In this report, we describe our 3 active competency questions for which we curated adequate data to provide meaningful answers.

Competency Question 1

Question: What are all the prerequisite courses needed to take the CSCI 4340 Ontologies course?

This competency question demonstrates the ability of the CRO to capture course prerequisite information, which is important to ensure that the recommendation system restricts which courses it recommends. To answer this question, we must identify the URI of the target course and then collect its prerequisite courses. We also need to iteratively look at the prerequisites of prerequisite courses to get a complete set of prerequisite courses. To answer this question, we can follow a chain of “hasRequiredPrerequisite” relations from the target course. The SPARQL query and results to answer this competency question are shown below.

```

prefix crs-rec: <https://tw.rpi.edu/ontology-engineering/oe2020/course-recommender/>
prefix crs-rec-ind: <https://tw.rpi.edu/ontology-engineering/oe2020/course-recommender-individuals/>
prefix lcc-lr: <https://www.omg.org/spec/LCC/Languages/LanguageRepresentation/>
SELECT DISTINCT ?prerequisiteCourseName
WHERE {
  VALUES ?targetCourseTag { "CSCI-4340" }
  ?targetcourse crs-rec:hasCourseCode [ lcc-lr:hasTag ?targetCourseTag ] ;
             crs-rec:hasRequiredPrerequisite+ ?prerequisiteCourse .
  ?prerequisiteCourse crs-rec:hasName ?prerequisiteCourseName .
}

```

prerequisiteCourseName
Computer Science I
Data Structures
Foundations of Computer Science
Introduction to Algorithms

Competency Question 2

Question: I am a rising senior and I want to take the smallest number of courses required to complete my degree. I also want to take “easier” courses whenever possible to allow for more time to plan for a future career. What courses can fulfill my remaining requirements?

This competency question addresses the notions of retrieving graduation requirements, identifying which courses could be used to fulfill those requirements, and recognizing which courses the student has already completed. It also incorporates some subjective notion of how “easy” a course is; for the sake of simplicity, we choose to define “easy” courses in this query as those that have a level of under 4000.

We answer this question based on course history data from one of the authors, Jacob. We answer this competency question by retrieving all candidate courses that can be used to fulfill graduation requirements encoded in our individuals ontology.

Because of the limited scope of implementation for this project, we only have encoded parts of the Humanities, Arts, and Social Sciences (HASS) core requirement for Computer Science majors. Additionally, we only curated courses from a few departments into our individuals ontology. As a result, we only retrieve 1 answer for our query. However, we can verify that it can be used to fulfill

the HASS core requirement and that it does not overlap with courses that Jacob has already completed. The SPARQL query used to obtain the result can be seen below.

```

prefix crs-rec: <https://tw.rpi.edu/ontology-engineering/oe2020/course-recommender/>
prefix crs-rec-ind: <https://tw.rpi.edu/ontology-engineering/oe2020/course-recommender-individuals/>
prefix lcc-lr: <https://www.omg.org/spec/LCC/Languages/LanguageRepresentation/>
SELECT DISTINCT ?recCourseName ?recCourseLabel
WHERE {
  # the URI for Jacob in our individuals ontology
  ?usrUri crs-rec:hasName "jacob" ;
    crs-rec:hasStudyPlan [ crs-rec:hasCompletedCourse ?completedCourseSection ] .
  ?completedCourseSection crs-rec:isCourseSectionOf ?completedCourse .
  ?requirement a crs-rec:Requirement ;
    crs-rec:hasCourseCodeRestriction ?ccRest .
  OPTIONAL { ?ccRest crs-rec:hasValidCourseCodeTag ?validTag }
  OPTIONAL { ?ccRest crs-rec:hasValidDepartmentCodeTag ?validDeptTag }
  OPTIONAL { ?ccRest crs-rec:hasValidLevelMax ?validLevelMax }
  OPTIONAL { ?ccRest crs-rec:hasValidLevelMin ?validLevelMin }
  ?recCourse a crs-rec:Course ;
    crs-rec:hasCourseCode ?recCourseCode ;
    crs-rec:hasName ?recCourseName ;
    rdfs:label ?recCourseLabel .
  ?recCourseCode crs-rec:hasLevel ?recCourseLevel ;
    lcc-lr:hasTag ?recCCTag ;
    crs-rec:hasDepartmentCode [ lcc-lr:hasTag ?recCCDeptTag ] .
  FILTER (?recCourse != ?completedCourse ) # filter completed courses
  FILTER (?recCourseLevel < 4000) # only select "easy" courses
  FILTER ( (?recCCTag = ?validTag) || (?recCCDeptTag = ?validDeptTag && ?validLevelMin <= ?recCourseLevel &&
    ?validLevelMax >= ?recCourseLevel)) # check that the course can satisfy restrictions
}

```

recCourseName	recCourseLabel
Introduction to Cognitive Science	COGS-2120

Competency Question 3

Question: I have taken CSCI 4340 Ontologies and CSCI 4020 Design and Analysis of Algorithms. What are some courses like CSCI 4340 Ontologies that I should take next fall?

This competency question addresses the ability of the CRO to represent schedules of courses (e.g., a semester schedule for Fall 2020) and infer completed courses based on the provided information. For example, because the Ontologies course itself has prerequisites, we can infer that the student has also completed those prerequisite courses. This competency question also touches on some elements of preference and topic areas covered by courses, but we omit the usage of similarity and recommendations because it is outside of the scope of evaluation for this report.

To answer the question, we first need to identify the URIs for the two courses. Next, we identify all prerequisite courses that we can infer as being completed by the student. We then collect all CourseSections for courses that are being offered in the Fall 2020 semester. Finally, we remove any candidate courses who have prerequisites that the student has not completed and courses that overlap with any of the courses the student has already completed. Because many answers in the individuals-ontology match these requirements, we apply a limit to the number of answers returned. The SPARQL query and results for this competency question are shown below.

```

prefix crs-rec: <https://tw.rpi.edu/ontology-engineering/oe2020/course-recommender/>
prefix crs-rec-ind: <https://tw.rpi.edu/ontology-engineering/oe2020/course-recommender-individuals/>
prefix lcc-lr: <https://www.omg.org/spec/LCC/Languages/LanguageRepresentation/>
SELECT DISTINCT ?validCourseName
WHERE {
  VALUES ?completedCourseTag { "CSCI-4340" "CSCI-4020" }
  ?completedCourse crs-rec:hasCourseCode [
    lcc-lr:hasTag ?completedCourseTag
  ] .
  ?completedCourse crs-rec:hasRequiredPrerequisite* ?completedCourseInferred .
  ?validCourseSec a crs-rec:ScheduledCourseSection ;
    crs-rec:hasSemester [
      crs-rec:hasTerm "fall" ;
      crs-rec:hasYear 2020
    ] ;
    crs-rec:isCourseSectionOf ?validCourse .
  ?validCourse a crs-rec:Course .
  ?validCourse crs-rec:hasName ?validCourseName .
  FILTER NOT EXISTS {
    ?validCourse crs-rec:hasRequiredPrerequisite+ ?filterPrereqs .
    FILTER(?filterPrereqs != ?completedCourseInferred)
  }
  FILTER (?validCourse != ?completedCourseInferred)
}
LIMIT 10

```

validCourseName
Introduction to Cognitive Science
Introduction to Linguistics
Introduction to Cognitive Neuroscience
Programming for Cognitive Science and Artificial Intelligence
Game AI
Learning and Advanced Game AI
Topics in Cognitive Science
Undergraduate Thesis
Master's Project
Master's Thesis

Discussion

Key Features

One feature that we considered essential in the development of the CRO was the ability to represent course information in such a way that it could directly be associated with actual course offerings at RPI. We have achieved this through our model design, distinguishing Courses, CourseSections, and

ScheduledCourseSections. The ScheduledCourseSection class includes a course registration number and schedule, which allows us to directly relate it to a real-world course that students can register for. The Course class then allows us to capture more general notions about a particular course that is persistent across semesters. CourseSections provide an intermediate level of granularity, which allows us to capture some notion of courses that will be offered in the future without tying it to a concrete realization of the course. Together, these classes enable our ontology to represent course information in all the ways necessary to allow a recommender system to generate recommendations for specific courses to take or plan out a series of courses in a plan of study.

Another key feature of the CRO is its modular representation of graduation requirements. Through the hasSubRequirement and isFulfilledBy relations, we can represent combinations of requirements like logical AND and OR operators, respectively. For example, a requirement having hasSubRequirement relations to requirement A and B corresponds to requiring that both A and B are fulfilled by the student. Similarly, a requirement with isFulfilledBy relations to A and B corresponds to requiring that either A or B is fulfilled. We also can represent combinations of courses that are not allowed through the hasRestriction requirement. Finally, we can relate requirements that are related but not directly in a hierarchical fashion through the canShareCreditsWith relation. These four cases allowed us to curate and model graduation requirements for our target major of Computer Science at RPI.

Lastly, we model student context and capture the basic information that we believe is required to enable personalized recommendations of courses and plans. We were able to model information that was relevant to hard constraints (namely the student's intended major, degree, and completed course information) as well as information pertaining to preferences or interests for soft constraints.

Value of Semantics

The use of semantic technologies provides our project with a number of benefits. One of these benefits is that it enables us to use inference and reasoning to fill in missing information. For example, we modeled cross-listings of courses (i.e., the same course but offered under different departments or distinguishing between graduate- and undergraduate-level) as a transitive and symmetric property. Within RPI's course catalog, course cross-listing information is often incomplete. Through CRO's modeling of this property, we can use reasoning to fill in complete cross-listing information for courses. Similarly, we can perform inference about what courses a student has already completed (using prerequisite information, as in Competency Question 3). We also can perform inference regarding what kinds of topics a student is interested in based on their course history and hierarchical relationships between topic areas.

Another major benefit is the ability to incorporate provenance of the information contained in the ontology. This includes provenance of the information source as well as when the information was collected. While in the scope of our project, we curated course information only from RPI's course catalog, the ability to include provenance provides us with greater flexibility in the future if we chose to incorporate different sources to expand our knowledge base of things like topic areas or courses offered at different institutions. The provenance of *when* information was curated also

could be useful in ensuring that different versions of course information are available to students and advisors.

A final benefit is that the semantics in our ontology allows us to explicitly express information about properties and classes through the use of annotations in RDF. This in turn makes our ontology capable of describing the course data in such a way that it is more interpretable to humans. Greater interpretability by humans is an important feature to have because it can help applications provide explanations about course recommendations to students. The interpretability of the ontology is also beneficial for if the CRO is to be applied to an institution other than RPI, as whoever is curating data for the new institution would need to understand the details and distinctions between different classes and properties. The rich semantics also can enable better interoperability between data that is modeled using the CRO.

Limitations

One of the CRO's limitations is the scope of graduation requirements that it captures. In this project we developed the CRO while only considering students aiming to receive a degree for a single major. Our current model does not inherently support any requirements surrounding dual majors or minors. Additionally, we only considered requirements for the Computer Science major in this project. Finally, in our representations of graduation requirements, we did not develop a flexible way to model changing requirements. For example, graduation requirements for Computer Science majors may change from year to year. In the current CRO, we would have to model this by having multiple different individuals for each year's version of graduation requirements. While our ontology can capture such changes in requirements, it is not done in a particularly elegant way and could be difficult to maintain.

Another limitation of the CRO is regarding course offerings. When students are trying to register for courses, there typically is an upper limit to the number of students that can enroll in a specific scheduled course section. In the CRO, we do not capture any of this type of information because it would require coordination to collect real-time information about course offerings. We defer such work to other active resources at RPI, and we do not incorporate it into the course recommendation process.

Lastly, this project has limitations in its scope and accuracy of course data curation. There were many situations where course prerequisite information presented by RPI's course catalog was ambiguous. This ambiguity often was caused by wordings of "and" and "or", which made it unclear whether some courses were *all* required, or some combination were required. It also was sometimes unclear which prerequisites were required and which were simply recommended. Within the timespan of this project, we were unable to produce a conclusive solution to this limitation.

Website

Key information about this project can be found on [our website](#). The website contains information pertaining to the design process of the project (including artifacts like the use case, terminology list,

and conceptual models), research-related information (like related work and re-use license), and resources such as a static demo of SPARQL queries and presentations. Links to all artifacts associated with this project, as well as prior iterations of various artifacts, will remain available and updated on the website.

Related Work

Several previous works have developed ontologies related to the main of universities and course structure. The Course Curriculum Syllabus Ontology (CCSO) [15] modeled information about university structure, course topics, and curriculum information. The CCSO had a greater focus on describing course syllabi and topics, but it did not provide structure to capture information such as course offerings or graduation requirements. The Academic Institution Internal Structure Ontology (AIISO) [11] was another ontology that primarily captured structural information about academic institutions. The BBC Curriculum ontology [16] described the national curriculum of the United Kingdom, but it did not extend to representing specific university courses or any sort of graduation requirements.

While several prior works related to the domain of university courses exist, a key difference between them and the CRO is the intended purpose of the ontologies. CRO was designed and developed with the use case of course recommendations at the forefront, while the other related ontologies appear to primarily be focused on simply describing the structure of universities and courses.

Future Work

The most obvious vein of future work is to expand the scope of data curation regarding courses and graduation requirements. This will primarily involve the development of new strategies to deal with irregularities in textual data that can be obtained from RPI's course catalog to extract full cross-listing information and more accurate prerequisite information. For graduation requirements, it is highly likely that manual curation will be necessary.

Once a more complete set of information about courses and requirements is available, the next step of this work will be to implement an actual recommendation system. The scope of this system can be the same as the scope for our use case. The goal for a prototype of this system would be to provide recommendations for a single semester to RPI students in their final year of school.

The development of such a prototype recommendation system will also require new evaluation metrics. We believe that two key metrics of evaluation for such a system would be to judge the correctness and usefulness of the generated recommendations. Correctness can be judged either by asking for expert opinion from RPI staff members who are experts in graduation requirements or by entering the suggested courses into RPI's system that automatically checks graduation requirements. Recommendations would be "correct" if they do not violate prerequisites, do not register multiple times for cross-listed courses, and if they can be used to successfully fulfill all graduation requirements. The "usefulness" of recommendations can be determined through user

studies involving both students and academic advisors. It might also be possible to provide some measure of usefulness by comparing generated course recommendations to real student registration patterns, if such data can be obtained.

After the above pieces of future work have been completed, it will be possible to explore expanded capabilities of the CRO and the prototype recommender system. Expansions to the CRO would include representing graduation requirements for minors and multiple majors. The prototype recommender system also can be improved to provide support for such expansions to the CRO. Depending on the success of the prototype recommender system, it might also be possible to consider expanding the overall system to support universities other than RPI as well.

Conclusion

In this report, we present the Course Recommender Ontology (CRO). The primary use case that motivated and guided the development of the CRO is to enable a recommender system capable of generating course recommendations for RPI students. The CRO captures semantics surrounding the concepts of courses, graduation requirements, and student profile such that it can provide personalized recommendations while adhering to hard and soft constraints on recommended courses. The CRO, individual artifacts associated with its development, and further information about the overall project, are made available through [our website](#).

We provide discussion on the contents of the CRO, including its primary features, value added compared to conventional systems, and limitations of its current state. Our strategies to model the various classes and relationships within CRO allowed us to create a flexible but expressive representation of information.

Acknowledgements

We would like to extend our thanks to all of the staff involved with the Ontology Engineering Fall 2020 course - Deborah L. McGuinness, Elisa Kendall, Sam Stouffer, Rebecca Cowan, Sabbir Rashid, and Shruthi Chari - for their guidance and support throughout the project.

References

- [1] Abdallah, S. A., & Ferris, B. (2010, July 23). The Ordered List Ontology. Retrieved September 11, 2020, from <http://smiy.sourceforge.net/olo/spec/orderedlistontology.html>
- [2] Academic Calendar. (2020). Retrieved September 16, 2020, from <https://info.rpi.edu/registrar/academic-calendar>
- [3] Association of Educational Publishers & Creative Commons. (2014, October 23). LRMI Version 1.1. Retrieved September 16, 2020, from <https://www.dublincore.org/specifications/lrmi/1.1/>
- [4] Brickley, D. (2015, April). Welcome to Schema.org. Retrieved September 12, 2020, from <https://schema.org/>

- [5] Degree Works. (2020). Retrieved September 20, 2020, from <https://info.rpi.edu/degree-works>
- [6] FOAF Vocabulary Specification 0.99. (2014, January). Retrieved September 16, 2020, from <http://xmlns.com/foaf/spec/>
- [7] Oh, S., & Rühle, S. (2009). DCMI: Home. Retrieved September 15, 2020, from <https://dublincore.org/>
- [8] RCOS. (2020). YACS. Retrieved September 30, 2020, from <https://yacs.cs.rpi.edu/schools>
- [9] "Rensselaer Polytechnic Institute - Acalog ACMS™", Rensselaer Catalog 2019-2020, 2019.[Online]. Available: <http://catalog.rpi.edu/>. [Accessed: 28- Sep- 2020]
- [10] Self-Service Information System. 2020. User Login. [online] Available at: <<https://sis.rpi.edu/>> [Accessed 27 September 2020].
- [11] Styles, R., & Wallace, C. (2008, May 14). Academic Institution Internal Structure Ontology Roles (AIISO Roles). Retrieved September 20, 2020, from <https://vocab.org/aiiso-roles/schema>
- [12] Styles, R., Wallace, C., & Moeller, K. (2008, September 25). Participation Schema. Retrieved September 16, 2020, from <https://vocab.org/participation/schema>
- [13] Xiaowen Huang National Laboratory of Pattern Recognition, et al. "Explainable Interaction-Driven User Modeling over Knowledge Graph for Sequential Recommendation." Explainable Interaction-Driven User Modeling over Knowledge Graph for Sequential Recommendation. Proceedings of the 27th ACM International Conference on Multimedia, 1 Oct. 2019, dl.acm.org/doi/10.1145/3343031.3350893.
- [14] Zhang, Yongfeng and X. Chen. "Explainable Recommendation: A Survey and New Perspectives." ArXiv abs/1804.11192, 2020.
- [15] Katis, Evangelos & Kondylakis, Haridimos & Agathangelos, Giannis & Kostas, Vassilakis. (2018). Developing an Ontology for Curriculum & Syllabus.
- [16] "Ontologies - Curriculum Ontology." BBC, BBC, www.bbc.co.uk/ontologies/curriculum. (2013).