ISSN 2278-3091
Volume 12, No.5, September - October 2023
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse021252023.pdf
https://doi.org/10.30534/ijatcse/2023/021252023

Supporting Curriculum Designers in Developing Balanced Outcome-based Programs using Knowledge Graphs

Ghadah Aldhalaan, Jawad Berri
Department of Information Systems
College of Computer and Information Sciences,
King Saud University, Riyadh Saudi Arabia
441200058@ksu.edu.sa, jberri@ksu.edu.sa

Received Date : August 29, 2023 Accepted Date: September 24, 2023 Published Date: October 06, 2023

ABSTRACT

It is essential for universities to accredit their programs to be recognized locally and globally and to maintain the quality and credibility of educational programs. For programs to become accredited, they must meet predefined quality standards for the intended accreditation institution. These quality standards are applied to program courses, and each course covers some requirements from the accreditation requirements, such as Student and Course Learning Outcomes, based on their specific accreditation commission. As programs and courses are added frequently at university programs, information about accreditation mapped to each course syllabus and program description is becoming more complicated. It must be maintained automatically since program accreditation must be renewed on a regular basis to guarantee the program's quality. In addition, supporting Curriculum Designers in finding gaps and imbalances in the proposed programs and courses is becoming necessary to alleviate the management of accreditation related tasks. In this paper, we propose a knowledge graph-based system that supports the development of balanced educational programs and helps detecting gaps and redundancies within the same university programs. The system provides an attractive graphical user interface that visualizes the curriculum components and accreditation requirements as interactive graphs. The system has been implemented using the graph database Neo4j and the results are analyzed using the Cypher query language.

Key words : Accreditation, Curriculum design, Knowledge Graph, Outcome Based Education.

1. INTRODUCTION

The interaction between different courses under the same or different programs at universities is becoming more complex as the current university curriculums' structure continues to expand and improve. Nowadays universities are accrediting their programs to ensure their quality, and these requirements are included in the curriculums as constraints which makes their design more complex and needs an efficient design to avoid deficiencies within a program but also within the same university curriculum programs.

Accredited programs must meet predefined quality standards for the intended accreditation institution. These quality standards are applied to program courses, and each course covers some requirements from the accreditation requirements, such as Student Outcomes (SO) and Course Learning Outcomes (CLO), based on their specific accreditation commission. Accreditation often involves a commitment to continuous improvement. Institutions must demonstrate not only that they meet minimum standards but also that they are actively working to enhance the quality of their programs. This requires a commitment to continuous improvement, which can be challenging to establish and maintain. Accreditation standards can change over time, and institutions must adapt to these changes. As programs and courses are added frequently at university programs, information about accreditation mapped to each course syllabus and program description is becoming more complicated. It must be maintained automatically since program accreditation must be renewed on a regular basis to guarantee the program's quality.

Moreover, curriculum designers face difficulties in discovering gaps and imbalances in the curriculum and need to be supported by specialized tools that help them achieve this goal. To fill these gaps, we introduce a knowledge graph-based system that captures, incorporates, and reviews relationships in detail, particularly underlying relationships hidden behind curriculums, and can then implement visual presentations of the complex relationships. The system provides an attractive graphical user interface that visualizes the curriculum components and accreditation requirements as interactive graphs allowing curriculum designers to develop and refine curriculums easily.

This paper is organized as follows: The next section provides a background of some important aspects and terminologies underlying this research. The third section provides an overview of relevant and related studies. Section four explains the methodology for implementing the knowledge graph-based system. Section five is a case study that illustrates approach using the knowledge graph constructed for the College of Computer and Information Science at King Saud University. In the concluding section, the findings of this study are presented, along with suggestions for future research directions.

2. BACKGROUND

To further understand the topic and its significance, this section presents the background knowledge of the research discussed throughout this paper.

2.1 Outcome Based Education

Outcome-based education (OBE) is an academic theory that centers each component of a school system on a set of objectives (outcomes) [1]. Each student should have accomplished the outcomes by the end of the educational period. Teachers expect students to exhibit demonstrable learning outcomes at the end of key learning experiences. Values, ideas, emotions, and psychological states of mind aren't among the outcomes. Rather, outcomes are measurable objectives of what students could do with what they understand and have achieved.

2.2 Program Accreditation

Program accreditation is a process that determines whether educational programs meet predefined quality standards. Accreditation is not permanent; it must be renewed on a regular basis to guarantee that the academic program's quality is well maintained 2. Accreditation standards specifically target student learning outcomes and student accomplishment as part of their measures.

2.3 Curriculum Design

Curriculum design is the process of planning and creating an educational curriculum, which is a structured set of learning experiences and content designed to achieve specific learning outcomes. Effective curriculum design is essential for providing a meaningful and organized educational experience for students. This process requires regular updates and revisions to keep it current and effective. It should be a dynamic and responsive process that evolves to meet the changing needs of students, to cope with latest technologies and the educational environment.

Furthermore, many university programs are accredited which puts more constraints on curriculums to meet quality standards. Managing curriculum outcomes at the program level, college and university levels requires nowadays fine management to detect overlaps and lacks of specific student outcomes in order to design efficient and optimized curriculums to save student time and decreases university budgets.

2.4 Knowledge Graphs

A knowledge graph is a representation of information that includes entities, relationships, and descriptions. Entities can be physical objects or abstract concepts, and relationships indicate the interaction between entities. Nodes represents entities while edges represent relationships between entities [3], [44].

As a result, the construction of a knowledge graph will greatly facilitate the accreditation curriculum maintainability, as well as make it easier for faculty to review, consult, and comprehend connected courses and outcomes across departments and view opportunities of improvements. Also constructing this knowledge graph will allow quality assessors to check the consistency of many aspects of different programs such as student and course outcomes and checks if every Course Learning Outcome (CLO) is associated with a Student Outcome (SO) and detects deficiencies and missing relationships. Furthermore, knowledge graphs facilitate discovering errors and unmatched entries of syllabi and manual entries and allow detecting imbalances in the distribution of CLOs an SOs.

3. RELATED WORK

Knowledge graphs have gained popularity in recent years due to their ability to represent complex relations between data. Knowledge graphs are semantic graphs that can capture subtle meanings that can inhibit inference in knowledge management systems. The development of applications using knowledge graphs enhances intelligent systems that aim to reason and recommend knowledge. By incorporating knowledge graphs into educational systems, institutions and educators can better harness the power of data and knowledge representation to create more effective and personalized learning experiences for students at all levels of education. Furthermore, educators and institutions can use knowledge graphs to design and optimize curriculum structures. By mapping out the relationships between course concepts and student outcomes, they can ensure that educational content is logically and cohesively organized within the curriculums [5], [6].

Aliyu et al. applied the knowledge graph to the education domain to support the management of courses. The relationship between courses and lecturers has been established in an Ad-hoc manner during the allocation of courses at the beginning of the semester. To test the functionality of this system, data from the Department of

Mathematical Sciences of ATBU was used. The system turned out to be successful in answering the questions like who taught a specific course in 2014 in the specific semester [7].

Yu et.al constructed a curriculum system that is based on a knowledge graph. To find the relationship between different university courses, the design and implementation are based on the Computer Science Department Syllabus of Miami University. The methodology for this graph system includes components: Constructing a knowledge graph model, acquiring multi-score entity knowledge, entity-relationship, and mapping of the knowledge graph [8].

In recent years, a massive development has been witnessed in MOOCs which stands for massive online open courses. This is a new model of learning and offers courses from top universities to learners in a friendlier format than traditional learning. Fu-Rong Dang et al. constructed a knowledge graph to describe the relationship among concepts and entities in the objective world in a structured way. In this study, the entities have been extracted and have been linked with Wikipedia entries and more [9].

Qin et al. have used knowledge graphs for knowledge analysis management. As database technology improves, so does knowledge about databases. The use of knowledge graphs in databases can also help teachers and students in the teaching and learning process and effectively improve the quality of teaching and improving learning and teaching in the field of database intelligence [10].

In order to recommend online courses, it is important to identify the long-term learning interests of users. Previous studies have shown that people's interest in learning changes dynamically over time. The long-term learning interests of online users can take two forms: changing interests and evolving interests. Changing interests reflect the dynamism and diversity of users' learning interests, even though users' interests remain stable over time. Knowledge graphs can create explicit and irrational knowledge relationships between courses, but a recommender system using knowledge graphs can better reflect and intelligently extend the semantic relationships between courses that motivate long-term learning interests. The proposed solution by Zhou et al. [11] can be used in online platforms to recommend suitable courses based on learners' dynamic interests, which will increase user engagement and reduce dropout rates. Furthermore, the proposed model can be used in online platforms to increase user engagement and reduce dropout rates.

4. SYSTEM DESIGN AND METHODOLOGY

In this section we present the system design and the methodology that has been developed to check inconsistencies in the design of curriculums.

4.1 System Architecture

The system's architecture is designed as a 3-tier architecture based on a client-Server system. The architecture and it components is shown in Figure 1.

- **Presentation Tier:** Represents the graphical user interface that the clients such as Faculty members, Curriculum Designers, and Reviewers from accreditation institutes will interact with through the web browser using a PC, mobile phone, or tablet.
- **Business Logic Tier:** Defines the functionalities of the system and includes Knowledge acquisition, preprocessing, knowledge extraction the components that make up the construction process of the Accreditation Curriculum Knowledge Graph.
- **Data Management Tier:** contains the database server where the system's information is stored.

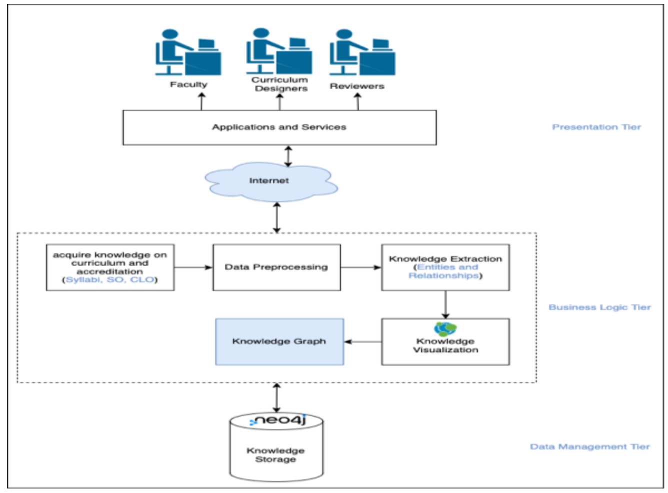

Figure 1: System Architecture

4.2 Query Design

We used predicate logic to express and formalize selected examples of queries that will be used later in the case study section. Predicates are functions that return Boolean values and have zero or more variables. As a result, and depending upon the value of its arguments, predicates could be true or false at any point 12[12]. In order to support the curriculum designer in his task to design and check any redundancies and inconsistencies, a set of queries have been designed during the elicitation phase. We present some of these queries and the corresponding predicate logic expressions in Table 1.

1. Retrieving courses that are prerequisite to other courses.
2. Retrieving all courses that cover a student outcome.
3. Returning all the Course Learning Outcomes of a specific program course, and if there are any prerequisite courses included.
4. Returning all the Course Learning Outcomes and Student Outcomes of some department courses, and if there are any prerequisite courses included.

Table 1: Predicate Logic Query Design

Query#	Predicate Logic Query Expression
1	*∃x, y (Course. name(x, c) ∧ prerequisites(x, y) ∧ c = 'Course Name' ∧ x≠y)*
2	*∃x (Course(x) → ∀y (CLO(y, x) ∧ (SO.name(y, c) ∧ c = "Student Outcome Code")))*
3	*∃x (Course(x) ∧ contains (x, c) ∧ c = 'Department Code Name' → ∀y (CLO (y) ∧ prerequisites(x, y))*
4	*∃x (Course(x) ∧ contains(x, c) ∧ c = 'Department Code Name' → ∀y, z (CLO (y) ∧ SO (z) prerequisites(x, y) ∧ covers(y, z))*

4.3 Knowledge Acquisition

The data used in for this research to develop a knowledge graph comes primarily from numerous course syllabi collected from the College of Computer and Information Science at King Saud University (CCIS - KSU). Course syllabi were collected from the following department programs that are Bachelors of Information Systems and Software Engineering. The two programs have common courses and both are ABET accredited. Our objective is to gain insight into which student outcomes and courses are related across the two programs which helps departments and the College improving the quality of their programs and identify redundancies and inconsistencies.

4.4 Preprocessing

Syllabi collected in a document form such as pdf and Microsoft word documents, information such as Course Name, Course Learning Outcomes, Student Outcomes, Credit Hours, Description, and Prerequisite courses are extracted from the documents. Some documents had missing data and are solved by imputation. During the data entry process, there were clear view that some of discrepancies and errors in the syllabi, such as a misspelling in the course contents, the absence of important relationships such as the connection between courses, CLOs and SOs, or other issues related to missing information.

4.5 Knowledge Extraction

Figure 2 represents the model including the concepts and relationships used in our system. It is as four entities, including their relationships and descriptions.
The model concepts are:

- **Program:** The program that offers the courses, in this project is the college of computer and information science for example, information systems program.
- **Course:** Courses that students need to pass in order to obtain their degree, named by the course code, for example "IS230: for database systems course". And contains other properties such as credit hours and details.
- **Course Learning outcome:** Defines the performances that faculty will evaluate to assess whether a student outcome has been met or not.
- **Student Outcome:** Specify what students should know and be able to do by the time they graduate. These are the abilities, knowledge, and behaviors that students gain as they go through the curriculum.

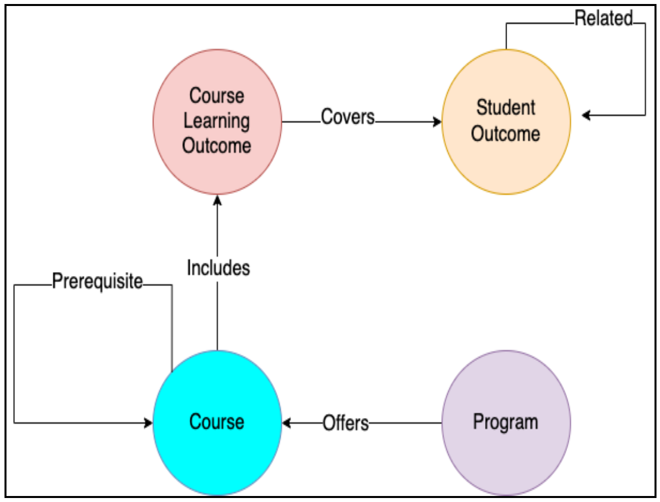

Figure 2: Entities and Relationships

The model includes the following relationships:

- **Related:** A student Outcome can be related to other SO from other commissions.
- **Covers:** A Course Learning Outcome covers a Student outcome within a course. It is noted that it is possible that more than one CLO covers the same SO within the same course.
- **Includes:** A Course includes one or more Course Learning Outcomes that are unique to the course.
- **Offers:** A Program offers a variety of courses.
- **Prerequisites:** A course can be a prerequisite for another course.

4.6 Knowledge Storage

After completing the preceding phases, we were able to discover the relationships between all entities in syllabi and the data is mapped into an appropriate knowledge graph using the right tools. We chose Neo4j [13], a prominent graphical database, to complete and store the data and construct the knowledge graph. Neo4j graph database offers traversals across complex hierarchies, find hidden relationships between items that are far apart, and explore the connections between nodes.

5. CASE STUDY

We constructed the knowledge graph as a scenario using the prior sections as well as data from programs at College of Computer and Information Sciences at King Saud University with around 60 nodes, and 84 relationships to try to spot problems and detect if there are any imbalances and issues of some of the programs at KSU.

5.1 Queries and Knowledge Visualization

For querying knowledge graphs, a variety of languages have been considered. In this paper we used the Cypher query language [14] as our language of choice, due to its relation to other languages such as SQL and Neo4j database.

Demonstrating examples of search results for example finding a specific student outcome and understanding which Course outcomes is related to it, in addition to viewing what course covers both the student and course learning outcomes. We listed some examples of queries below. The purpose of visualization is to demonstrate the findings of the current queries, such as the courses, relationships between nodes, CLOs, and other insights. We want to use these visuals to discover the meaningful relationship between courses and their outcomes and detect imbalances if any.

5.2 Queries

Four queries are proposed in this section to illustrate how redundancies and inconsistencies can be detected from a specific course. Tables 2, 3, 4 and 5 present the queries in predicate form and then encoded into Cypher. Also, Figures 3, 4, 5 and 6 display the result graphs of the queries defined.

1. Retrieving courses that are prerequisite to other courses. In this example, we are interested to retrieve the prerequisite courses of the course IS324 "Modern Applications Development". The query retrieved only one prerequisite course that is IS230 "Introduction to Database Systems".

Table 2: Query 1

Query Type	Statements
Predicate logic	*∃x, y(Course.name(x, n) ⋀ prerequisites(x, y) ⋀ n = 'IS324' ⋀ x ✔≠y)*
Cypher	***MATCH*** *(c1:course{name:'IS324'})-[r:prerequisite]->(c2:course)* ***RETURN*** *c1,c2*

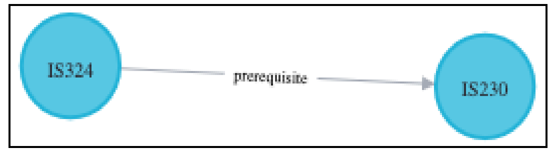

Figure 3: Results of Query 1

2. The second query retrieves all courses that cover Student outcome 6 "SO6 - Support the delivery, use, and management of information systems within an information systems environment."

Table 3: Query 2

Query Type	Statements
Predicate logic	*∃x(Course(x) →∀y(CLO(y, x) ⋀(SO.name(y, c) ⋀ c = "SO6")))*
Cypher	***MATCH*** *(c:course)-[:includes]->(cl:CLO), (cl)-[:Covers]->(s:SO)* ***WHERE*** *s.Name="SO6"* ***RETURN*** *c,cl,s*

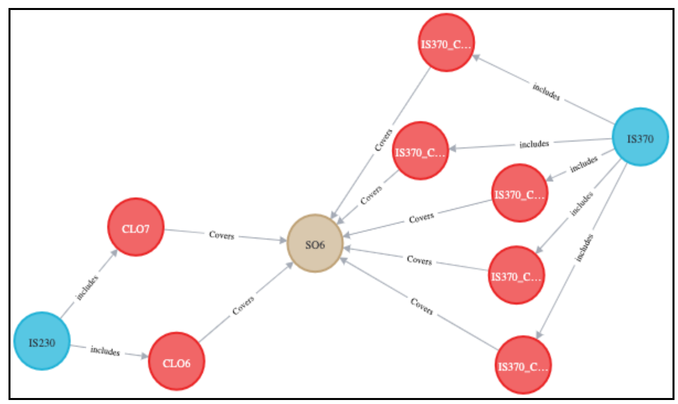

Figure 4: Results of Query 2

3. Query 3 aims to return all the Course Learning Outcomes of a subset of courses in the Information Systems department. The query retrieves and displays also any prerequisite courses of the courses retrieved.

Table 4: Query 3

Query Type	Statements
Predicate logic	*∃x, y(course.name(x, n) ⋀(CLO.name(y, c) ⋀ c = "IS") ⋀)prerequisites(x, y) ⋀ n = 'IS324'))*
Cypher	***MATCH*** *(c:course)-[r:includes]->(cl:CLO)* ***WHERE*** *c.name* ***CONTAINS*** *'IS'* ***RETURN*** *c, cl*

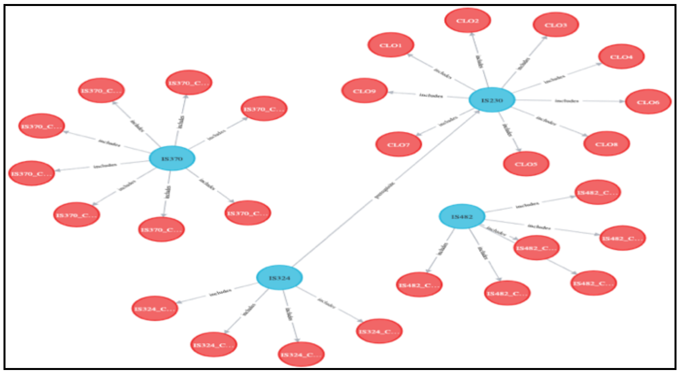

Figure 5: Results of Query 3

4. Additionally, in relation to the previous search result, returning all the Course Learning Outcomes and Student Outcomes of the Information Systems department courses, and if there are any prerequisite courses included.

Table 5: Query 4

Query Type	Statements
Predicate logic	*∃x, y (course.name(x, n) ∧ prerequisites(x, d) ∧ n = 'IS324'))*
Cypher	***MATCH** (c:course)-[:includes]->(cl:CLO), (cl)-[:Covers]->(s:SO)* ***WHERE** c.name **CONTAINS** 'IS'* ***RETURN** c,cl,s*

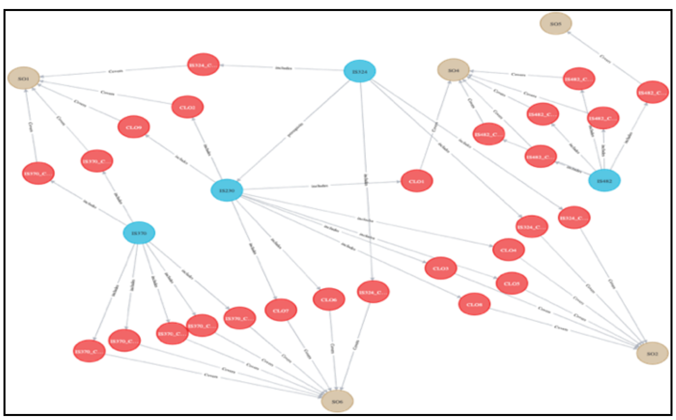

Figure 6: Results of Query 4

5.3 Outcomes

From the knowledge graph in Figure 7 we can observe that some courses cover more Student Outcomes than others and there is an imbalance between the different relationships, for example Student Outcome 5 is only covered by one Course Learning Outcome and Student Outcome 6 is covered by 7 Course Learning Outcomes.

Constructing this knowledge graph allowed us to identify many imbalances of SOs and CLOs and inconsistencies in programs that can be discussed with instructors and course designers and ultimately update and propose solutions to have a better design of courses and programs. Some of these remarks are explained in the following points:

- Checking if every course and Course Learning Outcome is associated with a Student Outcome, in addition detecting missing relationships.
- Detecting imbalances in the distribution of CLOs an SOs. For example, in Figure 4 SO6 is covered by two CLOs from the course IS 230 on the other hand SO6 is covered by five CLOs from the course IS 370 and the same thing goes for Figure 7, SO 5 is covered by one CLO while SO 5 is covered by seven CLOs.
- Usually going through syllabi manually without a computed graph representation leads to difficulties detecting errors and unmatched entries that are solved by constructing this knowledge graph and standardizing the content of every course syllabus.
- Supports quality control committees to assess how well their programs are covering the requirements of the accreditation programs.

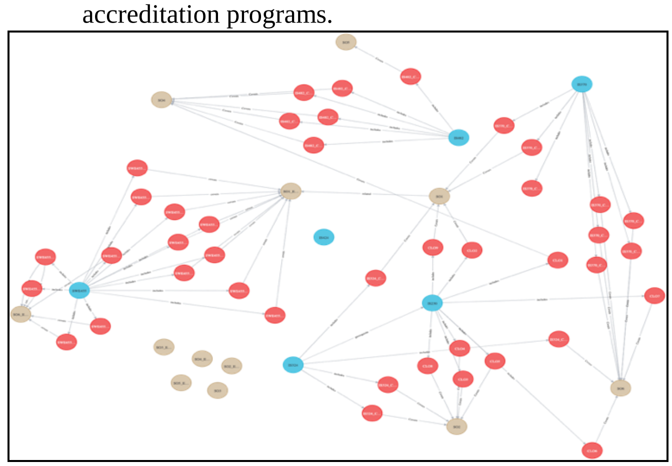

Figure 7: Knowledge Graph Overview.

5.4 Discussion

The system developed includes a representation of curriculum programs in knowledge graphs and a set of queries which are used to retrieve information about CLOs, SOs and courses in general in order to check errors, redundancies and inconsistencies within a program. These queries can be integrated within more complex modules to check fine gaps that cannot be detected manually at the program or College levels. For instance, there are some SOs that need to be covered for each level in the program. Also, some SOs may be addressed by many courses and should be restricted to only the most relevant ones. The extension of the system is currently considering identifying the needs of curriculum accreditation from curriculum designers and specialists in accreditation. Their expertise needs to be gathered and addressed in the next release of the system which will include modules that assist them in their task to develop programs complying with accreditation requirements at the program and institution levels.

6. CONCLUSION

This paper presented a system that addresses curriculum accreditation using knowledge graphs. The developed system constructs a knowledge graph to represent and store detailed information extracted from course syllabi. Also, the system includes a set of queries to retrieve information related to courses and programs that allows identifying redundancies and inconsistencies. The application of the knowledge graph resulted in unifying the syllabi and ensuring the completion of all the accreditation requirements.

Suggestions for future research consist in expanding the knowledge graph including comprehensive data from other programs to provide more insights, improvement opportunities, and detect imbalances across programs within the same institution which helps provide a bigger picture for the accreditation committees and authoritative faculties to make relevant changes to the curriculum. The system extension focuses on introducing specialized modules that integrate basic queries to retrieve and display information at a larger scale than the program allowing curriculum designers to address gaps at the institutions' level. Furthermore, including syllabi from other universities into a one integrated graph would allow to benefit accreditation institutes to monitor multiple universities programs.

REFERENCES

1. K., Premalatha. **Course and program outcomes assessment methods in outcome-based education: A Review.** *Journal of Education*, vol. 199, no. 3, pp. 111–127. 2019.
2. ABET, **What Is Accreditation?** Retrieved April 18, 2022, from https://www.abet.org/accreditation/what-is-accreditation/eligibility-requirements/
3. A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, J. E. Labra Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A.-C. Ngonga Ngomo, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, **Knowledge Graphs**, *ACM Computing Surveys*, vol. 54, no 4: 71:1-71:37, 2021.
4. S. Ji, S. Pan, E. Cambria, P. Marttinen and P. S. Yu, **A Survey on Knowledge Graphs: Representation, Acquisition, and Applications**, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494-514, Feb. 2022.
5. P. Chen, Y. Lu, V. W. Zheng, X. Chen, B. Yang, **KnowEdu: A System to Construct Knowledge Graph for Education**, *IEEE Access*, vol. 6, pp. 31553-31563, May 2018.
6. J. Berri, **Knowledge Graph-based Framework for Domain Expertise Elicitation and Reuse in E-Learning**, *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 12, pp. 291-296, 2021.
7. A., Ismail, D., Kana, A., Salisu. **Development of Knowledge Graph for University Courses Management**, *International Journal of Education and Management Engineering*, vol. 10. pp. 1-10, 2020.
8. Y., Xiaobing, S., Mike, C., Han, Y., Runming. **Design and Implementation of Curriculum System Based on Knowledge Graph**, in *Proc IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, pp. 767-770, 2021.
9. F.-R., Dang, J.-T., Tang, K.-Y., Pang, T., Wang, S.-S., Li, X., Li. **Constructing an educational knowledge graph with concepts linked to Wikipedia**, *Journal of Computer Science and Technology*, vol. 36, no. 5, pp. 1200–1211, 2021.
10. Y., Qin, H., Cao, L., Xue. **Research and application of knowledge graph in teaching: Take the database course as an example**, *Journal of Physics: Conference Series*, vol. 1607, no. 1, 012127, 2020.
11. J., Zhou, G., Jiang, W., Du, et al. **Profiling temporal learning interests with time-aware transformers and knowledge graph for online course recommendation**. *Electron Commer Res*, 2022. https://doi.org/10.1007/s10660-022-09541-z
12. **Predicate logic** - the Stanford University infolab. (n.d.). Retrieved May 25, 2022, from http://infolab.stanford.edu/~ullman/focs/ch14.pdf
13. **What is a graph database?** - developer guides. Neo4j Graph Data Platform. (n.d.). Retrieved May 10, 2022, from https://neo4j.com/developer/graph-database/
14. **Cypher query language - developer guides.** Neo4j Graph Data Platform. (n.d.). Retrieved May 10, 2022, from https://neo4j.com/developer/cypher/