An ACM\IEEE and ABET Compliant Curriculum and Accreditation Management Framework

Manar Salamah Ali
Computer Science Department
King Abdulaziz University
Jeddah, Saudi Arabia

***Abstract*—Following methodological and systemized approaches in creating course syllabi and program curriculums are very crucial for assuring the coherence (correctness, completeness, consistency, and validity) of curriculums. Furthermore, designing coherent curriculums have a direct impact on achieving curriculum outcomes. For institutions seeking accreditation, presenting evidence of curriculum coherence is mandatory. In this paper, a general framework architecture for curriculum and accreditation management is proposed. Furthermore, we propose a detailed design for a knowledge base that comprises of: a) the ACM\IEEE body of knowledge for the Computer Science Department, b) course syllabi, and c) course articulation matrices. We show how to utilize the proposed knowledge base in the quality improvement life cycle, in ABET accreditation, and as a significant step towards curriculum coherence.**

***Keywords*—Curriculum coherence; body of knowledge; accreditation; knowledge base design; ABET**

I. INTRODUCTION

In recent years, educational institutions are much concerned with following world standards for creating their educational program curriculums. Also, most institutions are seeking world accreditations from world accrediting institutions such as the Accreditation Council for Business Schools and Programs (ACBSP) [1] or the Accreditation Board for Engineering and Technology (ABET) [2].

The Body of Knowledge (BoK) in undergraduate programs provides core knowledge areas in the field and guidelines for creating curriculums. It also provides a detailed specification of what content should be included in an undergraduate program [3].

Periodical assessment of curriculums based on learning outcomes, accreditation standards, technological changes, and professional requirements benefits significantly from the bodies of knowledge [22, 23]. Furthermore, coherent alignment of curriculum components such as learning outcomes, assessment methods, teaching methods, and program outcomes increases student success as much as two standard deviations [13].

For these reasons, it is necessary to follow methodological approaches for the course and curriculum designs. These approaches must be compliant with the standards of the bodies of knowledge and accrediting institutions. Furthermore, automating the necessary processes for creating curriculums and evaluating its outcomes is significant for accurate and consistent assessments and evaluations.

A plethora of approaches for assessing and evaluating curriculum content and outcomes can be found in the literature. Nevertheless, most of these approaches assume manual-conducted processes when it comes to building curriculum content, which can result in the poor design of curriculums and jeopardize its coherence. There appears to be a lack of focus on proposing methods that measure the extent of compatibility and alignment of the curriculum content with the BoK. The BoK provides a consistent platform for curriculum and student outcome evaluation. Hence, there is a need for management frameworks that provide holistic and systemized solutions and utilize the BoK as a source for curriculum design, assessment, and accreditation processes.

The purpose of this work is three-fold. The first objective is to introduce a BoK-based curriculum management framework purposely designed to fine-tune the coherence of program curricula. The second objective is to provide the design of a knowledge-based system for supporting curriculum and accreditation management. The third objective is to demonstrate the use of the proposed management framework for accreditation and continuous improvement of the curriculum.

The first computer science BoK, as a standard, was delivered as a result of a joint task between ACM and IEEE organizations back in June of 1882 as an attempt to set standards and facilitate the assessment of educational programs. Since then, the ACM\IEEE produced three consecutive standards for the CS program: 2001, 2008, and 2013, respectively.

In this paper, we provide general framework architecture for managing the CS2013-BoK, course\curriculum design, and course articulation matrices. An articulation matrix design is proposed as the first step towards fully automated accreditation management. In this work, we propose a general knowledge base that is necessary for the continuous quality improvement of curriculums and for accreditation tasks. The work in this paper is compliant with both ACM|IEEE and ABET standards. The proposed management framework and the knowledge base will have a direct impact on curriculum coherence and a direct role in auto-generating reports for ABET: criteria 3 = “Student Outcomes”, 4 = “Continuous Improvement”, and 5 = “Curriculum” of the Self Study Report (SSR) [4].

The paper is organized as follows: Literature review is presented in Section II. The primary motivation of this research is discussed in Section III. An enhanced continuous improvement cycle model is introduced in Section IV. The proposed framework architecture is discussed in Section V. In Sections VI and VII, the requirements and design of the BoK and the course design modules are discussed. The articulation matrix as a tool for assessment is discussed in Section VIII. Related systems and conclusions are discussed in Sections IX and X, respectively.

II. LITERATURE REVIEW

The quality of academic programs is highly related to efficient and coherent curriculums where optimal content delivery is achieved[12]. Also, empirical results show that there is a direct relationship between curriculum coherence and student achievements of outcomes [13, 16, 24]. Many factors directly affect curriculum coherence, such as: 1) misalignment of program outcomes and accumulative course learning outcomes, 2) content redundancy, 3) content inconsistency, and 4) incomplete content with regard to core knowledge in the underlying area and required competencies. Achieving curriculum coherence requires different types of analysis applied to its content to ensure that content gaps and overlaps are addressed and remedied.

Applying improvements to the content of curriculums is driven by two interrelated processes: continuous improvement process and accreditation. The continuous improvement process is an ongoing sustainable method to improve the overall quality of an academic program in terms of the performance of individual courses, student learning, and program outcomes. The improvement process is incremental and iterative. According to ABET, the cycle encompasses iterative enactments of three main tasks: assessment, evaluation, and improvement. In assessment, data that is necessary for evaluation is collected from different resources such as direct and indirect assessment methods. In evaluation, the collected data is interpreted and analyzed to measure the level of student attainment with regard to outcomes. Consequently, improvements are applied at the course and program level according to the evaluation results [14]. At the course level or curriculum level, improvements may necessitate revisiting course topics and learning outcomes when a lack of curriculum coherence is evident.

Accreditation is a review process where institutions are required to provide evidence that their educational programs meet defined standards of quality. The continuous improvement of curriculums is an integral part of accreditation. For example, institutions must provide proof of the different analysis methods they are using to improve the quality of their program curriculums.

One of the main approaches for increasing the quality of education is the standardization of bodies of knowledge from which program curriculums are constructed. Some examples include Common BoK (CBK) for security management[5], the Business Analysis BoK (BABOK)[6], Landscape Architecture BoK (LABOK)[7], and ACM\IEEE (CS2013) for computer science[3].

Different methods have been proposed to analyze and assess curriculum content. For example, in [21], the authors used a network model to represent curriculum mapping. Curriculum mapping is a process used to identify gaps, overlaps, and misalignments by indexing the curriculum's constituent entities and applying appropriate methods to analyze its content. In comparison, the work in [19] follows a mixed method for evaluating program performance using tree representations for different curriculum and program associated entities. The authors in [25] introduced a risk management framework for providing data-driven decision-making to curriculum and program quality improvement. Curriculum coherence is evaluated in [12] using ontologies and natural language processing techniques.

Current approaches and analysis techniques focus on the existing content of curriculums without measuring the compatibility of the content with the standard requirements of the scientific knowledge in the underlying domain. The alignment of the curriculum with the BoK is significant to the development and improvement processes [18]. As a result, there is an evident gap in the literature. There is a need for proposing general and flexible frameworks, which uses the BoK as the source for curriculum contents and a benchmark for any further improvements and assessments.

III. MOTIVATION

The primary motivation of this work is to encourage curriculum and accreditation committees within academic programs to: 1) integrate the BoK in their processes of course and curriculum design, 2) create a knowledge base for the BoK, and 3) automate the process of course and curriculum design, 4) automate the process of assessment, evaluation, and improvement, and 5) enrich the knowledge base with periodical assessment and evaluation data.

The manual construction or update of the course\curriculum design might suffer from the following serious problems:

1) Curriculum incoherence: the curriculum can suffer from inconsistencies, redundancies, or incompleteness in terms of topics and learning outcomes. For example, the curriculum might suffer from overlapped topics, outdated topics, or missing core topics.

2) Difficulty or possible inaccuracy in retrieving statistics about courses, topics, or outcomes.

3) Difficulty or possible inaccuracy in writing ABET Criterion 3, 4, and 5 in the SSR report.

4) Potential difficulty\inaccuracy in complying with a BoK standard such as ACM\IEEE.

5) Difficulty in regular revisions of the program curricula

The first problem could be caused by course designers during the process of course design, while problems 2-5 are challenging issues that face program accreditation committees and quality assurance personnel.

To prevent such potential problems and inaccuracies, it is necessary to automate the process of creating the course design. In addition, automation is an essential step towards the full automation of the assessment and evaluation processes.

In the following sections, the proposed framework architecture and design are presented.

IV. CONTINUOUS IMPROVEMENT CYCLE

In this work, we suggest an enhanced model of the continuous improvement cycle. As seen in Fig. 1, the model includes the BoK as a compulsory source for the course and curriculum improvements. The BoK acts as a source for core knowledge areas, topics, and learning outcomes as well as a guideline for coherent and correct curricula.

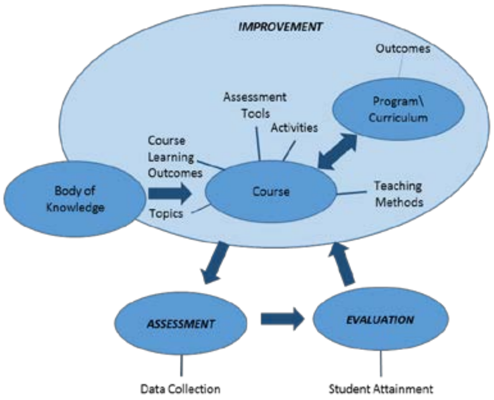


Fig. 1. An Enhanced Continuous Improvement Cycle.

V. FRAMEWORK ARCHITECTURE

The necessary modules of the proposed framework are illustrated in Fig. 2. In the BoK module, the content of the BoK is stored in a database. The curriculum builder module automates the process of designing courses and stores individual course designs in a database. These modules must be supported with flexible and user-friendly interfaces for entering and editing data related to the BoK and course design. The stored data will hold as a knowledge base for assessment and evaluation-related tasks. The accreditation management module may include different tools that support the accreditation process according to institutions' needs.

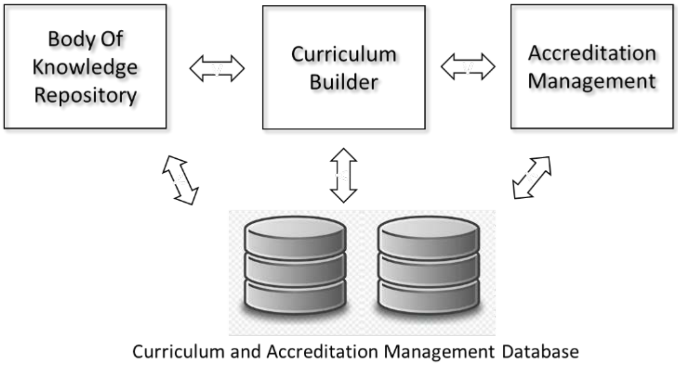


Fig. 2. Framework Architecture.

A group of senior students has developed a prototype system for storing the BoK and for building course syllabi (see Acknowledgement). The system was developed on Microsoft SQL Server 2014 Management Studio, user interfaces designed with Visual Studio 2015, and the back-code programming in ASP.Net. Snapshots of the system are listed in Appendix A (Fig. 9 to 11).

For increased generalization, in this paper, we provide a general Database (DB) design for the BoK and curriculum builder modules to help interested developers implement them in their chosen platform environment. The proposed design is applicable to CS programs. However, it can be easily modified to serve other programs such as Information Technology, Information systems, Software Engineering, or Computer Engineering.

In the following sections, the design of the BoK and the curriculum builder are discussed in detail. For each module, the data requirements, the DB design in ER diagrams and relational notation, and implementation tips are provided.

VI. BODY OF KNOWLEDGE REPOSITORY MODULE

A. Data Requirements

The data requirements for the BoK has been elicited from the computer science CS2013 report [3]. The detailed discussion of the requirements is listed in the following subsections.

1) Body of knowledge structure

In the latest CS2013 report, the CS BoK is organized into a hierarchical structure. The BoK has 18 Knowledge Areas (KA), and each KA is decomposed into Knowledge Units (KU). Within each KU a set of topics and their expected Course Learning Outcomes (CLO) are provided. The topics and their CLOs are categorized into core-tier1, core-tier2, and electives. The CS2013 guidelines offer great flexibility in consolidating topics into courses and curricula, and institutions must develop their methods of combining topics from the BoK into courses.

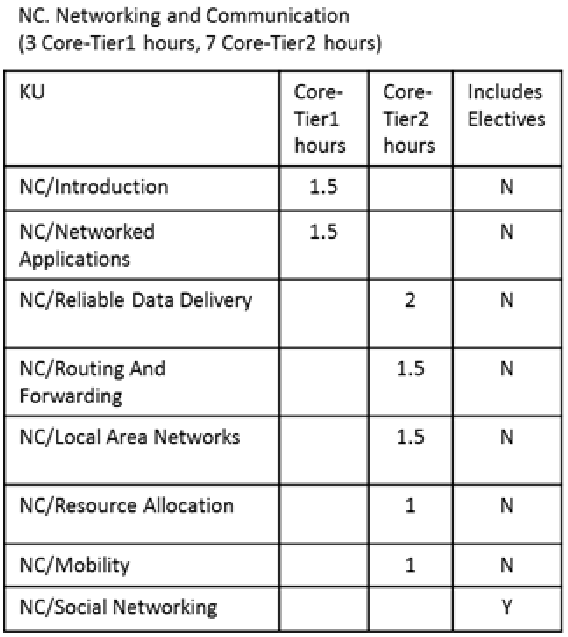
The curriculum-topic recommended coverage is as follows: 100% coverage of topics in core-tier-1, 90-100% coverage of topics in core-tier-2, and significant coverage of elective topics.

2) Topic coverage

The basic unit for topic coverage is an “hour”, which represents the time required to present the topic material in a traditional lecture. For each KA and KU, the minimum required coverage hours for tier1 and tier2 are provided. In addition, it is stated whether the KU has elective topics or not. The topic coverage scheme is illustrated in Fig. 3.

3) Mastery of CLOs

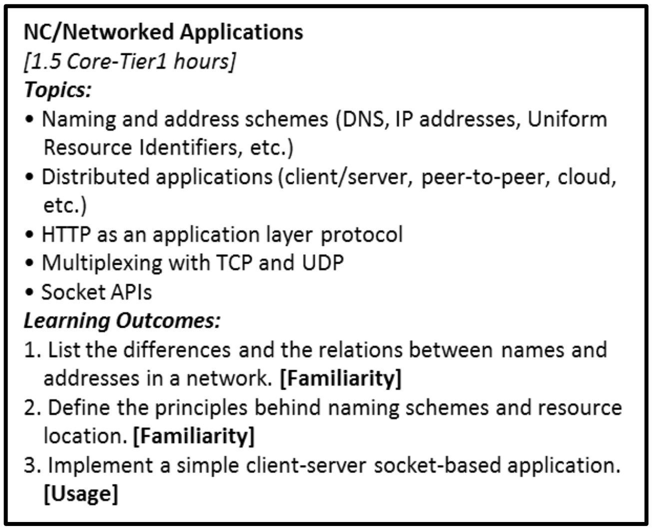
The CLOs in the CS2013 are classified according to three mastery levels: familiarity, Usage, or Assessment. However, each institution can apply different classification for the CLO mastery levels, such as blooms taxonomy. In Fig. 4, the topics and CLOs of the “Networked Applications” Knowledge Unit is demonstrated.



NC. Networking and Communication
(3 Core-Tier1 hours, 7 Core-Tier2 hours)

KU	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
NC/Introduction	1.5		N
NC/Networked Applications	1.5		N
NC/Reliable Data Delivery		2	N
NC/Routing And Forwarding		1.5	N
NC/Local Area Networks		1.5	N
NC/Resource Allocation		1	N
NC/Mobility		1	N
NC/Social Networking			Y

Fig. 3. Coverage hours for KA=”NC” and its Constituent KUs. (CS2013).



NC/Networked Applications
[1.5 Core-Tier1 hours]
Topics:

- Naming and address schemes (DNS, IP addresses, Uniform Resource Identifiers, etc.)
- Distributed applications (client/server, peer-to-peer, cloud, etc.)
- HTTP as an application layer protocol
- Multiplexing with TCP and UDP
- Socket APIs

Learning Outcomes:

1. List the differences and the relations between names and addresses in a network. **[Familiarity]**
2. Define the principles behind naming schemes and resource location. **[Familiarity]**
3. Implement a simple client-server socket-based application. **[Usage]**

Fig. 4. Topics and CLOs in KA=”NC and KU=”Networked Applications”. (CS2013).

4) Cross referencing

In the BoK, topics can be shared between more than one KU. Hence, some of the KUs\topics are cross-referenced with other KUs in the BoK. For example, the Information Assurance and Security (IAS) Knowledge Unit has 9 combined core hours. However, there are 63.5 more hours distributed over other KUs such as OS\security or SDF\Development Methods. Hence, related and relevant topics may exist in different KUs. Thus, this an important issue that must be put into consideration when designing the DB.

B. Database Design

The ER diagram for the BoK is provided in Fig. 5, and its corresponding mapping in relational notation is represented in Table I. All the relations are normalized to third normal form.

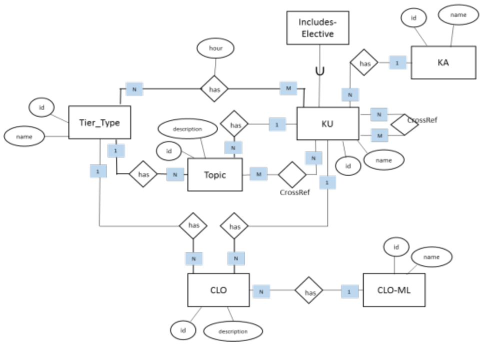


Fig. 5. The ER Diagram for Storing the BoK.

TABLE I. DESIGN OF BODY OF KNOWLEDGE DB IN RELATIONAL NOTATION

Relation	**description**	**Foreign Key**	**Referenced relation**
TIER-TYPE(<u>id</u>,name)	An enumeration relation for storing the types of tiers: core-tier1, core-tier2, and elective.		
CLO-MasteryLevel(<u>id</u>,name)	An enumeration relation for storing the different CLO mastery levels like Familiarity, Usage, and Assessment.		
KA(<u>id</u>, abbreviation, name)	Master Relation for storing the KAs, their abbreviations, and complete name.		
KU(<u>id</u>, name , ka-id)	Master Relation for storing the knowledge units: universal id in the system, name of unit, and the KA it belongs to.	ka-id	KA.id
TOPIC(<u>id</u>, description, tt-id, ku-id)	Master relation for storing the topics, with their universal system id, description, the tier type of the topic, and the KU it belongs to.	ku-id	KU.id
		tt-id	TIER-TYPE.id
CLO(<u>id</u>, description,tt-id,ku-id, cloMl-id)	Master relation for storing course learning outcomes, its universal id in the system, its tier type, the KU it belongs to, and the course mastery level.	ku-id	KU.id
		tt-id	TIER-TYPE.id
		cloMl-id	CLO-MasteryLevel.id
KU-TIER-HOUR (<u>ku-id</u>,<u>tt-id</u>,hour)	Cross Reference relation between KU and TIER-TYPE to store the number of hour coverage for each tier within a specific KU.	ku-id	KU.id
		tt-id	TIER-TYPE.id
Include-Elective(<u>ku-id</u>)	Specialization of KU, stores the id of KUs that have elective topics	ku-id	KU.id
KU-CrossRef(<u>ku-id</u>, <u>ku-cr-id</u>)	Cross Reference relation-Recursive M:N relationship of KU	ku-id, ku-cr-id	KU.id
Topic-Crossref(<u>t-id</u>, <u>Ku-id</u>)	Cross Reference relation between TOPIC and KU	t-id	TOPIC.id
		ku-id	KU.id

C. *Implementation Tips*

The following are some implementation tips for the development of the BoK repository:

- The interfaces for storing the BoK shall allow inserting, editing, deleting KAs, KUs, topics.
- For cross-referenced KUs and topics, users must follow the guidelines provided by the CS2013 report.

It is important to note that topics and CLOs outside the CS2013 report can be added to the BoK. Also, CLO mastery levels can be different than the ones provided in the report.

VII. COURSE\CURRICULUM DESIGN

Each institution retains its own policies for course\curriculum design and auditing. For example, how the courses are formed, who is responsible for creating and revising the syllabi, what are the auditing guidelines, and how often auditing is conducted. In Fig. 6, we present a workflow for course design. We also suggest the knowledge base necessary for automating the process. The design is represented in two parts: course syllabus design and articulation matrix.

In the proposed design, the main users are:

1) The curriculum committee is the group of people allocated by the college or department for designing and overseeing curriculum-related tasks.
2) Course coordinators\course designers are instructors allocated by the college or department for designing individual courses.
3) An accreditation committee is a group of people allocated by the college or department for conducting accreditation related tasks.

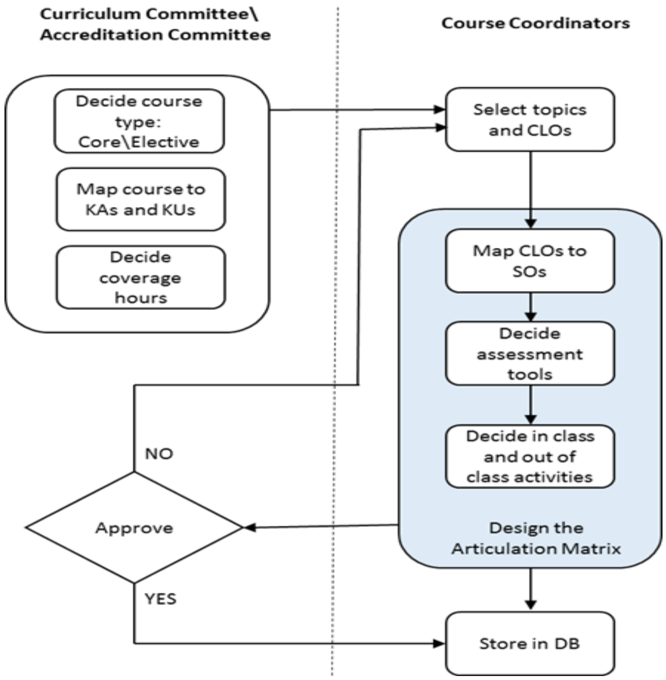


Fig. 6. Course Design Workflow.

D. *Course Syllabus Design*

In the following subsections, the design of the course syllabus is discussed.

1) *Data requirements*

For each course, the system shall keep a record of its id, name, number, type (core\elective), credit hours, and the prerequisite course(s). For each course, the course is mapped to specific KAs, KUs, topics, and CLOs from the BoK.

2) *DB design*

The ER diagram for the course syllabus is provided in Fig. 7, and its corresponding mapping in relational notation is represented in Table II. All relations are normalized to third normal form.

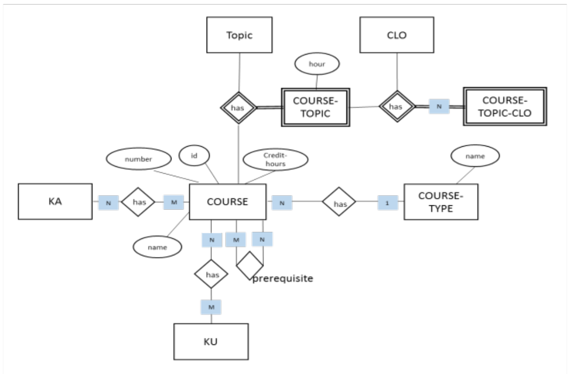


Fig. 7. The ER Diagram for Course Syllabus.

TABLE II. DESIGN OF THE COURSE SYLLABUS DB IN RELATIONAL NOTATION

Relation	description	Foreign Key	Referenced Relation
COURSE-TYPE(id, name)	An enumeration relation for course types "Core" or "Elective"		
COURSE(id, code, number, name, ct-id)	Master relation for storing course information, a universal system id, code, number, name, credit hours, and course type.	ct-id	COURSE-TYPE.id
COURSE-KA(c-id,ka-id)	Cross reference relation for storing the mapping of course to KAs	c-id	COURSE.id
		ka-id	KA.id
COURSE-KU(c-id,ku-id)	cross reference relation for storing the mapping of course to KUs.	c-id	COURSE.id
		ku-id	KU.id
COURSE-TOPIC(c-id, t-id,hour)	Weak relation for storing the mapping of course to topics and coverage hours for each topic	c-id	COURSE,id
		t-id	TOPIC.id
COURSE-TOPIC-CLO(c-id,t-id,clo-id)	Weak relation for storing the mapping of topics to CLOs which are specific for the designated course	(c-id,t-id)	COURSE-TOPIC(c-id,t-id)
		clo-id	CLO.id
COURSE-PREREQUISITE(c-id, c-pre-id)	cross reference relation for storing the prerequisite of courses	c-id, c-pre-id	COURSE.id

3) Design tips

Depending on the specific institutional and accreditation requirements, modifications can be applied to the design in Fig. 7. For example:

1) If the institution has a different way of calculating taught hours, relative attributes may be added to the relationship between COURSE and topics.

2) If course coordinators prefer to apply a maximum number of hour-coverage on the mapping between courses and their constituent KAs\KUs, relevant attributes may be added to the relationship between COURSE and KA and KU, respectively.

E. Articulation Matrix Design

A critical task of the accreditation process is to measure the performance of the courses and check if they achieve their predefined objectives. The starting point would be to use coursework design tools such as articulation matrices or course blueprints to map course outcomes to the different assessment tools (exams, quizzes, projects) in the course. In this paper, we refer to the articulation matrix, which is applied in the Accreditation Integration & Management System (AIMS) in the Faculty of Computing and Information Technology, King Abdulaziz University [8]. AIMS is a tool to support a sustainable multi-accreditation academic quality system (see Section VIII).

TABLE III. A SIMPLIFIED ARTICULATION MATRIX. (AIMS)

			Assessment Tools									
NO	CLOs	ABET Student Outcomes	Quiz	Exam	Comprehensive Final Exam	Project (Individual)	Group Project	Homework Assignments	Graded Lab Work	Lab Exam	Student Work Portfolio	Formal Presentation
1	Apply concepts in ER modelling notation to model a real world problem	2		x								
2	Evaluate a proposed decomposition to determine the degree of its normal form.	1	x	x				x				

An articulation matrix is a tool used by course coordinators for structuring the detailed design of coursework. The structure includes elements of course design such as: assessment tools, in-class activities, and out of class activities. A simplified structure of the articulation matrix is illustrated in Table III. For linking course performance with ABET, CLOs are mapped to a set of 5 ABET Student Outcomes (SO) [2]. The mapping is used to compute the overall student attainment in each individual SO.

1) Data requirements

For each course, it is required to store the different assessment tools in the course and their percentage of marks out of 100%. For each CLO in a course syllabus, the CLO is associated with assessment tools such as exams, projects, etc. Each CLO and assessment tool pair is associated with the week it is assessed in, and the percentage of marks for the assessment. The same CLO can be evaluated in more than one assessment tool; hence different assumptions can be applied.

In this section, we only provide the requirements and DB design for the "assessment tools" in the matrix. However, a straight forward modification can be applied to add course activities or any other attributes as necessary.

2) DB design

The DB design of the articulation matrix is illustrated in Fig. 8, and its corresponding mapping in relational notation is represented in Table IV. All relations are normalized to third normal form.

F. Implementation Tips

The following implementation tips apply to the course design module (syllabus and articulation matrix):

1) Two different interfaces with relevant authorizations for both curriculum\accreditation committee members and course coordinators must be provided.

2) Features for editing course content.

3) Automation for the workflow in Fig. 6 is recommended.

4) Features for querying the knowledge base must be provided. The queries must cover the necessary statistics about the courses and curriculum, such as: number of courses covering a specific topic, the number of courses covering a specific SO, or the depth of coverage of specific SO.

5) Features for auto-generating course syllabi must be added.

It is essential to apply data entry features that automatically assist in preserving curriculum coherence. We provide here some examples.

1) To avoid redundancies in topics and CLOs, provide the course coordinator with up-to-date statistics about the courses already covering these topics and CLOs.

2) To avoid over coverage of topics, KAs, or KUs, provide course coordinators with up-to-date statistics

about the coverage hours of these attributes in the curriculum.

3) To avoid choosing topics that are not relevant to the course, design pull-down menus for selecting topics from the KUs and cross-referenced KUs specified for the course.

4) The use of database assertions and triggers will help maintain design policies and guidelines. For example: maintaining a percentage of "Familiarity" CLOs in a curriculum within a specific range or monitoring the depth of coverage at different granularity levels (KA, KU, or topic).

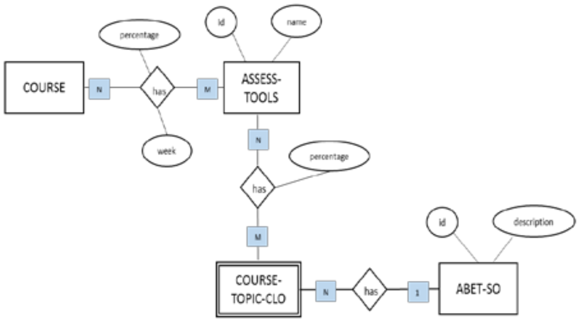


Fig. 8. The ER Diagram for the Articulation Matrix.

TABLE IV. DEISGN OF THE ARTICULATION MATRIX IN RELATIONAL NOTATION

Relation	Description	Foreign Key	Referenced relation
ABET-SO(id,description)	Enumeration relation for storing ABET SOs. Example:(1,"Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions")	–	–
ASSESSMENT-TOOLS(id,name)	Enumeration relation for storing different assessment tools. Example: (1,"Exam"),(2,"Quiz")	–	–
COURSE-ASSESSMENT(c-id,at-id,week,percentage)	Cross reference relation to store the mapping of courses to assessment tools with the percentage of each assessment out of 100	c-id	COURSE.id
		at-id	ASSESSMENT-TOOLS.id
COURSE-TOPIC-CLO-ASSESSMENT(c-id,t-id,clo-id,at-id,percentage)	Cross reference relation to store the percentage of each CLO assessment in a given assessment tool and the week its being conducted in	(c-id,t-id,clo-id)	COURSE-TOPIC-CLO(c-d,t-id,clo-id)
		at-id	ASSESSMENT-TOOLS.id
COURSE-TOPIC-CLO(c-id,t-id,clo-id,so-id)	Modify the relation in table 3 by mapping the CLO in each course to the corresponding ABET SO	(c-id,t-id)	COURSE-TOPIC(c-id,t-id)
		clo-id	CLO.id
		so-id	ABET-SO.id

VIII. ACCREDITATION MANAGEMENT

Once the correct and complete design of courses is stored, different accreditation components can be added to the framework. These components will help accreditation committees in the evaluation process of their programs. These components are to be designed based on the evaluation and accreditation techniques deployed by colleges and institutions.

The proposed knowledge base in this paper will serve as a baseline for assessment and evaluation components that are necessary for the continuous improvement cycle. According to the additional add-on components, other data may be captured and stored. For example, after each course offering, it will be required to store student results in exams and course work and then used in assessment calculations. To avoid potential analysis paralysis problem, a careful selective approach must be followed to decide which data will be used -and consequently stored- for assessment, evaluation, and as evidence for accreditation.

Another category of tools can be used for curriculum analytics. For example, the Curriculum Analytics Tool (CAT) in [15] generates the competency scores for the entire curriculum across cognitive and progression levels.

The existence of a knowledge base will empower the curriculum and accreditation personnel with limitless capabilities, especially with data-driven decision making based on accurate and up-to-date statistics and information about the curriculum and its outcomes. This will allow the authorized personnel to apply necessary changes and improvements based on valid evidence.

A developed system of the proposed framework will significantly support the documentation needed for accreditation. We provide examples from the ABET SSR report in the following points:

- Criterion 3 (Student outcomes): in this criterion, the implemented system shall auto-generate reports of the curriculum to SO mappings, CLO to SO mappings, individual courses to SO mappings, and course syllabi.
- Criterion 4 (Continuous Improvement): in this section, the implementation of the improvement life cycle is documented. Reports about direct assessment attainment scores at SO, course, and program level should be auto-generated. Also, the data-driven decisions for curriculum tuning must be reported. Any tools developed in the framework must be documented as evidence.
- Criterion 5 (Curriculum): Different curriculum analysis reports should be auto-generated. For example, evidence of breadth and depth of advanced computing topics or depth of mastery levels can be auto-generated.

IX. RELATED SYSTEMS

Within academic institutions, different tools and systems have been developed to aid in the process of continuous improvement, accreditation, and curriculum-based analytics.

For example, AIMS provides an ABET-compliant web-based system for course articulation matrices and course assessment. The CLOs in the matrix are filled manually by course coordinators through a programmed Excel sheet. At the end of each semester, all student grades in selected assessment tools and selected outcomes are archived. The system then publishes statistics about student attainment in these outcomes. This will allow instructors to provide a detailed analysis of these outcomes and provide suggestions for improvements. The system also offers different accreditation reporting features. The absence of integration with BoK is a significant drawback in AIMS. The analysis that is based on the relationship of outcomes and curriculum content is done manually.

Curriculum analytical tools, on the other hand, are tools that adjust the alignment between program and course level competencies. As an example, a curriculum analytical tool was developed in the Department of Electrical & Computer Engineering University of New Mexico to improve graduation rates [17]. The tool models sequences of courses in the curriculum as directed graphs and allocate patterns where failing a course will delay students' graduation, then reform the curriculum accordingly.

As for accreditation support tools, expert comparison of commercially available tools such as EvalTools [9], CLOSO [10], and WEAVEonline [11] can be found in [20]. EvalTools provide a course management system, outcome-based assessment, and reporting features. CLOSO is an outcome-based assessment tool related to ABET accreditation. Both EvalTools and CLOSO do not support the continuous improvement life cycle. WEAVEonline provides an accreditation management system with partial support for the continuous improvement life cycle.

X. CONCLUSION

The accreditation and the continuous improvement of curriculums and are two different but parallel and interrelated processes. They both need 1) systemized and methodological approaches to achieve them, and 2) acquisition of comprehensive curriculum and accreditation related knowledge base. There is a need to propose analytical methods to measure the compatibility of existing curriculums against the underlying BoK. In this paper, we focus on the importance of integrating the BoK in the continuous improvement cycle and discuss its impacts on the coherence and consistency of curriculums. A general framework model for curriculum and accreditation management is proposed. The database design of the BoK repository, the curriculum, and the workflow for designing courses are provided. We also discuss the means of linking the proposed design with accreditation tasks.

ACKNOWLEDGMENT

Special thanks to Raghad AlSaati, Abeer Harakaty, Maram Abuzaid, and Hanouf AlAhmadi for their contribution in developing a prototype of the BoK repository and the curriculum builder modules.

REFERENCES

[1] Accreditation Council for Business Schools and Programs (ACBSP) Available: https://acbsp.org/

[2] Accreditation Board for Engineering and Technology (ABET). Available: https://www.abet.org/

[3] ACM\IEEE (CS 2013) Report, Available: https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf or http://ai.stanford.edu/users/sahami/CS2013/.

[4] ABET Criteria. Available: https://www.abet.org/accreditation/accreditation-criteria/criteria-for-accrediting-computing-programs-2019-2020/#GC3

[5] Common Body of Knowledge (CBK) Available: https://www.isc2.org/Certifications/CBK

[6] Business Analysis Body of Knowledge (BABOK) Available: https://www.iiba.org/standards-and-resources/babok/

[7] Landscape Architecture Body of Knowledge Study Report (LABOK), Available: https://www.asla.org/uploadedFiles/CMS/Education/Accreditation/LABOK_Report_with_Appendices.pdf.

[8] Accreditation Integration & Management System (AIMS), Available: https://fcit.kau.edu.sa/aims/home.php.

[9] EvalTools. Available: https://iu.myschooling.net/EvalTools6/EvalTools/

[10] CLOSO. Available: http://www.smart-accredit.com/

[11] WEAVEonline. Available: https://app.weaveonline.com/login.aspx

[12] A. S. Barb and N. Kilicay-Ergib, "Applications of Natural Language Techniques to Enhance Curricular Coherence, " Complex Adaptive Systems Conference with Theme: Leveraging AI and Machine Learning for Societal Challenges, vol. 168, pp.88-96, 2020.

[13] D. Bateman, S. Taylor, E. Janik, and A. Logan, "Curriculum coherence and student success," Champlain College CEGEP, vol. 22, no. 5, pp. 8-18, Dec 2007.

[14] R. M. Felder and R. Brent, "Designing and teaching courses to satisfy the ABET engineering criteria," Journal of Engineering Education, vol. 92, no. 1, pp. 7-25, Jan 2003.

[15] S. Gottipati and V. Shankararaman, "Competency analytics tool: Analyzing curriculum using course competencies," Education and Information Technologies, vol. 23, no. 1, pp. 41-60, Jan 2018.

[16] T. Hatzakis, M. Lycett, and A. Serrano, "A programme management approach for ensuring curriculum coherence in IS (higher) education," European Journal of Information Systems, vol. 16, no. 5, pp. 643-657, 2007.

[17] G. L. Heileman, M. Hickman, A. Slim, and C. Abdallah, "Characterizing the complexity of curricular patterns in engineering programs," in ASEE Annual Conference & Exposition, Columbus, 2017.

[18] G. Mayo, W. Wu, T. McCuen, R. R. Issa, and D. Smith, "Implementation of the BIM body of knowledge (BOK) framework for program planning in academia," in Proceedings of the 12th BIM Academic Symposium & Job Task Analysis Review, pp. 2-9, Mar 2018.

[19] K. Mohiudddin, A. Islam, S. Mohd, and M. Shariff, "Evaluation of an Academic Program: The Case of Computing Accreditation Commission Framework in Higher Education," International Journal of Emerging Technologies in Learning (iJET), vol. 14, no. 11, pp. 70-91, 2019.

[20] A. Namoun, A. Taleb, and M. Benaida, "An Expert Comparison of Accreditation Support Tools for the Undergraduate Computing Programs," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 9, no. 9, Aug 2018.

[21] K. E. Willcox and L. Huang, "Mapping the CDIO curriculum with network models," CDIO, 13th International CDIO Conference, pp. 78–88, June 2017.

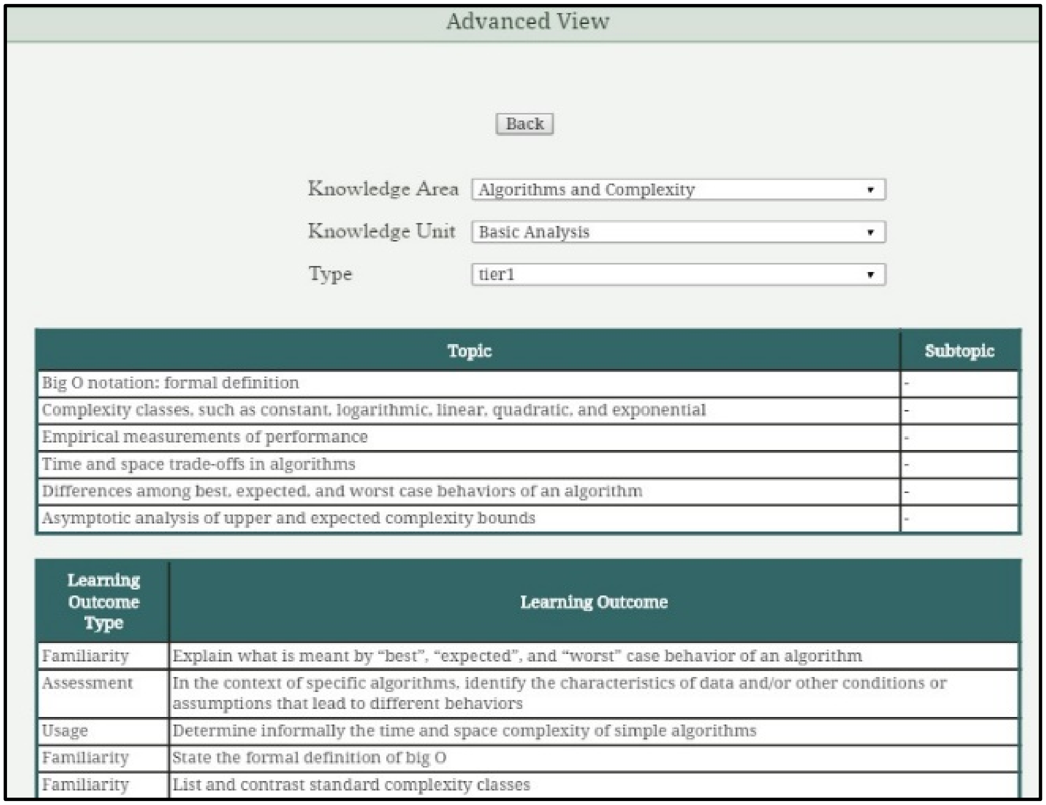
[22] P. A. Quezada-Sarmiento et al., "Body of Knowledge Model and Linked Data Applied in Development of Higher Education Curriculum," in Science and Information Conference, vol. 943, pp. 758-773, April 2019.

[23] E. Rata, "Knowledge-rich teaching: A model of curriculum design coherence," British Educational Research Journal, vol. 45, no. 4, pp. 681-697, May 2019.

[24] W. H. Schmidt and R. S. Prawat, "Curriculum coherence and national control of education: issue or non-issue?," Journal of curriculum studies, vol. 38, no. 6, pp. 641-658, Dec 2006.

[25] W. Y. Wong and M. Lavrencic, "Using a risk management approach in analytics for curriculum and program quality improvement," 6th International Conference on Learning Analytics & Knowledge, , 1st Learning Analytics for Curriculum and Program Quality Improvement Workshop Edinburgh, APR 2016.

APPENDIX



Knowledge Area: Algorithms and Complexity

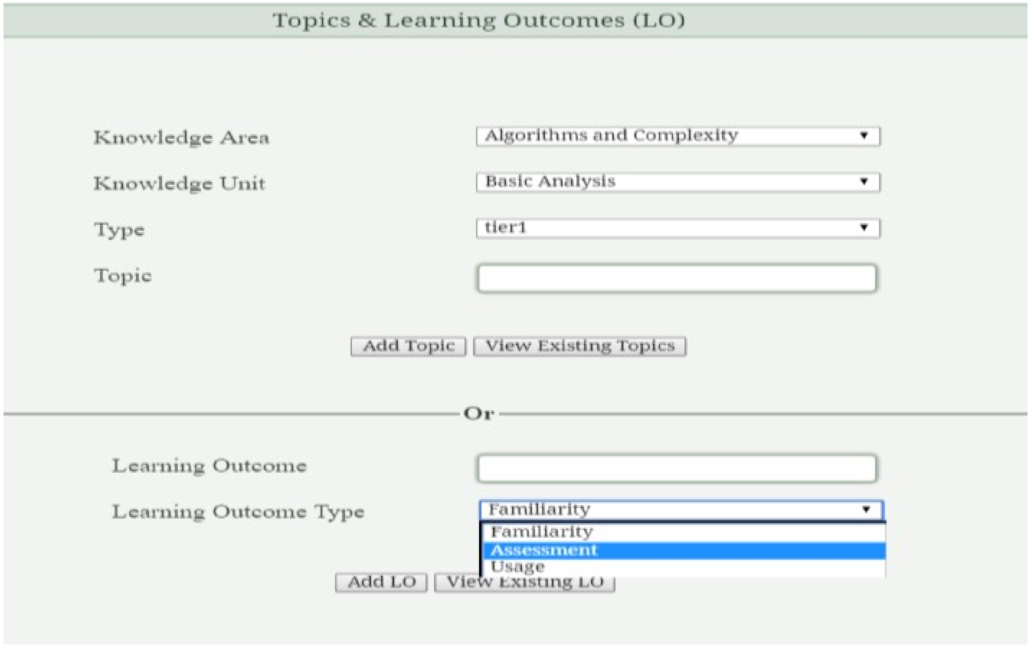
Knowledge Unit: Basic Analysis

Type: tier1

Topic	Subtopic
Big O notation: formal definition	-
Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential	-
Empirical measurements of performance	-
Time and space trade-offs in algorithms	-
Differences among best, expected, and worst case behaviors of an algorithm	-
Asymptotic analysis of upper and expected complexity bounds	-

Learning Outcome Type	Learning Outcome
Familiarity	Explain what is meant by “best”, “expected”, and “worst” case behavior of an algorithm
Assessment	In the context of specific algorithms, identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors
Usage	Determine informally the time and space complexity of simple algorithms
Familiarity	State the formal definition of big O
Familiarity	List and contrast standard complexity classes

Fig. 9. Browsing the Body of Knowledge.



Knowledge Area: Algorithms and Complexity

Knowledge Unit: Basic Analysis

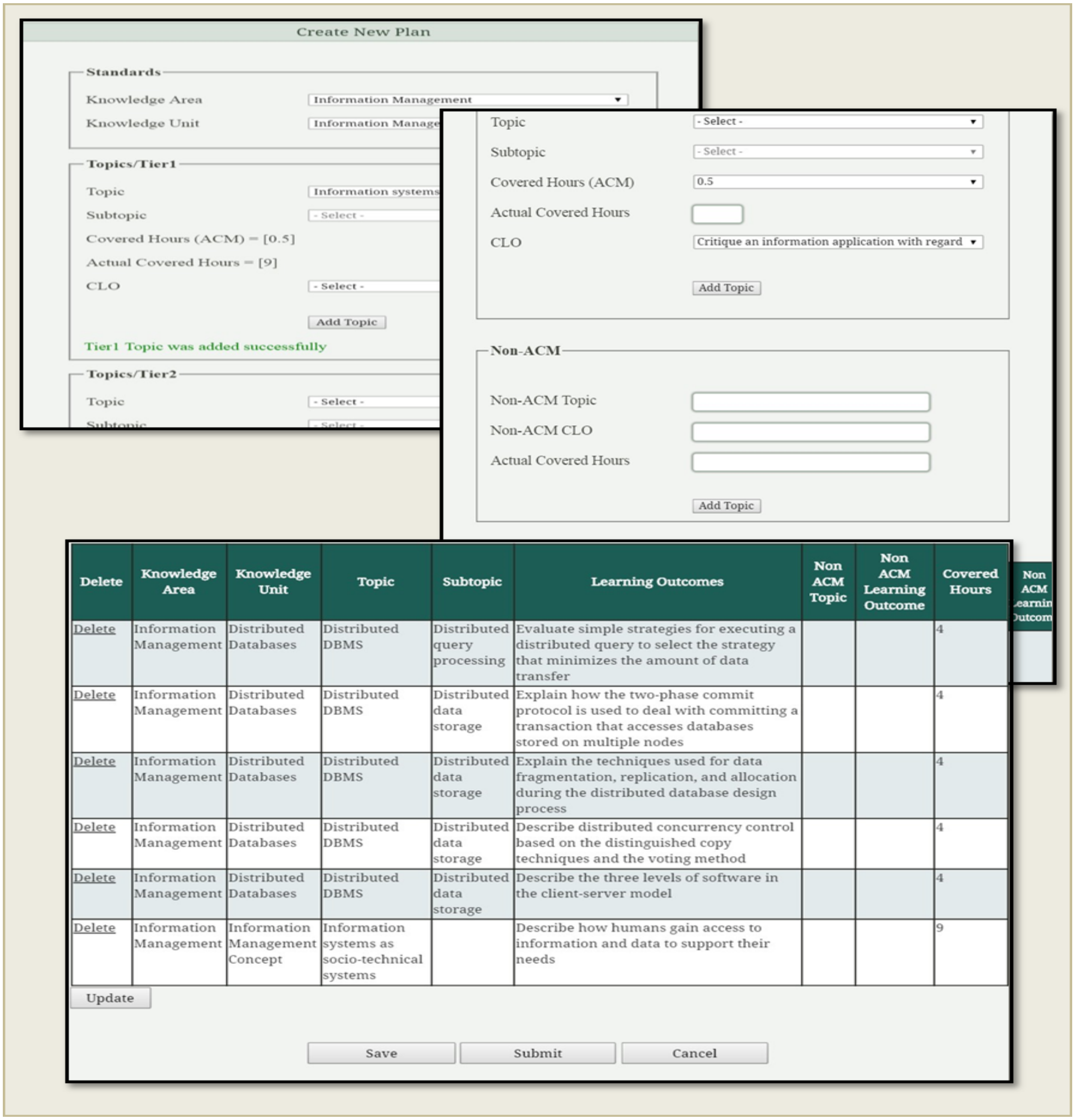
Type: tier1

Topic:

Learning Outcome:

Learning Outcome Type: Familiarity

Fig. 10. Storing the Body of Knowledge.



Create New Plan

Standards

Knowledge Area: Information Management
Knowledge Unit: Information Manage

Topics/Tier1

Topic: Information systems
Subtopic: - Select -
Covered Hours (ACM) = [0.5]
Actual Covered Hours = [9]
CLO: - Select -

Tier1 Topic was added successfully

Topics/Tier2

Topic: - Select -

Topic: - Select -
Subtopic: - Select -
Covered Hours (ACM): 0.5
Actual Covered Hours:
CLO: Critique an information application with regard

Non-ACM

Non-ACM Topic:
Non-ACM CLO:
Actual Covered Hours:

Delete	Knowledge Area	Knowledge Unit	Topic	Subtopic	Learning Outcomes	Non ACM Topic	Non ACM Learning Outcome	Covered Hours
Delete	Information Management	Distributed Databases	Distributed DBMS	Distributed query processing	Evaluate simple strategies for executing a distributed query to select the strategy that minimizes the amount of data transfer			4
Delete	Information Management	Distributed Databases	Distributed DBMS	Distributed data storage	Explain how the two-phase commit protocol is used to deal with committing a transaction that accesses databases stored on multiple nodes			4
Delete	Information Management	Distributed Databases	Distributed DBMS	Distributed data storage	Explain the techniques used for data fragmentation, replication, and allocation during the distributed database design process			4
Delete	Information Management	Distributed Databases	Distributed DBMS	Distributed data storage	Describe distributed concurrency control based on the distinguished copy techniques and the voting method			4
Delete	Information Management	Distributed Databases	Distributed DBMS	Distributed data storage	Describe the three levels of software in the client-server model			4
Delete	Information Management	Information Management Concept	Information systems as socio-technical systems		Describe how humans gain access to information and data to support their needs			9

Fig. 11. Designing a Course from the Body of Knowledge.