International Journal of Engineering Education Vol. 35, No. 2, pp. 645–657, 2019
Printed in Great Britain
0949-149X/91 $3.00+0.00
© 2019 TEMPUS Publications.

Analytical Tool for the Modelling and Simulation of Curriculum: Towards Automated Design, Assessment, and Improvement*

DILEK DÜŞTEGÖR
Department of Computer Science, College of Computer Science & IT, Imam Abdulrahman Bin Faisal University, Saudi Arabia, PO Box 243, Dammam, 31441, Saudi Arabia. E-mail: ddustegor@iau.edu.sa

Continuous quality improvement cycle is essential in educational systems allowing institutions to meet the evolving needs of the market. As such, it is required by all accreditation agencies. Curriculum revision is a critical step of this cycle. This study proposes a modelling paradigm to automate the design, analysis and improvement of curriculum. Based on proven theoretical principles, this novel graph-based approach captures both pre-requisite and cognitive dependencies among courses, enabling an optimal learning environment for students. The presented tool allows an easy and fast analysis of the impact of potential course revisions on all other courses, hence enabling a better continuous quality improvement process, thus providing benefits to many stakeholders in the education system, namely managers, instructors, students and employers. The proposed modelling paradigm is explained and illustrated on a capstone project course offered in the College of Computer Science and IT.

Keywords: accreditation; curriculum development; curriculum design; quality assurance; engineering education; automated tool

1. Introduction

Continuous quality improvement is fundamental for enhanced quality and sustainability, as a system that does not improve will sooner or later become outdated. Educational systems are no exception to this rule, particularly in an era of global competitiveness where programs' standardization and harmonization are constantly sought especially for mutual recognition. Continuous quality improvement is also required by many accreditation organizations around the world, e.g., (ABET).

In broad terms, the continuous quality improvement cycle can be defined as a sequence of course revision based on observations and feedback gathered over some period of time. The repeated set of steps consists of first defining program learning outcomes ensuring alignment with the mission of the institution, translating these program outcomes into relevant student learning outcomes for each course, then measuring them through sound assessments, and comparing with desired criteria or standards. The loop is closed through careful evaluation of comparisons results, identification of potential causes for undesirable outcomes and hence proposed modifications as potential improvements. These revisions most of the time will imply modifications of courses, that can vary from minor changes in an individual course (e.g., new text-book, adjusted pacing to spend more time on a particular topic) to major changes in the curriculum (e.g., introduction of a new course). Courses in a curriculum are intrinsically connected to each other, hence modification in a course can affect other courses. It is necessary to investigate its potential impacts before implementing any revision. Such an investigation necessitates a global view of the curriculum instead of a localized view of a particular course. Given the large of number of courses in a curriculum, and various types of dependencies among them, conducting this analysis manually would be extremely time-consuming and prone to errors. Studies even report how faculty members have negative views regarding accreditation, as they perceive the related activities as an extra work overwhelmingly disrupting them from other important objectives [1]. An automated solution would allow a faster and more accurate evaluation of program curricula, while encouraging faculty's involvement in the process.

A crucial step of developing an automated tool is to choose an appropriate modelling paradigm, which captures all the necessary information for a sound and complete analysis, and yet is not cumbersome and allows fast processing. In this research, a novel Colored Petri Nets (CPNs) based modelling paradigm is presented which allows a systematic analysis of curricula and helps the decision making of university administrators. Indeed, CPNs can capture various type of dependencies among courses, while allowing an easy and fast analysis of the impact of potential course revisions on all other courses. The management may use this model to verify the optimal alignment of courses of an existing curriculum, or a curriculum under modifi-

* Accepted 12 December 2018.

cation for example in the context of the continuous quality improvement procedure, for the benefit of many stakeholders in the education system. The purpose of this paper is thus to present a modelling paradigm that integrates proven educational theoretical principles to automate the design, analysis and improvement of curriculum.

2. Related literature

In the following, the literature is reviewed with respect to three aspects of the proposed framework. The first section identifies important course-dependencies through pertinent educational theories. Then, important works on continuous quality improvement cycle are summarized highlighting the gap that this current research is aiming to fill. Finally, studies utilizing Petri Nets for curriculum modeling are summarized, showing that classical Petri Nets are not sufficient to capture the necessary information.

2.1 Courses dependencies

It is common while designing an individual course to make assumptions about the background and ability of students. A wrong assumption can result in failure for under prepared students, boredom for over prepared students, and frustration for the instructor in both cases [2]. Therefore, it is important to understand how courses are related with each other.

The most known type of dependency is the pre-requisite dependency. This can be defined as "the skills and information necessary to succeed in a given instructional unit within a curriculum" [3]. Typically, a pre-requisite requires students to take classes before a course. With the increasing demand from employers for innovation, creativity and entrepreneurship, universities have been more and more interested in higher cognitive skills like analysis and design. Many studies propose to increase these skills through group work [4], problem-solving based learning [5], or lately design thinking [6]. However, according to Cognitive Load theory [7, 8]—a theory that stipulates that learning is impaired when the load imposed on the student by a particular learning task exceeds the limited capacity of working memory—the acquisition of higher cognitive skills can be optimized with the learner's familiarity with the topic [9, 10]; that is to say when students need to recall a new topic, then describe, and finally to apply it throughout the curriculum, then acquisition of higher analytic ability related to this topic is facilitated. In this research, this phenomenon is called cognitive dependency.

2.2 Continuous quality improvement and curriculum verification

There are numerous studies in the literature addressing different aspects of the continuous quality improvement cycle. Many proposed solutions are tailored to specific departments. Turhan et al. describes the process followed by the Computer Engineering and Software Engineering programs at Atilim University, granting the institution European accreditation [11]. Alidrisi presents a methodological approach to construct a study plan, and successfully applies the method on the Department of Industrial Engineering at King Abdulaziz University [12]. Min et al. showed that assessment and evaluation towards continuous improvement can be deliberately and systematically conducted in the context of an Industrial Engineering program [13]. An overall rubric-based framework is applied by the computer engineering department at King Saud University in [14]. Brumm et al. present a competency-based outcomes assessment system for the Agricultural Engineering Program at Iowa State University [15]. Abu-Jdayil and Al-Attar discussed the direct assessment methods used in the Chemical Engineering Program for a continuous improvement process at the United Arab Emirates University [16]. In [17], authors shared how they assessed the program based on outcomes, an experience ending with the ABET accreditation at the EE Department. In [18], Alarifi et al. proposes a "Software Engineering curricula evaluation and development process that focuses on filling the software skill gaps in the industry". In [19], Mills et al. describe a "two-level framework for information systems curriculum design, assessments and improvements". These studies define in detail the various steps of continuous quality improvement; however, the proposed processes are not fully automated, and mainly rely on manual work from committees.

In an attempt to automate the process, semantic representation is proposed to model a complete curriculum using OWL ontology, thus enabling the continuous improvement process [20]. A web based interface is presented in [21] to support the course assessment process. In the web-based software presented in [22], the authors also consider cognitive dependency and acknowledge that in a chain of courses with increasing expertise, the cognitive level should increase. Similarly, in [23], authors compare IS 2002 and IS 2010 with respect to pre-requisites and concluding depth of knowledge, and they show that depth of knowledge is reached with pre-requisites especially if these are usefully implemented. Nevertheless, the proposed methodologies lack a mechanism to verify cognitive dependencies between prerequisite courses. There also exists a few commercial products promising a ready-made curriculum mapping solution, e.g., Rubicon Atlas (by Rubicon International), Illios (open source software by the University of California, San Francisco), eMed (by the University of New South Wales), and Prudentia© (by the University of Notre Dame). Once again, none of these software products are based on a curriculum framework that verify cognitive dependencies.

On the other hand, studies that focus on cognitive dependencies expressing program outcomes with action verbs following Bloom's taxonomy [24], like Besterfield-Sacre et al who propose a universal framework applicable to all engineering disciplines [25], Almarshoud suggesting a framework tailored to some electrical engineering courses [26], or Holmes et al. proposing a mapping software tool that streamlines and standardizes the competency mapping process [27], do not address design and implementation of remedial action through courses revision.

Finally, many studies also focus on the design of curriculum, a highly complex problem given the many possibilities of arranging courses per semester and per year. Lately, researchers abstracted the curriculum design issue as an optimization problem and proposed various heuristic applications from computer science. For example, Wang uses a genetic algorithm to arrange an optimal curriculum [28]; in [29], authors present a curriculum support engine as a web-based application where visual representation of courses per year / semester along with pre/co-requisite dependency facilities the design, revision, evaluation and update of curriculum; in Wang et al. the curriculum evaluation problem is addressed using a fuzzy-logic based model to assess its rationality, where rationality is defined as some ratio of freely elective vs elective and required courses [30]. However, these approaches only work where courses for a program curriculum are already set, and does not allow checking for incoherence that may result from changing a course content.

2.3 Petri net modelling of curriculum

Petri-nets have been widely applied for the modeling and evaluation of a variety of workflow processes. They have also been used to model educational systems as the existing causal dependencies can be considered as a workflow system. Petri-net based modelling of a curriculum has been shown useful in investigating its consistency in terms of prerequisite dependency and other institution specific constraints (e.g., the number of times a course can be taken, or the maximum number of years allowed in the program) [31]. In [32], authors discuss how to construct Petri net models of individual courses, the whole curriculum, a test, and student learn-flow. They have also been widely applied to web-based education for modelling of e-learning [33–35], or to construct individual learning paths on the basis of generalized competencies systems [36, 37]. While all past studies model pre-requisite dependency, none of them consider the cognitive dependency.

3. The proposed modelling for curriculum design, assessment and improvement

The previous section demonstrated the serious gap existing between methods of curriculum design and proven theoretical principles in instruction. This research proposes a framework pedagogically sound to fill that gap, while enabling automation of the process.

3.1 Proposed curriculum framework

Combining both types of dependencies presented in the previous section, we can say that a pre-requisite should enable the student acquires a higher level of cognitive ability in the course. In other words, the usefulness of having pre-requisites comes from the potential to increase the student's "depth of knowledge". This can be enabled by ensuring that a course and its pre-requisite courses not only cover some common set of topics, but the cognitive skills of these common topics are also incrementally aligned. That is to say, in order to minimize cognitive load on students, lower cognitive skills are mastered before higher cognitive skills are introduced, hence an optimized learning environment is created.

The simplified curriculum in Fig. 1 is representing five courses C1, C2, C3, C4, and C5, where C1 is a pre-requisite course to C4, and both C2 and C3 are pre-requisite courses to C5. Courses C1, C2, and C3 are all offered in the same semester, and C4 and C5 are offered in the following semester. In this example, the two sub-graphs (C1, C4), and (C2, C3, C5) form two dependency chains.

Topics and skills can be expressed in various formats depending on codes that are used to describe them in the syllabus, hence this is institu-

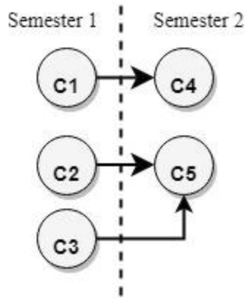


Fig. 1. Course dependency chain.

tion-dependent. In this research paper, they are defined based on the course learning outcomes (CLOs) as a couple (T, CL), where T stands for the topic and CL stands for the cognitive level that is associated with each CLO. After successful completion of a course, students will acquire the skills that are described in the course CLOs, hence we can say that the Post-Conditions of a course are identical to its CLOs. On the other hand, the Pre-Conditions for a course reflect the incremental alignment of cognitive skills with courses in its dependency chain. They are defined according to the cognitive level of the CLOs. Bloom defines six major categories of cognitive levels, from the simplest namely Knowledge, through Comprehension, Application, Analysis, Synthesis, and Evaluation which is the most complex [24]. Some institutions may prefer that students are explicitly taught all the cognitive levels until the program objective is reached (e.g., for a program objective that targets the application level skills in topic A, the student will have to go through courses covering Knowledge level to topic A, then Comprehension level, then finally Application level), whereas another institution may deem their students capable of skipping some steps (e.g., students can be taught Knowledge, Comprehension and Application level skills in topic A in one semester). The exact approach to apply here is institution-dependent.

Continuing with the same illustration, Table 1 shows hypothetical CLOs for the courses depicted in Fig. 1. The column Post-Conditions is identical to the column CLOs. The Pre-Conditions are defined in the third column of Table 1. In this example it is assumed that students do not need explicitly to go through lower cognitive levels, therefore courses C1 and C3, which aim for Comprehension level mastery, do not have pre-conditions. Similarly, courses C2 and C4 also do not have pre-conditions, as they aim for Application level mastery. However, course C5, which aims to teach Analysis level mastery, necessitates the student to have an existing Application level mastery before taking the course.

Since course alignment is aimed to provide an optimized learning environment, it is crucial early in the design stages of a curriculum to ensure that courses are defined accordingly. Also, later in the life cycle, this alignment needs to be preserved even after curriculum modification. The verification of alignment is an iterative process including the following steps:

```
Process Topic/Course/Curriculum Alignment Verification
1. Repeat, for each course $C_i$ in the curriculum:
   a. List topics T that will be covered
   b. For each topic $T_j$:
      i. Identify the cognitive level $CL_{(i,j)}$ that is aimed
      ii. Define the post-condition applying the institution policy
      iii. Backtrack in the pre-requisite dependency chain until either the topic $T_j$ is found or until the chain ends
      iv. If the topic $T_j$ is found in the pre-requisite dependency chain
         1. Identify the cognitive level $CL_{(k,j)}$ that is mastered
         2. If the topic was covered with a cognitive level that satisfies the institution requirements (e.g., $CL_{(k,j)} = CL_{(i,j)}$), then this topic is aligned
   c. If all topics are found to be aligned, then this course is cognitively aligned with its pre-requisite courses
2. If all courses are found to be aligned, then the curriculum is aligned
```

The curriculum evaluation for alignment of courses is a tedious process and a manual implementation would be extremely time-consuming and prone to errors. Hence the need for an automated analytic method.

3.2 CPN and CPN tools

As explained in previous sections, a complete curriculum model needs to capture both pre-requisite and cognitive dependencies of courses, as both are crucial in enabling optimal learning conditions. Petri nets are a powerful modeling tool with a well-defined mathematical foundation and a corresponding graphical representation that makes them easy to understand while enabling carrying out simulations and verifications [38]. However, classical Petri Nets only allow one type of token, therefore it is not possible to use them for modeling both pre-requisite and cognitive dependency. On the other hand, Colored Petri Nets (CPNs) combine the capabilities of Petri nets with the capabilities of a

Table 1. Learning outcomes and corresponding pre- and post-conditions

Course	CLOs	Pre-conditions	Post-conditions
C1	Topic A, Comprehension	None	Topic A, Comprehension
C2	Topic B, Application	None	Topic B, Application
C3	Topic C, Comprehension	None	Topic C, Comprehension
C4	Topic A, Application	None	Topic A, Application
C5	Topic B, Analysis	Topic B, application	Topic B, Analysis

high-level programming language, allowing the definition of multiple data types [39].

CPNs visual representation consists of a bipartite graph where places are connected through directed edges to transitions. Places represent system states, and transitions describe events that are changing the system states. Each place in a CPN can be marked with one or more colored tokens, that is to say a token with a data value attached to it. Both the number of tokens and their colour per places define the state of the system. The transition is enabled when (1) all input places are marked with tokens of the type that is associated with the respective input edge, (2) variables from different input edges evaluate conflict-free, and (3) the transition predicate evaluates to "true". After the transition is fired, the tokens are re-distributed according to incoming and outgoing arc values, to reach a new state [38]. There exist several software packages to construct and manipulate CPN models. The author used CPN Tools to edit, simulate an analyze CPN models in this research [40].

3.3 CPN modelling of courses

In the proposed modeling paradigm, the CPN model represents a dependency chain in a curriculum with the cognitive dependencies modeled as colored tokens. The set of courses in this dependency chain determines the transitions in the bipartite graph. In other words, each transition represents the action of taking a course. The status of students in this system is defined in terms of semesters in the program, therefore places stand for the level (e.g., Fall of Freshman, Spring of Freshman, Fall of Sophomores). An incoming arc to a transition list the skills acquired so far in the curriculum (union of post-conditions for all courses taken so far). An expression guarding each course ensures that the transition is enabled only when prerequisite skills necessary for that course have already been acquired in past courses (when preconditions are satisfied). This is to enforce cognitive dependency. An outgoing arc from a transition to a place enumerates the skills that are newly acquired after passing this course (post-conditions), in other words they produce the number of colored tokens representing these new skills and propagate them to the connected place. This model, through the colored tokens, enables propagation of the achieved skills, only allowing the firing of transitions where pre-conditions are fulfilled.

The bipartite graph representing the dependency chain ending with the course C5 in Fig. 1 is depicted in Fig. 2. The CPN model contains four places (drawn as ellipses), namely Level 0 that is the initial state, Level 1 that represents the student status after completion of the 1st semester, Level 2 that represents the student status after completion of the 2nd semester, and pre[C5] that is used to explicitly define the pre-conditions of course C5; three transitions (drawn as rectangular boxes), namely C1, C3, and C5 that stand for the three courses encountered in the dependency chain ending with C5; and directed arcs connecting places and transitions. The state level 1 is defined by acq_C2 (skills acquired after passing course C2, that is to say the couple (B, Application) standing for Application level of Topic B) and acq_C3 (skills acquired after passing

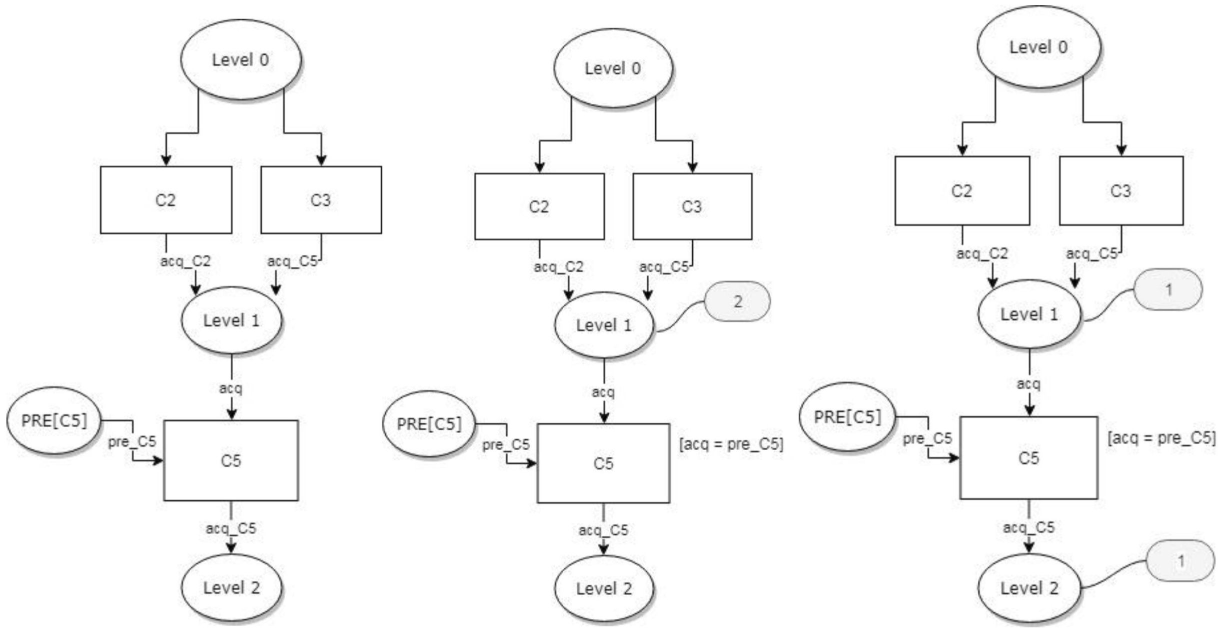


Fig. 2. (a) CPN model of C5 in Fig. 1, (b) Marking after one run of simulation, (c) Final marking.

course C3, that is to say (C, Comprehension)). Similarly, Level 2 is defined by acq_C5 (that is to say (B, Analysis)), where Level 0 models no acquired skills yet.

A transition executes by removing tokens from its input places (places succeeding to the transition) and it adds tokens to its output places (places following the transition). Arc expressions, which are indicated in the arcs, define the colors of the tokens that are moved from input places to output places as a consequence of the transition happening. One can see in Fig. 2, that in order for transition C5 to occur (in other words, in order for a student to enroll to course C5), the student must have accumulated from the first semester tokens equivalent to pre_C5, that is to say (B, Application) which corresponds to the pre-conditions of C5. If this condition is not fulfilled, the transition C5 will not be fired, indicating that cognitive dependency is not fulfilled, as C5 is not aligned with previous courses.

A marking of the CPN model is defined in terms of the tokens on the places. In Fig. 2 (a), the marking shows no tokens in any place, hence this is the initial marking. An execution of a CPN model is described in terms of sequence of runs, indicating the transitional markings. Fig. 2 (b) shows the marking of the model in Fig. 1, after one run which corresponds to one semester in the curriculum. One can understand that transitions C2 and C3 have been fired, consequently the tokens (B, Application) and (C, Comprehension) have been placed into place Level 1, hence the number 2 indicating these two tokens. Fig. 2 (c) shows the marking of the model in Fig. 2 (b), after another semester. We understand that transition C5 has been fired by consuming the token (B, Application), which is the pre-condition to take course C5, and a new token (B, analysis) has been propagated to the place Level 2, indicated with the circle 1 on its top. This is in fact the final marking as there is no further possible transactions.

Note that each transition holds all the necessary and sufficient information regarding the course it represents in terms of pre- and post-conditions. Hence, this is a decentralized modeling approach where modifying a course in a curriculum requires changing the corresponding place only without altering the rest of the model. Also, since this research is not about individual students learn-flows, the set of acquired skills does not need to be modeled per course and can be modeled per semester without any loss of generality. Thus, this CPN model is always a linear net with no backward arcs hence the number of reachable states is over-bounded by |C| x |S| with C the set of all courses in the net, and S the set of semesters included in the net, which allows preventing from the state explosion problem [38].

3.4 *Interpretation of simulation results*

CPN models can be used to simulate the behavior of systems, investigate different scenarios, and analyze system properties using the state space method [40]. The state space method consists of executing the CPN model and automatically generating all reachable states (called a state space). Analyzing its state-space, the system can be verified to be deadlock free. One can also check reachability of some given state [38]. When these properties are not satisfied, the state-space analysis also provides a counterexample, that is to say a system execution that shows why the property does not hold, which can be very helpful in correcting the problem.

The state-space analysis of a CPN-modeled curriculum gives very valuable feedback about the alignment of topics and courses in the dependency chain. Starting from the initial marking (e.g., Fig. 2), if the final state can be reached (for example, Fig. 2 (c)), this indicates that topics covered in courses in the pre-requisite dependency are covered with increasing cognitive skills. On the other hand, the presence of a deadlock proves that there is a problem in the curriculum, either in terms of ordering of courses (perquisite-dependency) or skills covered in courses (cognitive dependency). A step by step simulation can identify when the deadlock is happening, that is to say, one can identify exactly the pre-requisite that have not been fulfilled. This information can be very helpful in the decision-taking to correct the problem, as explained in the next section.

3.5 *Decision support for curriculum modification*

The CPN based curriculum modeling paradigm that has been introduced in detail in the previous section can be used by university managers (e.g., deans and department heads) and faculty members for various purposes.

As part of a continuous quality improvement cycle, a course revision may suggest modifying skills (either topic, or cognitive level, or both) covered in a certain course. However, we understand from pre-requisite and cognitive dependencies that a modification applied to a course can seriously impact other courses. The CPN modeling and state space analysis allows identifying from a global perspective which other courses (if any) will be impacted by a local modification in a course. A deadlock will happen when at least one pre-requisite skill necessary to enroll to a course is not fulfilled. Two types of deadlocks can be distinguished: (1) a mismatch is when the topic in the pre-requisite has been taught but covering a lower cognitive level than the one required, (2) a broken link happens when the topic in the pre-requisite has never been

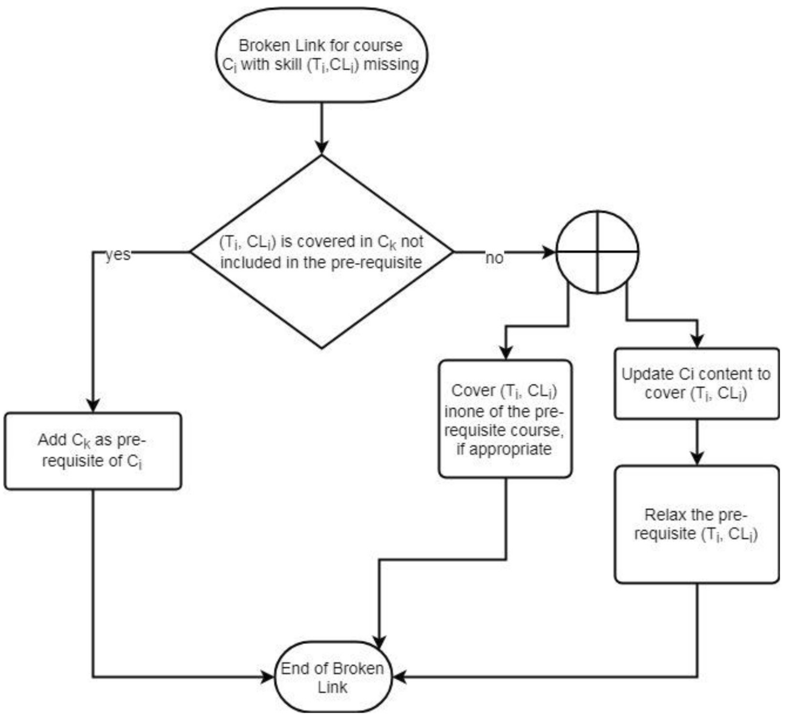

Broken Link for course C_i with skill (T_i, CL_j) missing
(T_i, CL_j) is covered in C_k not included in the pre-requisite
yes
no
Add C_k as pre-requisite of C_i
Cover (T_i, CL_j) inone of the pre-requisite course, if appropriate
Update Ci content to cover (T_i, CL_j)
Relax the pre-requisite (T_i, CL_j)
End of Broken Link

Fig. 3. Flowchart to remedy a broken link in the dependency chain.

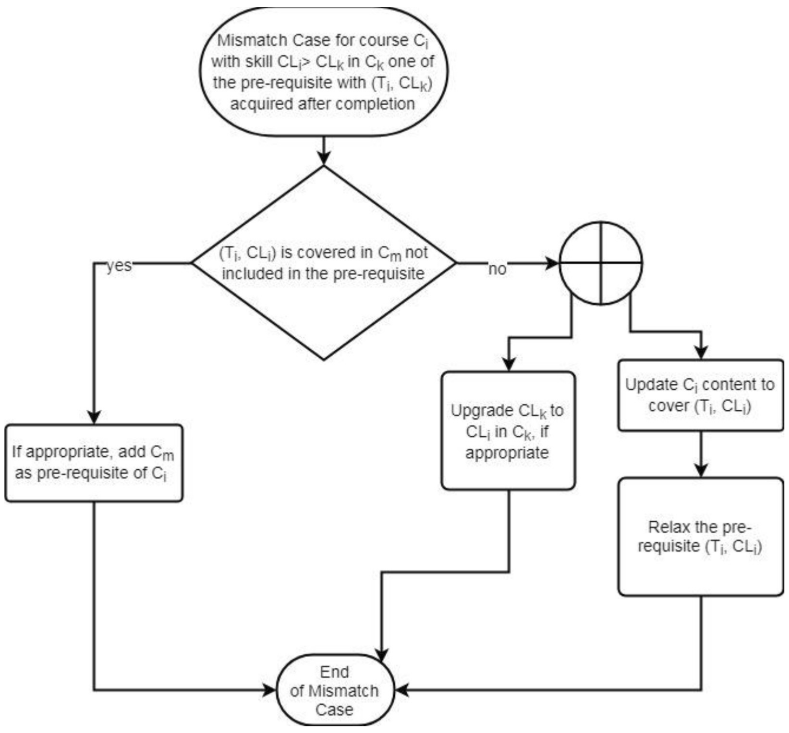
taught in the past courses. Hence, identified deadlocks not only indicate an issue, but also provide valuable information to design a solution, and protect from making things worse while trying to fix some issues identified through the quality improvement cycle.

A broken link indicates a very serious problem as the related topic has not been taught in courses that appear earlier in the dependency chain. It is possible that this topic is covered in a previous course, but this previous course is not included in the pre-requisite chain. If this is the case, then the solution is as easy as adding this previous course in the pre-requisite chain. This is necessary to restore the broken link. On the other hand, if the topic related to the broken link has really not been covered previously, then either this topic can be added to a previous course in the pre-requisite chain, or the necessary teaching arrangement needs to be done to make sure that this topic can be covered up to a higher cognitive level in the course with the broken link. Then the broken link will be restored by relaxing the pre-requisite (since the course will cover all the cognitive levels up to the aimed upper level), see Fig. 3 for the flowchart.

A mismatch indicates that the related topic has been taught in courses that appear earlier in the dependency chain, but the reached cognitive competencies are lower than expected. It is possible that this topic is covered in a previous course, but this previous course is not included in the pre-requisite chain. If this is the case, then this previous course needs to be investigated and checked if it deserves to be in pre-requisite chain or not. If not, then either the cognitive level of past coverage needs to be adjusted, or the necessary teaching arrangement needs to be done to make sure that this topic can be covered up to a higher cognitive level in the course with the mismatch. Hence, the mismatch will be fixed by relaxing the pre-requisite. See Fig. 4 for details of the flowchart for the mismatch case.

The presented CPN model can also be used as a tool to identify issues in an existing curriculum. Indeed, the analysis of the CPN model of a curriculum will identify misalignment problem of courses and their skills with respect to dependencies that intrinsically tie them together. Sometimes, faculty observe lack of readiness of students while taking a course, but it is hard to identify exactly where is the weak link in past courses and hence to design an action plan towards a solution. In the next section, a real-life example illustrates these second type of use of CPN modeling of curriculum. In any case, the type of the deadlock reveals very valuable information about the source of the problem, and guides in addressing it.

The system diagram in Fig. 5 pictures the continuous quality improvement cycle for a typical four-year engineering curriculum (following ABET main guidelines). The proposed CPN



Mismatch Case for course C_i with skill CL_j> CL_k in C_k one of the pre-requisite with (T_i, CL_k) acquired after completion
(T_i, CL_j) is covered in C_m not included in the pre-requisite
yes
no
If appropriate, add C_m as pre-requisite of C_i
Upgrade CL_k to CL_j in C_k, if appropriate
Update C_i content to cover (T_i, CL_j)
Relax the pre-requisite (T_i, CL_j)
End of Mismatch Case

Fig. 4. Flowchart to remedy a mismatch in the dependency chain.

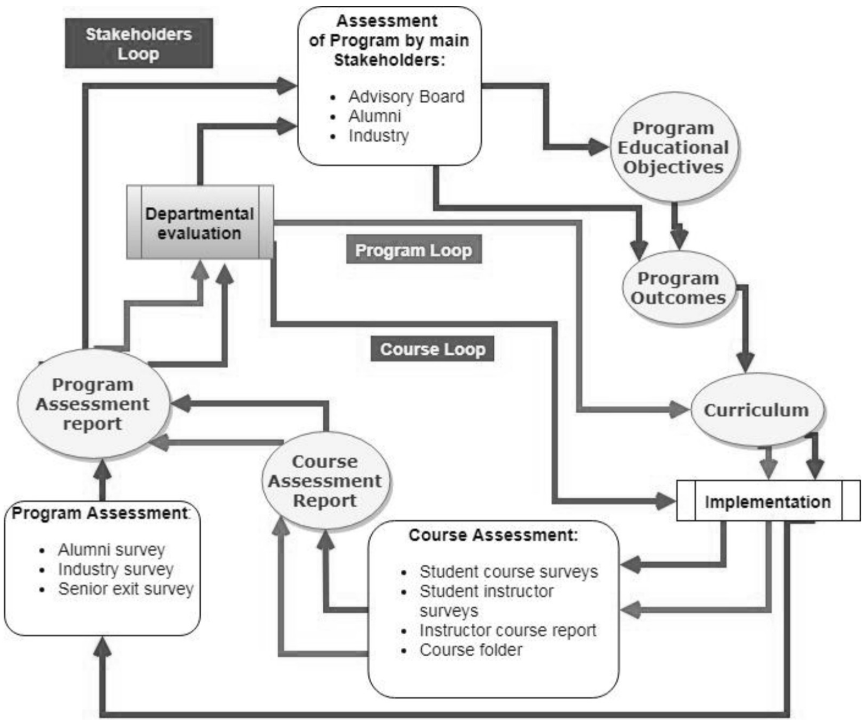


Fig. 5. System diagram of the continuous quality improvement cycle for a typical four-year engineering curriculum.

model can easily be integrated into the continuous quality improvement cycle, as part of the departmental evaluation process and decision making, for analyzing and maintaining any engineering curriculum.

4. Practical evaluation

Capstone projects are completing the engineering educational program through a learning experience integrating many of the student outcomes, espe-

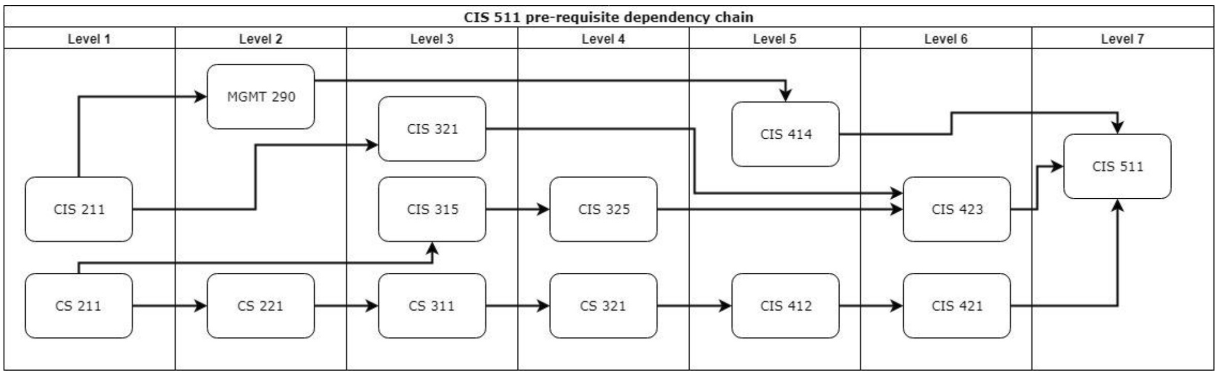


Fig. 6. CIS511 pre-requisite dependency chain.

cially aiming higher cognitive skills increasing students' employability like creativity and critical thinking [41]. For this reason, CIS 511, one of the capstone project course at the author's College (College of Computer Science and IT), has been analyzed to evaluate the proposed modeling paradigm.

4.1 Courses description

The capstone project course spans two semesters, with CIS 511 Project Proposal, followed by CIS 521 Project Implementation. In CIS511, students work in groups of 4 or 5 to analyze and design a real-world or large-scale system. Students report and present their project plan, system requirements and final design to an evaluation committee seeking their approval before moving on with CIS 521, that is to say the implementation.

Students work under the supervision and guidance of a faculty advisor that they meet regularly, but the course format is different from classical courses as it does not have formal lectures. Therefore, the challenge of the course is not in learning new skills. Rather, the challenge is to integrate a wide range of technical and soft skills acquired throughout the curriculum, in the context of a large-scale problem.

Although students encountered several courses with a project component in the past years, jury members lately argued about students' preparedness for CIS 511. They especially witnessed weaknesses in (1) project idea selection, (2) modeling, (3) reporting abilities, (4) presenting skills, and (5) understanding the difference between project and product, although all of these skills were covered in previous courses such as IT Project Management, Systems Analysis and Design, and Technical Reports.

The dependency chain of CIS 511 involves 14 courses (see Fig. 6). In order to build the CPN model of CIS 511, the necessary skills and acquired skills need to be identified for all courses present in its dependency chain. As explained in section 2, these skills can be expressed in various format depending on codes that are used to describe them in the syllabus, hence this is institution-dependent. This course belongs to an ABET accredited college, thus the ABET Outcomes have been adopted as Program Outcomes, describing knowledge, and skills that students acquire as they evolve through the program. Consequently, for all courses in the curriculum, their Course Learning Outcomes are mapped to the Program Outcomes. Thus, in this research skills are defined as a two-tuple (PO, CL), where PO represents the corresponding Program Outcome (defined from A to Z), which are sometimes further refined as sub-outcomes (defined with numbers); and CL represents the cognitive level which is derived from the action verb that is used in the course learning objectives analyzed from a contextual perspective of the sentence meaning. For the necessary skills, it is assumed that students do not need explicitly to go through lower cognitive levels (knowledge, comprehension, and application), whereas higher cognitive skills (analysis and beyond) necessitate that students are accustomed to the topic by lower cognitive skills in previous courses. However, this is only true for structured courses, which are lecturing students. For CIS 511, which is not a structured course with lecturing, and where the challenge lies in the integration of existing skills towards solving a large-scale problem, it is assumed that the necessary skills are identical to the skills that will be acquired. Table 2 shows the resulting listing of necessary and acquired skills for all courses involved in the CIS 511 dependency chain.

4.2 The Colored Petri Net model corresponding to the capstone project course

The CPN model corresponding to the capstone project CIS 511 (Fig. 7) contains 8 places corresponding to the status in terms of completed semesters (here called levels), and 14 places that are used

Table 2. Necessary and Acquired skills for courses in the CIS 511 dependency chain

Courses	Necessary Skills	Acquired Skills
CIS 211		E2, Comprehension G1, Comprehension
CS 211		C1, Application
MGMT 290	D2, Application	D2, Analysis
CS 221	C1, Analysis	C1, Synthesis
CIS 321	B4, Analysis C1, Analysis	B4, Synthesis C1, Synthesis
CIS 315		I2, Comprehension
CS 311		B1, Application C2, Application I2, Comprehension
CIS 325	C1, Analysis	C1, Synthesis
CS 321	C2, Analysis	C2, Synthesis I1, Application I2, Comprehension
CIS 414		B1, Knowledge B2, Comprehension D2, Application
CIS 412	C1, Analysis	B1, Application B2, Application B4, Comprehension C1, Synthesis
CIS 423	C1, Analysis	C1, Synthesis D1, Application
CIS 421	B2, Analysis	B2, Synthesis B3, Application B4, Application C1, Application C3, Application D1, Application I2, Comprehension J1, Comprehension
CIS 511	B1, Synthesis B2, Synthesis B4, Application C1, Synthesis D1, Application D2, Application E2, Application F1, Synthesis F2, Application G1, Knowledge H2, Application J2, Evaluation	B1, Synthesis B2, Synthesis B4, Application C1, Synthesis D1, Application D2, Application E2, Application F1, Synthesis F2, Application G1, Knowledge H2, Application J2, Evaluation

to represent the prerequisite skills for the 14 courses that need to be passed before enrolling to CIS 511. Therefore, there are 14 transitions representing these 14 courses. The arcs connecting level places to transitions, along with arcs from prerequisite places to transitions ensure that the prerequisite skills, as specified in the second column of Table 2, have already been acquired in previous courses. The arcs connecting transitions to level places update the new set of skills, as specified in the third column of Table 2, which are acquired after completing a course. Fig. 7 also corresponds to the initial marking, while the final marking is reached when the current curriculum allows enrolling to the capstone project CIS 511 with all its prerequisites fulfilled by previous courses, in other words ensuring an optimized learning experience in terms of cognitive load.

4.3 Results and discussions

The simulation of the CPN model of CIS 511 encountered in total eleven deadlocks, revealing issues in the existing curriculum. Obviously, the final marking is not reachable with the existing pre-requisite and cognitive level dependencies. Table 3 lists for each deadlock the discrepancies that were encountered between the required skills and available skills, indicating whether the issue is a broken link or a mismatch.

Among these results, the deadlocks directly related to CIS 511, namely 4 broken links and 2 mismatches (line 6 to 11 in Table 3), are congruent with results previously obtained from a tedious manual backtracking of the same course [42]. Moreover, the modeling and automated analysis presented in this research also identified the issues related to other courses (lines 1 to 5 in Table 3) that the manual analysis failed to bring to light, namely broken links for MGMT 290, CSI 321, and CS 321, and mismatches for CS 221, and CIS 421. These five courses will benefit from a closer analysis of their pre-requisite skills, especially in view of previous courses. Overall, the comparison of manual analysis and CPN based analysis shows that the later method is not only correct, but it is also complete, while enabling a faster processing without risk of human mistake.

Note here that after each deadlock, the pre-requisite constraints have been relaxed by changing the concerned course in the CPN model, in order to continue exploring the state space and identify other possible deadlocks. However, one could elaborate an action plan using the algorithm explained in section 2.3.3 (Fig. 3, and 4) to address all these deadlocks, but this would be an institution-related solution and hence is beyond the scope of this current study that focuses on the modeling and analysis method.

5. Limitations and future research

The described modelling paradigm and analysis methodology is certainly faster and human-error free as compared to a manual analysis of dependencies in curriculum. In that sense, it offers a powerful tool to assist curriculum designers to ensure a coherent and harmonious curriculum.

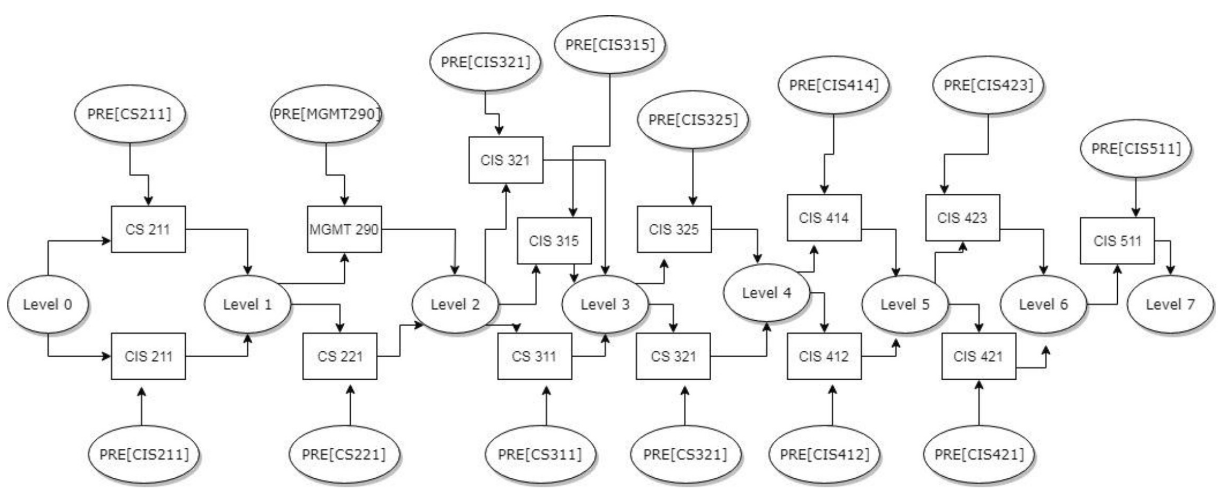


Fig. 7. CIS 511 CPN model.

Table 3. Deadlocks encountered in the simulation of the CIS 511 CPN model

	Course	Required skills	Available skills	Issue
1	MGMT 290	D2, Application	D2, not covered yet	Broken link
2	CS 221	C1, Analysis	C1, Application	Mismatch
3	CIS 321	B4, Analysis	B4 not covered	Broken link
4	CS 321	C2, Analysis	C2, not covered yet	Broken link
5	CIS 421	B2, Analysis	B2, Comprehension	Mismatch
6	CIS 511	B1, Synthesis	B1, Knowledge	Mismatch
7	CIS 511	E2, Application	E2, Comprehension	Mismatch
8	CIS 511	F1, Synthesis	F1, not covered	Broken link
9	CIS 511	F2, Application	F2, not covered	Broken link
10	CIS 511	H2, Application	H2, not covered	Broken link
11	CIS 511	J2, Evaluation	J2, not covered	Broken link

The CPN model is generic enough for various institutions to decide of their own definition of what is a skill, and apply their own standards in terms of pre-requisite and cognitive dependency. Nevertheless, the human and organizational aspects within the faculty of the program is very important and should not be ignored. Regarding the instructor, it is necessary to have a mechanism that ensures that faculty in previous, especially pre-requisite courses actually teach to the course learning objectives. Furthermore, regarding the students, it is also essential to have an assurance of learning plan to track student progress on achieving course objectives. Otherwise, the analysis results would be hypothetical pending rigorous teaching and assessment activities. The described approach needs to be considered complementary to existing mechanisms ensuring teaching strategies and assessments means conform to what is expected.

Finally, instead of using a Colored Petri Net simulation software, it would be useful to develop a dedicated software for curriculum verification, enabling an easier modification of the curriculum CPN model for even faster analysis. This follow-on implementation is currently on-going in the Computer Science department of the author's institution.

6. Conclusion

This study presented a CPN-based modeling paradigm capable of capturing both pre-requisite and resulting cognitive dependencies in a curriculum, and consequent analysis methodology that offers a powerful tool to assist curriculum designers to ensure a coherent and harmonious curriculum. On the one hand, the integration of pre-requisite and cognitive dependency ensures, based on proven theoretical principles, a better learning environment. On the other hand, this tool can be of great benefit especially during continuous quality improvement activities, to assess the impact on the rest of the curriculum of modifying a course content. An identified deadlock, whether it is a mismatch or a broken link, reveals very valuable information about the source of the problem. A comprehensive case analysis enumerated the possible action-plan to take in case of deadlock to overcome them.

The proposed method is generic enough to be adjusted to various institutions, allowing them to decide of their own definition of what is a skill, and apply their own standards in terms of pre-requisite and cognitive dependency. Yet, the described approach does not pretend to replace existing mechanism ensuring that teaching and assessment standards are met, but rather needs to be considered complementary.

The proposed modelling paradigm is explained on a simple example and illustrated on a capstone project offered in the College of Computer Science and IT to verify the efficiency of the presented approach. The obtained results are more comprehensive than a manual analysis, while being automated hence faster and more accurate.

References

1. M. Ohland and J. McNeil, The Influence of ABET Accreditation Practices on Faculty Approaches to Teaching, *Int. J. Eng. Educ.*, **32**(3A), pp. 1151–1159, 2016.
2. R. M. Diamond, *Designing and assessing courses and curricula: a practical guide*. Jossey-Bass, 2008.
3. A. Vuong, T. Nixon and B. Towle, A Method for Finding Pre-requisites in a Curriculum, *EDM*, 2011.
4. R. J. Mills, K. A. Lawless and J. A. Pratt, Training groups of end users: Examining group interactions in a computer-based learning environment, *J. Comput. Inf. Syst.*, **46**(3), pp. 104–109, 2006.
5. B. Martz, J. Hughes and F. Braun, Creativity and problem-solving: Closing the skills gap, *J. Comput. Inf. Syst.*, **57**(1), pp. 39–48, 2017.
6. Nikunj P. Dalal, Higher-Order Thinking in Mis, *J. Comput. Inf. Syst.*, **34**(4), pp. 26–30, 2016.
7. J. L. Plass, R. Moreno and R. Brünken, *Cognitive load theory*, Cambridge University Press, 2010.
8. T. Van Gog and F. Paas, Cognitive Load Measurement, in *Encyclopedia of the Sciences of Learning*, Boston, MA: Springer US, 2012, pp. 599–601.
9. B. B. Morrison, B. Dorn and M. Guzdial, Measuring cognitive load in introductory CS, *Proc. tenth Annu. Conf. Int. Comput. Educ. Res.—ICER '14*, no. July 2014, pp. 131–138, 2014.
10. J. Sweller and P. Chandler, Why some material is difficult to learn, *Cogn. Instr.*, **12**(3), pp. 185–233, 1994.
11. C. Turhan, G. Sengul and M. Koyuncu, A Comprehensive Assessment Plan for Accreditation in Engineering Education: A Case Study in Turkey, *Int. J. Eng. Educ.*, **31**(5), pp. 1270–1281, 2015.
12. H. Alidrisi, Development of a Study Plan for Industrial Engineering Program Using Interpretive Structural Modeling Technique, *Int. J. Eng. Edu*, **31**(5), pp. 1410–1418, 2015.
13. K. J. Min, J. Jackman and D. Gemmill, Assessment and Evaluation of Objectives and Outcomes for Continuous Improvement of an Industrial Engineering Program, *Int. J. Eng. Educ.*, **29**(2), pp. 520–532, 2013.
14. H. A. AL-Twaijry, M. Mekhallalati, H. R. Abachi and G. Muhammad, A Rubrics Based Quality Improvement Methodology for ABET Accreditation, *Int. J. Eng. Educ.*, **28**(6), pp. 1266–1273, 2012.
15. T. J. Brumm, S. K. Mickelson, B. L. Steward and A. L. Kaleita, Competency-based Outcomes Assessment for Agricultural Engineering Programs, *Int. J. Eng. Educ.*, **22**(6), pp. 1163–1172, 2006.
16. B. Abu-Jdayil and H. Al-Attar, Curriculum assessment as a direct tool in ABET outcomes assessment in a chemical engineering programme, *Eur. J. Eng. Educ.*, **35**(5), pp. 489–505, Oct. 2010.
17. S. A. Al-Yahya and M. A. Abdel-halim, A Successful Experience of ABET Accreditation of an Electrical Engineering Program, *IEEE Trans. Educ.*, **56**(2), pp. 165–173, May 2013.
18. A. Alarifi, M. Zarour, N. Alomar, Z. Alshaikh and M. Alsaleh, SECDEP: Software engineering curricula development and evaluation process using SWEBOK, *Inf. Softw. Technol.*, **74**, pp. 114–126, Jun. 2016.
19. R. J. Mills, K. Hauser and J. A. Pratt, A Comprehensive Two-Level Framework for Information Systems Curriculum Design, Assessment and Improvement, *J. Comput. Inf. Syst.*, **48**(4), p. 1, 2008.
20. A. Alfaries, CURONTO: A Semantic Model of the Curriculum for Program Assessment and Improvement, *Int. J. Eng. Educ.*, **30**(5), pp. 1083–1094, 2014.
21. W. Ibrahim, Y. Atif, K. Shuaib and D. Sampson, A Web-Based Course Assessment Tool with Direct Mapping to Student Outcomes, *Educ. Technol. Soc.*, **18**(2), pp. 46–59, 2015.
22. W. Hussain, M. F. Addas and F. Mak, Quality improvement with automated engineering program evaluations using performance indicators based on Bloom's 3 domains, in *2016 IEEE Frontiers in Education Conference (FIE)*, pp. 1–9, 2016.
23. J. H. Reynolds, D. R. Adams, R. C. Ferguson and P. M. Leidig, Programming in the IS Curriculum: Are Requirements Changing for the Right Reason?, pp. 1–6, 2016.
24. B. S. Bloom, M. D. Englehart, E. J. Furst, W. H. Hill and D. R. Krathwohl, *Taxonomy of Educational Objectives: The Classification of Educational Goals, Handbook I: Cognitive Domain*, 1956.
25. M. Besterfield-Sacre et al., Defining the outcomes: a framework for EC-2000, *IEEE Trans. Educ.*, **43**(2), pp. 100–110, May 2000.
26. A. F. Almarshoud, Developing a Rubric-Based Framework for Measuring the ABET Outcomes Achieved by Students of Electric Machinery Courses, *Int. J. Eng. Educ.*, **27**(4), pp. 1–8, 2011.
27. D. W. Holmes, M. Sheehan, M. Birks and J. Smithson, Development of a competency mapping tool for undergraduate professional degree programmes, using mechanical engineering as a case study, *Eur. J. Eng. Educ.*, **43**(1), pp. 126–143, 2018.
28. Y. Z. Wang, A GA-based methodology to determine an optimal curriculum for schools, *Expert Syst. Appl.*, **28**(1), pp. 163–174, 2005.
29. H. Hamam and S. Loucif, Web-based engine for program curriculum designers, *IEEE Trans. Educ.*, **52**(4), pp. 563–572, 2009.
30. B. Wang and J. Zhang, Curriculum System Evaluation Model for University's Professionals Based on Fuzzy Set, *IERI Procedia*, **2**, pp. 414–419, 2012.
31. R. Carvajal-Schiaffino and L. Firinguetti-Limone, Petri Net Based Modelling of a Career Syllabus, *Int. J. Comput. Commun. Control*, **9**(4), p. 397, Jun. 2014.
32. A. Juhasova, I. Kazlov, G. Juhas and L. Molnar, How to model curricula and learnflows by Petri nets—A survey, in *ICETA 2016—14th IEEE International Conference on Emerging eLearning Technologies and Applications, Proceedings*, pp. 147–152, 2016.
33. Z. Baohua, Research on Evaluation of E-Learning Modelling Based on Petri Nets, in *2008 International Conference on Advanced Computer Theory and Engineering*, pp. 699–703, 2008.
34. D. C. Borges, H. B. Neto and J. N. de Souza, Work in progress—Petri Nets as applied to the modeling of E-learning cooperative systems, in *2010 IEEE Frontiers in Education Conference (FIE)*, p. F1D–1–F1D–3, 2010.
35. Z. Balogh and M. Turčáni, Possibilities of Modelling Web-Based Education Using IF-THEN Rules and Fuzzy Petri Nets in LMS, Springer, Berlin, Heidelberg, pp. 93–106, 2011.
36. A. Shukhman, M. Motyleva and I. Belonovskaya, Individual learning path constructing in multilevel regional educational system, *2013 12th Int. Conf. Inf. Technol. Based High. Educ. Training, ITHET 2013*, pp. 2–5, 2013.
37. A. Shukhman, M. Motyleva and I. Belonovskaya, Individual learning path modeling on the basis of generalized compe-

tencies system, in *2013 IEEE Global Engineering Education Conference (EDUCON)*, pp. 1023–1026, 2013.
38. K. Jensen and L. M. Kristensen, Introduction to Modelling and Validation, in *Coloured Petri Nets*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 1–12, 2009.
39. W. Reisig and Wolfgang, *Petri nets: an introduction*. Springer-Verlag, 1985.
40. CPN Tools Homepage .
41. M. Steiner et al., Preparing Engineering Students for Professional Practice: Using Capstone to Drive Continuous Improvement, *Int. J. Eng. Educ.*, **31**(1), pp. 154–164, 2015.
42. M. Elhussein and D. Düştegör, Tracing Back the Chain, in *Proceedings of the 2017 9th International Conference on Education Technology and Computers—ICETC 2017*, vol. Part F1346, pp. 56–60, 2017.

Dilek Düştegör received her BS and MS degree in Computer Engineering from Boğaziçi University (Istanbul, Turkey) and PhD degree from USTL (Lille, France), where she specialized in the supervision of networked systems using graph-based algorithms. She is presently with the Computer Science Department of Imam Abdulrahman Bin Faisal University (Dammam, Saudi Arabia) where she founded the Unit for Quality Assurance of Exams. Her current research investigates the application of artificial intelligence and advanced modelling techniques for technology enhanced education.