System Level Approach for Computer Engineering Education*

MUHAMMAD RASHID
Department of Computer Engineering, Umm Al-Qura University, Makkah, Saudi Arabia. E-mail: mfelahi@uqu.edu.sa

The capstone design course is a culminating course that offers opportunities for students to acquire technical and soft skills in the context of a design project. Conventional curricula and lecture-based teaching methodologies are facing problems to address the challenges posed by industry-oriented projects. Consequently, this paper proposes a system level approach to the curriculum and integrates content-based learning with problem-based learning. Content-based learning consists of coherent delivery of core concepts and systematic laboratory sessions. Problem-based learning addresses the technical skills (such as problem formulation and system design) as well as the soft skills (such as communication skills, working in teams, lifelong learning and ethical/professional responsibility). The Digital System Design course in the Computer Engineering discipline serves as a case study. Assessment through student feedback and the analysis of quantitative data shows a significant improvement in student outcomes.

Keywords: system model; coherency of contents; systematic laboratory sessions; problem-based learning; soft skills

1. Introduction

The capstone design course is the culmination for the previous three years of the undergraduate curriculum, where students work in teams to design, implement and test their solutions for a specified real-world problem. They learn and apply the engineering design process: defining functional requirements, system analysis, design space exploration, and physical prototyping within limited time frames. The goal of this course is to develop student communication (oral and written), and analytical, design, and project management skills through a team-based design experience [1–5].

However, there is a gap between the skills developed during the first three years of engineering education and the skills required for effective capstone design experience [6–7]. Although, the capstone design experience provides a good platform for polishing the skills to transfer to industry, the development of these skills in engineering graduates should not be deferred or postponed until the end of their studies and must be considered throughout their engineering program. Consequently, the issue of skills (student outcomes) is getting increasingly important in engineering education accreditation programs [8–10].

For example, ABET-accredited engineering programs must help students to develop specific outcomes [8]. According to a survey [11], the most important ABET outcomes (i.e., skills) for professional practice are teamwork, communication, data analysis and problem-solving. The survey also highlighted another set of important skills such as ‘math, science, and engineering skills’, ethics, lifelong learning, design, and engineering tools. Another study [30] suggests that 75% of long term job success depends on soft skills (such as communication, teamwork, creativity, and problem-solving) and only 25% on technical knowledge.

Conventional curricula in the CpE domain [16–25] focus primarily on individual courses such that the integrated delivery of knowledge and design skills is often overlooked. Furthermore, laboratory practices are provided in a discrete and non-coherent manner. Consequently, these scattered courses and laboratory practices never provided an overall vision of CpE education. Similarly, the pure lecture-based teaching methodologies are not addressing the development of soft skills (such as communication, teamwork, and creativity, and problem-solving).

To address the issue of skills development for effective capstone design experience, problem-based learning is an attractive solution. It centers on the introduction of real-life problem to prepare future engineers. Recently, several problem-based techniques have been proposed in the paradigm of Electrical and Computer Engineering [26–30]. However, there are certain limitations:

1. students face difficulty in beginning their projects without the coherent vision of contents,
2. the problem is already defined by the instructor and students do not go through the problem identification and formulation phase,
3. it is difficult for students to perform an in-depth study without acquiring the necessary knowledge.

This paper proposes a system level approach to the curriculum and integrates content-based teaching methodology with problem-based learning. A Digi-
tal System Design course in the CpE domain is taken as a case study. However, the proposed approach is not restricted to a particular course or discipline and can easily be applied to other disciplines with minor changes. The main features of the proposed methodology are:

1. System model of the curriculum
2. Coherent delivery of the course contents
3. Systematic laboratory practicals
4. Development of technical and soft skills through problem-based learning.

The system model of the curriculum is constructed by using Y-chart methodology [36, 37]. It implies that the curriculum starts by presenting component level courses and concludes by merging them into system level courses. Component level courses focus on hardware and software. On the other hand, system level courses provide an integration of hardware and software. The system model of CpE core courses holistically binds the component level and system level courses by providing forward and backward references, facilitating the coherent delivery of course contents. Students’ understanding in the course is reinforced by systematic laboratory practices. A laboratory practical is an organized investigative experience. Three stages have been defined to ensure systematic delivery of design skills: procedure-based, semi-independent and completely-independent.

After the contents-based learning (coherent vision of core contents and systematic laboratory practices), the problem-based learning phase of the proposed methodology begins. It is a four-step process:

1. Multidisciplinary students’ teams identify problems related to the course contents in the form of proposals.
2. After the acceptance of proposals from the instructor, they explore alternate solutions of the identified problems.
3. They design, implement and test their solutions.
4. Finally, they present their work in the form of technical reports and oral presentations.

The assessment of the proposed methodology is made through feedback from students and the analysis of quantitative data for SOs. Assessment results show that students readily accepted the proposed methodology and recognized its benefits. A significant improvement in achievement of SOs was observed.

The rest of this paper is organized as follows: Section 2 highlights the pressing concerns and associated challenges in the CpE domain. Section 3 presents state-of-the-art and explicitly states the novelty of this paper. A proposed system level approach is presented in Section 4. To illustrate the viability of the methodology, Section 5 summarizes the assessment results. Section 6 discusses the results and, finally, Section 7 concludes the paper.

2. Setting the stage

This section describes the challenges associated with the skills development in the CpE domain.

2.1 Scope, complexity and evolution of the CpE domain

The scope and complexity of the CpE domain have changed dramatically over the past several decades. The use of computers started for defense applications in 1930s [12]. Soon, mainframe computers broadened the scope to include commercial applications. In parallel, exponential growth of VLSI technology increased the computing power. Similarly, the combination of information technologies and telecommunications opened the way for the information society. Another very important but less visible revolution started with the emergence of embedded systems technologies. More than 95% of the chips produced today are for embedded systems technologies. More than 95% of the chips produced today are for embedded applications [14]. The rapid evolution of computer engineering requires an ongoing review of the teaching methodologies [15].

2.2 Emergence of system level design

Rapid evolution of technology and multidisciplinary advances lead to another very important revolution. This less visible revolution has a focus on systems rather than programs and circuits. Consequently, the term system level design has been evolved [36, 37]. Since the expected growth rate of design productivity in conventional methodology is far below that of system complexity, System Level Design (SLD) has emerged as a new design methodology in the last two decades or so. Consequently, students should be educated as broadly as possible. They are no more circuit designers or computer programmers. They must first be educated as systems designers [13].

2.3 Coherent delivery of contents and systematic delivery of laboratory practicals

In conventional approaches [16–25], courses are offered as discrete building blocks so that a curriculum is nothing more than a collection of discrete courses. The non-coherent pedagogical approach falls short because the knowledge of a particular course is not properly grounded in context (overall curriculum) [22]. Students should feel the contents of a course like different components of a complete system. Similar to isolated theory courses, there
have been scattered and non-coherent efforts to deliver the engineering design experience in a laboratory. Currently, most laboratory practices are being designed in isolation and are not interrelated with regards to integrated delivery of design skills [25]. We have stated in the introductory part of this paper that a laboratory practice is not merely the manipulation of equipment but is also an integrated investigative experience. Consequently, laboratory practicals should be integrated horizontally as well as vertically for the development of skills required by the effective execution of capstone design experiences.

2.4 Problem-based learning

Our previous discussion in this section showed that it is important for engineering education to re-evaluate the conventional lecture-based teaching methodology and consider incorporating learner-centered teaching. Problem-based learning (PBL) is one such approach and has the potential to help students to cope with the challenges of the complexities of the field and problems that they will face in their future careers [26–35]. The objectives of PBL consist of promoting active learning, communication, and collaborative skills, open-ended inquiry, real-life problem solving, critical thinking, and the desire to learn for one’s lifetime. Despite the importance of PBL in engineering education, there are certain limitations. To summarize, the PBL allows students to go beyond factual knowledge and to apply concepts to real-life situations.

3. Related work

Section 3.1 reviews related work, while the innovative points about our approach are described in Section 3.2.

3.1 Literature review

The related work is divided into two categories: (1) the system level approach to the curriculum, and (2) problem-based learning.

3.1.1 System level approach to the curriculum

The pioneer work was proposed by the name of integrated learning in [18]. The objectives included, but were not limited to, improved communication and team skills, increased design content, the development of lifelong learning skills, better integration of curriculum elements, an increased knowledge of other engineering and non-engineering disciplines, and the development of societal understanding and social responsibility. To achieve the objectives of integrated learning, various techniques and facilities requirements were proposed in [19] and [20] respectively. Based on these proposed techniques and facilities, coherent delivery of the contents was not proposed.

A curriculum was proposed for the Electrical and Computer Engineering (ECE) program in [22]. The emphasis in the proposed curriculum was on: a purposeful liberal education, a solid grounding in fundamentals, a wide-ranging problem-based education and a hands-on learning experience. However, an integrated approach for the delivery of design skills was not presented. An integrated approach for the delivery of design skills through a 3-year design project is presented in [17]. The students are exposed to industrial tools in the course of their studies. The limitation of this approach is that a lot of time is spent in mastering the CAD (Computer Automated Design) tools and design methods. Furthermore, it may require a large infrastructure before implementing the curriculum.

Another approach for the development of an engineering curriculum with a specific focus on students’ design skills is presented in [25]. The entire curriculum is divided into three different stages: a structured design experience, a guided design experience, and an open-ended design experience. The objectives, implementation mechanism, and the expected outcomes of each stage are outlined. However, the primary focus in [25] is on laboratory activities. The coupling of laboratories with corresponding theory courses is not mentioned.

The holistic development of the Computer Engineering curriculum using the Y-chart methodology was presented in [39]. While multiple courses in the curriculum were integrated by using forward and backward references and the cohesiveness between laboratory practicals were provided by horizontal and vertical integration, the detailed description of a single course implementing the proposed concepts was not outlined.

To summarize, existing methods focused on the coherent vision of contents or design skills across the curriculum and did not implement the concepts by taking the case study for any single course. Furthermore, the concepts of coherent vision were never combined with problem-based learning.

3.1.2 Problem-based learning

Problem-based learning (PBL) is a non-conventional approach that focuses on the introduction of real-life problem. The use of PBL, where students focus on problem-solving competency rather than factual technical knowledge, is increasing because of its suitability to prepare future engineers [26–35].

A project-based graduate telecommunications engineering design course was proposed in [26]. The course was adaptable/flexible based on industry
requirements, program requirements and level of guidance/supervision provided to the students. The project work, presented in [27], highlighted programmable, configurable and dedicated processors, memory technology and organization, peripherals, and the design and implementation of complete embedded systems. Image processing application is taken as a case study.

Similarly, [28] described a lab session-based course on hardware/software (HW/SW) co-design for basic public key (RSA) application. The project followed a step-wise approach with assignments that build on each other. Students were required to make their own decisions about partitioning between hardware and software, the interface design, and the optimization goals. While the work in [26–28] provided technical as well as soft skills, they target a particular application. The relationship of course content with other topics was not considered.

In [30, 31] students were exposed to a series of semester-long projects that cover fundamental topics in digital system design. These projects were carried out in groups, thereby developing the students’ abilities to work in teams. Similarly, active learning environment were proposed in [32–35] to provide industry relevant skills early in the students’ career. First, the laboratory classes provided guided experimentation with standard laboratory material and then students worked in teams on multiple projects selected by the instructor.

3.2 Novelty of the proposed work

We summarize the novelty of the proposed approach in the following points.

- Section 3.1 shows that the state-of-the-art in the CpE curricula recognizes the benefits and needs of system level understanding for effective execution of capstone design projects [15, 21, 22]. However, little work has been done to develop the skills required for system level understanding.
- IEEE and ACM guidelines [38] and ABET accreditation requirements [8] do not ensure the coherent delivery of content. The proposed approach in this paper describes the coherent delivery of CpE education.
- Existing literature on coherent vision did not implement the concepts for a single course. This paper details the coherent delivery of contents by taking the Digital System Design course as a case study.
- In traditional methods, the development of soft skills was generally deferred to a capstone design course. The proposed approach prepares the students for capstone design experience by developing the required skills.

4. Proposed system level approach

This section describes a system level approach to the CpE curriculum and integrates coherent delivery of contents with problem-based learning.

4.1 System level approach to the CpE curriculum

The system level approach to the curriculum is viewed as a system to mold all the technical evolutions in a field into a unified study that proceeds from concept to ultimate capstone design experience. In this paper, we are only concerned with the CpE core courses, focusing on the engineering of computing systems. An abstract view of the system model for CpE core courses, with the help of the Y-chart, is shown in Fig. 1. The Y-chart methodology is used in the design of electronic systems [36, 37]. In this paper, we incorporate the idea of Y-chart methodology to develop the system level view of CpE discipline.

Beginning with fundamental courses in circuit design and programming, as well as progressing through a sequence of system level courses, students are exposed to CpE concepts that build upon one another to culminate in a senior-level capstone design experience. We term the courses related to circuit design and programming as Component Level courses. On the other hand, System Level courses provide a meaningful integration of hardware and software. A detailed view of the system model for CpE core courses with the help of Y-chart is shown in [39].

4.2 Integration of contents-based learning with problem-based learning

Once the system model of the curriculum is defined, any course can be taught by integrating the coherent vision of contents with problem-based learning as shown in Fig. 2.
Contents-based learning consists of the coherent
development of course contents and delivery of
systematic laboratory practicals. Problem-based
learning involves the development of technical as
well as soft skills. Consequently, the objectives of
proposed methodology are given as:

1. Delivery of a coherent vision of the contents
2. Systematic laboratory practicals
3. Development of technical and soft skills
   through problem-based learning (Fig. 3).

Table 1 maps different objectives of the integrated
methodology on ABET student outcomes (SOs),
identified as criteria a–k. ‘NA’ in Table 1 shows
that the methodology is not addressing the SOs ‘h’
and ‘j’.

In the following sub-sections, the proposed meth-

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
<th>Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>An ability to apply knowledge of mathematics, science and engineering</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>An ability to design and conduct experiments, as well as to analyze and interpret data</td>
<td>2</td>
</tr>
<tr>
<td>c</td>
<td>An ability to design a system, component, or process to meet the desired needs</td>
<td>1</td>
</tr>
<tr>
<td>d</td>
<td>An ability to function on multidisciplinary teams</td>
<td>3</td>
</tr>
<tr>
<td>e</td>
<td>An ability to identify, formulate, and solve engineering problems</td>
<td>3</td>
</tr>
<tr>
<td>f</td>
<td>An understanding of professional and ethical responsibility</td>
<td>3</td>
</tr>
<tr>
<td>g</td>
<td>An ability to communicate effectively</td>
<td>3</td>
</tr>
<tr>
<td>h</td>
<td>A broad education necessary to understand the impact of engineering solutions in a global and societal context</td>
<td>NA</td>
</tr>
<tr>
<td>i</td>
<td>A recognition of the need for, and an ability to, engage in life learning</td>
<td>3</td>
</tr>
<tr>
<td>j</td>
<td>A knowledge of contemporary issues</td>
<td>NA</td>
</tr>
<tr>
<td>k</td>
<td>An ability to use the techniques, skills, and modern engineering tools necessary for engineering practice</td>
<td>2</td>
</tr>
</tbody>
</table>
odology is illustrated by taking the case study of a Digital System Design course that merges programming and circuit design courses. It provides backward references to other component-level courses, for example how logic gates are implemented by using diodes and transistors. Similarly, it provides forward references to other system level courses, for example various combinational and sequential circuits are discussed in the Digital System Design and then reused in other system level courses.

The syllabus of the course is designed to include all of the three objectives. Coherent vision of the contents (objective 1) and systematic laboratory practicals (objective 2) are delivered during the first nine weeks of a 14 week semester. The last 5 weeks are reserved for problem-based learning (objective 3).

4.2.1 Contents-based learning

The coherent vision of Digital System Design (DSD) is based on: (1) Coherent vision of the DSD contents, (2) Relationship of the DSD contents with other courses, and (3) Systematic laboratory experiences.

4.2.1.1 Coherent vision of the course contents

The case study in this paper, Digital System Design, is related to the computation part as shown in Fig. 4. This is the point at which the two elements, hardware and software, combine. Forward and backward references to other courses in the curriculum are used to help students tie the concepts together.

The coherent view of the Digital System Design course, shown in Fig. 4, illustrates that the contents are divided into specification (application capturing) and implementation parts (architecture characteristics). Systems are specified behaviorally or structurally in a hardware description language (HDL). Similarly, systems are implemented either by using fixed logic (for example, ASICs) or programmable logic (for example, FPGAs).

Behavioral description can be combinational or sequential. In the case of combinational behavior, truth tables are used. Sequential behavior can be

Fig. 4. Holistic view of Digital System Design contents.
described either by Finite State Machines (FSMs) or High Level State Machines (HLSMs), depending upon the abstraction level. For example, HLSMs are used at Register Transfer Level (RTL). Structural descriptions are based on behavioral descriptions. Combinational logic is used for pure combinational behavior. Sequential behavior with FSMs are structurally described with the controller, which in turns consists of a Register and combinational logic. Sequential behavior with HLSMs at RTL are structurally described with a processor, which in turns consists of multiple data path components (for example, comparator, multiplexor, adder etc.) and controller. Finally, two different types of processors are custom and programmable.

4.2.1.2 Relationship of DSD contents with related courses

Figure 5 shows the relationship of DSD contents with the Computer Architecture (CA) course. The two major elements of programmable processors in the CA course are instruction set architecture (WHAT to do by a computer system) and micro-architecture (HOW to implement instructions). Microarchitecture further divides into data path components and controller. Consequently, the backward references are provided to DSD contents.

Figure 6 shows the relationship of DSD contents with the Microprocessors course. The two major elements in the microprocessors course are instruction-set architecture (RISC or CISC) and interfacing with peripherals (Memory and I/O). Interfacing circuitry can be implemented either by fixed or programmable logic. Consequently, the backward references for instruction set architecture and interfacing circuitry are provided to CA and DSD contents respectively.

4.2.1.3 Systematic delivery of laboratory practicals

Section 4.2.1.1 described the coherent delivery of core contents (first part of the contents-based learn-
ing). This section will describe the systematic laboratory practicals (second part of the content-based learning). In other words, the technical knowledge is taught in lectures (theory classes) and the technical skills are taught in the laboratory classes, divided into three categories: Procedure-Based (PB), Semi-Independent (SI) and Completely-Independent (CI). The laboratory experiments are listed in Table 2.

4.2.1.3.1 Procedure-based
Table 2 shows that the first five weeks of laboratory activities are procedure-based. During this period, students get enough practice for simulation and synthesis of combinational as well as sequential circuits. The main objective of procedure-based design experience is to bring students’ attention to the concept and practice of using engineering tools for the simulation and synthesis of digital systems. The case study in this paper used the ISE design environment and Xilinx FPGA boards. The delivery of procedure-based design is limited to motivating and exciting students’ interest in the design practice of the digital systems. It uses comprehensive and well-defined step-by-step procedures, delivered through carefully developed experiments. The main outcome of this stage is the realization of digital design.

4.2.1.3.2 Semi-independent
Table 2 shows that the weeks 6–9 of laboratory activities are semi-independent. During this period, students first design the data-path components and controller, and then merge them in the form of a processor at RTL. The main objective of semi-independent design experience is to start students exercising digital design by themselves. Students’ confidence is built by making them autonomous. They achieve the assigned tasks with the little guidance from the lab instructor. The instructor does provide guidance but not in the form of a step-by-step procedure to accomplish the desired tasks. The intention is to prepare students for problem-based learning.

4.2.1.3.3 Completely-independent
Procedure-based and semi-independent activities in the laboratory finally culminate in completely-independent activities. Table 2 shows that the last five weeks (10–14) of laboratory activities are completely independent. This stage is actually the problem-learning phase of the proposed methodology, where students identify, formulate and solve an engineering problem related to the contents of the course. The main objective of this stage is to make students completely independent in taking all the design process decisions by themselves, with almost no guidance. The completion of this stage will ensure the development of lifelong learning abilities in students. It will prepare the students for the capstone design experience.

4.2.2 Problem-based learning
Section 4.2.1 described the contents-based part of the integrated methodology. The time taken by the contents-based learning is approximately nine to ten weeks such that the last four to five weeks are reserved for problem-based learning. This section will describe the various facets of problem-based learning. In this phase, students work in the form of different teams. They use hardware description language (Verilog in this case study) to simulate and synthesize their designs. However, coherent delivery of the contents enables the students to identify various alternate solutions of the same design problem. Typical examples are a microprocessor-based solution (Microprocessor course) and circuit design using fixed logic ICs (Digital Logic Design course).

Students define the problem in terms of well-defined objectives that are achievable within a maximum of four to five weeks. The problem-based learning can give the students the opportunity to demonstrate the required SOs. The selected
problem is based on the course contents. To summarize, the main steps are:

- Prepare a project proposal identifying the problem and implementation schedule.
- Suggest alternate solutions to the problems.
- Design and implement the solution.
- Technical report writing and oral presentation.

The above mentioned steps are executed by multiple design teams of students. Often these design teams consist of students from different departments, for example Computer Engineering and Electrical Engineering. Consequently, students are engaged in open-ended learning while working in a cooperative learning environment (meeting ABET criteria d). Students explicitly outline the design steps to execute the complete design, distribute the design steps among the team and finally integrate and test the overall system for the design validation (meeting ABET criteria e). The instructor also highlights the issues of professional and ethical responsibility while working on projects (meeting ABET criteria f). Finally, students present their work in the form of technical reports and oral presentations (meeting ABET criteria g). The projects designed by multiple students’ groups during Spring 2014 are listed below:

- Design of a digital chess clock controller
- Design of a traffic light controller
- Design of a smoke detection system
- Design of a vending machine.

5. Assessment of proposed system level approach

A comprehensive assessment was performed to assess the students’ performance in SOs as shown in Fig. 7. The assessment was carried out in two different ways: feedback from students and analysis of quantitative data. Feedback from students is further distributed between two categories: personal opinion collected through a questionnaire and the SOs-based student survey. Similarly, analysis of quantitative data is further distributed between two categories: ABET self-study and average comprehension level in SOs.

The proposed methodology was introduced in Spring 2014. The results of the proposed methodology were compared with the previous data (Fall 2013), termed as the base study, where the proposed methodology was not introduced. The comparison between the base study and the proposed methodology is shown as C1 (comparison of ABET self-study) and C2 (comparison of average comprehension level) in Fig. 7. Furthermore, the quantitative data related to the ABET self-study and the average comprehension level in SOs were compared with feedback from students, shown as C3 and C4 in Fig. 7 respectively.

5.1 Feedback from students

The objective was to reflect the students’ feedback on the validity of educational methodology. The feedback from students was gathered through anonymous and voluntary survey forms. In the

![Fig. 7. Assessment plan for the proposed system level approach.](image)
first part, shown in Table 3 and discussed in Section 5.1.1, students were asked questions related to their personal opinions and experiences. The second part, discussed in Section 5.1.2, assessed the course effectiveness in terms of SOs.

5.1.1 Personal opinions through questionnaires

A survey of personal opinions on validity of the course was performed at the end of the course. Thirty questionnaires were completed (4 out of 34 students did not fill out the questionnaires). Table 3 shows an overall average of 3.61 (agreeing more than being neutral) from the 30 responses collected. Furthermore, most of the answers cluster between 3 and 4, which show a uniform opinion distribution.

Students were generally agreed that the methodology was interesting and the coherent vision of the contents helped them in binding different concepts (Q1 and Q2). Students also showed a positive consensus on facilities in the laboratories and supervision from the laboratory instructor (Q4 and Q5). However, it can be observed that students were less confident about the skills and the time required for problem-based learning (Q3 and Q6).

5.1.2 SOs-based survey

The summary of the SOs-based student survey is shown in Table 4. The first row listed all the SOs, while the second and third rows show the students’ response for the corresponding SOs assessed in the course. The second row, represented as ‘Percentage for Criterion’ and abbreviated as ‘PFC’, shows the percentage of students who marked 70% or higher percentage in the corresponding SO. The third row, represented as ‘Average Comprehension Level’ and abbreviated as ‘ACL’, shows the average percentage for the corresponding SO collected from the 30 responses.

For example, the SO ‘a’ has two columns. The data for ‘PFC’ (second row) shows that 73.3% of the students marked SO ‘a’ equal to or higher than 70%. Similarly, the data for ‘ACL’ (third row) shows that the average percentage for SO ‘a’ is 68.67%. It is obvious from Table 4 that the value of ‘PFC’ ranges from 63.3% to 80%. Similarly, the value of ‘ACL’ ranges from 66.67% to 74.7%. The data in the second and third rows of Table 4 will be compared with the quantitative data in Section 5.2.1 and Section 5.2.2 respectively.

5.2 Analysis of quantitative data

The quantitative data is collected by assessing the SOs through written exams, technical reports and presentations. For example, in written exams, students were asked to design a digital component or system for a given problem (SOs ‘a’ and ‘c’). In the laboratory examination, students were required to simulate and synthesize their design by using the tools available in the laboratory (SOs ‘b’ and ‘k’). Similarly, technical reports and presentations were used to assess the performance during problem-based learning (SOs ‘d’, ‘i’ and ‘g’). The analysis of quantitative data is organized into two different forms: ABET self-study in Section 5.2.1 and determination of comprehension level in Section 5.2.2.

5.2.1 ABET self-study

Any program applying for the ABET accreditation prepares an ABET Self-Study Report. A Self-Study Report is a quantitative and qualitative assessment of the strengths and limitations of the program being reviewed. ABET self-study results for Digital System Design are given in Table 5. The first column in Table 5 lists the SOs, while the remaining columns display assessment results for the SOs. Assessment results are shown for two different semesters. The data in the second and third columns, termed the base study, were collected in the Fall 2013 where the

Table 3. Summary of student’s opinion showing their satisfaction level. Students were asked to response on a 5-point scale: 5—Strongly agree, 4—Agree, 3—Neutral, 2—Disagree, 1—Strongly disagree

<table>
<thead>
<tr>
<th>Questions</th>
<th>Score (Mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1. The course was interesting and I would like to introduce it to other students.</td>
<td>3.78</td>
</tr>
<tr>
<td>Q2. The coherent vision of the contents was useful and helped in knowledge retention.</td>
<td>3.82</td>
</tr>
<tr>
<td>Q3. The skills acquired in the lab classes are important for project development.</td>
<td>3.3</td>
</tr>
<tr>
<td>Q4. The lab facilities were sufficient to carry out the projects.</td>
<td>3.95</td>
</tr>
<tr>
<td>Q5. The supervision from laboratory instructor during the projects was sufficient.</td>
<td>3.72</td>
</tr>
<tr>
<td>Q6. The available time was suitable for the problem-based learning.</td>
<td>3.10</td>
</tr>
<tr>
<td>Average</td>
<td>3.61</td>
</tr>
</tbody>
</table>

Table 4. Summary of SO-based survey. Students were asked to mark each SO in terms of percentages. Average PFC (%) = 66.66, Average Comprehension Level (%) = 68.62

<table>
<thead>
<tr>
<th>SO</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
<th>j</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFC (%)</td>
<td>73.3</td>
<td>63.3</td>
<td>66.6</td>
<td>70.0</td>
<td>66.6</td>
<td>70.0</td>
<td>80.0</td>
<td>NA</td>
<td>72.50</td>
<td>NA</td>
<td>66.66</td>
</tr>
<tr>
<td>ACL (%)</td>
<td>68.67</td>
<td>68.0</td>
<td>66.67</td>
<td>72.0</td>
<td>68.0</td>
<td>71.4</td>
<td>74.7</td>
<td>NA</td>
<td>68.30</td>
<td>NA</td>
<td>68.62</td>
</tr>
</tbody>
</table>
The proposed methodology was not introduced and the data in fourth and fifth columns were collected in Spring 2014 where the proposed methodology was introduced. The total number of students in the base study and the proposed methodology were 36 and 34 respectively.

The data in each semester (base study as well as proposed methodology) is further divided into two columns. For example, the SO ‘a’ has two columns: M(a) and P(a). M(a) is the marks that were allocated to questions used in the assessment of SO ‘a’. P(a) is the percentage of students who achieved 70% or higher marks in SO ‘a’. Marks allocated to questions addressing a particular SO vary semester to semester, depending upon the instructor teaching the course and his/her priorities. Comparison of the proposed methodology with the base methodology in Table 5 shows a significant improvement in all the SOs. The base methodology did not address many SOs, such as ‘d’, ‘e’, ‘f’, ‘g’, ‘i’. The proposed methodology addressed these SOs through problem-based learning.

Finally, the comparison of the last column in Table 5 with the second row of Table 4 (SOs-based students’ survey) shows the consistency of results. The value of ‘ACL’ collected from students’ responses ranges from 63.3% to 80% (as shown in Table 4), while the value of ‘ACL’ obtained from the analysis of quantitative data ranges from 53% to 65% (as shown in Table 6).

### 6. Discussion

In contrast to the conventional approaches (e.g. the base methodology in this paper), the system level approach to engineering education, presented in this paper, ensured that the knowledge and tools may change, but the engineering approach to problem solving is relatively constant. Therefore, emphasis should be made on lifelong learning through strong and broad foundation.

Comparison of the proposed methodology with the base methodology showed a significant improvement in all the SOs. The proposed approach allowed students to go beyond the factual knowledge and apply concepts to real-life situations (improvements in SO ‘a’). Coherent vision of multiple courses in the curriculum as well as the coherent delivery of the course contents increased the system level understanding among students. Consequently, students enhanced the understanding of the relationships between parts of a heterogeneous design and its interaction with the physical environment by focusing on systems (improvements in SO c and e). Similarly, horizontal and vertical integration of the laboratory practicals help students to tie concepts together. Coherency among different laboratory practicals within a single course was provided in a systematic way (improvements in SO ‘b’ and ‘k’).

At the same time, the base methodology did not address many SOs such as ‘d’, ‘e’, ‘f’, ‘g’, ‘i’. The proposed methodology addressed these SOs through problem-based learning and included professional practice as an integral component. These practices cover a broad range of activities, including lifelong learning (SO ‘I’), professional and ethical responsibility (SO ‘f’), written and oral communic-
tion (SO ‘g’) as well as working as part of a team (SO ‘d’).

In existing problem-based learning techniques [26–30], students may face difficulty in beginning their projects. In the proposed approach, the coherent vision of contents helped students to increase their confidence in starting their projects and they were able to perform an in-depth study after acquiring the necessary knowledge. Similarly, the problem was not defined by the instructor and students have gone through the problem identification and the formulation phase (SO ‘e’). Feedback from students and analysis of quantitative data suggested that the contents in other courses of the curriculum should be presented in a coherent way such that every course should start from the system model of the entire curriculum.

However, the time duration for systematic delivery of laboratory practicals was not sufficient. An increase in the time duration for semi-independent experience may result in the development of more interesting projects from students. Similarly, students found problem-based learning very interesting. Consequently, it may be started a little earlier in the semester.

7. Conclusion

This paper proposed a system level approach to engineering education and integrated coherent delivery of contents with problem-based learning. The coherent delivery of course contents was based on the system model. During the problem-based learning, students worked in teams to identify and solve a problem relevant to the core contents. They submitted technical reports and oral presentations. Finally, the proposed approach was assessed comprehensively by collecting the feedback from students as well as the analysis of quantitative data. Assessment results revealed significant improvements in student outcomes.

References


Muhammad Rashid received his Bachelor’s degree in electrical engineering from the University of Engineering and Technology, Peshawar, Pakistan, in 2000, his Master’s degree in embedded systems design from the University of Nice, Sophia-Antipolis, France, in 2006, and his Ph.D. degree in embedded systems design from the University of Bretagne Occidentale, Brest, France, in 2009. He is an Assistant Professor with the Computer Engineering Department, Umm Al-Qura University, Mecca, Saudi Arabia. Prior to joining Umm Al-Qura University, he worked with Thomson Research and Development, Paris, France, and the Advanced Engineering Research Organization, Wah Cantt, Pakistan. His research interests mainly include electronic design automation for embedded systems and engineering education.