Journal of Engineering Education Transformations,
Volume No. 37, January 2024 Special Issue, eISSN 2394-1707

Dublin Descriptors for Problem Scenarios in Problem Based Learning

Michael Rohan Swaminathan[1], Prakash Hegade[2] and Ashok Shettar[3]
[1,2,3]KLE Technological University, Hubli.
[1]01fe20bcs201@kletech.ac.in
[2]prakash.hegade@kletech.ac.in
[3]ashok@kletech.ac.in

***Abstract*— Problem-based learning has been one of the major pedagogies that support various learning styles, utilizing problems as a key element in studies. While there are many ways to solve a problem, Computational Thinking can be one of them. Typically, we formulate learning outcomes and design assessments to attain these goals. However, there are cases where a single outcome may not suffice, especially with multi-layered problems. In such scenarios, we propose using Dublin Descriptors and evaluate the study using system design problems. Two case studies were designed, requiring students to reflect on the components of computational thinking. Reflections in the second case study were designed according to the written Dublin Descriptors. We conducted qualitative and quantitative analyses on data from 37 students in two case studies to identify themes related to Bloom's and Dublin Descriptors, using In Vivo, Descriptive, and Focus coding in two rounds of analysis. Statistical inferences suggest that the method holds promise for cases requiring advancement beyond Level 4. Dublin Descriptors show promise in cases where information needs to be communicated from ideas to inferences.**

***Keywords*—Computational Thinking; Dublin Descriptors, Problem-Based learning; Reflections, System Design**

***JEET Category*—Research**

I. INTRODUCTION

ENGINEERING education is experiencing a transformation due to changing teaching methods, the need for employability, and the reach for internationalization. This has been an ever-evolving phenomenon. The pedagogies have been evolving with time to meet the needs of tech-savvy students. One popular approach that has gained attention in recent years is Problem-Based Learning (PBL), which engages students in real-world problem-solving experiences across different fields of study, being inspired from the medical background (Barrows, 1998). Originally used in the medical field, PBL has now expanded to various disciplines with the goal of developing critical thinking, teamwork, and self-directed learning skills. Several methods like one day many problems in PBL have been explored (Hegade, 2019). At the same time, engineering education institutions worldwide are grappling with the importance of standardization and transparency in learning outcomes, as emphasized by the Dublin Descriptors (Masaev et al., 2020).

The Dublin Descriptors, which originated from the Bologna Process have become widely accepted globally as a framework for defining and comparing education qualifications (Logman & Kautz 2021). These Descriptors outline five dimensions of learning outcomes; knowledge and understanding practical application of knowledge, critical thinking skills, communication abilities and learning skills. They play a role in ensuring that qualifications from different institutions and countries can be compared and meet established standards. However, incorporating the Dublin Descriptors into PBL an approach known for its flexibility and adaptability poses a challenge.

The case studies developed for these classes not only immerses students in authentic and complex problem-solving scenarios but also enable them to connect with principles they already know or can explore further. Case studies have long been recognized as educational tools because they simulate real world challenges and encourage, in depth analysis (Van Dijk et al., 2001). Whether it’s case studies or any pedagogy, usually in engineering, we write the learning outcomes using Bloom’s taxonomy and rarely employ any other approach (Krathwohl, 2002). This work proposes to use the Dublin Descriptors.

Through empirical analysis and expert insights, we endeavour to distil not only the principles of effective case study design but also the strategies to align these pedagogical instruments with the Dublin Descriptors without compromising the essence of PBL. As the landscape of engineering education continues to evolve, the ability to navigate the intricate intersection of PBL and the Dublin Descriptors is paramount in ensuring that graduates are not only proficient but also adaptable, critical thinkers ready to navigate the complexities of our rapidly changing world.

The paper is further divided into the following sections: section II presents the literature survey, section III presents the research design, Section IV presents the methodology, section V presents the results and data analysis, section VI presents the discussion followed by the conclusion in section VII.

II. LITERATURE SURVEY

This section reviews the literature in the fronts of PBL, Computational Thinking, Reflections, Dublin Descriptors, and System Design. PBL is an approach that involves students in complex life problems that require investigation, critical thinking, and collaboration (Hmelo-Silver, 2004; Hung, Jonassen, & Liu, 2008). PBL operates on the belief that learning is a constructive process where learners solve relevant and meaningful problems (Duncker & Lees, 1945; D'Zurilla & Goldfried, 1971). The main goal of PBL is not only to develop specific knowledge and skills but also to cultivate general competencies like problem-solving abilities, self-directed learning skills, and metacognition (Anderson, 1993).

PBL has gained acceptance and extensive research across fields of study and educational environments, particularly in medical and engineering education (Prince, 2004; Garrison, 1997). To assess the effectiveness and outcomes of PBL, researchers commonly have employed case studies as an approach. Case studies involve in-depth investigations of multiple cases within real-world settings (Gerring, 2004; Baxter & Jack 2008). These studies provide descriptions of the phenomena under investigation and shed light on the underlying processes and mechanisms. Moreover case studies also allow for comparisons. Contrasts between cases or different aspects within a case as well as the identification of common patterns and themes across cases have been discussed (Masaev et al., 2020).

However it is important to acknowledge that case studies have limitations. These limitations include challenges in generalizing findings to contexts due to the nature of each sample having potential biases in data collection and analysis as well as complexities and ambiguities in interpreting results (Gerring, 2004; Baxter & Jack 2008). In a research conducted (Lohman & Finkelstein, 2002) impact of PBL cases on student's problem solving abilities was examined. They designed two types of cases; one with structure and guidance and another that was open ended and less structured. The findings indicated that students who engaged with the open-ended cases demonstrated improvements in their problem solving skills compared to those who worked on the structured ones. Additionally the researchers observed that student's prior knowledge and motivation played a role in their problem solving performance. The studys conclusion suggests that PBL cases should be tailored to meet learners' specific needs and goals while simultaneously challenging their existing knowledge and skills. Numerous similar studies have been conducted in the past.

Computational Thinking (CT) has been integrated with engineering case studies (Jona et al., 2014). CT has several aspects that align with PBL characteristics. Its components: decomposition, pattern recognition, abstraction, and algorithms can be used as a means to solve a problem in PBL pedagogy (Selby & Woollard, 2013). The principles align with PBL design and structure (Wing, 2006). Reflections help us develop critical thinking (Williams, 2001). Effectiveness of PBL and CT have been studied (Hegade et al., 2023).

System design problems provide a holistic system development capability and can be layered according to the depth that one needs to explore. It is difficult to write a single learning outcome when the problem has many layers of learning, and it is challenging to do justice to the learning process. Literature presents an opportunity to design system design case studies using Dublin Descriptors and examine their effects on student learning and cognitive skills.

III. RESEARCH DESIGN

This section presents the employed research design for the proposed work.

A. *Interpretive Framework*

Our interpretive framework is pragmatism. The ontological belief is that Dublin Descriptors are useful, and evaluating their effectiveness can be beneficial. To construct knowledge, this work uses the inductive approach. Considering the sampling method used, the work has its own limitations as it is confined to the study of one classroom. The work reflects both the researcher's and the participant's perspectives. The work uses a multi-method approach for data analysis. Both qualitative and quantitative methods are employed for analysis.

B. *Research Question*

We formulate our research question based on the literature survey, addressing problem-based learning, computational thinking and Dublin Descriptors, and reflection elements.

RQ: What are the effects of using Dublin Descriptors for system design learning outcomes in a problem-based learning environment using computational thinking?

We explore this by designing a case study that explicitly asks participants to use computational thinking elements to reflect upon, and another case study where the reflections are further customized to bring out the implications from the cyclic Descriptors.

C. *Model*

Based on the formulated research question and the support from the literature survey, a model was designed to guide the case study. This model is presented in Figure 1.

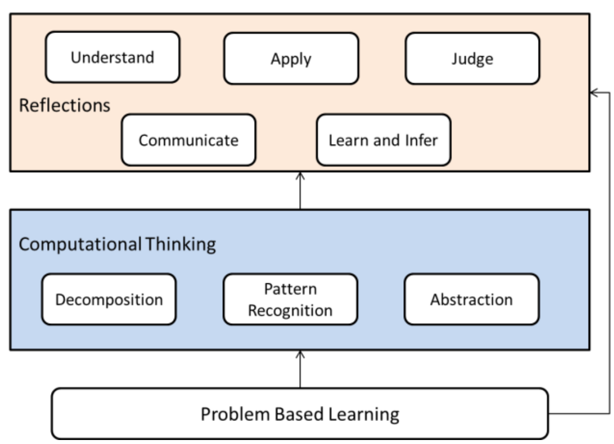

Fig. 1: Model for system design and Descriptors

Computational thinking serves as a framework for problem-solving in problem-based learning. For a system design problem, students reflect on aspects such as understanding, application, judgment, communication, and inference, which are derived from the Dublin Descriptors. The questions are designed to reflect on these elements.

One of the available guiding tools at our disposal for writing better intended learning outcomes is Dublin Descriptors. Dublin Descriptors are identified by (Adam, 2004).

- Knowledge and understanding
- Applying knowledge and understanding
- Making judgments
- Communication skills
- Learning skills

Dublin Descriptors are cyclic Descriptors written for three cycles. The cycles can be as broad as a university program ranging from bachelor's to philosophy or for a project at a course level. Each cycle of Descriptors also grows with complexity. They have been used to write the learning outcomes in higher education qualifications (Gudeva et al., 2012). Following is the Dublin Descriptor set written for the generation of case studies, as presented in Table 1.

TABLE I
DUBLIN DESCRIPTORS FOR CASE STUDIES

Indictor	Cycle	Descriptor
Knowledge and understanding	1	have demonstrated knowledge and understanding of pre-requisites required to solve a case study
	2	have demonstrated knowledge and understanding that extends and enhances the prior knowing's
	3	have demonstrated knowledge and understanding that can assist in writing new algorithms
Applying knowledge and understanding	1	can apply their knowledge and understanding in making informed arguments
	2	can apply their knowledge and understanding in solving problems in the field of study
	3	can apply their knowledge and understanding in solving problems in new or unfamiliar environments.
Making Judgments	1	have the ability to gather and interpret relevant data
	2	have the ability to integrate knowledge and make judgments
	3	have the ability to analyze and reflect the inferences
Communication Skills	1	can communicate information and ideas
	2	can communicate conclusions and rationale
	3	can communicate problems, solutions and inferences
Learning Skills	1	have developed the learning skills to solve the problem
	2	have developed the learning skills to infer ideas
	3	have developed the learning skills to generalize ideas and principles

D. Sampling

A consent form was shared with all the students, stating that the data would be used for research purposes. With their signed consent and permission, following university guidelines, data was collected for the research work. Data was collected from 37 students who were enrolled in a course on Model Thinking from the School of Computer Science and Engineering department and had completed their second year at KLE Technological University, Hubli. The sampling technique used was purposive sampling (Sharma, 2017). The data collection included survey forms to be completed and case study solution sheets to be submitted in the form of online documents.

IV. METHOD

This section describes the objective and the case study design for the data collection.

A. Objectives

The objectives of the work are listed below in Table 2, formulated on the basis of the research question design.

TABLE II
OBJECTIVES

ID.	Objective
OB_1	To understand and comprehend system design problems using computational thinking
OB_2	To reflect on the different parameters of computational thinking
OB_3	To use Dublin Descriptors to write the course learning outcomes.

The objectives were used as a guide to formulate the case study and questions for reflection.

B. System Design Case Study

There were two questions designed on the system design problem: one on Ride sharing system and another on OTT platform.

The question on Ride Sharing System had the following parts to answer: Design a ride sharing service system (Example: Ola, Uber, etc.). Discuss the major components of this system and its operational design. Following are the guiding questions to answer: a. How can you decompose this system? b. What patterns do you observe in this system? c. What are the abstractions that you observe? d. What are the major components and algorithms that you can use in building such application?

The question on OTT Platform had the following parts to answer: Design a new OTT platform (Example: Netflix, Amazon Prime, etc.). Discuss the major components of this system and its operational design. Following are the guiding questions to answer: a. How can you decompose this system into different parts? What algorithms and data structures do you need to build each of them? Present an overall design. b. What patterns do you observe in this system? What potential problems do you observe and how would you solve them? Will these patterns help you in integrating the different parts of the system? If you are taking some decisions in the system design, explain why. c. What are the abstractions you observe for different business use cases? Can you optimize this system? What algorithms would you use? Can you generalize this for other applications? What major principles do you observe?

Question 2 was designed according to the Dublin Descriptors, which were tabulated in Table 1. Question 1 will be evaluated based on the learning outcomes written by Bloom's Taxonomy, and Question 2 will be evaluated as per the Dublin Descriptors.

V. Results and Data Analysis

This section presents the results and the data analysis. The submissions made by 37 students were qualitatively and quantitatively analyzed. The submitted answer sheets were graded based on the designed criteria. The results are presented in the following subsections. Two rounds of coding were carried out on the case study data for qualitative analysis. In the first round, major analysis was performed using in vivo coding and descriptive coding. In in vivo coding, we use the exact phrases and words collected from the case study experiences and perspectives, and this is suitable for a system design problem (Manning, 2017). Along with in vivo, descriptive coding was also employed. In descriptive coding, we code the passages according to the topic of interest (Holton, 2017). For the second round, focused coding was used. Focused coding was employed as we needed to categorize the related themes and merge them (Stuckey, 2015). The paper presents the summary and analysis of the coding methods.

A. First Round Coding

To comprehend the reflections conveyed by the students, we follow a methodical approach to inspect and identify the various themes that would provide a holistic representation of their reflections. During the qualitative coding phase, we utilized appropriate coding methods in the context of a case study analysis to identify the different themes present in the provided reflections. As the foundational principles of computational thinking were used in the reflections, the codes were mapped to it during the first round of coding. A few of the samples from student's reflections that was coded were mapped using Computer Science and Engineering (CSE) conceptual understanding. Thus, the two suitable coding methods utilized during the first phase were in vivo and descriptive coding.

In Vivo Coding, also known as "verbatim coding," or "inductive coding", in literature gets its name from its connection to living things. In Vivo coding works best as a first reading of the data, which not only creates a summary if read in order, but also a framework from each source which should be later combined with a higher level of second coding across all the data (Saldaña, 2021). This method of coding involves extracting words or short phrases from the participants authentic language found in data records capturing "the terms used by the participants themselves". It is an approach for qualitative studies, especially those that prioritize and respect the voices of participants. In Vivo Coding is particularly useful during grounded theory's Coding phase. Can complement other coding methods. Its distinct value is evident in ethnographies focused on youth, where it helps amplify marginalized voices and enrich our understanding of their cultures. Furthermore In Vivo Coding finds relevance in action and practitioner research by emphasizing interpreting participants terms within their lives than, through academic or professional lenses. In the context of a case study that explores the transition, from service to life In Vivo Coding plays a vital role in capturing participants authentic expressions without any changes. For example, one participant vividly described the challenges by saying, "I felt really anxious when I first left the military." Here the In Vivo Code "ANXIOUS" accurately represents the participants words. Another participant compared the adjustment process to learning a language and expressed it as, "Getting used to civilian life feels like mastering an entirely different language." In this case the In Vivo Code "MASTERING A LANGUAGE" captures their unique expression. These examples demonstrate how In Vivo Coding enables us to preserve participants genuine language providing a nuanced understanding of their experiences during analysis in this case study. Utilizing in vivo coding proves effective in maintaining alignment, with participants responses avoiding biases and ensuring relevance.

Here are a few samples of the student reflections that have been coded using In Vivo coding: "An effective database design ensures efficient data storage and retrieval, contributing to a streamlined user experience.", this was coded as USER EXPERIENCE as the end result of the process leads to a smooth user experience while using the ride-sharing system. Another student's reflection states, "The user service serves as a central repository for all user-related data and offers APIs for retrieving or modifying this information.", is coded as USER SERVICE, as it is the direct interpretation of the reflection and the major process that is spoken of. The student reflection, "Microservices Architecture: The system is decomposed into distinct microservices, each responsible for a specific functional area.", was coded as MICROSERVICES ARCHITECTURE, which is a major decomposition technique and CSE theoretical concept used in system design. It preserves the exact opinion of the entire statement. Another reflection, "User-Centric Design: Content recommendation algorithms prioritize user preferences, enhancing engagement and satisfaction.", was coded as USER-CENTRIC DESIGN, which is another decomposition technique, which focuses on the system design catering to the user's needs and requirements. The student reflection, "The abstractions and algorithms used for OTT platforms can be generalized to other applications" is coded as GENERALIZATION, and "UI frameworks (React, Angular, etc.) for creating interactive and responsive user interfaces" is coded as USER INTERFACE, and so on.

Descriptive Coding is a method of coding that effectively captures the topics present in qualitative data passages. Often referred to as "topic coding", it involves using words or phrases (nouns) to summarize the central themes of the data. This approach is versatile and can be applied across types of qualitative studies. Descriptive Coding forms a foundation for analysis by creating a vocabulary for the data providing valuable insights into the overall focus of the study. It produces a categorized inventory or summary of the data contents, which serves as groundwork for coding cycles and more advanced analysis. This method is particularly useful in documenting and interpreting material products and physical environments within contexts. By focusing on products experienced by participants in their lives, this approach enhances understanding (Saldaña, 2021). In a case study analyzing the impact of a company restructuring on employee experiences, Descriptive Coding is applied to distill key themes from qualitative data passages. Through Descriptive Coding, these themes are succinctly captured as codes: Job Security Concerns, Impact on Work Routines, Communication Disruptions, Feelings of Disconnection, and Uncertainty about Roles and Responsibilities. Each code represents the main thought or idea within the transcript, "Employees' expressions reveal concerns about job security, heightened anxiety, and disruptions to their work routines. Some participants express a sense of disconnection from the organization, citing communication challenges. Additionally, there is a prevailing uncertainty regarding the new roles and responsibilities introduced by the restructuring." These descriptive codes provide a structured and categorized representation of the employees' experiences, serving as a foundational step for subsequent analysis and interpretation in the case study.

In our research, we were able to identify the descriptive codes that emerged from the reflections, from both the case studies. Some of the descriptive codes that were picked out through the coding phase were: "System Interaction", "Systemic Overview", "Pattern-based Algorithm" and so on. These codes illustrate a comprehension of the reflections in a conceptual manner. A few of the samples are: "In any ride sharing system, a user would request for a cab using an app or API", is coded as SYSTEM INTERACTION, as it represents a user-system interaction for a particular task. The student reflection, "A designer ridesharing service system, similar to Ola, Uber, involves multiple components working together to provide a seamless and efficient experience for both riders and drivers", is coded as SYSTEMIC OVERVIEW, as it presents a holistic view of the ride-sharing system. Another reflection, "Optimize recommendation algorithms by incorporating machine learning techniques to enhance content suggestions", is coded as PATTERN-BASED ALGORITHM, as the content suggestions are amplified by understanding the patterns of user's usage using machine learning algorithms, which aligns with principles and theories within the field of CSE. The student reflection, "The user, content, and activity data could be sharded across multiple databases to distribute the load and enhance scalability." is coded as DISTRIBUTED SYSTEM, which outlines a key concept in CSE theory. Another reflection, "Implementing content caching and utilizing a CDN ensures faster content delivery and reduced load on the backend servers.", is coded as SYSTEM STABILITY, as Content Delivery Network (CDN) operates at the intersection of CSE principles. Lastly a reflection by a student, "Main goal is to create a scalable, user-friendly, and reliable platform for streaming movies and TV shows with personalized user experiences", is coded as CONCEPTUAL MOTIVATION as it outlines the major aim of their study, and so on.

Answers were analyzed to identify the themes. A sample of a few themes assigned is presented in Table 3 below. Several such themes were classified form the case studies. 32 such reflections were identified from the case studies.

TABLE III
SAMPLE THEMES AND CODES

Student Reflection	Theme	Code
In any ride-sharing system, a user would request for a cab using an app or API.	Pattern	System Interaction
The user interface is a critical element as it serves as the primary interaction point between riders, drivers, and the system.	Abstraction	User Interface
The system is decomposed into smaller, independent services, each responsible for a specific functionality.	Decomposition	Microservices Architecture
You start looking for something specific with a search engine.	Algorithms	Data-based Algorithm

The summary of identified themes is presented in Table 4 below from the set of 32 reflections. All others were categorized into one of the 32 identified reflections.

TABLE IV
GENERALIZATIONS

Theme	Code
Pattern	Conceptual Motivation Governing Rules System Interaction System Optimization Trends User Behavior User Experience User Workflow
Decomposition	Design Blueprint Distributed System Interacting Entities Mapping System Microservices Architecture Simplify The Complexity System Architecture System Integration System Operation User-centric Design Visual Application
Abstraction	Generalization System Management System Scalability System Security System Stability Systemic Overview User Interface User Service
Algorithms	Algorithmic Optimization Data-based Need-based Pattern-based Principle-based

B. Second Round Coding

In the second round of coding, emphasis is placed on the importance of focused coding to distill and consolidate data into meaningful categories, thereby facilitating a nuanced analysis that unveils patterns, relationships, and key themes essential for a comprehensive understanding of the system design case study (Saldaña, 2021). The codes derived from the first round of coding are organized into their relevant groups. Conceptual mappings were identified from the themes and codes. The answers described the system as a whole being interconnected. The line between the four components of computational thinking was blurred when the system was seen from a holistic perspective. The final themes that emerged are listed in Table 5. From the sample codes mentioned in the first round of coding, “user experience”, “user-centric design”, “microservice architecture” and “user service” are grouped under APPLICATIONS with other codes like “visual application” and “mapping system”, since these codes represents the system applications that are included in the designing of the system. Similarly, “generalizations”, “conceptual motivation” and “systemic overview” are categorized under COMPREHENSION as they illustrate the design of the system in a broader aspect. INTERACTIONS include the codes such as “user interface”, “system stability” and “system interactions”. “Distributed system” is grouped under INFERENCES, as they draw insights from the system design. Lastly, “pattern-based” is grouped under JUDGMENTS as the algorithm makes a final decision based on the patterns identified. The answers finally resulted in five major themes, which can be seen below.

TABLE V
FOCUSED THEMES

SI. No.	Theme
1	Comprehension
2	Judgments
3	Inferences
4	Interactions
5	Applications

The coding led to the design of quantitative evaluation criteria.

C. *Quantitative Analysis*

Quantitative analysis, built upon the insights gleaned from two rounds of qualitative analysis, forms a robust and comprehensive approach to understanding complex phenomena. The initial qualitative analysis serves as the foundation, unraveling rich narratives and uncovering nuanced patterns within the data. This qualitative groundwork allows for the identification and categorization of key themes, which are then translated into measurable variables for quantitative exploration.

Following are the parameters for quantitative analysis and corresponding themes, as presented in Table 6

TABLE VI
QUANTITATIVE ANALYSIS PARAMETERS

Criteria	Points	Description	Themes
Decomposition	8	Evaluate how effectively the system is decomposed into its constituent parts.	Comprehension. Applications
Pattern Recognition	8	Assess the identification and articulation of recurring patterns within the system design.	Interactions
Abstraction	6	Evaluate the use of abstraction in the system design.	Inferences
Operational Design	6	Examine how well the operational aspects of the system are explained.	Judgments, Comprehension
Communication of Conclusions and Rationale	8	Assess how well the conclusions and rationale behind design choices are communicated.	Inferences
Data Gathering and Interpretation	6	Evaluate the effectiveness of data gathering and interpretation in the case study.	Inferences, Judgments
Integration of Knowledge	3	Assess how effectively knowledge is integrated to make informed judgments in the system design.	Interactions
Communication of Problems, Solutions, and Inferences	6	Evaluate how effectively problems, solutions, and inferences are communicated.	Inferences, Applications
Usage of Diagrams	3	Assess the effective use of diagrams to illustrate system components and interactions.	Applications, Interactions
Feasibility and Realism of Design	2	Evaluate whether the proposed design is feasible and realistic.	Applications, Judgments

D. *Case Study Scores*

Table 7 below presents the overall scores of the two case studies designed.

TABLE VII
OVERALL SCORE

Section	Ride Sharing	OTT
Average	29.8919	31.8649
Variance	33.3397	26.5493
Standard Deviation	5.77405	5.15260

The two studies were validated using a t-test with a spreadsheet application. The formula was employed to produce the results. The hypotheses were written as follows:

Null: Two group means are equal.

Alternative: Two group means are not equal.

The results obtained can be seen in Table 8 below.

TABLE VIII
T-TEST RESULTS

T-Statistic	Scores
P(T<=T) two-tail	0.130479244
t Critical two-tail	1.993463539

As the p-value = 0.130479244 is greater than the value of significance (two-tailed test) = 0.05, we therefore accept the null hypothesis and reject the alternative hypothesis. We can thus conclude that the two group means are equal.

E. Student Feedback

Feedback was collected from the students regarding two questions. The first question was about the overall course delivery, and the second one captured the learning experiences related to the system design problem. The two sets of feedback are presented in Figures 2 and 3 below.

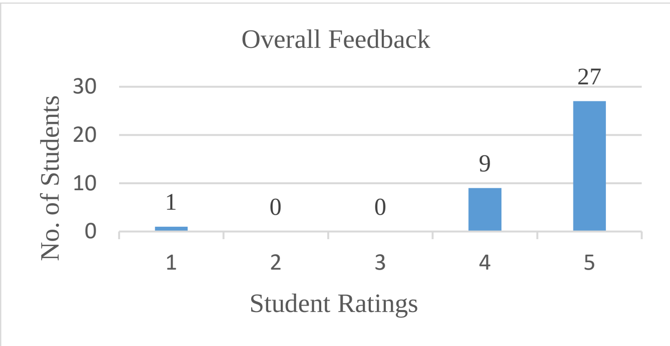

Fig. 2: Overall course feedback

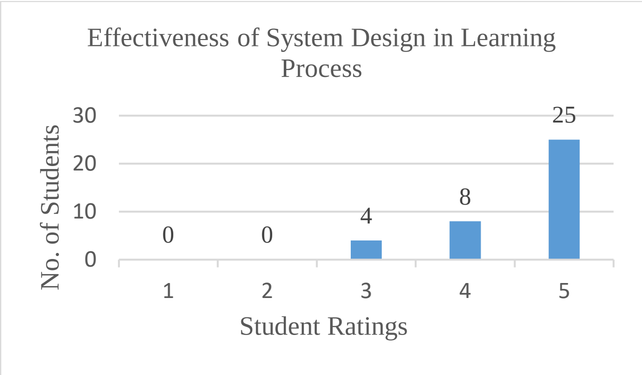

Fig. 3: Learning effectiveness of case study

We can note, from the feedback given by 37 students, that the learning process has been satisfactory.

VI. Discussion

This case study helps us arrive at intriguing discussion points. Though the mean values are almost the same, and we have concluded that the two-group means are equal. In order to explore this further, the themes generated were revisited for a more in-depth analysis of each case study. Table 9 below presents the number of themes that originated from each case study.

TABLE Ix
Themes from Each Case Study

Type	Number of Themes
Ride Sharing	10
OTT	18
Common to Both	4

The numbers clearly show that Dublin Descriptors help capture more themes than Bloom's Taxonomy. We can observe from Table 9 that OTT alone has led to 18+4 themes. This means that students are better able to connect to real-world applications.

With this study, we can conclude that when we want to study a concept with a layered approach, with incremental growth from solving a problem to generalizing it, then Dublin Descriptors are most suitable for the problem. Because it requires thinking in three cycles, complexity can be introduced in a phased manner. Information can be communicated from ideas to inferences. As evidence, if we observe the grading, it was done based on the following parameters for both problems, as shown in Table 10:

TABLE X
Scores on Identified Parameters

Criteria	Ride-Sharing System	OTT Platform
Decomposition	6.054	6.297
Pattern Recognition	3.757	3.811
Abstraction	3.757	3.622
Operational Design	5.135	5.622
Communication of Conclusions and Rationale	3.541	3.838
Data Gathering and Interpretation	1.378	1.595
Integration of Knowledge	3.541	3.811
Communication of Problems, Solutions, and Inferences	1.243	1.919
Usage of Diagrams	0.405	0.243
Feasibility and Realism of Design	1.081	1.108

The case study on the OTT Platform showcases an advantage when it comes to evaluating system design in a more balanced way compared to the Ride Sharing System. The incorporation of Dublin Descriptors in the OTT Platform offers a framework that encompasses not only computational thinking but also operational design, communication and knowledge integration. This inclusive approach, evident through higher mean scores for aspects indicates that Dublin Descriptors are more effective in capturing the multifaceted nature of the system design problem compared to the Ride Sharing Systems narrower focus on computational thinking. The versatility of Dublin Descriptors makes them a valuable tool, for conducting evaluations of system designs ultimately enhancing the overall quality of the assessment process. Students' feedback has been positive, indicating that the process has been effective in their learning. Thirty-three out of 37 students were satisfied with the system design problem.

VII. Conclusion

For problems that require iterative learning, Dublin Descriptors prove to be effective. They highlight several themes in students and enable them to connect with real-world problems. We started this research work with the objective of understanding and comprehending system design problems using computational thinking, which we achieved through two case studies. We aimed to reflect on the different parameters of computational thinking, and we accomplished this using Bloom and Dublin Descriptors. Additionally, we found that using Dublin Descriptors to formulate the course learning outcomes was effective for incremental Problem-Based Learning (PBL) case studies.

References

Adam, S. (2004, July). Using learning outcomes. *In Report for United Kingdom Bologna Seminar* (pp. 1-2).

Anderson, J. R. (1993). Problem solving and learning. *American psychologist, 48*(1), 35.

Barrows, H. S. (1998). The essentials of problem-based learning. *Journal of Dental Education*, *62*(9), 630-33.

Baxter, P., & Jack, S. (2008). Qualitative case study methodology: Study design and implementation for novice researchers. *The qualitative report, 13*(4), 544-559.

D'zurilla, T. J., & Goldfried, M. R. (1971). Problem solving and behavior modification. *Journal of abnormal psychology, 78*(1), 107.

Duncker, K., & Lees, L. S. (1945). On problem-solving. *Psychological monographs*, *58*(5), i.

Jona, K., Wilensky, U., Trouille, L., Horn, M. S., Orton, K., Weintrop, D., & Beheshti, E. (2014, January). Embedding computational thinking in science, technology, engineering, and math (CT-STEM). In *future directions in computer science education summit meeting, Orlando, FL*.

Garrison, D. R. (1997). Self-directed learning: Toward a comprehensive model. *Adult education quarterly, 48*(1), 18-33.

Gerring, J. (2004). What is a case study and what is it good for?. *American political science review, 98*(2), 341-354.

Gudeva, L. K., Dimova, V., Daskalovska, N., & Trajkova, F. (2012). Designing Descriptors of learning outcomes for Higher Education qualification. *Procedia-Social and Behavioral Sciences*, *46*, 1306-1311.

Hegade, P. (2019). One-Day Many-Problems: A Problem Based Learning Approach. *Journal of Engineering Education Transformations*, *33*(1), 1.

Hegade, P., & Shettar, A. (2023). Effectiveness of Computational Thinking in Problem Based Learning. *Journal of Engineering Education Transformations*, *36*.

Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn*?. Educational psychology review*, *16*(3), 235-266.

Holton, J. A. (2017). 15 From Grounded Theory to Grounded Theorizing in Qualitative Research. *The SAGE handbook of qualitative business and management research methods: History and traditions*.

Hung, W., Jonassen, D. H., & Liu, R. (2008). Problem-based learning. *Handbook of research on educational communications and technology*, *3*(1), 485-506.

Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into practice*, *41*(4), 212-218.

Logman, P. S. W. M., & Kautz, J. (2021, May). From Dublin Descriptors to implementation in Bachelor labs. In *Journal of Physics: Conference Series* (Vol. 1929, No. 1, p. 012065). IOP Publishing.

Lohman, M. C., & Finkelstein, M. (2002). Designing cases in problem-based learning to foster problem-solving skill. *European Journal of Dental Education*, *6*(3), 121-127

Manning, J. (2017). In vivo coding. *The international encyclopedia of communication research methods*, *24*, 1-2.

Masaev, S. N., Dorrer, G. A., Vingert, V. V., Yakimova, E. A., & Klochkov, S. V. (2020, November). Dublin descriptors. In *Journal of Physics: Conference Series* (Vol. 1691, No. 1, p. 012021). IOP Publishing.

Prince, M. (2004). Does active learning work? A review of the research. *Journal of engineering education*, *93*(3), 223-231.

Saldaña, J. (2021). The coding manual for qualitative researchers. sage.

Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition.

Sharma, G. (2017). Pros and cons of different sampling techniques. *International journal of applied research*, *3*(7), 749-752.

Stuckey, H. L. (2015). The second step in data analysis: Coding qualitative research data. *Journal of Social Health and Diabetes*, *3*(01), 007-010.

Van Dijk, L., Van Der Berg, G. C., & Van Keulen, H. (2001). Interactive lectures in engineering education. *European Journal of Engineering Education*, *26*(1), 15-28.

Williams, B. (2001). Developing critical reflection for professional practice through problem-based learning. *Journal of Advanced nursing*, *34*(1), 27-34.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.