Framework for Generation and Evaluation of Assessment Instruments

Submitted in partial fulfilment of the requirements
of the degree of

Doctor of Philosophy

by

Rekha Ramesh

Roll no: 114388002

Supervisor:
Sridhar Iyer (IIT Bombay)

Co-supervisor:
Sasikumar M. (IIT Bombay)

Interdisciplinary Program in Educational Technology

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

2017

Dedication Sheet

Dedicated

to

my Family

and

the Almighty

Thesis Approval Sheet

This thesis entitled **"Framework for Generation and Evaluation of Assessment Instruments"** by Rekha Ramesh is approved for the degree of Doctor of Philosophy.

Examiners

____

____

____

Supervisor (s)

____

____

____

Chairman

____

Date : 17/10/2017

Place : Mumbai

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Rekha

(Signature)

Rekha Ramesh

(Name of the student)

114388002

(Roll No.)

Date: 17/10/2017

Abstract: Assessment is an integral part of instruction and it has a profound influence on what students' study, how much they study and how effectively they study. The design of an assessment instrument (AI) is one of the major components of assessment process. If students are subjected to random and unfair AIs, then assessment may not serve the intended purpose. Hence, the quality of any educational assessment exercise depends on the quality of AI used. Process of evaluation of AI quality brings lot of subjectivity into it as the benchmark can vary from person to person. Manual evaluation of quality by considering all the parameters is a very cumbersome task.

With this background, the broad research objective of the thesis is "How to improve the quality of AI and how technology will assist in this process? There are two ways to improve the quality of AI, namely (i) Evaluating the quality of teacher generated AI and providing feedback or (ii) Automatically generate the AI, so that the quality is ensured at the time of creation itself. The first approach is the major focus of my thesis. Pursuing these two approaches resulted in the design of two frameworks for the generation and evaluation of AI.

There are many quality parameters of AI. Based on literature, the measure of alignment of AI against the Learning Objectives (LOs) of the course is adopted as the quality of AI in our work. Implementation of frameworks resulted in the development of tools, namely, Instrument Quality Evaluator (IQuE), Teacher Training Module (TTM) and AI Generator (AIGen).

IQuE measures the quality of AI in terms of its alignment with the LOs of the course. An ontology based Knowledge Representation (KR) mechanism is designed to integrate the contents of syllabus, LOs and AI. Content and cognitive level information are extracted from LOs and questions using simple Natural Language Processing (NLP) techniques. Measure of alignment is formulated based on the commonalities and differences in concepts covered and cognitive levels from LOs and questions respectively. IQuE provides two types of outputs; a numerical measure of alignment and its visual representation. It also estimates the utility of each question indicating its contribution towards LOs. Accuracy of IQuE was tested with large number of samples (N=1000) and the accuracy with respect to content and cognitive level alignment are 91.2% and 93.23% respectively.

Using IQuE, a TTM is developed that can be used to train teachers to write good assessment questions against given LOs. It has a multistage environment and is supported by a formative feedback mechanism that gives the feedback about the alignment of teacher written questions against a system displayed LOs.

We have built a prototype version of AIGen that facilitates automatic generation of AI from the teacher entered AI specification (AIS) and tagged question repository. In the preliminary investigation, it was found that the generated AIs were 80% compliant with the corresponding teacher entered AIS.

The context of the research work is AIs designed for written examinations in a typical university scenario in engineering curriculum. All the samples for the study are taken from the *Data structures* course in engineering curriculum.

Keywords: *Quality of Assessment Instrument, Ontologies, Alignment, Automatic Assessment Tools, Instructor interfaces.*

Table of Contents

List of Figures

List of Tables

Abbreviation Notation and Nomenclature

AI	Assessment Instrument
AIGen	Assessment Instrument Generator
CAM	Content Alignment Matrix
CAS	Content Alignment Score
IEEE LOM	Learning Object Metadata
IEEE	Institute of Electrical and Electronics Engineers
IMS QTI	IMS Question and Test Interoperability specification
IQuE	Instrument Quality Evaluator
KR	Knowledge Representation
LAM	cognitive Level Alignment Matrix
LAS	cognitive Level Alignment Score
LO	Learning Objective
LOA	LO Annotator
LOIA	LO and Instrument Annotator
QR	Question Repository
TTM	Teacher Training Module

Chapter 1

Introduction

Assessment plays a crucial role in teaching and learning. It is an ongoing process aimed at understanding and improving the level of students' learning (Rowntree, 1977). It involves generating and collecting evidence of a learner's attainment of knowledge and skills and judging that evidence against defined standards. Assessment is a broad term defined as a process for obtaining information that is used for making decisions about students, curricula and educational policy (Nitko, 2001) (SQA Handbook, 2009).

Assessment has many different purposes. It plays a major role in understanding how students learn, their motivation to learn and how teachers teach. Assessment offers information to students about the knowledge, skills, and other attributes they have acquired after successfully completing course work and academic programs (Heywood, 2003). It provides opportunities for the academic community to engage in an introspection of its set learning goals, in order to determine the degree to which these goals correspond to student and societal needs. It is a mechanism to evaluate if students' activities, products, or performances coincide with the academic community's expectations (Brown, 2004) (SQA Handbook, 2009). It helps academic units to understand the dimensions of student learning that can be used to improve students achievement and the educational process (Heywood, 2003) (Pellegrino, Chudowsky & Glaser, 2001).

For the employers, assessment provides information about the potential of job applicants. It is also a mechanism for selecting students for higher studies and may be a factor in the reputation and financial security of institutions in higher education. Certification to conform that a student has achieved a particular level of performance is another important purpose of assessment. This is indicated in terms of the outcome of assessment which can be in the form of simple ‘pass’ or fail’, ‘competent’ or ‘not yet competent’, or it can be a certificate showing grade, marks, or percentages. Students give utmost importance to it as this directly reflects on their employability and prospective earnings (Rowntree, 1977).

Assessment is embedded in the learning process. It is closely interconnected with curriculum and instruction (Frye, McKinney & Trimble. 2007). Given at the beginning of a course, assessments help in estimating the prior knowledge of students (diagnostic assessments). This facilitates the instructor to know where to begin the instruction and/or identify areas of remediation that must be addressed. Frequent assessments during the course help the teacher and students see the progress of learning and help identify problem areas where students need more help or time (formative assessments). Given at the completion of instruction, assessments tell students how much has been learned by the end of the term (summative evaluation) (Kennedy, Hyland, & Ryan, 2006) (J. Biggs, 2003) (Krathwohl & Anderson, 2002). They provide the basis for making judgments on the grades to be assigned to each student.

Assessment is a complex process that starts with determining the objectives of the assessment, selecting proper type of Assessment Instrument (AI), selecting set of questions for the instrument, conduction of assessment, and finally grading. For a particular course, assessment objectives are determined by the LOs defined for the course. The LOs are statements that define the expected competencies the students should achieve after the successful completion of a course (Heywood, 2003).

The design of an AI is one of the major components of assessment process. If students are subjected to random and unfair AIs, then assessment may not serve the intended purpose. Hence, while generating an AI, its quality needs to be evaluated. This should be done based on the perspectives of both the examiner and examinees. Fairness, accuracy, consistency and elimination of bias are very important while selecting questions for the instrument. One of the ways to do this is to get a feedback from the examinees and other teachers related to the same course. However, this is possible only after the examination is over.

It has been widely reported that the AIs designed to test the students' achievements in engineering education in most of the universities in India are of poor quality (Banerjee & Muley, 2007) (Phatak, 2015). We conducted a preliminary study investigating the quality of AIs of various courses offered as part of engineering curriculum from various universities in India. A quality metrics was formulated which includes various parameters such as

- Coverage of syllabus
- Coverage of cognitive levels as per the LOs of the course
- Difficulty level of questions in AI and
- Distribution of marks.

It was found that most of the questions were not aligned to LOs defined for the course. 75% of the questions were catering to only lower order thinking skills whereas the LOs were expecting higher order thinking also.

With the growing number of students in most of the institutions, a single course is usually handled by multiple teachers and often *there is a shortage of experienced faculty and domain experts*. Hence, it cannot be guaranteed and may be practically impossible that AIs will be designed only by experienced teachers. This makes the situation even worse. Assessment being the lone measure of competence in the current education system of our country, it has a decisive role in career building of students. Further, it is an irreversible process and involves a lot of effort and expenditure on the part of organization or university. All these makes ensuring the quality of AI an important concern.

Even though many of the *universities provide guidelines for creating an AI, most of it is concentrating on the structural information* such as number of sections, sub sections, questions within a section or subsection, the type of questions, the marks distribution, etc. This does not ensure the quality of instrument. There are *no proper procedures or constraints specified concerning the quality of an AI.* Although the general notion of quality is fairly well understood by teachers, formal mechanism for ensuring it is lacking. Hence the *quality of AI varies depending on the individual teacher's experience and expertise in the course and the instrument setting process.*

Process of evaluation of instrument quality brings lot of subjectivity into the process as the benchmark can vary from person to person. Manual evaluation of quality by considering all the parameters is a very cumbersome task. Today, even *experienced teachers have to spend a lot of time and effort in manually ensuring the quality* of AI. It is more difficult for novice teachers. They lack the expertise in creating quality AI as they are not trained for that and they do not have the experience. Hence, there is a need for formalizing and automating the process of evaluating the quality of AI. All the factors that led to pursuing this research work are summarized in Fig. 1.1.

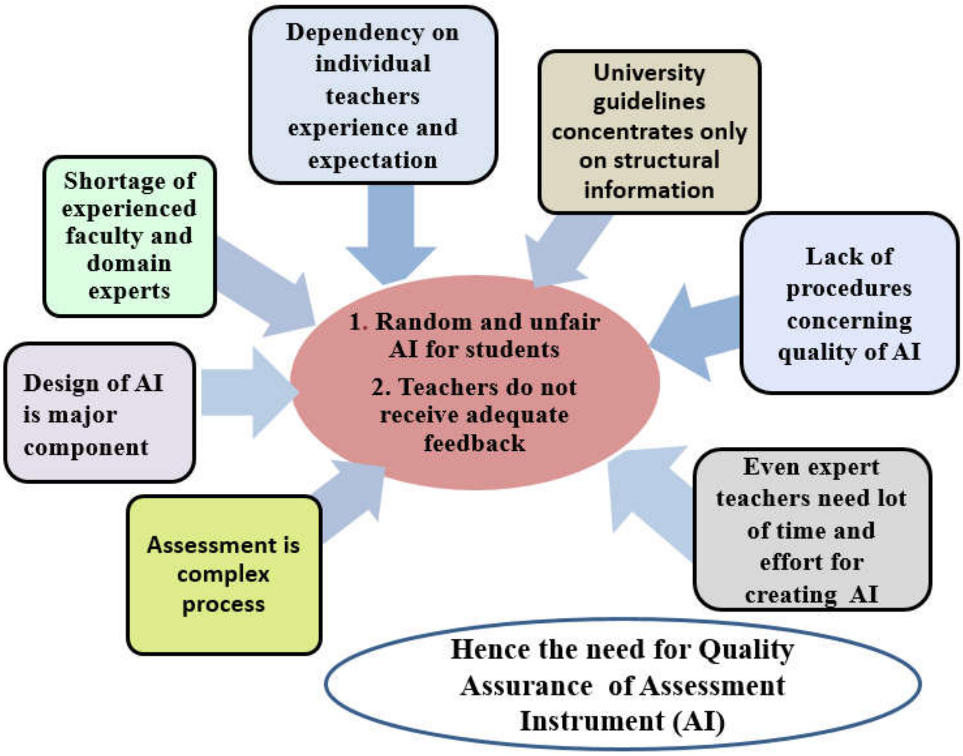


Figure 1.1. Factors that affect the quality of AI

1.1 Research Objectives

The focus of the research is on finding what technology interventions can be designed to address the problem of improving the quality of AI and to what extent they can solve this problem. This led to further investigation of the solution approaches and formulation of research objectives. The journey of the research work is depicted in Fig. 1.2.

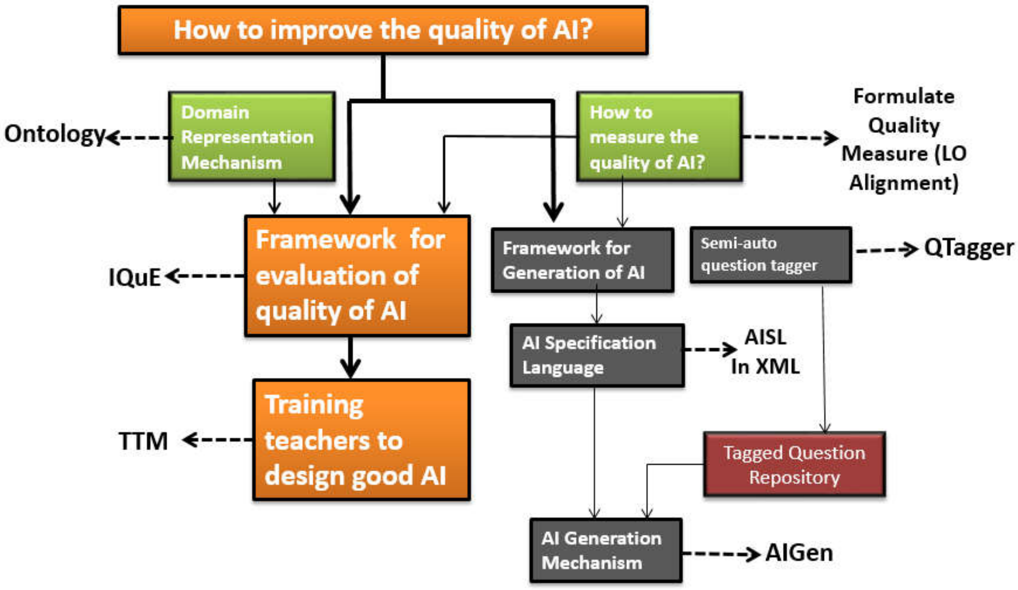


Figure 1.2. Journey of research work

The research work started with the objective of improving the quality of AI. To improve the quality, we need a measure for the quality. Hence the next step involved exploration of parameters defining the quality of AI. Literature surveyed indicated various parameters such as proper proportions of cognitive level, difficulty level, type of question, distribution of questions among the different sub topics allotted for assessment, distribution of marks across the instrument, etc. (Agrawal, 2008). Some of the parameters such as proper proportions of cognitive level and distribution of questions among the different sub topics constraints are satisfied if the AI is aligned with the LOs of the course for which the instrument is designed (Krathwohl, & Anderson, 2002). For defining the quality of AI, one should find solutions to questions such as

- Does it cover the syllabus in a 'fair' way?
- Are the questions properly aligned with the learning objectives of a course?
- Is there redundancy among the questions?

There are two ways to improve the quality of AI (i) Let the teachers create AI and then the system will evaluate its quality along different dimensions before it is given to students. (ii) Providing an AI Generation framework that will contain an interface for teachers to enter AI Specification (AIS) and automatically generating an AI compliant with AIS.

In the first method, AI can be revised till it meets the expected standards. The major focus of the research work is in this direction. Given an AI, how to formalize mechanisms for evaluating its quality along dimensions such as adequate coverage of syllabus and fairness to LO? Thus, the primary research objective is ***designing a framework for evaluating the quality of AI in terms of its alignment with the LOs of the course.*** The framework is operationalized by building an Instrument Quality Evaluation (IQuE) tool that will automatically measure the quality of AI. Teachers can use this tool to evaluate the quality of AI created by them before it is given to students. IQuE will give them feedback about the alignment and teacher can then improve AI and check again for alignment. This can be repeated for many number of cycles till the alignment of the AI is proper.

It has been reported that teachers have difficulty in setting aligned assessment questions corresponding to a given LO. Hence, a teacher training module (TTM) to train the teachers is required. Thus, the second research objective is to ***design and develop a training module (TTM) that will train teachers to set good assessment questions against a given LO***. This results in TTM tool that uses IQuE to check the alignment of teacher set questions against the system displayed LO and gives feedback about the alignment.

In the second method, the quality of AI can be ensured at the time of creation itself. AIS will capture constraints and requirements for the development of AI. Hence, the third research objective is ***to design an AI Generation framework which will take a teacher entered specification and produce an AI***. We have built a prototype model of such a system which is called AIGen (Nalawade & Ramesh, 2016). The AIGen needs a tagged question repository of the domain so that the system will interpret the AIS and intelligently select the questions from the repository. Creating tags is difficult and time consuming. Hence, a Semiautomatic Question Tagging system (QTagger) which takes an assessment question as input and attempts to identify various tags such as Bloom's level, type of question, difficulty level, and the content or topic of the question is implemented (Rekha et al., 2014).

1.2 Research Methodology

The primary research objective of this work is to improve the quality of AI. Hence the broad Research Question (RQ) to be answered is "How to build a framework for the generation and evaluation of AI that will improve the quality of AI?"

The framework is built using the design and development research (DDR) method with three distinct phases (Richey Klein, 2005).

1) **Need and context analysis phase:** In this, synthesis of literature reviewed from various sources was done to reinforce the need and establish the context of framework
2) **Design and development phase**: It includes the iterative cycles involved in the development of IQuE system with an attached TTM module and a prototype model of AIGen system.
3) **Summative Evaluation phase**: In this phase, all the products are evaluated for their accuracy, usability and usefulness. Both qualitative and quantitative analysis methods are used for evaluation of the products.

The *Data Structures* course of second year engineering curriculum is chosen as the domain and all samples are taken from that. For testing the accuracy of IQuE, AIs of various universities within and outside India are taken as samples. Teachers from various colleges affiliated to Mumbai University participated in the usability and usefulness study of all systems.

1.3 Scope and Limitations of Thesis

Currently, the scope is limited to engineering curriculum as all the samples for the study are taken from the *Data Structures* and similar courses in engineering. We believe that the system can work on a range of domains and courses other than engineering curriculum. But formal testing has not been done. Similarly, the type of assessment is restricted to written examinations in a typical university scenario where teachers generate the AI for the prescribed syllabus and students write the examination at the designated time and venue. Hence, all other types of assessments such as oral examinations, laboratory tests, project, etc. are beyond the scope of our work.

There are many quality parameters of AI such as proper proportions of cognitive level, difficulty level, type of question, distribution of questions among the different sub topics allotted for assessment, distribution of marks across the instrument, time allotted to students to solve the questions in AI, ambiguity in wording of questions, its alignment with the learning objectives of the course, etc. But the emphasis of framework is on alignment problem not on other aspects of quality.

Domains which are more abstract and non-textual such as Mathematics or Drawing are not in the immediate focus of this research work. As the aim is to have an automated solution, we tried to investigate how much and what aspects of the system can be addressed in a machine processable way. Each question and LO from the AI is machine processed to extract content and cognitive level information from it. The system will not be able to analyse certain types of questions in which there is no indication of any concepts involved. For example, in question such as "Solve y=f(x)" there is no way the system can automatically find what concept from mathematics course is involved in it. Same problem occurs, if it is a video content question or a picture question such as "Given a picture, interpret it"

The domain ontology is the backbone of the framework and hence comprehensive domain ontology must be available for the domain. A guideline for creating ontology for a domain is part of the framework.

1.4 Thesis Contribution

Thesis has contributed to the area of educational assessment specifically towards improving the quality of AI. The major contributions are in the form of educational technology tools such as IQuE, TTM and AIGen that can facilitate teachers in the evaluation and generation of AI. There are some by-products that got developed as supporting systems to major ones. Following are the contributions from this thesis.

- A framework to evaluate the quality of AI
- An Instrument Quality Evaluator (IQuE) tool that realizes the framework. IQuE measures the quality of AI in terms of its alignment with the LOs of the course.
- Teacher training module (TTM) that can be used to train teachers to write good assessment questions against given LOs.
- A language to define AI specification (AIS) which encompasses its structural as well as behavioural aspect.
- A framework for the automated generation of AI.
- A prototype model of a tool (AIGen) that realizes this framework. AIGen interprets the teacher given AI specification and selects the question from a question repository and automatically generates the AI.
- Software based support system (QTagger) which suggests metadata for a given question.
- Guidelines for building a domain ontology to represent the syllabus.

- Formulation of alignment measure of AI in terms of its content and cognitive level.
- Design of a visual representation of alignment.

1.5 Structure of Thesis

This thesis is organized as follows:

Chapter 2 discusses the literature surveyed for research work. It presents the background and related work carried out to get a deeper insight towards the need and solution approach for answering the research objectives formulated in Section 1.1. It mainly concentrated on nature and type of assessment, role of AI in assessments, quality parameters of AI, formulating the quality measure of AI and finding the appropriate technology intervention that can address the problem of generating and evaluating the quality of AI.

Chapter 3 provides the solution approach followed to achieve the research objectives. It describes the framework designed for generation and evaluation of AI. Chapter 4 gives the details of the research methodology adopted for the complete thesis. Chapter 5 establishes the need and context analysis for all the objectives of the thesis. Chapters 6, 7, 8 and 9 describe the design, development and evaluation of IQuE, TTM and AIGen tools respectively. Chapter 10 connects the research objectives to the achieved solutions and discusses the generalizability of the solution and scope and limitation of thesis. Chapter 11 concludes the thesis by providing contributions and future research directions.

Chapter 2

Review of Literature

As the area of research is educational assessment, the exploration started with the understanding of all the dimensions of assessment domain. Systematic domain literature survey was carried out which included journal papers, assessment handbooks, relevant web sites and conference papers. The survey concentrated on the assessment process as a whole and has delved into fundamental questions about educational assessment, such as its purposes; which kinds of knowledge and skills should be assessed; how well current assessments are fulfilling the various demands placed on them; and what are the methods and guidelines suggested for the best practices in assessment. As the focus is specifically on improving the quality of AI, the literature was also surveyed on guidelines for setting a good AI established by different universities within and outside India. In spite of having such guidelines, various reports and articles were found expressing concern about the poor quality of instrument. This survey helped in understanding the complex process of creating a good quality AI as well as the amount of effort and time teachers have to spend in it. This further reinforced the necessity of building a framework for generation and evaluation of AI.

Further exploration was done in various parameters defining the quality of AI in order to formulate a measure of quality. Similarly, literature was surveyed to identify a knowledge representation (KR) mechanism for the domain.

In order to train teachers in writing aligned assessment questions against given LOs, we wanted to build a training module. Even though we could not get any existing system directly matching to our requirement, the literature survey was done in this direction to explore various elements of general tutoring systems to integrate into our system.

For an automated AI generation, an interface where the user can enter AI specification (AIS) and a question repository from which the relevant questions can be selected as per the specification are needed. For this, we explored the literature on existing model of AIS also called a blueprint as well as different tags/metadata that can be attached to each question in the repository.

2.1 Educational Assessment Theory and Practice

This section explores the assessment domain with respect to its purpose on different stakeholders, the different methods of assessing, what should be done with the results of assessments and what are its implications on students.

Even in early 1970s, researchers found that assessment influenced students most, not the teaching. Rowntree stated that "If we wish to discover the truth about the educational system, we must first look to its assessment procedures" (Rowntree, 1977). He provided a framework for the exploration of assessment domain and identified five dimensions of assessment that should be carefully thought about before the actual assessment procedure is carried out. These five dimensions are (i) Why assess? (ii) What to assess? (iii) How to assess? (iv) How to interpret? and (v) How to respond? If a teacher has answers to all the above questions in a clear and unambiguous manner, then a valid AI that perfectly suits to the requirement can be designed. An assessment framework that provides an overall outline or plan to guide the development of assessment tests, questionnaires, and procedures is crucial in determining the contents of an assessment instrument (Anderson & Morgan, 2008). The assessment needs to be 'fit-for-purpose'. It should enable evaluation of the extent to which learners have learnt and the extent to which they can demonstrate that learning (Brown, 2004). Teachers need to consider not just what we are assessing and how we are doing it, but also, why. Brown has designed some interesting tips on "How to use assessment to prevent learning!" that make teachers and students think about some of the behaviours that can actually get in the way of students' learning. Nitko provides complete coverage of educational assessment including developing plans that integrate teaching and assessment, crafting objective, emphasizing the importance of valid and reliable AI and creating them and evaluating and grading assessments (Nitko, 2001). It is now well understood that assessment has a profound influence on students' learning but not so well understood that institutional structures and procedures have an equally profound influence on teaching and learning and the way learning is assessed. Often changes are made in structures and procedures that pay little attention to either cognitive and personal development of the students or to the way in which they learn. Heywood in his book suggested an integrated model for assessment-curriculum-learning-teaching process (Heywood, 2003). Every assessment, regardless of its purpose, rests on three pillars: a model of how students represent knowledge and develop competence in the subject domain, tasks or situations that allow one to observe students' performance and an interpretation method for drawing inferences from the performance evidence thus obtained (Pellegrino, Chudowsky & Glaser, 2001). As per Gibbs, assessment makes more difference to the way that students spend their time, focus their effort, and perform, than any other aspect of the courses they study, including the teaching. He elaborates the pedagogic principles underlying the use of assessment to support learning (Gibbs, 2010). Many universities provide assessment handbooks that give comprehensive guidelines for instructors to construct tests and assess student achievement more accurately and also procedures for successfully administering tests, analysing test questions, and assigning grades. They look at the principles of assessment and bring together information on assessment from the rich literature on assessment in general as well as on best practice in assessment (Assessment Handbook, 2012), (Davis, 2002), (Guide to Assessment, 2009), (Handbook on Testing and Grading, 1991).

2.2 Quality of Assessment Instrument

The quality of any educational assessment exercise depends on the quality of the instruments used. If these instruments are poorly designed, the assessment can be a waste of time and money. Anderson and Morgan clearly identify the characteristics of a good question that contributes to the AI. According to them, good questions are clear, relevant to the curriculum and focused on aspect of learning. They provide engaging and genuine tasks that are fair to all types of students (Anderson & Morgan, 2008). As per Sadler, there are many judgmental processes involved related to instrument design process such as to what extent the course objectives are met and why certain set of questions are appropriate for inclusion in AI (Sadler, 2005). The National Council of Educational Research and Training (NCERT), an autonomous organization of the Government of India has established guidelines for setting a good AI. It defines the following attributes to be associated with each question to be selected in the AI:

- Question id
- Objective tested by the question
- Specification on which the question is based
- Topic covered
- Question type
- Marks allotted
- Approximate time required for answering
- Estimated difficulty level

Validity is considered as the most important quality measure of a good AI (Agarwal, 2008). Validity is confirmed when the AI is aligned to the objectives of the course (Nitko, 2001). The assessment tasks are the key components of the teaching system and they should be aligned with the learning activities assumed in the intended outcomes (Biggs, 2003). Assessment becomes valid when the evidence of student's achievement clearly matches against the objectives of assessment (Brown & Glasner, 1999) (Gibbs & Rowntree, 1999) (Thorpe, 2000).

Many researchers have emphasized that the most important quality parameter is alignment of AI with LOs of the course. Hence, the next objective was how to measure this. As per Krathwohl, LOs are usually framed in terms of (i) some subject matter content and (ii) a description of what is to be done to that content. LOs typically consist of noun or noun phrase which form subject matter content and verb or verb phrase that define the cognitive processes involved in it (Krathwohl, 2002). Hence, alignment of AI with the LOs of the course is proposed as a measure of quality in this work. The content and cognitive processes involved in each question from an AI and each LO in the set of LOs are considered as parameters for alignment measure.

2.3 Technology Requirements for Automation

The main components associated with creating quality assessments are syllabus, LOs and AI. To automate the task of measuring alignment of AI with the LOs of the course, a KR mechanism is needed to represent the syllabus for which AI is to be designed. The KR has to be machine parsable and capable of effectively capturing the complex structure of syllabus. LOs and questions are framed based on the contents of syllabus. The KR should have the capability to connect these components. Then, there should be an automated extraction method that identifies relevant contents from LOs and questions. Finally, a formula that will measure the alignment from the extracted information is required. Following subsections elaborate the literature surveyed in this direction.

2.3.1 Ontology as a Knowledge Representation Mechanism

As mentioned before, there are specific requirements expected of the KR mechanism that will be selected. It has to be machine parsable and also capture structure of the syllabus involving hierarchical and dependency relations among various concepts from the syllabus. In pursuit of such a KR mechanism that will satisfy all the requirements, it was found that there is already a considerable research dealing with the application of ontological engineering to education field. The researchers have classified them into curriculum or syllabus ontology creation, ontology-based learning object organization, and ontology-based learning content retrieval (Mizoguchi, 2004) (Mizoguchi & Bourdeau, 2015). Ronchetti and Sant have presented a case study on a human centered approach to developing a tool for managing, inspecting and monitoring curriculum using an ontology based representation of the curriculum (Ronchetti & Sant, 2010). The authors conclude that the visualization of curricula via ontologies is a good possibility to represent the structure and the dependencies between courses clearly and allow one to share, exchange, reuse, analyze and extend curricula. Chi Y. reports a study where an ontological mechanism is utilized in terms of a knowledge intensive approach to create general course sequence for planning the learning from the information integrated from multiple textbooks (Y. Chi, 2010). This was done in an e-learning environment where '*one course multiple text*' is a popular phenomenon to provide flexibility to learners. The design of the integrated learning ontology conceptualizing multilevel knowledge structures such as curriculum, syllabus, learning subject and materials was proposed by (Chung & Kim, 2016) (Fok & Ip, 2007). Fok and Ip in their book, address the issues and methodologies in the design and construction of education ontologies and discusses the necessities of such an ontology that can help in organizing, retrieving and recommending educational resources for personalized learning.

2.3.2 LO and Question Analysis

There is extensive literature survey available on the usage of Natural Language Processing (NLP) techniques in various fields of study such as machine translation, natural language text processing and summarization, user interfaces, multilingual and Cross-Language Information Retrieval (CLIR), speech recognition, artificial intelligence, and expert systems. Chowdhury has compiled use of NLP in all such applications (Chowdhury, 2003). The central task for natural language text processing systems is the translation of potentially ambiguous natural language queries and texts into unambiguous internal representations on which matching and retrieval can take place (Liddy, 1998). Liddy's explanation exactly matches with the requirements of our research work. The content and cognitive level information has to be extracted from questions and LOs. These have to be mapped to an ontology for further processing. As LOs and questions are written in natural language which is English in the present context, NLP techniques seemed to be a suitable option for extraction. Apart from extracting words, we are also interested in the part-of-speech tags of words and specific phrases that can act as question indicators to identify cognitive processes involved in LOs and questions.

2.4 Automated System for Evaluation of AI

One way to improve the quality of generated AI is to evaluate its quality before it is given to students. We would like to find how much difficulty is there in manually ensuring the quality and what are the technology requirements if the process is to be automated.

2.4.1 Difficulty in Manually Ensuring Quality

It is generally agreed that designing an AI is a complex process and lot of time and effort are needed to manually ensure the quality of AI (Aldabe, et.al, 2006). Many teachers dislike preparing and grading exams because of its cumbersome nature and complex process (Davis, 2002). With the growing number of universities in India and number of students in each university, often there is shortage of experienced faculty and domain experts (Banerjee & Muley, 2007). Analysis of the AIs of the engineering curriculum from many of the universities in India revealed that they are of poor quality catering only to lower order thinking skills. Teachers have difficulty in framing LOs as well as writing assessment questions against those LOs (Reddy & Mahajan, 2016). Lack of formal training for teachers in engineering education makes thing even worse. These facts have repeatedly come as a major concern in internal reviews within the universities, panel discussions in academic conferences, keynote addresses and talks of eminent academicians and media reports. To make the life of students and teachers easier, wherever possible we must make best use of the available technologies to make assessment more efficient (Brown et al., 1994).

2.4.2 Formulating the Measure of Alignment

To measure the content alignment, we looked at literature involving an information theoretic definition of similarity which is frequently used in information retrieval systems. In such systems, similarity measure is needed to retrieve relevant documents from web matching the keywords from the user given query.

Xia and Yihong have applied the formula based on commonalties to calculate the similarity between two concepts in ontology (Xia & Yihong, 2006). Lin has presented a variety of definitions of similarity in terms of information theory where a similarity measure is derived from a set of assumptions. These assumptions are based on the intuition that the similarity between A and B is related to their commonalities and differences and the maximum similarity is reached when A and B are identical. A and B can be ordinal values, feature vectors consisting of words and strings or taxonomies (Lin, 1998). Lin’s assumptions are valid in the scenario of finding the measure of content alignment between LO and question. The commonalties and differences in the concepts covered by them can be used to find the measure of alignment.

2.5 Teacher Training System

As mentioned previously, teachers face many challenges while framing aligned questions for their AIs against a set of LOs of the course. Hence, the decision of building a tool was taken that can train teachers to write good assessment questions. Writing aligned questions against given LOs has multiple scenario of varying complexities. They are scenarios where a teacher has to write

- Single question against a given LO
- Multiple questions against a given LO
- Multiple questions against given set of multiple LOs.

Hence, literature on game design where there are multiple levels and each progressive level is more challenging and complex than the previous ones was searched. Elements of “gamification”—the use of game-like mechanisms applied to teaching and learning environment increases motivation or engagement (McClarty et. al., 2012). Another feature of digital games is that they inherently force the player to master a concept in order to advance. Games should present players with challenges in order to maximize engagement (Kiili, 2005). This philosophy of game design can be applied to TTM to design multiple stages representing multiple scenarios. The functionalities such as dashboard and score board can be adopted to display players’ progress and motivate them to perform better and retain their interest till the end (Shaffer et al., 2009).

Every type of tutoring system requires a feedback mechanism that signals a gap between a current level of performance and some desired level of performance or goal (Song & Keller, 2008). Resolving this gap can motivate higher levels of effort. Feedback is usually presented as information to a learner in response to some action on the learner’s part. It comes in a variety of types (e.g., verification of response accuracy, explanation of the correct answer, hints, worked examples) and can be administered at various times during the learning process. Such type of feedback is called as formative feedback (Shute, V. J., 2008). It can effectively reduce the cognitive load of a learner, especially novice or struggling learners (Paas, Renkl, & Sweller, 2003).

2.6 Automated Generation of AI

A good quality AI has proper blend of items (questions) guided by various parameters such as cognitive level, difficulty level, type of item, distribution of marks across the question paper, etc. A system to automatically generate AI should have a mechanism to take teachers request of assessment and AI specification and then select the questions from a question repository. The literature survey concentrated on following aspects; existing AI Generation systems, elements needed to specify requirements of the desirable characteristics of AI, tags that are required with a question to facilitate the automatic generation of AI.

2.6.1 Existing AI Generation Systems

There are many systems that automatically generate questions for the AI. They use either NLP, machine learning and item models to generate multiple choice or fill in the blanks questions from the text corpora. Some of them are listed here. Goto et. al., proposed a system to generate multiple cloze questions from text using NLP techniques (Goto, T., et.al. 2010). ArikIturri is an automatic question generator based on corpora and NLP that constructs fill-in-the-blank, word formation, multiple choice and error correction type of questions (Aldabe et.al, 2006). Deane, P., & Sheehan use frame semantics for Natural Language Generation of Math Word Problems (Deane, P., & Sheehan, K., 2003). Gierl et.al. proposed item models to automatically generate multiple choice questions where the content specialists identify and structure the content of test (Gierl, et.al., 2012). Michael in his paper provides a complete coverage of e-assessment module of Moodle that supports 20 question types (O'Rourke, 2011). The other attributes related to questions like content or cognitive level is put by the user if needed. In AIGen, we want an automatic AI generator where selection of questions is driven by AI specification provided by the teachers.

2.6.2 Assessment Instrument Specification

There are some instrument specifications described in the literature. They are usually called as blueprint, Table of Specification (TOS) or test grid in the literature. For example, the blueprint is defined as a three-dimensional chart which shows the placement of each question with respect to objective and the content area that it tests (Assessment Handbook, University of Ulster, 2012). The TOS suggested by Ahmed and co-authors determines the format and design of test items and consists of individual weightage of the test items based on content and Bloom's level (Ahmed et. al., 2013). A plan designed for a specific assessment is given in operational terms in the form of blueprint which consists of proportion of items against content, item type and cognitive level (C. Kan Kan, 2010). Central Board of Secondary Education (CBSE) of India provides similarly structured Blueprint for all the subjects up to 12^{th} standard (CBSE web site, 2013) (CBSE Teachers' Manual, 2013). A step by step procedure of developing AIS (test Blueprint), the different parameters to be considered and the values of those parameters are explained in detail with an example (Anderson & Morgan, 2008). All these specifications provide only the properties associated with the questions that is to be put into AI. None of them defines the structural aspect of AI like the number of sections, subsections, total number of questions and sub-questions, optional questions, total marks and time duration.

In order to facilitate automatic creation of AI, we want our AIS to contain both structural and property information in it. We adopted four dimensions for the AIS; content or topic for the assessment, cognitive level defined by the revised Bloom's taxonomy, difficulty level and question type. For each dimension, there are predefined set of values. For example, cognitive level, there are six predefined values like Recall, Understand, Apply, and Analyze, Evaluate and Create.

2.6.3 Metadata Associated with a Question in a Question Repository (QR)

For automated generation of AI, a question repository is needed, so that system can interpret AIS and select relevant questions from it. For this the questions should be annotated with tags/ metadata that can match the search criteria.

The questions vary widely one from another in their characteristics and use (Gall, 1970) (Porter et al., 2011). The authors have described two procedures for describing an item's content. In fixed category scheme, the content is divided into topics, subject matter areas, or instructional objectives. In keywords based methods, the item can be associated with any number of user defined tags. According to Currier, in order to support efficient retrieval, questions must be described with appropriate metadata like standard IEEE LOM (S. Currier, 2007).

Many Learning Management System (LMS) provide the facility of tagging questions while forming questions for Quiz. For example, Moodle and Totara allows each question to be annotated with question type and category / topic to which it is associated (MOODLE, 2014), (Parchure, 2006)), (totaralms, 2014). Test and Surveys option in Blackboard LMS allows users to manually add metadata such as categories, topics, levels of difficulty, and keywords to each question (blackboard, 2014). In the Test and Quizzes Tool of Sakai LMS, user can tag each question with its question type that includes essay, multiple choice, fill in the blank etc. (sakai, 2014).

There are internationally accepted standards for question / item and repositories like IMS QTI. It provides commonly used question types such as multiple choice/response, true and false, image hot spot, fill in the blanks, select text, slide, drag object/target, order objects, match items and connect points (Smythe, & Roberts, 2000). Graesser & Person tagged questions with metadata such as question type (short answer questions, long answer questions), degree of difficulty (high, medium, low) and deep reasoning or knowledge deficit questions on the basis of Bloom's level. (Graesser & Person, 1994). They used these tags for classifying the questions. Marbach & Sokolove categorized students generated questions into six categories by tagging them with respect to the concepts involved in them from simple concepts to highest level of scientific research question. Denny and his co-authors used difficulty level (high, medium, or low) tags to operationalize the quality of questions generated (Denny et.al., 2009).

Varieties of questions are needed for different types of assessments and instructional strategies. Even though existing question bank management systems provide the facility of associating user defined tags to a question, they are insufficient and are to be manually put by the user. Hence, they are not suited to the needs for AIGen. This motivated us to go for a semi-automated tagging system. From the literature surveyed and commonly used set of tags recommended by teachers, we presently focus on four sets of tags namely, cognitive level, difficulty level, question type and content / topic.

2.7 Summary of Literature Surveyed

From the extensive literature survey done, it has been observed that assessment is a vast and interesting area of research. Figure 2.1 gives the overview of areas of literature survey done

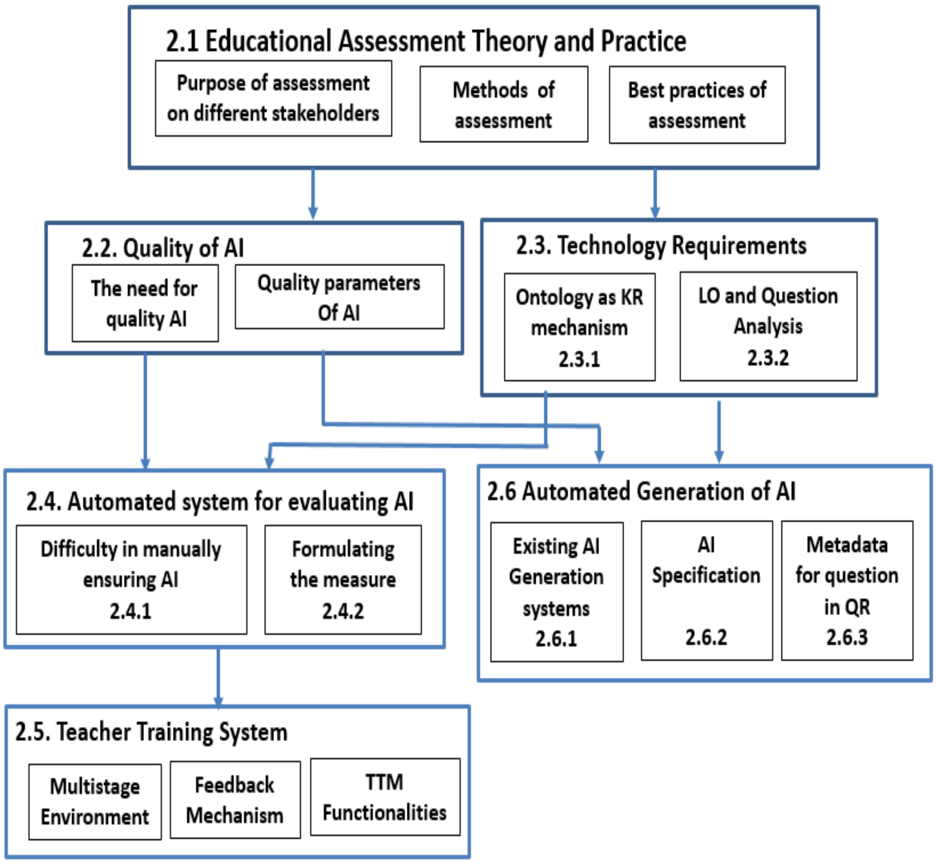


Figure 2.1 Different areas of literature surveyed

Initially, the literature survey was done in different aspects of educational assessment and quality of AI. This helped in exploring the existing work done in this area and establish the validity of research objectives introduced in chapter 1 of thesis. Then the literature survey was extended to include areas that finds specific solutions to the research objectives. The major findings from the synthesis of the literature surveyed is explained in chapter 5: need and context analysis. Even though the researchers have worked on various dimensions of assessment instrument and the process of assessment itself, to the best of our knowledge, very little or no work is done in designing the framework for generation and evaluation of AI.

Chapter 3 discusses the overall solution approach followed in this thesis to address the research objective of improving the quality of AI formulated in chapter 1.

Chapter 3

Solution Approach

This chapter explains the approach followed to address the research objectives formulated in Section 1.3. The research work is focused on improving the quality of AI. From the literature surveyed (Section 2.2), it was found that most important requirement of an AI is that it should be aligned to LOs of the course (Biggs, J., 2003). Hence, in our work, quality of AI is taken as the measure of its alignment against the LOs of the course. As per Krathwohl, every LO incorporates a set of topics/concepts and the level of competency expected to be achieved by students in those concepts (Krathwohl & Anderson, 2002). Cognitive levels defined in Bloom's taxonomy are widely used to represent different levels of competencies in LO. The process of measuring the alignment that determines the quality of AI can be broken down into two components such as the measure of content alignment and cognitive level alignment.

As described earlier, there are two ways to improve the quality of AI, namely

- Evaluating the quality of AI generated by teacher and providing feedback

- Automatically generating AI so that the quality is ensured at the time of creation itself.

The Overall solution approach is shown in Fig. 3.1. These two approaches are discussed in the following sections.

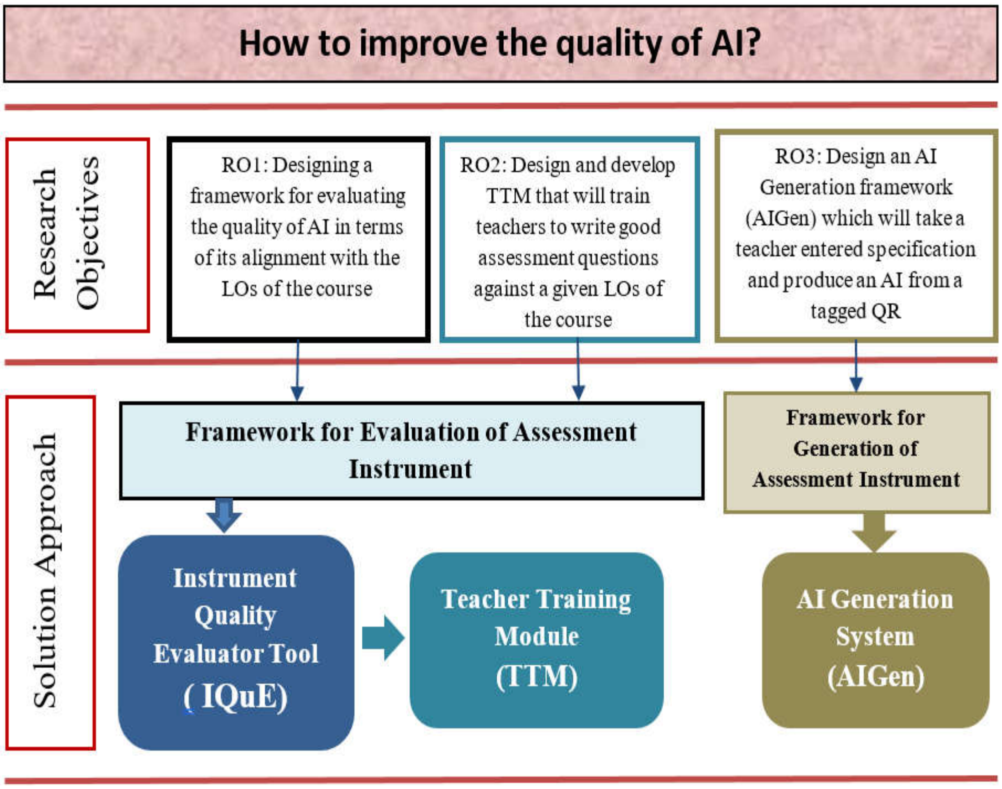


Figure 3.1. Solution Approach

3.1 Framework for AI Quality Evaluation

The design and development of a framework for AI quality evaluation is a solution to first research objective which is "How to design a framework for evaluating its quality in terms of its alignment with the LOs of a course?" The proposed approach and the overall process of AI quality evaluation is shown in Fig. 3.2.

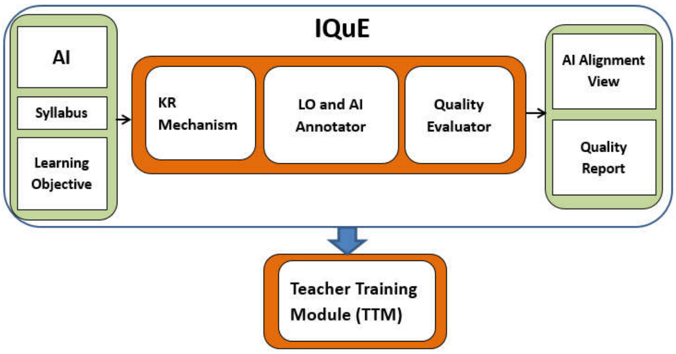


Figure 3.2 Framework for AI Quality Evaluation

In order to design a framework to evaluate the quality of AI, we need a mechanism that captures the expectation from a course. Course expectations are typically available in prescribed *syllabus and the set of LOs*. Syllabus captures typically the list of concepts, but generally, they do not characterize the cognitive level. LOs characterize the cognitive level, but they may not necessarily cover every topic in syllabus. Hence, ideally a combination of both is needed.

As the framework intends to automate the evaluation task, it needs a *KR* which is in a machine parsable form that will integrate the relevant knowledge from syllabus and LOs.

LOs and questions from AI are assumed to be framed in natural language (English). The content and cognitive level information has to be extracted from them and mapped to the KR mechanism. This is done by the *LO annotator and the AI annotator*. Thus, we get the integrated KR mechanism where the information from syllabus, LOs and AI are overlaid. The extraction and mapping process forms the major component of the framework. For automatic extraction, we chose to apply basic NLP techniques.

The next step was to formulate the measure of alignment from the integrated KR mechanism. This is done by the *Quality Evaluator*. The amount of similarity in the content and cognitive level information from LO and AI is considered to calculate the measure of alignment. An information theoretic approach was used to derive the formula.

Finally, we explored the possible ways for the representation of output of alignment. Teachers can easily and quickly analyse AI if they are provided with a visual representation of alignment. The textual result with a numeric measure of alignment will help them in deeper analysis. Hence, we finalized on both visual and textual output to show the extent of alignment between an AI and a corresponding set of LOs. The visual output is called as *AI Alignment View* and textual output is called as the *Quality Report*.

The framework is operationalized as IQuE tool. Detailed working of IQuE is explained in chapter 6 and 7.

Teachers are the targeted users of IQuE. They will input their AI to IQuE and test its quality. Based on the report generated they will revise the questions in AI and then recheck the quality. This cycle may be repeated till the quality is improved to the satisfaction of the teacher. It has been reported that they have difficulty in writing aligned assessment questions corresponding to a given LO. IQuE can also be used to make teachers better AI designers. This resulted in the design of *Teacher Training Module (TTM)* that can be used to train the teachers to write good assessment questions against a given LO/LOs. Once the teachers are trained, the AI they design will be almost always acceptable to IQuE and the revision cycles may take less time.

TTM follows formative feedback mechanism and has three stages. Teacher goes through all the stages one by one sequentially. In stage 1, teacher has to write a single question against a given LO. In stage 2, the teacher has to write multiple questions against a given LO and in stage 3 he has to write multiple questions against a set of LOs. The IQuE checks the alignment of the teacher written questions against the displayed LOs and gives corrective or advisory feedback. Depending on the nature of the feedback, the teacher is told to modify the question or move to the next LO in the same stage. Detailed working of TTM is explained in chapter 7.

3.2 Framework for AI Generation

One of the ways the quality of AI can be improved is by automatic generation of AI. As per researchers, a good quality AI has proper blend of items (questions) guided by various parameters such as cognitive level, difficulty level, type of item, distribution of marks across the question paper, etc. as per the requirement of the teacher. The proposed approach and the overall process of automatic AI generation is shown in Fig. 3.3.

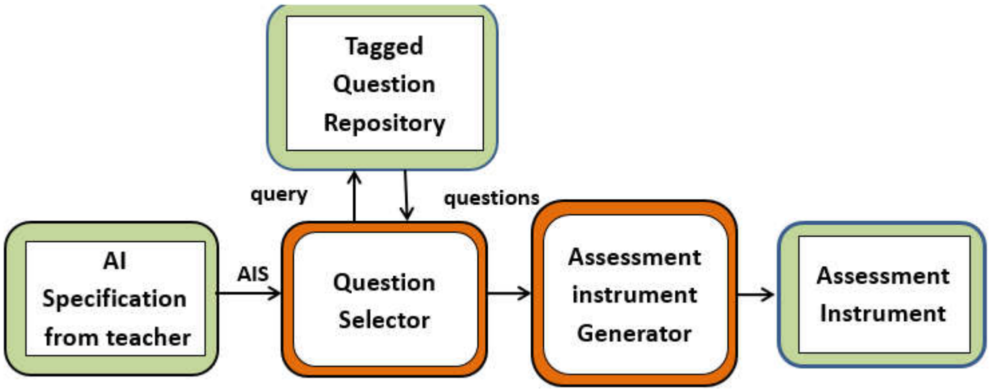


Figure 3.3 AI Generation Framework

A system to automatically generate the AI should have a mechanism to take teachers request of AIS. The AIS includes specification of structural as well as the property information of the AI. Structure of AI incorporates information about nested structure covering sections, questions, sub questions and the choice of options among questions or sub questions for the candidates. Property information includes learning objective of a question, proportion of questions in different cognitive levels, proportion of questions in different difficulty level, proportion of question types, distribution of questions among topics assigned for the assessment, marks associated with a question, etc. All these properties can be associated with section, questions or sub questions in an AI. Detailed description of components and structure of AIS is available in Section 9.1 of chapter 9.

The AIGen also needs a mechanism to read AIS and select questions from a repository to be placed into AI. In order to select, we require questions that are tagged with the values same as that of parameters in AIS. Manually tagging questions is a cumbersome and laborious task. Often, existing question repositories do not have tags for everything defined in the AIS. Hence, we have built a semi-automatic question tagging system which can suggest appropriate tags for each question and then store in the repository. Detailed working of creating a semantically tagged question repository is explained in Section 9.2 of chapter 9.

With a well tagged question repository and AIS, it is theoretically possible to generate an AI. The question selector of the framework interprets the AIS and select the most appropriate questions from the repository to be put into AI.

As a proof of concept of AI generation framework, we have built a AIGen tool that takes a subset of AIS and uses a tagged question repository and generates an AI. We call this as AIGen version 1. This has to be expanded in future to cover the full AIS.

3.3 Summary

Evaluation of quality of AI and automatic generation of AI are the two ways in which the quality of AI can be improved. Framework for AI quality evaluation is operationalized as IQuE tool and framework for AI generation is operationalized as AIGen tool. Major focus of our work is in the design and implementation of IQuE. In the framework for AI generation, a lot of effort has been put in characterization of AIS. The AIGen system is currently considering only subset of AIS. The quality of AI produced by the automated AIGen system depends on the AIS entered by the teacher as well as the availability of enough number of questions. Hence, it cannot in general guaranteed to be totally aligned to the LOs defined for the course. To check and ensure the alignment, it can be fed to IQuE. The feedback provided by the IQuE can then be used to either modify the AIS and generate a new version of AI or manually make the corrections in the AI itself. Chapter 4 discusses the specific research questions and research methodology adopted for our work.

Chapter 4

Research Methodology

Chapter 2 of the thesis reviewed literature related to the research problem area and presented broad research questions related to improving the quality of AI. Chapter 3 proposed two solution approaches to achieve this research objective in the form of designing a framework for (i) Evaluation of quality of AI and (ii) Generation of AI. It also presented the rationale and details of determining the components of both the frameworks.

Keeping in mind the broad research objective, the appropriate research design and overall research methodology was adopted. The broad research objective is "How to improve the quality of AI?". This is further broken down into following three research objectives as defined in chapter 1.

1) Designing a framework for evaluating the quality of AI in terms of its alignment with the LOs of the course.
2) Design and develop a training module that will train teachers to write good assessment questions against a given LO.
3) Design an AI Generation framework which will take a teacher entered specification and produce an AI from a tagged item repository.

The implementation of solution approaches resulted in the development of IQuE, TTM and AIGen tools.

4.1 Research Questions

In the process of planning the implementation of the solution approaches, we have formulated the following research questions. The questions are categorized into different classes depending on which aspect of the study it caters to. Some are positioning questions which help problem identification and need analysis and others are design questions that are pertaining to design and implementation of the systems. There are also types of questions that need to be answered during the formative and summative evaluation of the developed systems. Table 4.1 lists all the questions in each of the three categories.

4.2 Why Design and Development (DDR) for this Work?

The development of IQuE, TTM and AIGen requires iterative cycles of requirement analysis, prototype building, formative and summative evaluations. Design and Development Research (DDR) methodology was found to be the most suitable research methodology for my work (Richey & Klein, 2005).

4.3 What is Design and Development Research (DDR) Methodology?

Seels & Richey define DDR as "the systematic study of designing, developing and evaluating instructional programs, processes and products that must meet the criteria of internal consistency and effectiveness" (Seels & Richey, 1994). There are two categories of developmental research, referred to as Type 1 and Type 2 (Richey & Klein, 2005). They vary in the extent to which the conclusions resulting from the research are generalizable or contextually specific.

Type 1 developmental studies focus upon a given instructional product, program, process, or tool. Typically, they address not only product design and development, but evaluation as well. At times, they may validate a particular design or development technique or tool. Type 1 refers to an approach in which the roles of designer and researcher coincides within a specific developmental context. Type 2 studies, on the other hand, focus upon a given design, development, or evaluation model or process. They may involve constructing and validating unique design models and processes, as well as identifying those conditions that facilitate their successful use.

Table 4.1Research Questions

Position	1) What role does the quality of AI play in achieving the intended outcome by learners? 2) What are the different parameters that determine the quality of AI? 3) What are the different problems usually found with the AIs of various universities in India? 4) What are the difficulties faced by the teachers in manually evaluating the quality of AI? 5) How can technology help in solving research problem of improving the quality of AI?
Design	<u>IQuE</u> 6) How to annotate the nodes of the domain ontology with the content and cognitive level information from LO and questions in AI? 7) How to devise a measure for alignment score of AI with the LOs of the course in terms of its content and cognitive level? 8) How to visually represent the alignment of AI with the LOs of a course? 9) What constraints should an ontology meet inorder to facilitate development of IQuE?
	<u>TTM</u> 10) What is an appropriate pedagogy model for online training of teachers about using IQuE? 11) How to design and develop training environment including suitable feedback mechanism?
	<u>AIGen</u> 12) What are the elements needed to specify requirements of the desirable characteristics of AI? 13) What tags are required to be associated with a question to facilitate AIGen? 14) How to design and develop a tool that automatically generates an AI from a teacher's specification using a question repository?
Evaluation	16) How usable is IQuE as perceived by the users? 17) How useful is IQuE as perceived by the users? 18) What is the users' perception about the effectiveness of feedback mechanism of TTM? 19) What is the accuracy of AIGen?

Type 1 studies may have an analysis phase, design phase, development phase, and a try out and evaluation phase. It would include phases directed toward first analysis, then prototype development and testing, and finally prototype revision and retesting as shown in

Fig. 4.1. DDR is cyclical in character: analysis, design, evaluation and revision activities are iterated until an appropriate balance between the intended and realization has been achieved.

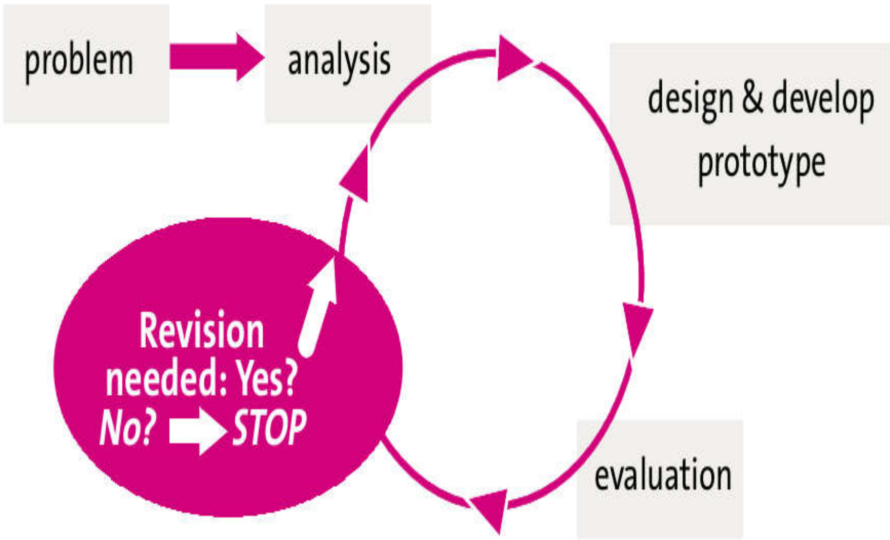


Figure 4.1 Iterations of systematic design cycles (Plomp, 2007)

In case of DDR, three phases; are distinguished

- Preliminary research which incorporates needs and context analysis, review of literature, development of a conceptual or theoretical framework for the study
- Development or prototyping phase: It is an iterative design phase consisting of iterations, each being a micro-cycle of research with formative evaluation as the most important research activity aimed at improving and refining the intervention
- Assessment phase: Summative evaluation to conclude whether the solution or intervention meets the pre-determined specifications. As also this phase often results in recommendations for improvement of the intervention.

McKenney illustrates this cyclical process to show a practical application of DDR to a research study undertaken by him to explore the potential of a computer to support curriculum materials development. He depicts each phase with number of loops. The loops within each phase indicate number of cycles in each phase where height and width of each cycle indicates number of participants included and time taken to complete the study done in that cycle (McKenney, 2001).

4.4 Application of DDR Methodology for the Current Work

Type 1 DDR method is suitable for the current research work as it includes the development of IQuE, TTM and AIGen tools and its formative and summative evaluation. Each of them includes phases directed towards first analysis, then prototype development and testing, and finally prototype revision and retesting. The analysis, design, evaluation and revision activities are iterated until the desired accuracy is reached.

We have adopted the McKenney model of DDR method with three distinct phases (McKenney, 2001):

1) **Need and context analysis phase:** In this, synthesis of literature reviewed from various sources was done to reinforce the need and establish the context of AI Generation and Evaluation Frameworks as well the corresponding tool such as IQuE, TTM and AIGen.
2) **Design and development phase:** It includes the iterative cycles involved in the development of AI quality evaluation framework and the corresponding tool (IQuE and TTM) and AI generation framework and the corresponding prototype model (AIGen).
3) **Summative Evaluation phase:** In this phase, all the products are evaluated for their accuracy, usability and usefulness

The research design containing details of sample, procedure and data analysis technique is shown in Table 4.2.

The interaction and dependency between different phases and research cycles in each of the phases is shown in the research design in Fig. 4.2. The number of cycles, iterations in each cycle and the mechanism for formative evaluation changes depending on the complexity of the system. Following sub sections briefly explain each of the phases.

Table 4.2 Phases of DDR for the current research work

Phases	Research Focus		Method of evaluation	Evaluation Instrument	Sample
Need and Context Analysis	Research cycle 1: Literature Review and need Analysis for IQuE, TTM and AIGen	1.Understanding the need for good quality AI	Synthesis of Literature Survey	-----	Research papers and Assessment Handbooks Experience Report from University Teachers Analysis of past years AIs
		2.Understanding teacher difficulty in manually generating and evaluating the quality of AI			
		3.Identifying appropriate Knowledge Representation Mechanism			
a) Design and Development of (IQuE)	Research cycle 2: Development of IQuE	Stage1: Creation of Domain Ontology	Qualitative	Rubrics	Domain ontology for DS course created by experts
		Stage2: Generation of LO and AI Annotated Ontology (LIAO)	Quantitative N=1000	Confusion Matrix	Pairs of an LO and questions from DS courses taken from different universities (Set 1)
		Stage3: Devising the AI Alignment Score			
	Research cycle 3: Refinement of IQuE:	Refinement in stage 1, stage 2 and stage 3	Quantitative N=1000	Confusion Matrix	Pairs of an LO and questions from DS courses taken from different universities (Set 1)
Summative Evaluation (IQuE)	Research cycle 4: Saturation of IQuE:	Accuracy of IQuE	Quantitative N=1000	Confusion Matrix	Pairs of an LO and questions from DS courses taken from different universities (Set 2)
		Usefulness and Usability of IQuE	Quantitative N=10	SUS survey	User responses to Likert's scale and open ended questions of survey
b) Design and Development of TTM	Research cycle 5: Development of TTM		Qualitative N=2	Content Analysis	Users (teachers) comments on the feedback generated by the system
Summative Evaluation (TTM)	Research cycle 6 Usefulness of TTM		Qualitative N=10	Content Analysis	Users (teachers) comments on the feedback generated by the system
c) Design and Development of (AI Generator)	Research cycle 7: Development of prototype model of AIGen	Accuracy of AIGen	Qualitative N=20	Rubrics	System generated AIs against a user given specification
		Usefulness and Usability of AIGen	Qualitative N = 5	SUS survey	User responses to Likert's scale and open ended questions of survey (Brooke, 1996).

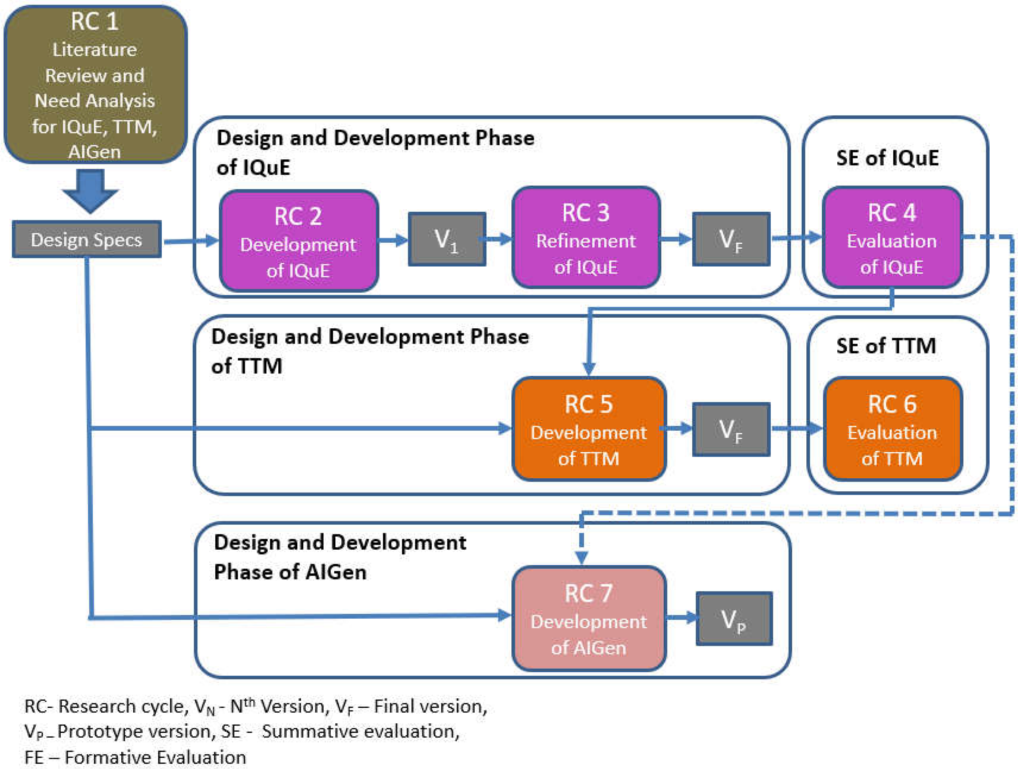


Figure 4.2 Research Design Model

4.4.1 Research phase 1: Need and Context Analysis

In this phase, the research problem is defined by first investigating the need and context analysis of the research problem. The first five research questions from Table 4.2 are pertaining to this phase. Finding answers to these questions not only created interest in the research project, but also helped in knowing the extent to which our notions of relevance are congruent with the perceptions and needs of practitioners (teachers). Extensive literature survey was done and it was found that evaluating the quality of AI is the most important concern of teachers where they spend a lot of time and effort in manually ensuring it. Automating this task would help teachers to a great extent.

After finalizing on the relevant research topic, the next step was to focus on particular aspect of design and development, identifying the technologies needed to develop and the evaluation procedures to be adopted. Hence, it was decided to focus on the alignment of AI to the LOs of a course as the quality measure in the framework for AI quality evaluation. Similarly, the decision of developing a teacher training module (TTM) also emerged during the need and context analysis as it was found that teachers face difficulty in writing aligned questions against a given set of LOs. Further the design challenges of framework for automatic generation of AI were identified.

Hence, outcome of this phase is the design specification to design and develop IQuE, TTM and AIGen. The scope and limitations of these systems were also identified. The broad literature survey is explained in chapter 2 of the thesis and the need and context analysis phase for each of the research objectives which forms the research cycle 1 of DDR is described in chapter 5.

Participants

Identifying the type of participants involved in each phase of the research work is also a major activity. There are often multiple types of participants in a given developmental research project, and if the study is conducted in phases, the participants may vary among phases (Richey & Klein, 2005). For our research work following participants were identified:

- Developers – build the systems (the researcher, interns hired for part development of the systems
- Evaluators – formative and summative evaluators (engineering college teachers who have taught *Data Structures* and educational technology experts)
- Organizations – where clients and users belongs (engineering colleges affiliated to Mumbai University)
- Learners and other types of users – end users of the system (all engineering college teachers)

4.4.2 Research phase 2: Design and Development

In this phase, the research procedures required to design and develop framework for AI quality evaluation and the corresponding systems IQuE and TTM and framework for AI generation and the corresponding system AIGen are determined. Most of the procedures are common to all the systems like the type of participants and research methodologies adopted in this phase.

Description of the components of the research model is explained in the respective sections below.

4.4.2.1 Design and Development of IQuE

The first prototype version of IQuE (V_1) is developed in the research cycle 2 of design and development phase based on the design specification obtained from cycle 1 as shown in Fig.4.2 and Table 4.1. The V_1 is subjected to rigorous formative evaluation using quantitative research method with N=1000 samples. The results of evaluation are then qualitatively investigated to find the scope for further improvement in the obtained accuracy. The issues were categorized as solvable and unsolvable by IQuE. Modifications were done in all the development stages of IQuE that led to research cycle 3 where accuracy of IQuE was refined. All the solvable issues were resolved and it was found in formative evaluation that the accuracy of IQuE increased significantly. We call this as final version V_F as the results got converged in two cycles for the given set of samples.

4.4.2.2 Design and Development of TTM

The first version of TTM (V_F) is developed in research cycle 5 as it uses all the components of the final version V_F of IQuE at the backend. The pilot testing with N = 2 was done during the formative evaluation of V_F using a qualitative research procedure which involved analysis of user comments on TTM feedback and interview responses. All the necessary changes suggested by users in the pilot study were incorporated into the V_F. The design and development of TTM involving research cycles 5 and 6 is explained in chapter 7 of thesis.

4.4.2.3 Design and Development of AIGen

The prototype version of AIGen is developed in research cycle 7 of DDR. AIGen uses some of the components of IQuE such as tagging engine and domain ontology. The system validity as well as usability and usefulness was done during summative evaluation using qualitative research methods. The design and development of AIGen involving research cycle 7 is explained in chapter 8 of thesis.

4.4.3 Research phase 3: Summative Evaluation

IQuE and TTM were subjected to summative evaluation phase as they were the final versions of the products. We have only the prototype version of AIGen and based on the results of the formative evaluation, much improvements need to be done in that. So currently AIGen is not subjected to any summative evaluation.

To establish consistency in the performance of IQuE, it was tested with a different set of N =1000 samples in the research cycle 4 of summative evaluation phase. The accuracy remained almost same with no further solvable issues detected. The IQuE was also tested for its usability and usefulness.

The summative evaluation of TTM was done in research cycle 6 of DDR. TTM was given to (N=10) instructors for actual testing. The methodology followed was same as in formative evaluation but these were only CS teachers and not ET experts.

The summative evaluation of IQuE and TTM are described in Chapter 7 and 8 respectively.

4. 5 Summary

In this chapter, the research objectives were further narrowed down into a set of 19 research questions. These questions are then classified into positioning, designing and evaluation questions based on what area of research work it is addressing. DDR was found to be the most suitable research methodology for my work. We have adopted the McKenney model of DDR with three distinct phases; Need and context analysis phase, Design and development phase and Summative Evaluation phase. The detailed explanation of need and context analysis phase which forms research cycle 1 of DDR is given in chapter 5. The details of descriptions of these phases are explained in chapters 6, 7, 8 and 9 respectively. Chapters 6 and 7 describe phase 2 (research cycles 2, 3) and phase 3 (research cycle 4) of IQuE. Each of the chapters 8 and 9 contain the description of complete tool TTM and AIGen involving both design and development phase as well as evaluation phase. The complete life cycle of IQuE tool involving research cycles 2, 3 and 4 is explained in chapter 6 of thesis. Similarly, TTM and AIGen are explained in chapter 7 and 8 respectively.

Chapter 5

Need and Context Analysis

This chapter focuses on the need and context analysis of IQuE, TTM and AIGen tool. This form the research cycle 1 of need and context analysis in DDR methodology. The need analysis focuses on position type of research questions which is shown in Table 4.1 a part of which is reproduced here as Table 5.1

Table 5.1. Position type of research questions

Position	1) What role does a quality of AI play in achieving the intended outcome by learners? 2) What are the different parameters that determine the quality of AI? 3) What are the different problems usually found with the AIs of various universities in India? 4) What are the difficulties faced by the teachers in manually evaluating the quality of AI? 5) How can technology help in solving research problem of improving the quality of AI?

The answers to questions 1, 2, 4 and 5 are derived from the literature surveyed which included assessment handbooks, research papers, experience report from university teachers, etc. The literature surveyed is already discussed in chapter 2. Synthesis of the literature survey that informs the need of the research study is explained in Section 5.1. In order to answer question 3, we analysed previous year AIs of various courses of Mumbai university and this is described in Section 5.2. Section 5.3 discusses how technology can be used to solve this problem.

5.1 Synthesis of the Literature Surveyed

From the extensive literature survey done, it has been observed that assessment is a vast and interesting area of research. It should be fit-for-purpose. Assessment is known to have a profound influence on what students study, how they study, how much they study and how effectively they study. *The quality of any educational assessment exercise depends on the quality of the instruments used*.

There are various quality parameters of AI such as proper proportions of cognitive level, difficulty level, type of question, distribution of questions among the different sub topics allotted for assessment, distribution of marks across the instrument, time allotted to students to solve the questions in AI, ambiguity in the wording of questions etc. From the literature, we found that the most important quality measure of a good AI is its validity. AI is valid if it measures the competencies that are expected to be achieved by the students after they complete the course. The expected competencies are provided by the LOs of the course. Hence, the validity is achieved when the AI is aligned to the objectives of the course. We are focusing on this aspect of quality. *The quality of AI is formulated as the measure of the alignment of AI with the LOs of the course*.

Teachers have to spend lot of time and effort in manually ensuring the quality. *This establishes the need for an automated mechanism to evaluate the quality of AI.* Hence, we focused our research work on building an automated tool to evaluate the quality of AI (IQuE).

We also found from the literature that teachers face a lot of difficulty in writing aligned assessment questions against a given set of LOs. Hence, using IQuE, we proposed to build a teacher training module (TTM) that the teachers can use to get trained, so that eventually they become better AI designers.

The other way to generate good quality AI is to have an automated system to generate a good quality AI against teacher specification. An attempt was made in this direction that led to design specification of AIGen. This prompted the requirements to create a structure for AI specification in order to take teachers request of assessment and build a semantically tagged question repository.

5.2 Confirming Findings from the Literature

The literature states that AIs designed for examinations in engineering education in many of the Universities in India are of poor quality (Phatak, 2015). There are many colleges affiliated to one university and examination is a centralized process. University conducts the examinations on behalf of all the colleges. Individual teacher has no control over it. University appoints the panel of examiners for each course (3-member panel in Mumbai University). They will design the AI for that respective course for all the colleges. AIs have fixed patterns with lots of internal options. The emphasis is on passing the examination and not excelling in that course. Questions are mostly descriptive and most of them cater only to lower order thinking skills.

In order to confirm the findings from the literature, we analysed the AIs of various courses in Mumbai University spanning over four years (2010 – 2013) to investigate the various shortcomings associated with them.

Sample:

We collected thirty AIs from Computer Engineering examinations of Mumbai University from year 2010 to 2013. Random sampling was used to take five AIs from each semester starting from semester 3 onwards. They were taken from different courses as shown in Table 5.2. All AIs were of 100 marks and 3-hour duration. There were total 7 questions each carrying 20 marks. Question 1 was a compulsory question. Out of remaining 6 questions, a student had to answer any 4 questions. There could be internal options within a question.

Data Collection

AIs of previous years for all semesters are easily obtained and freely downloadable from Mumbai University website or any of the college websites affiliated to the university. One sample AI from each course is given in Appendix D.

Table 5.2. Sample of AIs considered for analysis

Course Name	Semester	Number of AIs
Data Structures	III	5
Operating System	IV	5
Computer Networks	V	5
System Programming and Compiler Construction	VI	5
System Security	VII	5
Distributed Computing	VIII	5

Analysis

Each AI was thoroughly analysed by two experts for each course who evaluated its quality based on the following parameters.

1) Distribution of cognitive level among the questions
2) Difficulty level
3) Content coverage
4) Question Type

The experts were teachers who were domain experts and who have taught the course for multiple times. Both the experts analysed individually and later agreed to a common set of values for all the four parameters by debating and discussing. We have adopted revised Bloom's taxonomy for the cognitive level. The difficulty level of a question can be low, medium or high and is calculated based on the number of concepts involved in it, its cognitive level and the difficulty of the concept itself as decided by experts. If a question has more number of concepts then it is considered as more difficult than a question with less number of concepts (Marbach & Sokolove, 2000) (Denny, Reilly & Simon, 2009). A question with higher cognitive level is considered to be more difficult than the questions with lower cognitive level (Amer, 2006). Some concepts are more difficult than others in the syllabus. Questions can be of three types such as,

1) A problem to solve or write a program

2) Descriptive questions that contain words such as 'describe', 'explain', write a short note on', etc. and they usually belong to 'Understand' cognitive level

3) Objective type question can be short answer questions, MCQs, fill-in-the-blanks, match the following, etc.

Results

From the analysis, following observations were made

1) 90% of the questions were from the lower order thinking skills (Recall, Understand and Apply) except in one or two cases where they were from Analyse level. But, the percentage of such questions never crossed 20%.
2) There were hardly any questions from evaluate and create cognitive levels in AIs ranging from semester 3 to semester 8 even though the LOs of the course demand it.
3) The questions that belong to 'Apply' level were stereotypical questions such as "Write a program…", "Apply a method to solve a problem", etc. which often get repeated in the subsequent AIs.
4) Significant number of questions were descriptive questions that belong to 'understand' cognitive level.
5) All the objective questions are recall level questions.
6) At the broad chapter level, the content coverage is not of much concern. All the AIs have 80 to 90% content coverage. But, if the weightage of each chapter is considered, then the questions covered by each chapter are not proportional to it. In that perspective, there is an unfairness in the content coverage. As two questions, each of 20 marks are optional, the students have the option of omitting some part of the syllabus and can still score marks in the exam.
7) Most of the questions were of low and medium difficulty levels as they contained only one or two concepts.

The results obtained were in accordance with what is observed from the literature. Hence, poor quality of AIs found in most of the engineering examinations is a serious issue faced by most of the universities. Our work of building automated tools for improving the quality of AI is a solution towards this issue. It may help even novices to produce good quality AIs which will ensure proper assessments for the students.

5.3 Potential of Technology to Solve the Problem

As mentioned in chapters 1,2 and 3, there are two ways to address the issue of improving the quality of AI (i) Let teachers create AI first and then the system will do the automatic evaluation of its quality along different dimensions before it is given to students. The system will give feedback to the teacher about the quality of AI. Teachers can then revise the AI till it meets the expected standards. (ii) Generate the AI automatically, so that the quality is ensured at the time of creation itself. This research is focused on these two approaches.

For creating quality assessments, the main components are syllabus, LO and AI. Our focus is on alignment of questions with LOs with respect to concept and cognitive level. For all our systems, we needed a KR mechanism to represent the syllabus for which the AI is to be designed. Syllabus is not merely a set of concepts but it has hierarchical structure, dependencies and relationships among the concepts in it. The KR mechanism should capture this. Further, as a requirement to our systems, it should be able to integrate the concept and cognitive level information extracted from LOs and questions in it. From the literature, we found that ontology is one such mechanism and is widely used (Mizoguchi, 2004) (Mizoguchi & Bourdeau, 2015). The ontology contains all the concepts related to a particular domain and relationship between them. It is a semantically connected network of concepts and is well suited to represent syllabus. It is also machine parsable. The automated systems can utilize it to extract or embed information in it. Looking at all these functionalities, ontology was chosen as the KR mechanism for all the systems.

5.4 Selection of Domain

Even though assessment is crucial in every field of education, we carried out the research work in the field of engineering education. The major reason for this decision is the familiarity of the course and a first-hand experience in teaching and assessing engineering students. Recently, there has been lot of concern about the poor quality of AIs in the engineering examinations which has been discussed in the motivation section of chapter 1.

Further, *Data Structures* from computer science curriculum has been chosen as the domain for the research work. It is a foundation course and is a prerequisite for many other courses. The primary focus of *Data Structures* course is for students to acquire the ability to translate a problem statement into an algorithm and then translate that algorithm into working code. Apart from Computer Science students, this course serves students of several other engineering disciplines. It is not only a transition course from problem solving and basic programming skills to more advanced topics in computer science but also as the final programming course for many students (Chinn, Prins & Tenenberg, 2003). It is widely used in education research papers from CS.

5.5 Summary

This chapter established the need and context of the research problem of improving the quality of AI. This forms the research cycle 1 of need and context analysis phase of DDR methodology. The findings from the literature were confirmed from the analysis of previous year AIs of various courses of Mumbai University examinations. The potential of using a technology to solve this problem is also discussed. The output of this phase is a design specification to build IQuE, TTM and AIGen. Chapter 6 describes the design, development and implementation details of IQuE.

Chapter 6

Design, Development and Implementation of IQuE

The IQuE is the result of operationalizing the framework for AI quality evaluation. The framework is explained in Section 3.1 of chapter 3 in the thesis. Although the literature discusses many different qualities of AI, we have selected the alignment of AI with the LOs of the course for our work (Agarwal, 2008). IQuE is a software tool to measure alignment of AI with a set of LOs of a course (Rekha, Sasikumar & Iyer, 2016). The following subsections provide the detailed explanation of design and development of IQuE and its implementation. As explained in chapter 4, DDR methodology is followed in developing IQuE. *Data Structures* course of BE curriculum of Mumbai University is used as the illustrative domain for our work. This domain is used for all examples and illustrations in this chapter.

6.1 Design and Development of IQuE (Cycle 1 of DDR)

IQuE takes syllabus, LOs of the course and domain ontology as input. When an AI of the same course is given as input, it calculates the measure of its alignment (alignment score) against the LOs of the course. It also provides a visual representation of alignment. The design of IQuE was done in three stages as shown in Fig. 6.1.

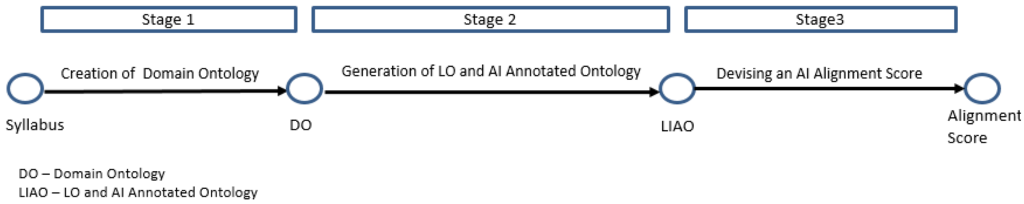


Figure 6.1. Development Stages of IQuE

6.1.1 Stage 1: Creation of Domain Ontology

The creation of domain ontology forms an important part of the development of IQuE. Ontology is used as a knowledge representation mechanism to represent the domain. It forms the foundation structure on which the knowledge extracted from LOs and questions is overlaid. Following sub-sections describe our notion of structure of ontology, its validation and the guidelines developed for transforming ontology to fit into the requirements of IQuE. Fig. 6.2 shows the overview of stage 1 processes.

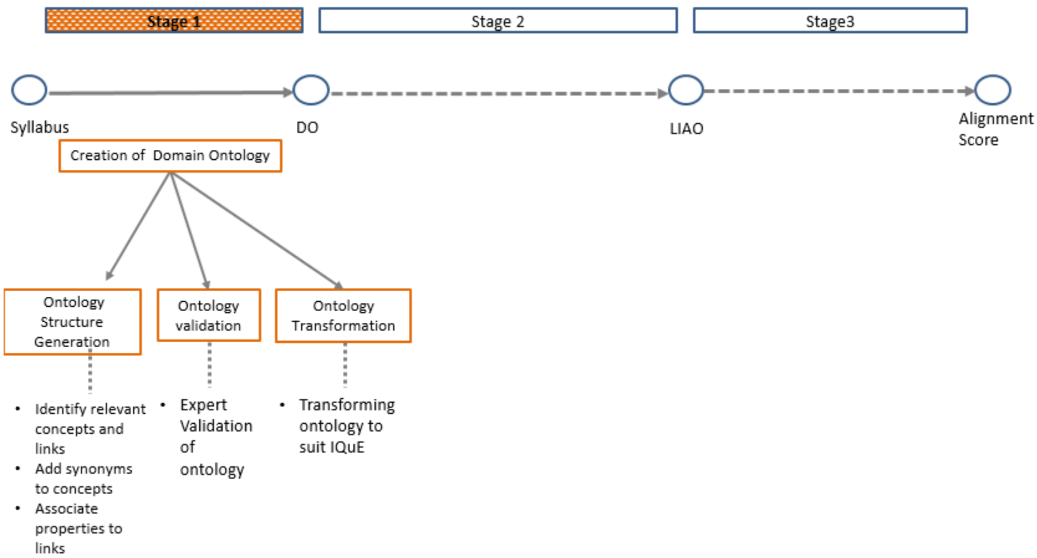


Figure 6.2 Overview of stage 1

6.1.1.1 Structure of Domain Ontology

The domain ontology captures a hierarchical structure of the domain and also dependencies among the various topics from the domain (Noy and Deborah, 2000). Fig. 6.3 shows part of domain ontology. The key elements of the ontology model adopted will now be explained.

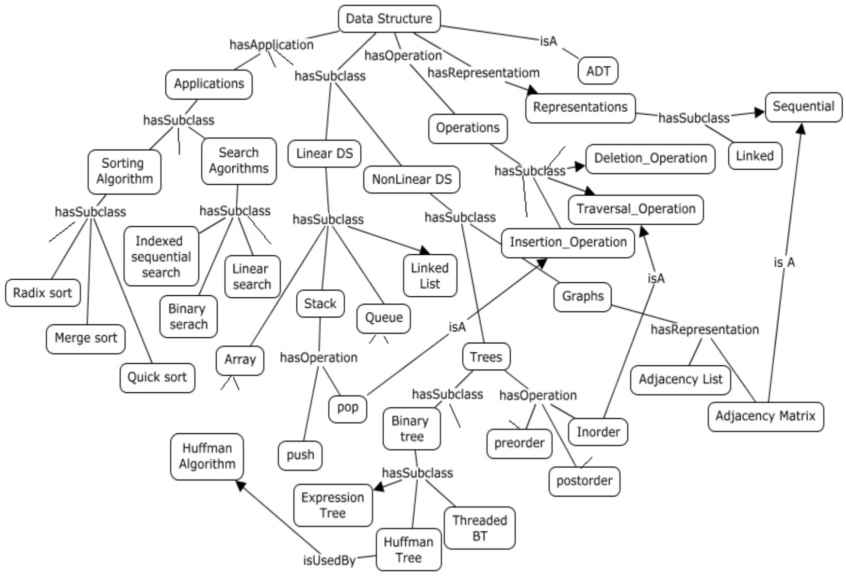


Figure 6.3. Part of Domain Ontology for *Data Structures* course

- **Concepts**: Every node in the ontology represents a concept/topic from the domain. The root node of the ontology tree is the name of the domain itself i.e. *Data structures*. All the major topics form the level 1 nodes in the ontology. The major topics can be further narrowed down to subtopics that form the subclasses in the ontology. The relationships include but are not restricted to hierarchy of concepts. The dependencies/ relationships between the concepts are shown using links. The concepts in the domain are finalized by compiling the contents of various standard textbooks and the *Data Structures* course contents of many different universities in India and abroad. The process of finalization of concepts involved many steps such as listing all the concepts from different resources, grouping all similar concepts and finding the hierarchy among them. It was possible to constrain all the concepts to a 5-level hierarchy (e.g. Data Structure → Non-Linear Data Structure →Trees →Binary Tree → Huffman Tree)

- **Links/Relations**: In a similar way, all possible relations among the concepts were listed. The relations with similar behaviours were grouped together and given a common name. These formed the links in ontology. All links are directed. In our ontology, we are assuming following links.

  - ‘hasSubClass’--- indicates one concept is a subclass of another concept
    e.g. Linear Data Structure *hasSubClass* Queue.

- “hasRepresentation” --- Representation of a data structure indicates how its elements are stored in memory. Elements are sequentially (contiguously) stored in memory in an Array. In linked list, contiguous memory is not allocated to store the elements, instead every element will contain the address of the next element in the list. Hence, they have linked representation. Some of the data structures like Graph can be represented in both ways; adjacency list which is a sequential representation and adjacency matrix which is a linked representation.

  e.g. Graph *hasRepresentation* ‘Adjacency List’ which is a *subClass* of Linked Representation.

- “hasOperation” ---There are many operations that can be performed on any data structure such as ‘insertion’, deletion’, ‘traversal’, etc. They form subclasses of the node ‘operations’ in ontology. The link ‘hasOperation’ connects the ‘data structures’ node to the node ‘operations’. Some of the data structures calls these operations by specific names. For example, ‘push’ and ‘pop’ are specific names given to insertion and deletion operations on stack. Ontology uses following hierarchy to depict this.
  e.g. Stack *hasOperation* Push and Push *isA* insertion operation.
- “hasApplication” --- Different kinds of data structures are suited to different kinds of applications in real world. They form subclasses of the node ‘Applications’ in ontology. The link *hasApplication* connects the ‘data structures’ node to the node ‘Applications’. The applications are specific to a data structures. This is depicted by having *hasApplication* link from the specific data structures.
  e.g. Binary tree *hasApplication* in Heap sort.
- “isA” --- One concept is a subclass of an another concept. Ideally, isA is an inverse relation to *hasSubclass*. But in our ontology mapping algorithm, the nodes connected by *hasSubclass* are traversed by default. To restrict this, they are kept separate.
  e.g. Every Data Structure *isA* ADT.
- “includes” --- One concept has many other parts included in it. In other words, to understand and implement one concept you need to understand other parts

e.g. Hash search *includes* Hash function, collision handling and collision resolution techniques.

The links are used to traverse the ontology to locate the neighbourhood nodes which are relevant in the ontology. The type of links decides what nodes are to be included for mapping. For example,

LO: Students should be able to demonstrate various operations on data structures

In this case, annotator will include all the nodes that indicate various ***data structures*** as well as all the nodes that are connected by ***hasOperation*** link from every data structure.

Some links can be considered as connecting a node to a set of properties. All its subclasses can inherit the properties of its super classes and can also override them by providing their own definitions of them. For example, various operations (***Insertion, deletion, display, traversal, and search***) can be considered as properties of ***Data Structures*** connected by ***hasOperation*** link. All the subclasses of it such as *stack*, queue, etc. can inherit those properties. Stack overrides insertion and deletion operation by providing its own definition of ***push*** and ***pop*** operation. Fig. 6.4 depicts this scenario.

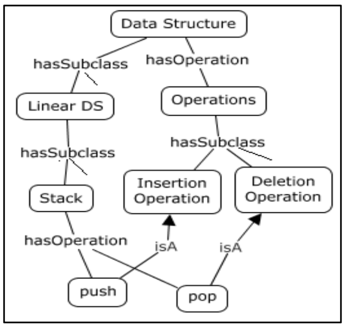


Figure 6.4 Property Overriding

The links can also have inverses. For example, in the statement "Heap sort uses binary tree" The *uses* link is an inverse link of 'hasApplication' as shown in Fig. 6.5. If any LO or question statement has word 'uses', then 'hasApplication' link will be detected automatically because of the inverse relation.

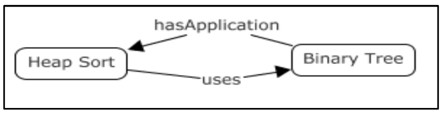


Figure 6.5 Inverse Links

There are many tools available for ontology development. Protégé is the most popularly used open source tool which is used for developing domain ontology for the IQuE system (Noy et. al., 2003).

- **Synonyms**

  Every node in the ontology is annotated with a set of alternate labels. Alternate labels form the synonyms or expanded forms of the common abbreviations that the teacher may choose to use while framing questions or LOs instead of the node name. For example, ***methods of traversing trees*** form the synonym for ***tree traversal operations*** and ***depth first search*** is an expanded form of ***DFS***.

6.1.1.2 Ontology Validation

The generated domain ontology was validated by 3 different experts. The experts were CS instructors who have taught *Data Structures* course multiple times and had more than 8 years of teaching experience in an engineering institution. They were given a syllabus of data structure course and were given the task of generating the ontology for it. Our ontology was compared with expert generated ontologies based on number of concepts, number of links and the type of links, the outgoing links from each concept and incoming links to each concept. There were no major differences and the minor ones were sorted out by discussing and then coming to the common consensus.

6.1.1.3 Transforming Ontology to Suit IQuE

Ontologies created by different authors follow different conventions depending upon the purpose for which they will be used. The model and conventions that we have adopted is explained in previous sections. The ontology of a course for IQuE represents syllabus contents and its structure. It should facilitate the mapping of identified concepts and relations from LOs and questions to the nodes and links in ontology.

For the system to accept the existing ontology, it needs to be fit into the predefined structure and satisfy certain properties. Otherwise, the given ontology needs to go through a transformation process so that it will be modified as per our requirement. Figure 6.6 represents the transformation process. The transformation process includes a set of simple guidelines to be followed so that only certain type of nodes and links are allowed. The first six rules are for nodes and the remaining are for the links. Rules are made as general as possible so that they can be adapted very easily to create domain ontology for any other course.

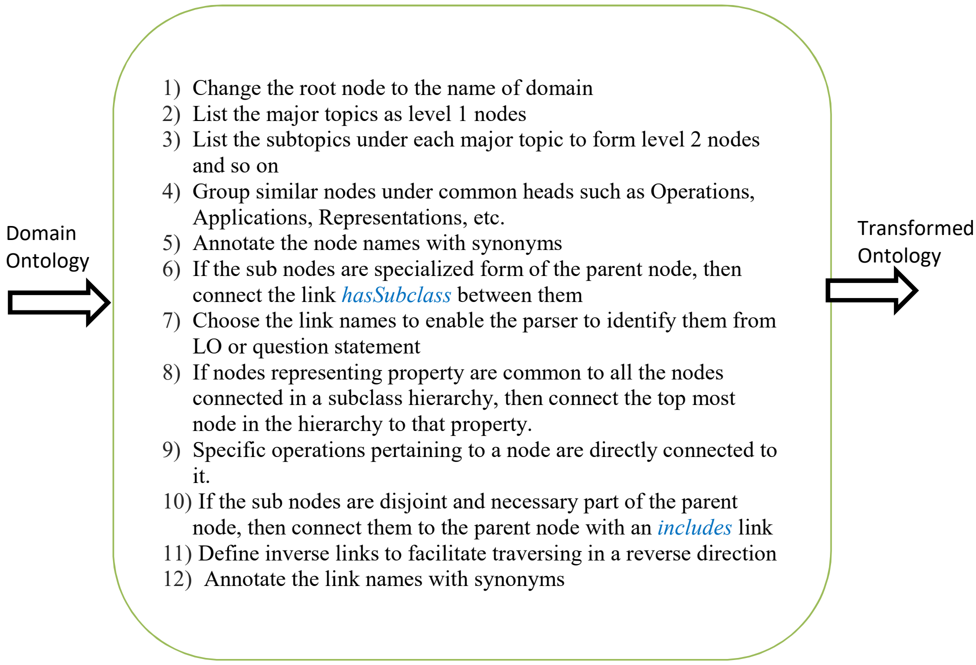


Figure 6.6 Guidelines for creating ontology

6.1.2 Stage 2: Generation of LO and Instrument Annotated Ontology

The stage 2 of IQuE involves the design of three major processing components: LO annotator, AI annotator and Generating AI alignment view as shown in Fig. 6.7. The input to stage 2 is domain ontology, a set of LOs of the course and an AI whose quality is to be evaluated. The information extracted from LOs and every question in AI are annotated to the nodes of the ontology and a graphical representation of LO and AI annotated ontology is generated as output which is called as AI alignment view.

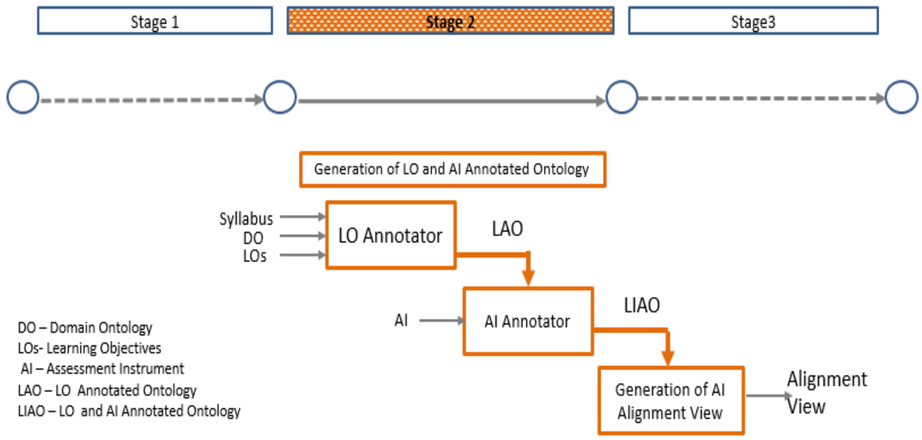


Figure 6.7. IQuE Stage 2: Generation of LO and Instrument Annotated Ontology

Inputs:

- **LOs:** The set of LOs are written in a simple text file *los.txt*. They are written one per line and there are no restrictions on the number of LOs. Any character other than text will be removed by IQuE at the time of preprocessing.
- **AI:** The AI contains a list of questions saved in *q.txt*. Like LOs, questions are also written one per line and there are no restrictions on the number of questions. The questions can be of any type such as short answer, long answer, essay type, MCQs etc. Any character other than text will be removed by IQuE at the time of preprocessing.
- **Ontology:** Ontology is created using Protégé tool (explained in Appendix A) and the generated OWL file is given as input to IQuE. Our notion of ontology is described in Section 6.1.1.

6.1.2.1 LO Annotator

The LO Annotated Ontology (LAO) is the output generated by the LO Annotator. The LO annotator takes the input from the syllabus and LOs and annotates the domain ontology as shown in Fig. 6.8 (Rekha, Sasikumar & Iyer, 2016).

LAO is an ontology based knowledge representation that incorporates the information from syllabus and a set of LOs into domain ontology. To create LAO, there is a need to specify the structure of domain ontology, syllabus and LOs. Our notion about the structure of domain ontology is already discussed in the stage 1 in the previous section.

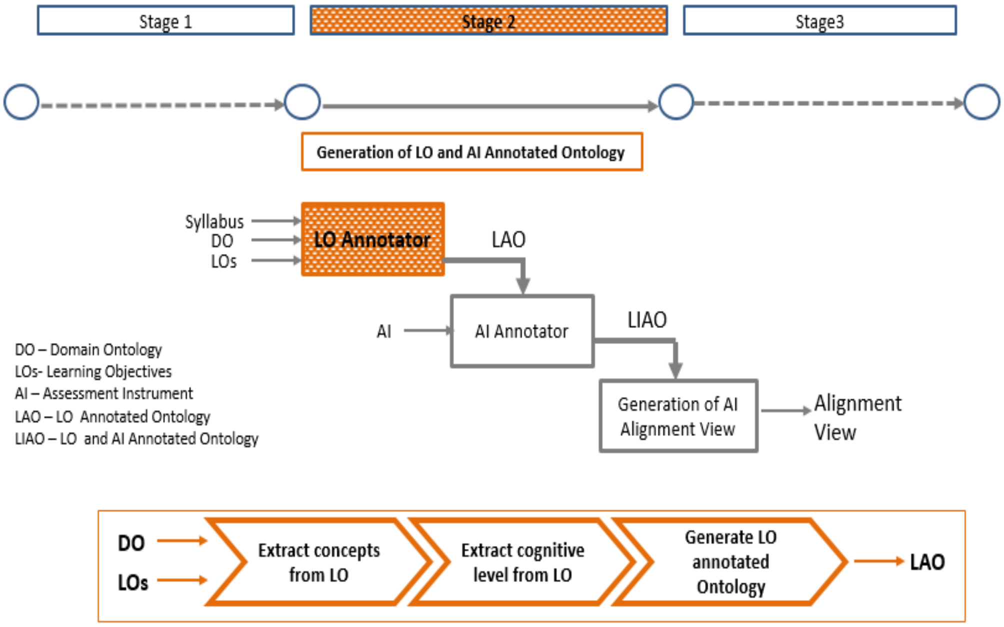


Figure 6.8 LO Annotator

Syllabus is considered as set of keywords which have a corresponding match to nodes of ontology. Fig. 6.9 shows how the domain ontology will look after mapping contents of syllabus to its nodes. The contents included in syllabus are shaded in grey.

Every course has a set of predefined LOs (l1, l2, l3…., ln) covering the entire syllabus. Every LO contains 2 attributes: a set of topics/concepts (c1, c2, …, cl) from the syllabus addressed by that LO and cognitive level defined by Bloom's Taxonomy. The key design element is how to automatically extract this information (concepts and cognitive level) from the LO text. This requires some NLP techniques. Initially, the LO statements are preprocessed using simple NLP techniques such as tokenization and Lemmatization (Manning et. al., 2014).

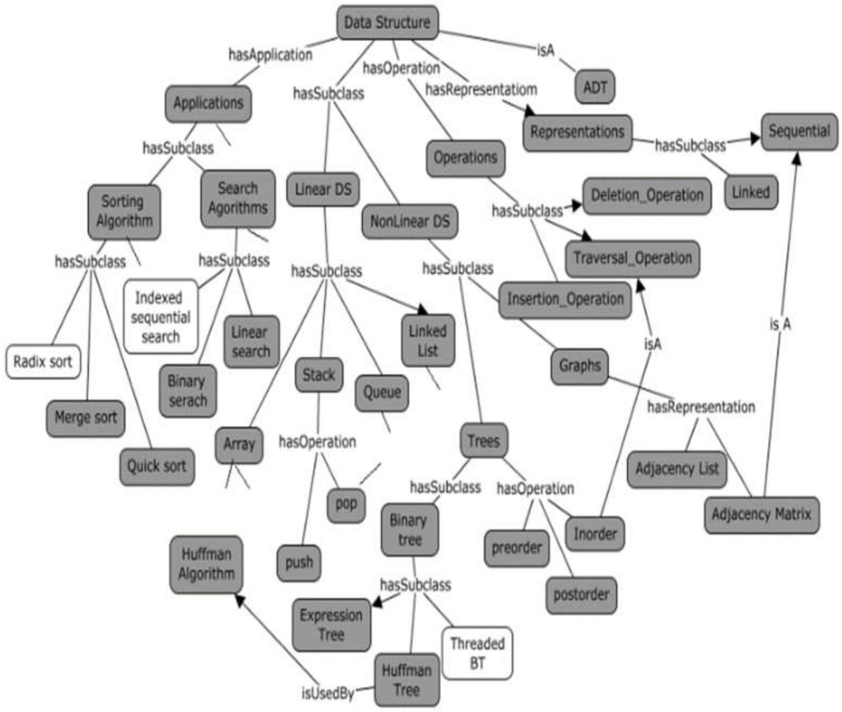


Figure 6.9 Part of domain ontology with syllabus contents mapped

The words or tokens are matched to the nodes/concepts of domain ontology. But the matching process is not direct. There are some issues and challenges in extracting concepts and cognitive level from LOs and mapping to the nodes of the ontology (Rekha, Sasikumar & Iyer, 2016). Following subsections explains the process of parsing the LOs to identify relevant concepts and cognitive level, mapping to nodes of ontology and traversing the links of ontology to mark related concepts.

6.1.2.1.1 Extracting Concepts from LOs

The issues which came up while processing the LOs to extract concepts from them and the solutions that were designed to solve them are described in detail below.

1) The concepts can be single or multiple worded. For example,

> *LO1: Students should be able to implement the Huffman coding algorithm using binary tree.*

Here, the concepts *Huffman coding algorithm* and *binary tree* are multi worded concepts.

How to identify a multi worded concept from an LO?

To extract multiple worded concept, N-grams algorithm can be used. It can find N-grams (sequences of 1 to N words) from LO which is exactly matching with the node names in the ontology. The algorithm starts with N = 1, find the single worded concept matching with node names in the ontology, then increment the value of N at each iteration. The maximum value of N is kept at N = 4 with assumption that there will not be a concept with N>4.

For example, in LO1, *Huffman coding algorithm* and *binary tree* can be found using N=3 and N=2.

2) Teachers can frame the LOs in many ways. Hence, concepts from the LOs may not exactly match with the node names in the ontology. For example,

> *LO2: Students should be able to demonstrate and implement different methods for traversing trees.*

In this case, the concepts ***methods of traversing trees*** will not directly map with the node name ***tree traversal operation*** in the ontology. Then, how to identify the concepts from LOs that are differently worded but are synonyms of the node names in the ontology?

To identify the concepts from LOs that are differently worded but are synonyms of the node names in the ontology, every node in the ontology can be annotated with a set of synonyms. This can solve the problem to some extent. Synonyms form possible alternative names that may be used in place of node names in the ontology while framing the LOs.

For example, in LO2, the concept ***methods of traversing trees*** will be an annotation to the node ***traversal operations*** in the ontology.

3) The concept that is identified may not be enough as it may be a higher-level concept in the ontology tree and the question in the AI may be from the subtree under that. Then, how to determine the alignment between the two is a major issue. For example,

> *LO3: Students should be able to implement various searching and sorting algorithms.*

Does this mean all the **searching and sorting algorithms** listed in the syllabus also need to be mapped? Suppose there is a question in the AI.

Q: Write a program to implement Merge sort. Show the steps to sort the following numbers. 99 22 88 66 40 10 34 52.

Is this question aligned with the above LO? Thus, apart from the concepts identified from LO, which are the other related nodes to be mapped in the ontology?

To locate other related nodes to be mapped in the ontology apart from the concepts identified from LO, we must explore neighborhood relations. For example, the concepts ***searching*** and ***sorting algorithms*** is an umbrella term used for all the searching and sorting techniques mentioned in the syllabus and which forms the nodes in the subtree below them in the domain ontology. Such nodes can be reached by traversing a ***hasSubclass*** relation from the explicitly found nodes. Hence, in this case, in addition to the explicitly found nodes ***searching algorithm*** and ***sorting algorithms***, we need to also map and color other related nodes in the ontology such as ***Linear search, binary search, index sequential search, Hash search, Shell Sort, Radix sort, Insertion Sort, Quick Sort, Merge Sort and Heap Sort***.

In such cases, the following decisions need to be taken by the system as it processes each LO

- Is there a need to find the related nodes?
- Which links to traverse in the ontology?
- What depth it should be traversed?

One of the ways to address the first issue can be to find hints from the LO text itself. After analysing many such LOs, it was found that there are some words typically found in these LOs such as **various, different, any, all, plural form** of a concept etc. If these words are found in LO, they indicate that the associated concepts act like slot variables. Then annotator can find all the valid concepts that can be substituted for these slot variables.

In the above example LO, the associated concepts to the word '*various', **searching algorithms** and **sorting algorithms*** will act like slot variables and all the searching and sorting algorithms will form valid concepts that can be substituted for them. Mostly, it suffices to traverse the ***hasSubClass*** links in the ontology from these variables until you reach leaf nodes in that subtree as shown in Fig. 6.10.

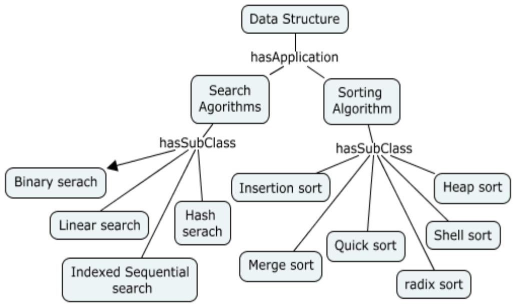


Figure 6.10 Part of domain ontology with syllabus contents mapped

4) Sometimes we need to identify the relation names from LOs that correspond to links in ontology to find the related nodes to the identified concepts to be mapped to node names in ontology. For example,

LO4: Students should be able to explain various operations on stack.

In this case, ***stack*** and ***operation*** are the only concepts that are explicitly identified from LO. There are other implicit concepts to be mapped to ontology that are connected through ***hasOperation*** link. How to identify relevant links from LO?

To identify the relation names from LOs that correspond to links in ontology, the link names are annotated with associated synonyms. Words or tokens from LO are matched with the link names and its synonyms. If there is a match, all the nodes connected to the identifying concept are considered for mapping by traversing the link. For example, in LO4, *stack* and *operation* are the concepts explicitly identified from LO. From the word '*operation'*, *hasOperation* link is identified. This link is traversed from *stack* and the nodes *push* and *pop* are included for mapping. The LO also includes a slot indicator *various* and it is associated to concept *operations*. So, the *hasSubClass* link from *operations* is traversed and the corresponding nodes such as *Insertion_Operation, Deletion_Operation, Search_Operation, Traversal_Operation and Display_Operation* are also included for mapping as shown in Fig.6.11. Now the *Insertion_Operation* is called *pus*h for stack which is indicated by *isA* link in the ontology. Similarly, the *Deletion_Operation* is called *pop* for stack which is indicated by *isA* link in the ontology. Hence, the words *push* and *pop* operations are retained along with all other operations except *Insertion_Operation* and *Deletion_Operation.* The process is shown in Fig. 6.11.

If the concept from LO do not have the identified link connected to it, then each of the super classes can be traversed to see whether they have the link. If they have, all the nodes connected to that link are considered for mapping by traversing that link.

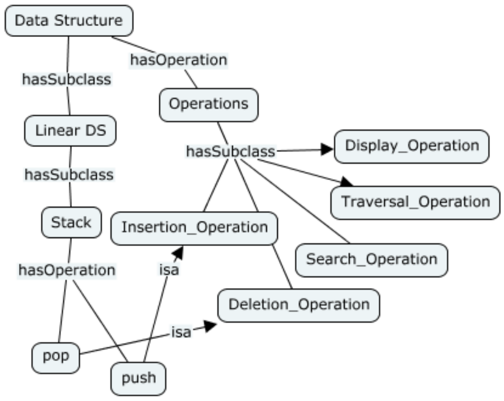


Figure 6.11 Identifying concepts by traversing link (hasOperation)

5) Sometimes explicitly identified concepts may be indirectly related to each other. They form isolated nodes when mapped to domain ontology. Isolated nodes are the nodes that are not connected to any other node in an LO or question statement. For example,

> *LO5: Students should be able to implement Huffman Coding Algorithm using Binary tree.*

In this case, ***Huffman Coding Algorithm*** *uses* ***Huffman tree*** *which is a* ***Binary tree****.* Do we need to color the intermediate nodes between isolated nodes?

The decision whether to color the intermediate nodes between the explicitly identified concepts depends on the distance among them. All the nodes are connected to each other at least through the root node ***Data Structure***. There may be multiple paths between two nodes. One of the solutions may be finding the shortest path and coloring all the nodes in it. The simplest assumption we have made is to consider only the 'one hop' connectivity. If there is only one intermediate node between two isolated nodes, then color that node otherwise keep it as it is. For example, In LO5, ***Huffman Algorithm*** and ***Binary Tree*** are isolated explicit nodes. They are connected by two paths, one longer path through the root node ***Data Structure*** and the other one hop path through the node ***Huffman Tree***. Thus, we color the node Huffman Tree in the ontology as shown in Fig. 6.12.

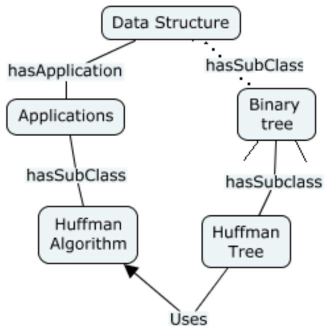


Figure 6.12 Isolated Explicit Nodes

Table 6.1 summarizes the issues identified and the proposed solutions that we have adopted in our system.

Table 6.1. Proposed solutions for challenges and issues in IQuE

Challenges and Issues	**Proposed Solution**
Identification of Multi worded concepts	N-Grams algorithm
Identification of differently worded concepts	Associating synonyms with the nodes in ontology
Finding related nodes to be mapped in the ontology other than the identified concepts	Introducing the concept of slot variable to decide whether to traverse the subclass hierarchy and include all the nodes in that.
Identification of differently worded links from LO	Links in ontology are associated with synonyms
Finding the nodes connected by links if the concept from LO do not have the identified link connected to it	Traverse the superclass hierarchy of concepts to find the connected link
Coloring the intermediate nodes between two explicit isolated nodes	Color the only intermediate node between them if there exists.

Functions: To formalize the process of expansion of explicitly identified concepts to get implicit nodes, we defined following functions.

Assumptions:

CL is set of concepts covered by LO and CQ is set of concepts covered by question. CQ[i] is any concept in CQ and CL[j] is any concept in CL. Both CQ[i] and CL[j] can be leaf node or an intermediate node in the ontology. If C is any concept either in LO or a question, then the following functions can be defined on it as shown in Table 6.2.

Table 6.2 Functions defined on concept 'c'

Function	Definition
syn (C)	C U all its synonyms as defined in ontology
expand (C)	C U all the implicit nodes as identified by the ontology traversal algorithm depending on the conditions present in the LO in which C is present.
syn_expand (C)	expand (C) U syn (C) for all i ϵ expand (C). It contains the synonym nodes and the nodes that we get after expanding every node from the syn (C).

The implicit concepts can be obtained

- By traversing the 'hasSubclass' link due to the identification of slot indicator or
- By traversing any other link explicitly identified in the statement or
- As connecting nodes between two isolated nodes.

Note: The design process of developing LO Annotator is spiral. Initially, some set of LOs were taken and processed and the concept mapper algorithm was devised. The output generated was then analyzed and accordingly the algorithm was revised to solve the issues involved in them. This process was repeated for another set of LOs till the algorithm stabilizes. But as the LOs are framed in natural language (here we assume it as English), there are some specific issues that the algorithm cannot handle because of the inherent ambiguity present in the language. For example,

> LO 6: *Students should be able to implement Hash search including various Collision handling and resolution techniques*

Here, the algorithm will identify only two concepts *Hash search* and *Collision handling*. The word collision is connected to both handling and resolution because of the 'and' connective. The algorithm is not able to find this.

Similarly, if there are multiple concepts after the slot indicator, then the algorithm associates the nearest one to it. But it may not be true always. For example, LO 7: *Students should be able to demonstrate and implement different tree traversing methods*

Here, the algorithm identifies *'different'* as the slot indicator and *tree* and *traversal operations* as concepts. Hence, the nearest one is *tree* and slot indicator gets associated to it. Thus, *tree* gets expanded to all its subclass hierarchy, however, the LO is intending different *traversal methods* such as *inorder, preorder and postorder*.

Such type of issues may be solved by using smarter NLP techniques. Before such LOs are given to the system, it goes through a preprocessing step where all such anomalies are resolved. For example, various *Collision handling and resolution techniques* will be transformed into *Collision handling and Collision resolution techniques.* Similarly, different tree traversing methods will be converted into different methods of traversing trees.

6.1.2.1.2 Extracting Cognitive Level from LOs

Revised Bloom's taxonomy forms the basis for cognitive level identification of an LO (Karthwohl & Anderson, 2002). Every level of Bloom's taxonomy namely, Recall, Understand, Apply, Analyze, Evaluate and Create is associated with an elaborate set of keywords. These keywords can be stored into a dictionary. The tokens are matched to the keywords in the dictionary and accordingly its cognitive level is identified. For example, consider the following set of LOs;

Students should be able to-

*LO8: Compare and contrast **stack** and **queue** data structure (Analyze)*

*LO9: Explain the working of **Expression Tree** (Understand)*

*LO10: Design a real-world application using **graph data structure** (Create)*

*LO11: Define **Abstract Data Type** (Recall)*

In LO1, LO3, LO6 and LO7, 'Implement' is a keyword associated with **B**loom's level Apply. Thus, the cognitive level is identified as 'Apply'. Similarly, the cognitive level of LO4 is 'Understand' as they contain keywords 'explain' and 'demonstrate. 'Compare', 'Explain', 'Design' and 'Define' are the keywords associated with Analyze, Understand, Create and Recall levels of Bloom's Taxonomy. Hence, all the concepts associated with

LO8-LO11 will be associated with ‘Analyze’, ‘Understand’, ‘Create’ and ‘Recall’ cognitive levels respectively.

If two tokens match with the keywords of two different Bloom’s level, then the higher level one is chosen as cognitive level of the complete LO. For example, in LO2, the keyword ‘Demonstrate’ is at ‘Understand’ level and ‘Implement’ is at ‘Apply’ level. Thus, the cognitive level of the LO2 is identified as ‘Apply’.

Once the cognitive level of LO is identified, all the concepts involved with this LO are annotated/ colour coded with this level. This means that the students are expected to achieve that level of competency in those concepts.

6.1.2.1.3 Generating LO Annotated Ontology

The annotator identifies the relevant concepts and the cognitive level associated with them from each of the LOs as described in previous sections. These concepts can then be mapped to the nodes of the domain ontology and color coded to generate LAO. Fig. 6.13 shows how the LAO will look after coloring all the relevant nodes.

The nodes whose corresponding concepts are part of syllabus are initially colored as black. The color code for a node has 2 parts: The left part and right part. The left part indicates concepts involvement in LO. **Black** indicates the concept is not covered by an LO. **Red** indicates covered at least by one LO. The right part indicates concepts involvement in question which is explained later. The nodes with red and black colors will indicate that the concept is within syllabus and is also addressed by an LO. Different shades of red color are used to indicate variations in cognitive level. Higher the level, darker is the shade.

The LAO provides the information about coverage of concepts in all the LOs and the expected cognitive levels from them while framing assessment questions aligned with them.

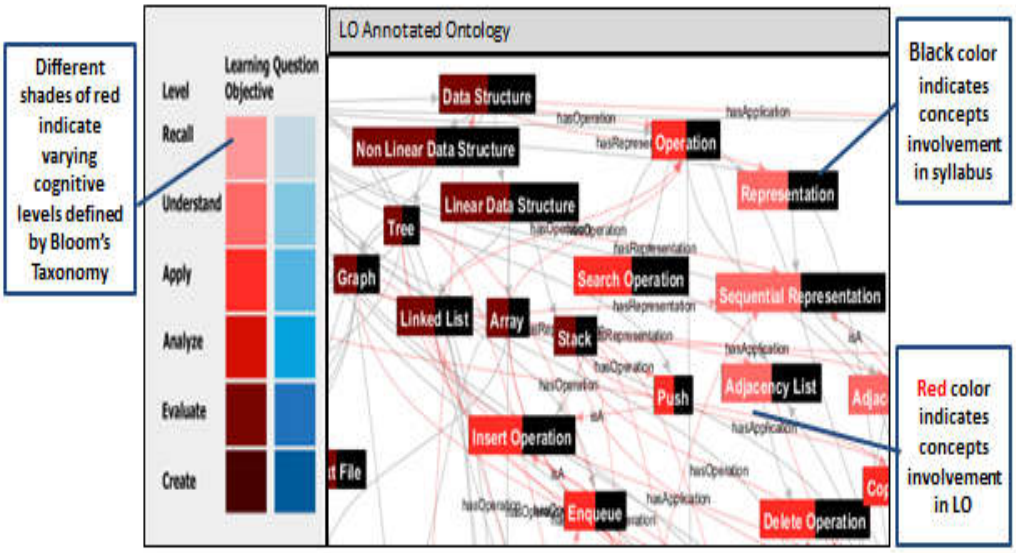


Figure 6.13 LO Annotated ontology with the example LOs mapped

6.1.2.2 AI Annotator

The AI annotator extracts information from an AI and incorporates into LAO. This becomes instrument annotated ontology (LIAO) (Rekha, Sasikumar & Iyer, 2016). The AI can be manually generated by a teacher or can be system generated with the help of teacher's specification and tagged question repository. It is assumed that AI contains a list of questions such as q_1, q_2, ……, $q_{n...}$ Other details from AI such as number of questions, number of sections and subsections, instructions about options, time duration, etc. are omitted as the current focus is on processing questions and extracting information from them. Hence, such details are non-significant for IQuE. Every q_i contains two attributes; a set of topics/concepts (c_1, c_2, c_3,…cl) from the syllabus addressed by that question and the level of competency required by students to answer that question. Following subsections describe the process of extracting concepts and cognitive level information from a question and the process of mapping these to the nodes of domain ontology. This process is shown in Fig 6.14.

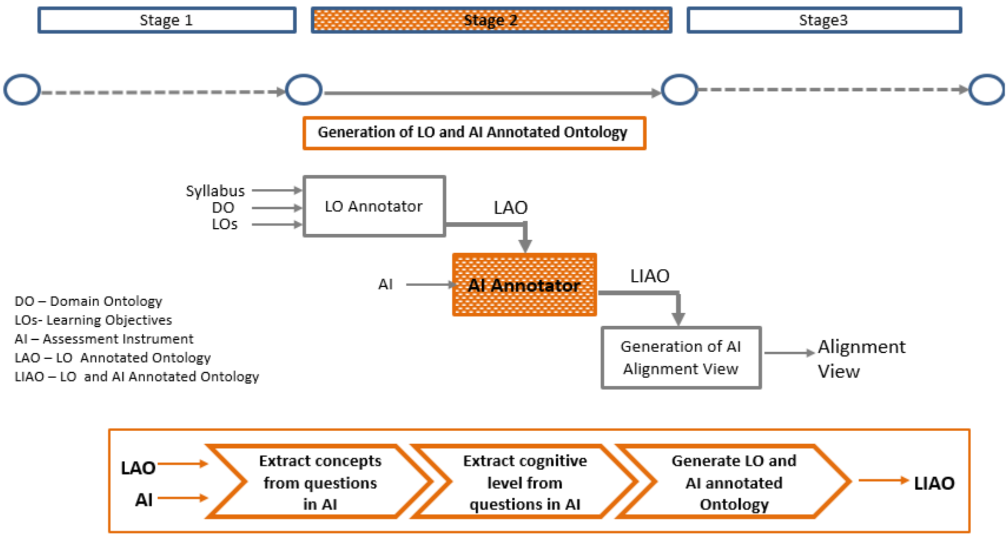


Figure 6.14 AI Annotator

6.1.2.2.1 Extraction of concepts from a question

The most challenging part of design of LIAO is extracting relevant concepts from a question that can be mapped to nodes of LAO. The LOs are limited in terms of their numbers and there is less variety as far as a course is concerned. But the questions that can be generated by teachers have no limits and they can be much more creative depending on the individual teacher. Thus, identifying concepts from them is a difficult task. We collected questions from past question papers of Mumbai and various other universities covering varied concepts and cognitive levels.

All the issues, challenges and their solutions in extracting concepts and cognitive levels from LOs described in Section 6.1 and summarized in Table 6.1 are equally applicable here. But there are additional challenges such as teachers may frame a question on the application of some concepts in data structure without even using the name of concept. For example,

Q: Imagine you have a web-site which serves files to thousands of users. It cannot service all requests and can only handle 100 at a time. It adopts a fair policy of serving 100 at a time in order of arrival. Select the most appropriate data structure to simulate such a system.

Here, the question is on the application of ***queue*** data structure but nowhere the word ***queue*** is used. Hence there are no explicit concepts available to map to the ontology. In such situation, how will the annotator find the associated concepts from a question?

There can be many ways to handle this such as using more sophisticated NLP techniques, using templates to find semantically close concepts from a question or manually annotating the question. Currently, this is done manually by modifying such questions before giving it to the system for parsing. Thus, there is a question preprocessing step where the teacher can edit and reframe the question to submit again.

6.1.2.2.2 Extraction of cognitive Level from a question

The first step in finding the cognitive level of a question is same as what is followed in case of LOs. Every token from a question is matched to the keywords associated with the Bloom's level stored in the dictionary and accordingly its cognitive level is identified. If the tokens/words in a question match with more than one action verbs at different levels of Bloom's taxonomy, then the higher one is chosen as the cognitive level of the question.

Apart from the available action words in each level of the Bloom's Taxonomy, some domain specific action verbs are added into the dictionary such as 'write a program', 'devise an algorithm', 'show a stepwise execution' in 'Apply', 'Evaluate' and 'understand' etc. For example, consider the following question.

> *Q: Write a 'C' program to form a circular linked list and display the contents in reverse order.*

Here, 'Write a program' is a keyword associated with 'Apply' and display is a keyword associated with 'understand'. Hence, the cognitive level of the question is 'Apply'.

Sometimes, the words in a question are misleading. For example,

> *Q: Describe how the position of pivot affects the performance of merge sort algorithm.*

In this case, 'describe' is a keyword associated with 'understand'. But along with the word 'how' it goes to a higher Bloom's level 'analyze'. We call such phrases as question indicators.

Finding the cognitive level of the question is much more challenging. The domain specific action verbs can be added by the domain experts at the beginning. But there can be minor variations in questions. For example, “write a program” can also be written in different ways in a question such as “write a ‘C program”, “write a program in C”, “Write programs that”, “write a recursive program”, “write a recursive routine”, “write a recursive function in ‘C’”, etc. Similarly, question indicators can be much more creative. Hence, more difficult. To some extent this can be solved by providing question templates. The AI annotator extracts phrases instead of action verbs from a question and then match with question templates. Variations are handled by partial matching.

6.1.2.3 Generating AI Alignment View

The instrument annotator identifies the relevant concepts and the cognitive level associated with each of the questions as described in the previous sections. These concepts can then be mapped to the nodes of the domain ontology and color coded to generate LO & IAO also called as AI Alignment view. Fig. 6.15 shows how the IAO will look after coloring all the relevant nodes. As discussed earlier, the color code for a node has 2 parts: The left part and right part. The right part indicates concepts involvement in question. The **black** here means that the concept is not covered by any question and **blue** indicates covered at least by one question. Different shades of **blue** indicate different cognitive levels defined by Bloom's taxonomy. Higher is the level, darker is the shade.

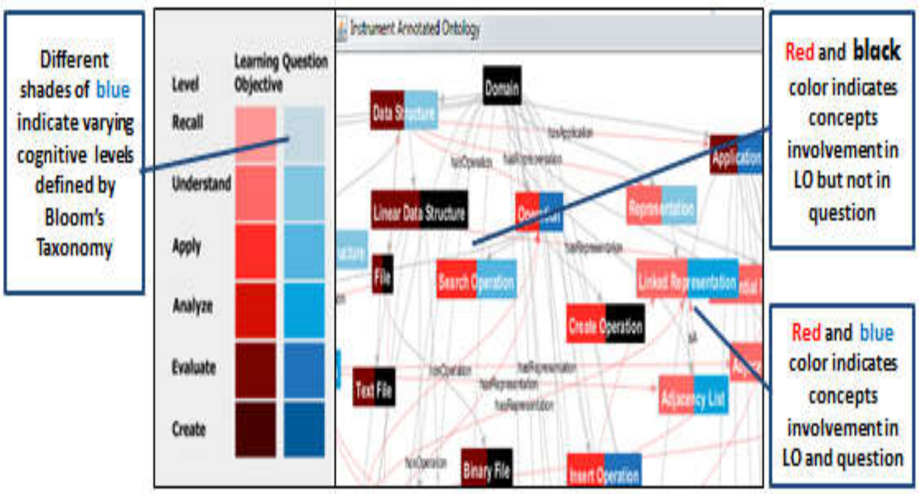


Figure 6.15 AI Alignment View

For every node in the AI alignment view which represents the concept in the domain, following information is associated with it such as

- Is the concept from the syllabus?
- Is it covered by any LO? If yes, what is the cognitive level expected from an assessment question that can be framed on that concept?
- Is there any assessment question that is asked on it? If yes, what is cognitive level of the asked question?

Based on this information, the following statistics can be gathered.

- Number of concepts within or out of syllabus
- Number of concepts within syllabus but there are no LOs covering them
- Number of concepts within syllabus but there are LOs covering them
- Number of concepts within or outside syllabus but no questions are asked on them
- Number of concepts within or outside syllabus but questions are asked on them
- Number of concepts covered by LOs at each cognitive level
- Number of concepts covered by questions at each cognitive level

From these statistics, the quality of AI can be evaluated. The difference in levels generated by LO and question on a given concept will amount to misalignment. These can be aggregated to measure the overall alignment between a set of LOs and AI. This is explained in the following section.

6.1.3 Stage 3: Devising an AI Alignment Score

This section explains the process of computing the measure of alignment of AI with a set of LOs of a course. It starts with calculating the alignment score between a question and LO and then extends it to a set of questions in AI and a set of Los as shown in Fig. 6.16.

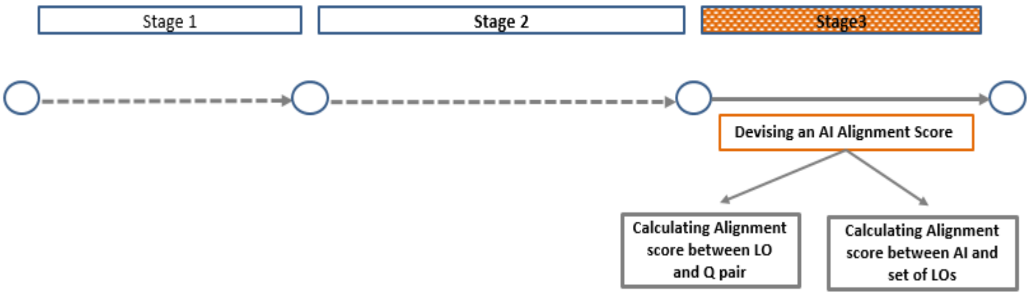


Figure 6.16. Overview of the processes involved in stage 3

6.1.3.1 Alignment score between question and LO pair

For an LO-question pair, the IQuE extracts the set of concepts and cognitive level covered by them and calculates the measure of content and cognitive level alignment between them.

a) Measure of content alignment

Let CL is the set of concepts covered by LO and CQ is the set of concepts covered by a question. The alignment of a question with an LO content wise, depends on the commonalities and differences in the number of concepts between them. To measure the content alignment, an information theoretic definition of similarity which is frequently used in information retrieval systems has been adopted (Xia & Yihong, 2006) (Lin, D., 1998). It can be computed using the formula:

$$Alignment\ (CQ, CL) = \frac{|CQ \cap CL|}{|CQ \cap CL| + \propto |(CQ|CL)| + \beta |(CL|CQ)|} - - - - - - - - - - - - - - -(1)$$

Where

$|CQ \cap CL||$ is the number of concepts common to both LO and question

$|(CQ|CL)|$ is the number of concepts only in question and not in LO

$|(CL|CQ)|$ is the number of concepts only in LO and not in question

α and β are constants in the range [0,1]. For now, we assume α=1 and β=1.

As per this formula, the measure of content Alignment (CQ, CL) lies in the range of [0, 1] with value 0 indicating no alignment, the value 1 indicating total alignment and others indicate partial alignment i.e. there is some amount of overlapping of concepts from the domain of interest.

Calculating$|CQ \cap CL|$, $|(CQ|CL)|$ and $|(CL|CQ)|$

The calculation of $|CQ \cap CL|$, $|(CQ|CL)|$ and $|(CQ|CL)|$ are not simple set operations as the concepts in CL and CQ may be at different hierarchy levels in the ontology. The set of concepts from both CQ and LO are compared and matched to get $|CQ \cap CL|$, $|(CQ|CL)|$ and $|(CQ|CL)|$. But the matching process is not straightforward. The LO usually covers a broad range of concepts from the syllabus. It may also include a concept and all the subclass concepts in its subtree. The question may be aligned to it even if it includes any one of the concepts in the subtree.

For example,

LO: Students should be able to implement various ***sorting algorithms***
Q: Write a typical recursive implementation of ***Quick Sort****. The implementation uses last element as pivot.*
Here, CL = {sorting algorithms}

CQ = {Quick sort}

The question is perfectly aligned to LO and the alignment value should be 1. But the formula, if applied directly, Alignment (CQ, CL) will give a value 0. To remove such anomalies, the sets CQ and CL need to be matched intelligently using the following process. The process has three cases:

Assume CQ[i] is any concept in CQ and CL[j] is any concept in CL. Both CQ[i] and CL[j] can be leaf nodes or intermediate nodes in the ontology. If C is any concept either in LO or a question, then following functions are defined on it. The functions syn (C), expand (C) and syn_expand (C) are defined in Table 6.2

The CQ[i] matches with CL[j] if-

Case1: CQ[i] ⋳ syn (CL[j]).

Example,

*LO: Students should be able to implement the **Huffman coding algorithm** using binary tree.*

*Q: What is Huffman Coding? Apply the **Huffman coding algorithm** and determine the* code *for the following characters whose frequencies are given.*

Case 2: CQ[i] ⋳ syn_expand (CL[j]).

Example,

*LO: Students should be able to understand the working of various **multi-way search tree structures**.*

*Q: Construct a **B tree** of order 5 for the following data*

Case 3: CQ[i] is a leaf node, CQ[i] is connected to CL[j] with an "isA" link.

Example,

*LO: Students should be able to demonstrate and implement different **methods for traversing trees***

*Q: Write a program to construct binary tree for the following **preorder** and **inorder traversal** sequences*

Every time CQ[i] matches with CL[j], increment the count of (CL∩CQ) and tag both in the respective sets. The count of untagged elements of CQ gives (CQ|CL) and CL gives (CL|CQ). Substitute in the formula to get measure of alignment.

b) Measure of cognitive level

For every pair of an LO and a question, IQuE finds its cognitive level. The cognitive levels of LO and questions are matched to get the cognitive level alignment matrix. The measure of cognitive level alignment is decided based on the following rules.

Rule 1: If there is an exact match in the cognitive level, then the value is 100 indicating a 100% alignment.

Example,

*LO: **Demonstrate** and **implement** different methods for traversing trees.*

*Q: **Write a 'C' program** to implement circular queue using array and perform insert and delete operation.*

Here, the LO and question are at 'Apply' level. Hence, the value is 100.

Rule 2: If there is a difference of +/- 1 in the cognitive level of LO and question, where plus (+) sign indicates LO is at higher level than question, then the value is 50.

Example,

*LO: **Understand** the working of various multi-way search tree structures.*

*Q: **Write** and explain **algorithm** to **delete** node from AVL tree*

Here, LO is at 'Understand' level and question is at 'Apply' level. So, there is a difference of 1 level and the value is at 50.

Rule 3: If there is a difference of more than +/- 1 in the cognitive level of LO and question, then the value is 0.

Example,

*LO: **Analyze** a given problem and **select** the appropriate data structures required to solve the problem.*

*Q: **Explain** the representation of polynomial using linked list with an example*

Here, LO is at 'Evaluate' level and question is at 'Understand' level. Hence, there is a difference of more than 1 level and the alignment value is at 0.

Ideally, there should be 2 classes; perfectly aligned with a value of 100 and not aligned with a value of 0. However, in the revised Bloom's Taxonomy, there is no strict boundary among the categories and they overlap one another (Krathwohl, & Anderson, 2002). To accommodate this fact, rule 2 is introduced which says that, if the cognitive levels of the question and LO are at adjacent levels in the taxonomy, then the measure of cognitive level alignment should be 50, indicating partial acceptance.

6.1.3.2 Calculating Alignment Score between Question Set (AI) and LO Set

The content alignment score is calculated for every question of AI paired with every LO in the set of LOs. This is substituted in Content Alignment Matrix (CAM). The values from CAM are aggregated to generate a single score called as Content Alignment Score (CAS). Similarly, the cognitive alignment score is calculated for every question of AI paired with every LO in the set of LOs. This is put in cognitive Level Alignment Matrix (LAM) and the corresponding single score cognitive Level Alignment Score (LAS). Overview of the processes involved in calculating the alignment score between AI and LOs set is shown in Fig. 6.17.

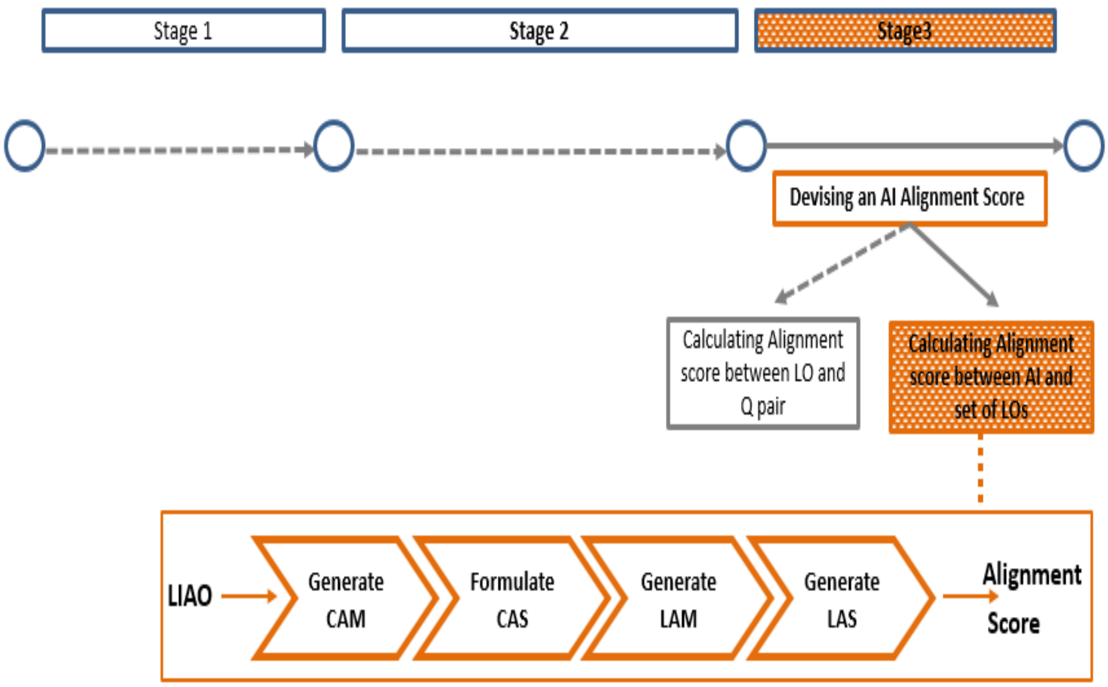


Figure 6.17. Overview of the processes involved in calculating the alignment score between AI and LOs set

Consider the following set of LOs and questions in AI as shown in Table 6.3.

Table 6.3: Sample AI and LO Set

LO set	**Questions in AI**
Students should be able to------ LO1: Implement various operations on Stacks LO2: Implement various operations on Queues LO3: Model stack as ADT LO4: Model queue as ADT LO5: Compare the performances of stacks and Queues LO6: Write programs that use stacks LO7: Write programs that use Queues	Q1: Write a program to implement circular queue. Q2: Define Abstract Data Type. Write an ADT for queue Q3: Write a program to implement circular queue. The following operation should be performed by the program (i) Creating the queue (ii) Inserting into queue (iii) Deleting from the queue (iv) Displaying all the elements of the queue. Q4: List any four applications of Queue and explain any one Q5: What is priority queue? Give an application for priority queue Q6: Distinguish between stack and queue

6.1.3.2.1 Generate the Content Alignment Matrix (CAM)

A part of CAM is generated is shown in Table 6.4, where rows are LOs and columns are questions and

$$CAM[i, j] = alignment(CL_i, CQ_j)$$

$$i \in L \text{ and } j \in Q$$

where L is set of LOs and Q is set of questions

Table 6.4 Content alignment matrix

LO \ Q	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**
LO 1	0	0	0.33	0	0	0.33
LO 2	0.5	0.33	1	0.33	0.33	0.33
LO 3	0	0.33	0	0	0	0.33
LO 4	0.5	1	0.33	0.33	0.33	0.33
LO 5	0.5	0.33	0.33	0.33	0.33	1
LO 6	0	0	0	0	0	0.5
LO 7	1	0.5	0.5	0.5	0.5	0.5

6.1.3.2.2 Formulate the CAS

We are interested in the overall alignment of the complete instrument with the LOs of the course. CAM gives individual alignment values which inform a single question is aligned to multiple LOs and multiple questions are aligned to a single LO. A question in AI may be contributing fully to an LO (with a score of 1) or partially (with a score < 1) and nothing to an LO (with a score of 0). The CAM does not tell that how much percentage of the concepts from LOs are covered by all the questions together. The CAS represents this value. To formulate CAS, the values from the CAM needs to be aggregated. The scores of only those questions which are useful are considered for aggregation.

The values of each row of CAM indicate the contributions made by each question towards that LO where maximum can be 1. The CAM [i, j] = 1 in any row indicates that $CL_i = CQ_j$ i.e. all the concepts in LO are covered by that question. All other questions corresponding to the remaining cell values less than 1 in that row are redundant in terms of content and they are not contributing anything more to that LO. Such questions are non-utility questions pertaining to that LO and can be made as 0. This form the first step towards generating Q-utility matrix.

The remaining non-zero values in any row CAM [i, j] indicate partial coverage of concepts towards that LO. Even though the individual alignment values are small and the corresponding questions cover only few concepts from LO, if questions are combined, it may result in higher alignment values as they may cover more number of concepts from LO. If there are N non-zero values, there are 2^N possible combinations of questions. Instead, in the first iteration we paired the maximum valued question in a row with every other non-zero valued question in that row and the combined score was calculated, If the combined score is less than the maximum, then the maximum value is retained and all other values are made zero. Otherwise both values are retained. In the next iteration, the next highest value is considered as maximum and the process is repeated. This continues till there are no further maximum values to be considered. The algorithm for generating question utility matrix is given in Fig. 6.18 and the generated Q-utility matrix is shown in Table 6.5.

```
Algorithm: Generate Q-Utility Matrix
  N=number of questions (columns);
  For every row i do following.
            For every jth element [i, j] in the row i {
                       if [i, j]==1 then
                       { [i,0 to j-1] = 0;
                         [i, j+1 to N] = 0; }
                        }
 For every row i do following.
   Create a single row matrix (isProcessed) with all values = 0;
    processCount = 1; //non-zero
    while (processCount > 0)  {
            processCount = 0;
            Find the [i, jMax] element in the row i where ([i, jMax] is max of all [i, j]) and
            (isProcessed [j] == 0);
            assign isProcessed [jMax]=1;
            For all element [i, j] in the row I  {
                 if j!=jMax AND isProcessed[j]==0 {
                    CScore = CalculateCombinedAllignemntScore([i , jMax], [i, j]);
                       if( CScore <= [i, jMax]) {
                             assign [i, j]=0;
                             processCount++;   } }
             } }
```

Figure 6.18 Q-utility algorithm

Table 6.5 Q-utility matrix

Q / LO	Q1	Q2	Q3	Q4	Q5	Q6
LO 1	0	0	0.33	0	0	0.33
LO 2	0	0	1	0	0	0
LO 3	0	0.33	0	0	0	0.33
LO 4	0	1	0	0	0	0
LO 5	0	0	0	0	0	1
LO 6	0	0	0	0	0	0.5
LO 7	1	0	0	0	0	0

From the Q-utility matrix, for every row, calculate the combined alignment score of all questions together that has a non-zero score. For example, from the first row Q3 and Q6 were combined and the combined aligned score with respect to LO1 was calculated. In this case, it comes out to be 0.66.

$CQ3 \cup CQ6$ = {queue, operation}

CL_1 = {operation, stack}

Alignment $((CQ3 \cup CQ6), CL_1) = (1)/(1+1+1) = 0.66$

This **combined alignment score** for each row indicates the maximum possible contribution that all the questions together will make towards that LO. The mean of all these values will be the content alignment score for the AI.

For the above Q-utility matrix, the maximum of each row is (0.66, 1, 0.66, 1, 1, 0.5, 1).

The mean is (0.66+1+0.66+1+1+0.5+1)/ 7 = 5.16/7 = 0.8314

Thus, the content alignment score of the above AI is 83.14%.

The alignment score measures the content alignment of AI with LOs. The higher the value, the higher is the alignment. The above score indicates to the teacher that the AI they have designed is covering only 83.14% of the concepts covered in the LOs. Lower values can result because of redundant questions (repeated questions asked on the same concepts), out of scope questions or some of the concepts being left out. In the example above, there are 2 questions asked on applications of queue but there are no specific questions on stacks for LO1 and LO3.

6.1.3.2.3 Generating Cognitive Level Alignment Matrix (LAM)

The cognitive alignment score for a pair is calculated using the set of rules defined in Section 6.1.3.1 (b). Given an AI and a set of LOs, for every pair of question and LO, the measure of alignment is calculated and then AI cognitive alignment matrix is generated as shown in Table 6.6, where rows are LOs, columns are questions and each cell contains the measure of alignment.

LAM [i, j] = cog_alignment (LL_i, LQ_j)

$i \in L$ and $j \in Q$ where L is set of LOs and Q is set of questions

where LL is cognitive level of LO and LQ is cognitive level of question

Table 6.6 cognitive level alignment matrix (LAM)

Q / LO	Q1	Q2	Q3	Q4	Q5	Q6
LO 1	100	100	100	50	0	50
LO 2	100	100	100	50	0	50
LO 3	100	100	100	50	0	50
LO 4	100	100	100	50	0	50
LO 5	50	50	50	0	0	100
LO 6	100	100	100	50	0	50
LO 7	100	100	100	50	0	50

6.1.3.2.4. Formulate the Cognitive Level Alignment Score (LAS)

The LAM [i, j] = 100 in any row indicate that $LL_i = LQ_j$ i.e. cognitive level of LO is same as the cognitive level of question. From Q-utility matrix in Table 6.5, it was found that

Q4 and Q5 are non-utility questions. Hence, the cognitive levels of such questions are excluded from LAM for calculating LAS. The maximum value for each row in LAM indicates the maximum contributions made by all the questions towards that LO.

$$\text{LAS} = \frac{\sum_{i \in L} Lo\_score(Li)}{|L|} ---------------- (2)$$

$$\text{LO\_score}(L_i) = \text{Max}_j\ (\text{LAM}\ [i, j]) -- (3)$$

$i \in L$ and $j \in Q$

Hence, the AI is 100% aligned to the given set of LOs.

The 100% score indicates to the teacher that in the AI they have designed, there is atleast one question which is cognitively aligned with each of the LO.

6.1.3.3 Reporting the Quality of AI

The final report of the quality of AI against a given set of LOs is in terms of its alignment with respect to content and cognitive level as well as the information about the non-utility questions. Thus, for the above AI, the quality report is

- The content alignment score is 83.14%
- The cognitive level alignment score is 100%

Utility of Questions

For computing CAS, we were focusing on the LOs and seeing how they are supported by the questions. Focusing on the questions in the AI gives another perspective, and we can define a utility measure of a question characterizing how much it contributes to various LOs. For example, Q4 and Q5 in the utility matrix contribute nothing to any LO. They can be discarded or can be replaced by some other useful questions. Utility of a question can be calculated by considering the maximum value of each column. Question 1 has a utility value of 1. The average utility of all the questions together is the mean of the individual questions utility value. From the above Q-utility matrix,

The average utility of all the questions together is 4/6 = 0.66

6.2 Implementation

The IQuE system is implemented in JAVA platform (JDK 1.8). Protégé tool is used for developing and modifying the domain ontology. Stanford CoreNLP Library is used to parse LOs, questions and extract action verbs, concepts and relations from text files (Noy, N. F., 2003). owlapi 4.0 Library in Java is used to parse ontology (Bechhofer, S., & Matentzoglu, N., 2014) and Graphstream 1.3 for visualizing the ontology.

6.2.1 Protégé implementation of Domain Ontology

Protégé includes machine-interpretable definitions of basic concepts in the domain and relations among them. Protégé helps to make domain assumptions explicit, to analyze domain knowledge, to enable reuse of domain knowledge and to separate the domain knowledge from the operational knowledge (Noy and Deborah, 2000) (Noy, N. F. et al., 2003). Protégé is used to build domain ontology for *Data Structures*. Hence, whatever modifications is done in the knowledge structure (ontology) will not affect the other part of the system. The output is in the form of OWL file which can be utilized by the JAVA program (Bechhofer, & Matentzoglu, 2014). The ontology creation process is discussed in Appendix-A

6.2.2 Implementation of IQuE

IQuE takes as input three files as input

- los.txt that contains the set of LOs for a course
- q.txt containing questions from an AI and
- data_structures.owl which is domain ontology

The *data structures* course of second year engineering curriculum is chosen as the domain here and all examples are taken from that.

Step 1: Upload the files

User can upload the three files: los.txt, q.txt and data_structures.owl corresponding to LOs, AI and ontology using an interface as shown in Fig. 6.19.

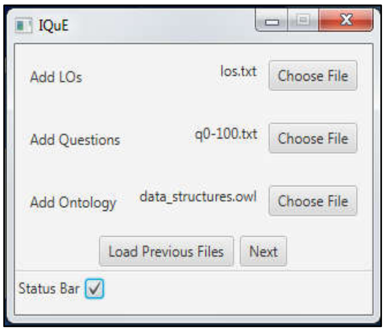


Figure 6.19 Interface to upload the input files

Step 2: Extract Concepts and cognitive level from LO or question

Each LO or question (LQ) is parsed from respective files and processed to find the concepts and cognitive level from it and then use this to color the nodes of the ontology. This complete process is explained in the following algorithm. Based on the issues, challenges and the solution discussed in Section 5.1.2, the concept mapping algorithm is shown in Fig. 6.20.

Algorithm Ontology Mapper

```
    stmt = read (LQ)
    LQ.stmt = preprocess (stmt)
    tagged_ stmt = POS_tagger (LQ.stmt)
    action_verbs = find_action_verbs (verbTags)
    c_level_LO = find_cog-level (action_verbs)
    LQ.relation = find_relation ( LQ.stmt)
    onto_concepts = extract_concepts (OWLfile)
    LQ.explicit = find_explicit_concepts (onto_concepts, LQ_stmt)
    LQ.implicit = onto_traversal (LQ.explicit, onto_concepts, LQ.slot)
    c_level_concepts = assign_cog_level (LQ.implicit, c_level_LO)
    color _code = assign_color (c_level_concepts)
     color_onto_nodes (LQ.implicit, color_code, OWLfile)
```

Figure 6.20 Ontology Mapper Algorithm

- Pre-processing steps: An LO or question is read and then converted into lower case and then lemmatized.

  Example,

  e.g. stmt = "*Demonstrate and implement different methods for traversing trees*."

  clean_stmt = lower_case (stmt)

e.g. clean stmt = "*demonstrate and implement different methods for traversing trees*."

lemma = lemmatize (clean_stmt)

e.g. lemma = "demonstrate and implement different method for traverse tree"

- Find Cognitive Level: First the preprocessed statement is tagged using Stanford coreNLP POStagger. Then the verbs are extracted from it. These verbs are matched with the action verbs stored in the dictionary corresponding to each Bloom's level and then the cognitive level is extracted from it.

  tagged_ stmt = POS_tagger (LQ.stmt)

  e.g. tagged_stmt = describe/VB how/WRB various/JJ data/NN structures/NNS are/VBP represented/VBN in/IN memory/NN

  These individual tokens are then separated into verbTags and noun tags. All types of verb tags like VB, VBN, etc. are stored in verbTags

  action_verbs = find_action_verbs (verbTags)

  E.g. action_verbs = [describe]

  c_level_LO = find_cog-level (action_verbs)

  E.g. c_level_LO = 'Understand'

- Extract Explicit and Implicit Concepts: Explicit concepts are directly obtained by matching the individual words or tokens from LQ_stmt to node names in ontology stored in onto_concepts. The function *find_explicit_concepts* involves the N-Grams routine. N-Grams algorithm is used for getting multi-worded concepts. For finding implicit concepts, slot and associated slot variable is found from the clean stmt. Then the onto_traversal algorithm is called to traverse the ontology.

  LQ.explicit = find_explicit_concepts (onto_concepts, LQ_stmt)

  LQ.implicit = onto_traversal (LQ.explicit, onto_concepts, LQ.slot)

  The onto_traversal algorithm is shown in Fig. 6.21.

```
Algorithm Onto_Traversal
//Find the slot indicator from the preprocessed statement
     LQ.slot = find_slot (LQ.stmt)
//Find the slot variable from the preprocessed statement
     for each slot_varible
          traverse hasSubClass link and include all connected nodes into LQ.implicit
     for each relation in LQ.relation
                      traverse the link and include all connected nodes into LQ.implicit
          if the LQ.implicit do not have the identified link connected to it, then
          {
              traverse each of the superclasses to see whether they have the link.
              If they have the link then
              {
                 traverse that link to include all the nodes connected into LQ.implicit
                 remove the redundant concepts from LQ.implicit
               }
           }
           If concepts in LQ.implicit are  isolated then
         {
               find the path between them
               if path length = 1 then
                      {
            include intermediate nodes into LQ.implicit }
         }
```

Figure 6.21 Ontology Traversal Algorithm

Figure 6.22 shows the output with an LO and its corresponding cognitive level, slot, slot variable, relation and all explicit and implicit concepts extracted by the onto_traversal algorithm. Because of the ‘implement verb’, the cognitive level is ‘Apply’. The word ‘various’ is a slot indicator here. Hence, the slot variable ‘operation’ is traversed in all its subclasses. The subclasses of concept ‘data structures’ is traversed because of its plural form which is also a slot indicator.

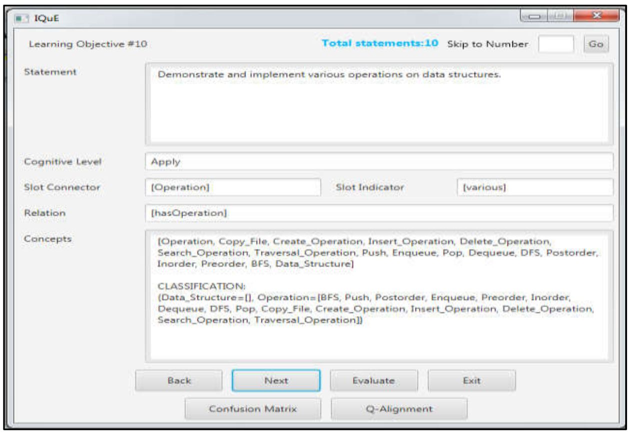


Figure 6.22 An LO with its associated extracted Information

Step 3: Assign cognitive levels to explicit and implicit concepts

Assign the same cognitive level to all the explicit and implicit concepts in LQ_implicit associated with a given LO.

c_level_concepts = assign_cog_level (LQ.implicit, c_level_LO)

Step 4: Find the color codes and assign them to corresponding nodes of the ontology.

color _code = assign_color (c_level_concepts)

color_onto_nodes (LQ. implicit, color_code, OWLfile)

Step 5: Generate the output of IQuE

Fig. 6.23 shows the interface to select the type of output the user wants.

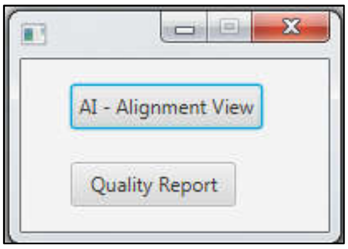


Figure 6.23 Interface to select the type of Evaluation

The user can select 'AI-Alignment View' which provides the graphical view of the alignment as shown in Fig 6.24 and/or 'Quality Report' which gives the content and cognitive level alignment scores as shown in Fig 6.25. The quality report also gives the narrative summary about the utility of questions.

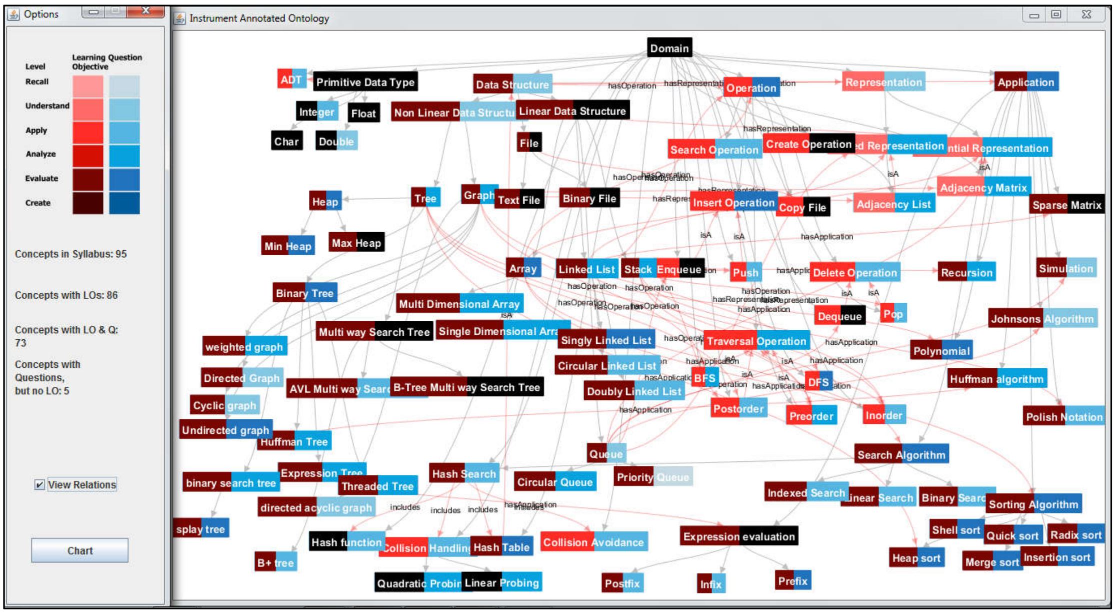


Figure 6.24 AI Alignment View

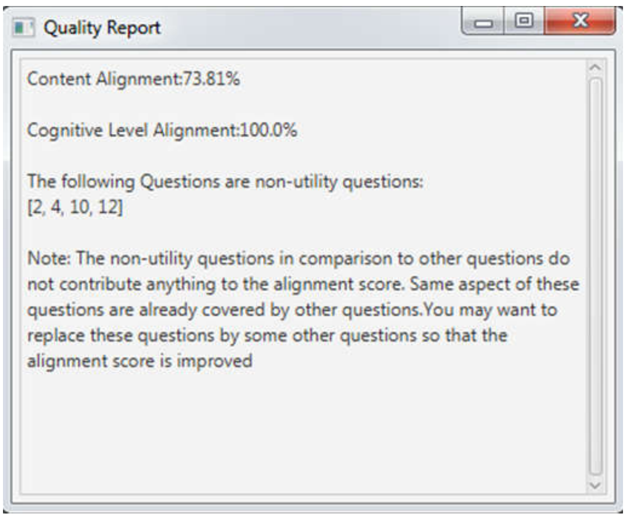


Figure 6.25 Quality Report of AI

Step 6: Testing the IQuE

From the interface shown in Fig. 6.22, the user can select the option of "Q-Alignment". He/she will get an interface to enter the values for 'α' and 'β' to substitute in the alignment formula as shown in Fig. 6.26. The option 'Generate' will automatically generate the excel spread sheet for Content and Cognitive level alignment matrix that can be used for testing the system.

Figure 6.26 Interface to enter values for α and β

From the interface shown in Fig. 6.22, the user can select the option of 'Confusion Matrix'. User will get an interface as shown in Fig. 6.27 where he/she must enter

1) An Excel file 'expert-teacher1.xlsx' containing expert generated alignment values for content and cognitive level alignment in two spreadsheets 'teacher1-concept' and 'teacher1-cog' respectively.
2) Select the option to generate the confusion matrix either for content or cognitive level.
3) Select an option either for generating 2-class or 3-class confusion matrix.
4) Select the appropriate spreadsheet from 'expert-teacher1.xlsx' as per the option selected for (ii).

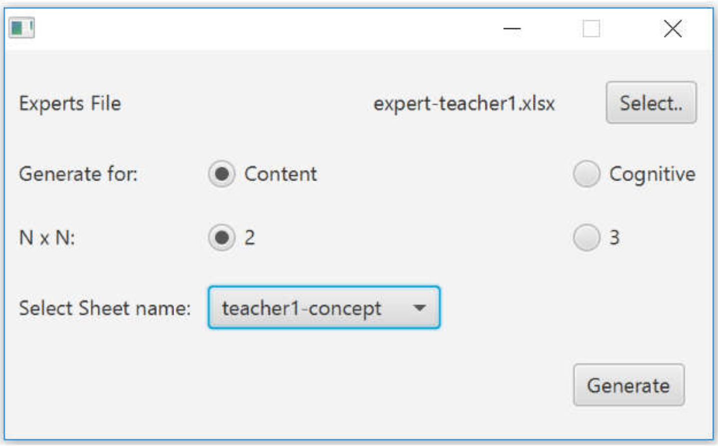


Figure 6.27 Interface to select option values to generate Confusion Matrix

6.3 Summary

The framework for quality evaluation captures the alignment of LO and AI in a course using ontology based representation. The framework incorporates three major steps, creation of domain ontology, generation of LO and AI annotated ontology and devising an alignment score. IQuE is an implementation of this approach. Ontology is created for the *Data Structures* domain. The concept and cognitive level information is extracted from the LOs and questions, and mapped to domain ontology to get LO and AI annotated ontology. IQuE provides three types of output; Numerical measure of alignment in terms of content and cognitive level, visual representation of alignment and intermediate output which shows the concepts, relations and cognitive level identified by IQuE for each LO and question. The refinement of IQuE after formative evaluation and its final summative evaluation is described in chapter 7.

Chapter 7

Refinement and Evaluation of IQuE

The design, development and implementation of IQuE is discussed in the Chapter 6. The IQuE was then subjected to formative evaluation to find its accuracy. The results of evaluation were quantitatively and qualitatively analysed to find the areas where further scope of improvement is possible. This led to refinement of IQuE which represents cycle 2 of DDR. Once all the refinements in respective stages were done, IQuE was summatively evaluated to get the final accuracy which represents cycle 3 of DDR. Following sections describe the process of formative evaluation, refinement and summative evaluation of IQuE.

7.1 Formative Evaluation of IQuE

Accuracy of IQuE was tested on following two aspects:

- Is the IQuE correctly measuring the alignment between a question and LO?
- How much is it deviating from the expert teachers' notion of alignment?

The first question was answered by performing a quantitative analysis of difference between the system generated and expert generated alignment values. Three experts provided the alignment values for both content and cognitive level. Experts were teachers who had,

- A teaching experience of more than 10 years
- Taught *Data Structures* course multiple times
- Been a member of curriculum revision committee of university and
- Designed the AI for the course in university examinations.

Later, together they came to a common set of alignment values by discussing over the differences.

Large number of samples of LOs and questions were presented to IQuE. It then calculated the measure of alignment between each pair of LO and question in terms of the content and cognitive level. These were then compared with expert generated values and a confusion matrix was generated. The accuracy was calculated from the matrix for both the content and cognitive level.

7.1.1 Samples

A pair of LO and question was considered as one sample for testing. A sample of size, N = 1000 consisting of 100 questions and 10 LOs were taken for first level of testing. The LOs were mostly adopted from ACM computer science curriculum (Sahami et. al., 2012) and later customized to suit the syllabus. Syllabus of *Data Structures* course of second year engineering curriculum of Mumbai University in India was selected as the domain. LOs were catering to different concepts from syllabus and they were corresponding to varying levels of Bloom's taxonomy ranging from 'Understand' to 'Evaluate' level as shown in Table 7.1.

Questions were collected from *Data Structures* course offered in different universities from India and abroad as well as from the competitive examination like Graduate Aptitude Test Engineering (GATE) for CS which is a national level engineering entrance examination. The syllabus of different universities varied in their contents. But we have taken questions from the topics from the Mumbai University curriculum. 270 questions were collected out of which 100 questions were randomly selected for first level testing. Questions were from varied topics and different cognitive levels.

Table 7.1. Representative LOs of *Data Structures* course

Students should be able to -----	
LO1	Model a given Data Structure as ADT
LO2	Describe how various data structures are represented in memory
LO3	Write programs that use data structures such as: arrays, linked lists, stacks, queues, trees, hash tables, and graphs.
LO4	Demonstrate and implement different methods for traversing trees.
LO5	Understand the working of different multi-way search tree structures
LO6	Implement the Huffman coding algorithm using binary tree
LO7	Implement hash tables, including collision avoidance and resolution techniques
LO8	Implement various searching and sorting algorithms
LO9	Analyze a given problem and select the appropriate data structures required to solve the problem.
LO10	Demonstrate and implement various operations on data structures

7.1.2 Generation of Confusion Matrix

When a text file of questions and LOs are given as input to IQuE, it generates an excel file consisting of CAM and LAM as two separate spreadsheets in it where a cell contains alignment values for each pair of question and an LO.

a) Confusion Matrix for Content Alignment

Confusion matrix is useful to compare the system generated alignment values with the expert rated alignment values. In order to generate a confusion matrix, we need to create some discreet classes. We have opted for two classes because it is simple and intuitive with a midpoint value of 0.5 as a separation criterion. Looking at exact values of numbers is meaningless as there is no universal agreement on what is the exact score for alignment. Hence, we need to decide at what point of time we can say that the decision of teacher's and system's alignment measures are comparable. It is possible to take more than two classes and some other threshold value as class separator instead of 0.5. We did some trials by taking threshold values of 0.4, 0.5, 0.6 and 0.8. The performance is approximately same for values of 0.3, 0.4, 0.5 and 0.6 but degrades with threshold values less than 0.4 and more than 0.7.

Table 7.2 shows a part of the content alignment matrix. It contains 1000 columns of questions and 10 rows of LOs. Each cell indicates the system generated content alignment measure for a corresponding pair of LO and question.

The system generated values were compared with expert generated values and a confusion matrix was generated. The confusion matrix can be generated considering two classes or three classes. For two classes of agreement, the levels are defined as [LOW ($0 \leq x \leq 0.5$) and HIGH ($0.5 < x \leq 1$)]. If the difference between the system generated and expert generated is more than 0.5, HIGH alignment is assumed, otherwise the alignment is LOW.

Fig. 7.1 shows the confusion matrix generated for two classes which shows the content alignment for N =1000. Out of 1000 samples from Fig.7.1 (a)

- True Positive (TP) = 37 (both expert and system gives HIGH alignment)
- True Negative (TN) = 764 (both expert and system gives LOW alignment)
- False Positive (FP) = 116(Expert gives LOW but the system says HIGH)
- False Negative (FN) = 79 (Expert gives HIGH but the system says LOW)

Overall accuracy for content alignment of IQuE =

$$\frac{(TP+TN)}{N} = \frac{(37+764)}{1000} = 80.1\%$$

The Tables from (b) to (k) represents the LO wise confusion matrix. Each of them had 100 samples of questions. Hence, (b) to (k) represents the performance of LO1 to LO10. The performance of all other LOs except LO3 and LO9 is very good. This reduces the overall accuracy of IQuE. The detailed explanation for this is provided in section 7.3.1.

LO\Question	1.Write a program to implement circular queue	2.Explain Linear and Non Linear data structure with example	3.What are the advantages of linked list over array? Write a program to implement stack using	4.Write a program to implement insertion sort. Show the passes of insertion sort for the	5.Give different searching techniques. Write a program to implement binary search
1.Model a given Data Structure as ADT.	0.5	0.5	0.5	0	0
2.Describe how various data structures are represented in memory.	0.5	0.5	0.5	0	0
3.Write programs that use data structures such as: arrays, linked lists, stacks, queues, trees, hash tables,	0.5	0.5	0.5	0.5	0.5
4.Demonstrate and implement different methods for traversing data structures.	0.5	0.5	0.5	0	0
5.Explain the working of different multi-way search tree structures.	0	0	0	0	0

Table 7.2 A partial snapshot of the excel worksheet showing the measure of content alignment

2) Confusion Matrix: Cognitive Level Alignment

Fig. 7.2 shows the system generated confusion matrix for cognitive level alignment. Each cell in the cognitive level alignment matrix (LAM) indicates the system generated cognitive level alignment measure for a corresponding pair of LO and question. The system generated measure of cognitive level alignment is compared with expert generated values for generating a confusio matrix. Out of 1000 samples from Fig.7.1 (a)

- True Positive (TP) = 358 (both expert and system gives HIGH alignment)
- True Negative (TN) = 489 (both expert and system gives LOW alignment)
- False Positive (FP) = 39 (Expert gives LOW but the system says HIGH)
- False Negative (FN) = 89 (Expert gives HIGH but the system says LOW)

N=1000		Program Generated	
		Low (0-0.5)	High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	**764**	**116**
	High (0.51-1.0)	**79**	**37**

(a) Combined

N=100		Program Generated	
		Low (0-0.5)	High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	93	1
	High (0.51-1.0)	2	4

(b) LO1

N=100		Program Generated	
		Low (0-0.5)	High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	92	3
	High (0.51-1.0)	2	3

(c) LO2

N=100		Program Generated	
		Low (0-0.5)	High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	55	2
	High (0.51-1.0)	34	9

(d) LO3

N=100		Program Generated	
		Low (0-0.5)	High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	90	6
	High (0.51-1.0)	1	3

(e) LO4

N=100		Program Generated	
		Low (0-0.5)	High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	96	0
	High (0.51-1.0)	4	0

(f) LO5

N=100		Program Generated	
		Low (0-0.5)	High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	**88**	**10**
	High (0.51-1.0)	**2**	**0**

(g) LO6

N=100		Program Generated	
		Low (0-0.5)	High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	**98**	**0**
	High (0.51-1.0)	**2**	**0**

(h) LO7

N=100		Program Generated	
		Low (0-0.5)	High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	**81**	**1**
	High (0.51-1.0)	**13**	**5**

(i) LO8

N=100		Program Generated	
		Low (0-0.5)	High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	22	65
	High (0.51-1.0)	11	2

(j) LO9

N=100		Program Generated	
		Low (0-0.5)	High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	**48**	**28**
	High (0.51-1.0)	**12**	**12**

(k) LO10

Figure 7.1 Confusion Matrix: Content level alignment

N=1000		Program Generated		
		Low (0-0.5)	High (0.51-1.0)	No Match
Expert (Teacher) Generated	Low (0-0.5)	484	39	23
	High (0.51-1.0)	89	358	7

Figure 7.2 Confusion Matrix: Cognitive level Alignment

The accuracy for the cognitive level alignment is (484+358)/1000 = 84.2%.

7.1.3 Result Analysis

We did a deeper analysis to find the reason behind less accuracy. This has resulted in some design changes in all the stages of IQuE. It resulted in Cycle 2 of DDR.

All the false positive (FP) and false negative (FN) cases were analyzed and then categorized initially in 2 classes

1) Can be fixed by refining any of the stages of IQuE
2) Extremely difficult or impossible to automate unless the IQuE is equipped with more intelligence or sophisticated NLP capabilities.

Table 7.3 shows samples of unique FP and FN and the corresponding pair of LO and question and justification of their membership to class 1 or class 2. If the sample belongs to class 1, then the last column provides the information about which stage of IQuE needs to be refined. In stage 3, Q-utility algorithm was modified to enhance the performance measure.

Table 7.3. Examples of FP and FN

LO	Question	Justification	Category
1.Model a given ***Data Structure*** as ***ADT***.	Q1. Write an ***ADT*** for ***Circular Queue***. Explain its primitive functions with example	The parser considers ***model'*** as noun instead of verb and the IQuE fails to identify its cognitive level of LO.	**(case 1)** Refining stage 2: Concept mapping procedure

LO	Question	Justification	Category
2.Write programs that use ***data structures*** such as: ***arrays, linked lists, stacks, queues, trees, hash tables***, and ***graphs***.	Q2. Write a program to implement ***insertion sort***. Show the passes of ***insertion sort*** for the following input.	This should be an exact match but IQuE gives CAS = 0. If a program uses ***data structure***, then it is an application of that ***data structure***. Here, ***insertion sort*** is an application and uses an ***array*** but it is not explicit.	**(case 1)** Refining stage 2: Concept mapping procedure
3.Demonstrate and implement different ***methods for traversing trees***.	Q3. Consider the ***graph*** at right. A vertex x is "finished" when the recursive call ***DFS (x)*** terminates. In what order are the vertices finished?	1)Ideally, the CAS should be zero but it is 0.33. As ***DFS*** is connected by 'hasSubClass' relation to ***traversal operation,*** it is included in the list of identified concepts by default. 2)IQuE is unable to recognize cognitive level because of the absence of BT action verbs	**(case 1)** 1)Refining stage 1: Structure of ontology **(case 2)** Refining stage 2: Cognitive level mapping procedure
	Q4. Check if a given ***array*** can represent ***Preorder Traversal*** of ***Binary Search Tree***	This is a perfectly aligned question. The 'hasRepresentation' link is extracted from the verb 'represent' which gets traversed and concepts ***Representation*** and then to '***Sequential Representation.***	**(case 2)** Currently not automated in IQuE
4. Understand the working of a ***multi-way search tree*** structures.	Q5. Discuss ***AVL trees***. Insert the following elements into a ***AVL search tree***: 27 23 25 29 35	This should be an exact match but IQuE gives CAS as 0.33. But lack of knowledge that working of any algorithm also involves their ***representations*** and ***operations*** on them.	**(case 2)** Refining stage 1: Structure of ontology
5. Implement ***hash tables***, including ***collision resolution*** and avoidance.	Q6. ***Hash*** the following in a table of size 11. Use any two ***collision resolution techniques***. 23 0 52 61 78 33 100 8	This should be an exact match but IQuE gives CAS as 0.25. It does not have enough knowledge that ***Hash search*** involves the construction of ***hash table***.	**(case 2)** Refining stage 1: Structure of ontology
	Q7. Explain the procedure of collision avoidance in hash table	This should be an exact match but IQuE gives CAS as 0.33. Unable to recognize the concept ***collision avoidance*** in LO	**(case 2)** Currently not automated in IQuE

LO	Question	Justification	Category
6. Implement various searching and ***sorting algorithms.***	Q8. Write a program to implement ***Merge sort***. Show the steps to sort the following numbers. 99 22 88 66 40 10 34 52	This should be an exact match but IQuE gives CAS as 0.5. IQuE is unable to recognize that both ***searching algorithm*** and ***sorting algorithm*** are important concepts and they are disjoint. So, if anyone is present, it should be considered as complete alignment	**(case 2)** Refining stage 2: Concept mapping procedure
	Q9. Explain ***indexed sequential search*** with an example	This should be an exact match but IQuE gives CAS as 0. The concept ***search algorithm*** is not recognized by IQuE because of 'and'.	**(case 2)** Currently not automated in IQuE
7.Analyze a given ***problem*** and select the appropriate ***data structures*** required to solve the ***problem***.	Q10. Write a program to convert given fully parenthesized ***infix expression*** to ***postfix expression***.	This should be an exact match but IQuE gives CAS as 0.5 as it is unable to recognize that conversion of infix to postfix is an ***application*** of ***stack*** as the concept ***stack*** is implicit here.	**(case 2)** Refining stage 2: Concept mapping procedure
8. Demonstrate and implement various ***operations*** on ***data structures***.	Q11. Write a non-recursive function for inorder traversal	This should be an exact match but IQuE gives CAS as 0.5. IQuE is not getting a match for ***data structures*** which is implicitly ***tree*** here.	**(case 2)** Refining stage 2: Concept mapping procedure

Refinement done in different stages is described in next section.

7.2 Refinement of IQuE (Cycle 2 of DDR)

The samples that resulted in FPs and FNs in the confusion matrix that were classified into class led to further refinement in development stages of IQuE. These are explained below.

7.2.1 Design changes in stage 1: Creation of ontology

There were some changes in the structure of the ontology.

- The link 'hasSubclass' is a relation from parent node to its child nodes while 'isA' is reverse link of that. Some of the nodes which were earlier linked by 'hasSubclass' relation was changed to 'isA' relation. For example, in the beginning, all the traversal operations like *DFS, BFS, preorder*, etc. was connected to the node *traversal operation* with a 'hasSubclass' in ontology. As per the ontology traversal algorithm, the explicitly identified concept and all the concepts connected to it by 'hasSubclass' get added by default if there is a slot indicator connected to it. But every *traversal operation* is associated with a specific data structure. For example, *inorder, preorder* and *postorder* are *traversal operations* on *trees*. Similarly, *BFS and DFS* are *operations* on *graphs*. And the question asked may be on any one of these. Consider the following pair of LO and question:

LO: Demonstrate and implement different methods for traversing trees.

Q: Write algorithm for Breadth First Traversal of the graph.

Here L = {[traversal operation], tree}

Q = {Breadth First Search, graph} ---- [*Breadth First Search* is the node name and *Breadth First Traversal* is its synonym]

The concept *methods of traversal* and all its subclasses get implicitly extracted from LO as there is a slot indicator 'different' connected to it. This is restricted by removing all the specific traversal operations from the subclass hierarchy and creating a reverse 'isA' link. Hence, in the above example, *Breadth First Search* will be connected to *traversal operation* by an 'isA' relation and the algorithm for calculating the alignment will match the *Breadth First Traversal* to *traversal operations.* Now, they will be derived from the relations: *Graph* 'hasOperations' *Breadth First Traversal* and it 'isA' *traversal operations*.

Such instances are present at many other places which were modified. *Adjacency matrix* was removed as a subclass of a *Sequential representation* and was connected by 'isA' relation. Another example (LO3, Q3) is from the Table 7.3.

- A concept of family of nodes was introduced. There are situations like one concept having many other parts included in it. In other words, to understand and implement one concept you need to understand other parts. Implementing *Hash search* includes implementing *Hash function, Hash Table collision handling* and *collision resolution techniques*. Consider an LO,

LO: Demonstrate and implement hash search.

Q: Using Linear probing and quadratic probing insert the following values into a Hash table of size 10. Show how many collisions occur in each technique: 99 33 23 44 56 43 19 85 78 54

The question is highly aligned with the LO. But there are no matching concepts and the system will flash low alignment. In such situations. the concept of family of nodes help. Another example is (LO5, Q6) from Table 7.3.

Some additional synonyms were added to some of the nodes in the ontology. *Methods of traversal* was added as synonym to *traversal algorithm*.

7.2.2 Design changes in Stage 2: Generation of LO and AI Ontology

Following are the major changes made in stage 2. They can be considered as additional constraints in ontology traversal algorithm.

For content level alignment,

- When an N-Grams is identified as a concept, all its N-1 grams should be removed from the list of identified concepts. Consider an LO and a question pair

  LO: Students should be able to implement various searching algorithms

  Q: Construct a Binary Search Tree from given post order expression.

  Here *binary search tree* is recognized as a concept using N=3. Then *tree* (N=1) and *binary search* (N=2) should not be included in the list of identified concepts. Otherwise, if this restriction is not put, then *Binary search* will also be recognized as a concept and will be matching with *searching algorithm* and will give a high alignment.

- The operations on data structures can be traced back to their specific data structures as they are connected to it by 'hasOperation' link even if they are not explicitly stated in the LO or question statement. In Table 7.2, the (LO8, Q11) pair illustrates this

  LO8: Demonstrate and implement various ***operations*** *on* ***data structures****.*

  Q11. Write a non-recursive function for inorder traversal

  The *inorder traversal* can be traced back to *tree* data structure as *inorder traversal operation* is connected to tree using 'hasOperation' link in the ontology. So, the *tree* can be added to the list of implicitly identified concepts and increase the CAS to 1 which is equal to 100% alignment.

- The process of matching the concepts from LO was done on a completely processed LO. But certain words like *linked list* get lemmatized to *link list* and are not identified as a concept by ontology mapper. Hence, we performed concept matching in two steps; first on a lemmatized statement and then on the original statement before lemmatization. Repeated words are then removed

For cognitive level alignment

- Apart from the action verbs associated with each of the cognitive levels in the Bloom's taxonomy, there are certain context or course specific action verbs such as insert, delete, print, write a program, etc. which frequently occur in the question stems. Most of them fit in 'Apply' category. To support the system for the correct recognition of cognitive level, these verbs were added to the list of action verbs in their respective levels.

- Certain words such as "Model", "Construct", etc. when they come in the beginning of the question statement are considered as nouns by POS Tagger. Consider

  Q: Model stack data structure as ADT

  In this, the verb 'Model' is wrongly tagged as noun by the POS tagger. If they are preceded by an explicit subject phrase appended by a subject like "You can", the POS tagger will be forced to classify them as verb. This solves our problem. Other example is (LO1, Q1) pair from Table 7.3.

- Questions in an AI are either a command or an interrogative sentence. If it is a command, action verbs in Bloom's taxonomy are helpful in classifying them into respective cognitive levels. We analyzed the questions with which the system was not successful in correctly classifying using action verbs. It was found that there are some question indicators which are commonly used by teachers to frame the questions which are not understood by the IQuE. The most frequently found question indicators are; " In what order", "What is the possibility", " How many", etc. They can be used to identify the cognitive level of questions. Consider the following question

Q: Consider the following graph. In what order are the vertices visited using DFS starting from vertex a? Where a choice exists, use alphabetical order. What if you use BFS?

In this case, the cognitive level of the question becomes 'Analyze' because of the question indicator 'In what order'. But the IQuE reports it as 'Recall' cognitive level because of the word 'Where'. Hence, such question indicators are added to the list of action verbs in the respective category of cognitive level. Another example is (LO4, Q4) from Table 7.2.

- Typically, in *Data Structures* course, the action verb 'create' is used at "Apply" level. Consider,

  Q: Write a program to create a linked list and perform the following operations (i) Insert into list (ii) Search for data (iii) Delete from list (iv) Display the list.

  We analyzed lot of AIs of *Data Structures* course generated for university examinations and concluded that 'create' should be pulled down to lower category of "Apply' level.

7.2.3 Design changes in Stage 3: Devising an AI Alignment Score

The major revision of stage 3 occurred in the Q-utility algorithm. In the earlier version, initial steps are same i.e. every row in CAM is scanned to find if there is alignment score of 1 and once found making all other scores in that row as zero. If it is not there, then all the questions that are corresponding to nonzero scores in that row were combined and the score is calculated. If the combined score is less than the maximum, the maximum score is retained and all others are made zero. But if we would have combined the questions pairwise instead of all together, the combined score may have been greater than the maximum in some cases and this will affect the final CAS of AI while aggregating the values from the Q-utility matrix. Consider a hypothetical example,

Suppose CL = $\{c_1, c_2, c_3, c_4, c_5\}$ where c_is are concepts and

CQ1 = $\{c_1, c_2, c_3\}$

CQ2 = $\{c_1, c_4, c_5, c_9)$

CQ3 = $\{c_5, c_6, c_7, c_8\}$

The CAM for this is

	Q1	Q2	Q3
L	0.6	0.25	0.125

In this case, by combining Q1, Q2 and Q3 the combined alignment score with the LO comes out to be 0.55 as CQ3 contains three out of scope concepts (concepts that are in question but not in LO). The combined score is less than the maximum. Hence, 0.6 will be retained in the Q-utility matrix and all others will be made zero.

But instead, if you combine Q1 and Q2, then the combined score will be 0.833 which is greater than 0.6. This flaw was detected during formative evaluation while analyzing the output of different sample AIs which led to refinement of the Q-utility algorithm.

7.2.4 Current Limitations of IQuE

There are some issues that are currently limitations of IQuE as they are extremely difficult or impossible to automate in the current approach. Currently, these issues are handled manually in the pre-processing step. Such issues are classified as follows.

7.2.4.1 The 'AND' Problem

Consider an example LO--

LO1: *Students should be able to implement various searching and sorting algorithms.*

Here, the algorithm will identify only one concept ***sorting algorithm***. The word 'algorithm' is connected to both ***searching*** and ***sorting*** because of the 'and' connective. The algorithm is not able to find this. One of the solutions is rewriting the LO as

LO1': *Students should be able to implement various searching algorithm and sorting algorithms.*

Now they will be identified as 2 separate concepts, and will be correctly processed.

7.2.4.2 Disjoint Concepts

In LO1' above, the concepts *searching algorithm and sorting algorithm* are disjoint. If a question is asked on any type of searching algorithms or any type of sorting algorithms, it should be considered as fully aligned question. But the algorithm considers them as two different concepts and this reduces the value of CAS. As a solution to this problem, such LOs can be split into smaller and more specific LOs. For example, LO1' can be divided into 2 LOs

LO1'(a): *Students should be able to implement various searching algorithms.*

LO1'(b): *Students should be able to implement various sorting algorithms.*

7.2.4.3 Slot Indicator and Multiple Concepts

If there are multiple concepts after the slot indicator, then the algorithm associates the nearest one to it. But it may not be true always. For example,

LO2: *Students should be able to demonstrate and implement different tree traversing methods*

Here, the algorithm identifies ***'different'*** as the slot indicator and ***tree*** and ***traversal operations*** as concepts. Hence, the nearest one is ***tree*** and slot indicator gets associated to it. So, ***tree*** gets expanded to all its subclass hierarchy. But the LO is intending different ***traversal methods*** such as ***inorder, preorder and postorder***. Thus, one solution is to rewrite the LO during pre-processing step. For example, LO2 can be rephrased as

LO2': *Students should be able to demonstrate and implement different methods of traversing trees*

7.2.4.4 Identification of Unwanted Concepts

Sometimes certain words in LOs and questions are misleading for the algorithm. They are used in a different context. But the algorithm identifies them as valid concepts or relations. Such concepts contribute negatively to CAS. For example, Consider a pair of LO and a question:

LO: *Demonstrate and implement different methods for traversing trees.*

Q: Check if a given array can represent Preorder Traversal of Binary Search Tree

In this case, the word 'represent' in the question forces the algorithm to identify 'hasRepresentation' relation. As far as the alignment is concerned, this link is useless. Further, this link gets traversed from every other concept and a few more concepts such as *Sequential Representation* gets added to the list and increases the non-matching concepts in the question and reduces the CAS.

7.3 Summative Evaluation of IQuE

All the stages of IQuE were refined as per the results of formative evaluation. Further, we wanted to find out how much improvement was there in its performance. Hence, IQuE was subjected to summative evaluation on the following aspects.

- Accuracy
- Usability and
- Usefulness of IQuE.

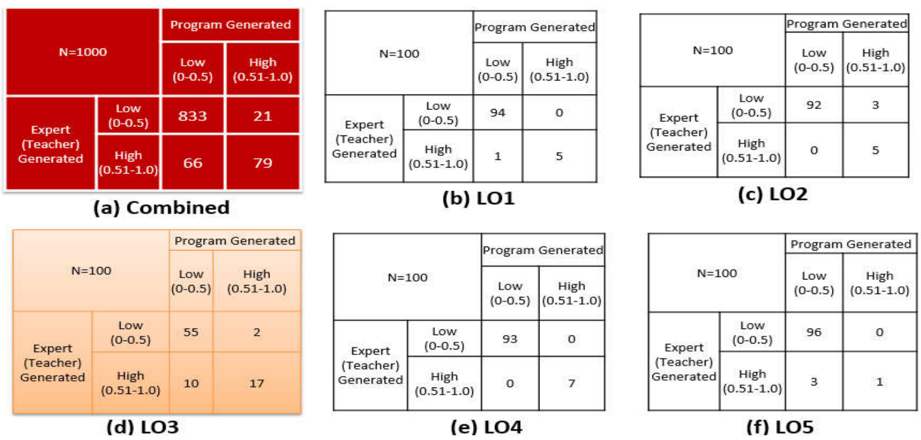
7.3.1 Accuracy of IQuE

The method followed for summative evaluation was same as that followed for formative evaluations. Same set of samples were taken for questions and LOs as we wanted to measure enhancement in the performance of IQuE after refining all the stages. Fig. 7.3 shows the confusion matrix generated for the content alignment for N =1000. Out of 1000 samples, as shown in Fig. 7.3 (a)

- TP = 79 (both expert and system gives HIGH alignment)
- TN = 833 (both expert and system gives LOW alignment)
- FP = 21 (Expert gives LOW but the system says HIGH)
- FN = 66 (Expert gives HIGH but the system says LOW)

Accuracy for content alignment of IQuE =

$$\frac{(TP+TN)}{N} = \frac{(79+833)}{1000} = 91.2\%$$



N=100		Program Generated Low (0-0.5)	Program Generated High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	**98**	**0**
	High (0.51-1.0)	**1**	**1**

(g) LO6

N=100		Program Generated Low (0-0.5)	Program Generated High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	**98**	**0**
	High (0.51-1.0)	**1**	**1**

(h) LO7

N=100		Program Generated Low (0-0.5)	Program Generated High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	**80**	**0**
	High (0.51-1.0)	**20**	**6**

(i) LO8

N=100		Program Generated Low (0-0.5)	Program Generated High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	64	14
	High (0.51-1.0)	16	6

(j) LO9

N=100		Program Generated Low (0-0.5)	Program Generated High (0.51-1.0)
Expert (Teacher) Generated	Low (0-0.5)	**63**	**2**
	High (0.51-1.0)	**14**	**30**

(k) LO10

Figure 7.3 Confusion Matrix (Summative Evaluation): Content Alignment

The accuracy increased by more than 10% after refining all the stages. The Tables from (b) to (k) represent the LO wise confusion matrix. The detailed analysis of LO wise performance was done to determine which of the LOs were reducing the overall accuracy of IQuE system. It was found that the performance of LO3 and LO9 are poor.

LO3: Write programs that use data structures such as: arrays, linked lists, stacks, queues, trees, hash tables, and graphs

LO9: Analyze a given problem and select the appropriate data structures required to solve the problem.

Deeper analysis indicated that there are some specific characteristics of these LOs that make identifying concepts from them difficult.

- These LOs are very broad. They cover huge set of concepts under them mostly covering the entire concepts in the domain, in this case 'data structures'
- They are at higher cognitive level (LO9 is at evaluate level).
- An aligned question against this LO is supposed to test the students' ability to solve an open ended/real world problem which involves the use of multiple concepts under the umbrella of explicitly identified concept (e.g. ***data structures*** in LO9). Moreover, such questions can be framed without the mention of any of the explicit concepts. For example, for a question framed against LO9, students are expected to select a specific data structure as a solution to the problem. Hence, a question with explicit mention of data structure is out of scope for it.

Thus, if further accuracy is to be increased, human annotation may be one of the solutions

b) Confusion Matrix: Cognitive Level Alignment

Fig. 7.4 shows the system generated confusion matrix for cognitive level alignment.

N=1000		Program Generated		
		Low (0-0.5)	High (0.51-1.0)	No Match
Expert (Teacher) Generated	Low (0-0.5)	512	30	3
	High (0.51-1.0)	49	399	7

Figure 7.4 Confusion Matrix (Summative evaluation): Cognitive level alignment

The accuracy for the cognitive level alignment is (524+399)/990 = 93.23%. The accuracy increased by 9 % after refining IQuE. In 1% cases from the remaining 6.8%, the system is not able to recognize the cognitive level of the question because there are no verbs matching to any of the action verbs associated with the cognitive levels of Bloom's taxonomy. In some cases, the action verbs associated with the question are misleading, as in

Q: Evaluate the following prefix expression " ++ 26 + - 1324".

'Evaluate' is the action verb associated with the 'evaluate' level of the Bloom's taxonomy but the question is only at 'Apply' level. Such interpretations have caused reduction in the inaccuracy.

7.3.2 Qualitative Analysis of Performance

We did a detailed qualitative analysis of the performance of IQuE. Every question and LO pair was analyzed to find the rationale followed by the algorithm to come to content and cognitive level alignment score. This further reinforced the logical correctness and the predictability of the IQuE. Table 7.4 provides the result of analysis for an example LO and 5 sample set of questions with different CAS values ranging from 0 (with no alignment) to value 1 (with complete alignment) as calculated by the system.

Table 7.4 Result analysis for an example LO

LO	Implement the ***Huffman coding algorithm*** using ***binary tree***.				
Questions	Q1. Write and explain algorithm to ***delete*** node from ***AVL tree***	Q2. Check if a given ***array*** can represent ***Preorder Traversal*** of ***Binary Search Tree***.	Q3. Write a program to construct ***binary tree*** for the following ***preorder*** and ***inorder traversal*** sequences	Q4. Write a program to create a ***binary search tree***. Show ***BST*** for the following input.	Q5. Explain the method of Huffman coding. Apply the ***Huffman Coding Algorithm*** and construct a ***Huffman tree***. Give the code for each symbol
CAS	0	0.2	0.33	0.5	1.0
CLAS	100	0	100	100	100
Analysis of CAS	None of the concepts are matching. ***AVL Tree*** is not a ***Binary tree.*** So, the content alignment is zero.	Question contains three concepts and ***Binary Search Tree*** is a ***binary tree***. The score is less because Q|L=2 and L|Q=1	Only ***binary tree*** concept is matching with LO. But the question has nothing to do with ***Huffman coding***. So, ideally, score should be zero. But it is 0.33 as Q|L=1 and L|Q=1	The question has only one concept BST which matches with the ***binary tree*** concept of LO. But the question has nothing to do with ***Huffman coding***. So, ideally, score should be zero.	Both question and LO has 2 concepts and they are matching. ***Huffman Tree*** is a type of ***Binary tree***. So, there is a perfect match.
Analysis of LAS	Because of the word 'delete', the cognitive level of question is 'Apply' and is same as the level of LO	Difference in cognitive levels of LO and question is more than 2. LO is at 'Apply' and question is at 'Evaluate' level	Cognitive levels are exactly matching. Both are at 'Apply' level	Cognitive levels are exactly matching. Both are at 'Apply' level	Cognitive levels are exactly matching. Both are at 'Apply' level

The LO from Table 7.4 covers 2 concepts ***Huffman coding algorithm*** and ***binary tree*** and its cognitive level is 'Apply'. The concepts identified by IQuE are shown in bold and Italics in the above Table. In case of content alignment, the IQuE assigns some score for partial coverage of concepts. It also penalizes for non-matching concepts in LO as well as in question. Currently the amount of penalty is same for both cases as value of α and β are kept equal to 1. But this can be changed by changing the values of α and β. In case of questions 2, 3 and 4, where ideally, the CAS should be zero, the IQuE is providing some value for partial coverage. To mask this behaviour, the IQuE considers that the question is highly aligned if the value is greater than 0.5.

7.3.3 Usability and Usefulness of IQuE

The IQuE was given to twenty users to study its usability and usefulness and SUS survey was administered (Brooke, 1996). The users were data structure domain experts out of which four were education technology (ET) researchers and remaining sixteen were only CS instructors from various engineering colleges in Mumbai. The average SUS score was found to be 73.4% for ET researchers and 85.53% for non-ET and CS instructors.

There were 10 Likert's scale questions in the survey questionnaire. Users were also told to provide an explanation of why they have chosen a particular response for each of the questions. The participants' responses to the open ended SUS questions were analyzed step by step using thematic content analysis. First the raw data (responses for each question from all the twenty participants) were individually analyzed and coded. Empty and meaningless responses were removed from the list. Coding for two sample responses is shown in column 2 of Table 7.5. The codes obtained from all responses of each question in step 1 were combined and then again analyzed. Similar meaning codes were then combined to form a single theme as shown in column 3 of Table 7.5. This process is repeated until all the themes are distinct and cannot be further combined.

Table 7.5 Example showing the steps of thematic content analysis of SUS responses for IQuE

Example Responses	**step1**	**step2**
1)I found it is very useful to prepare the assessment instrument. It took away all the headache of alignment of questions with the LOs while setting the assessment instrument.	*took away all the headache of alignment of questions with the LOs*	*Checking alignment of AIs with LOs is made easy (2)*
2)As the system is very helpful in designing a paper mapping with the course outcome which is bit difficult manually.	*Manual checking of alignment is difficult*	

At the end of thematic content analysis, we ended with 9 distinct themes which included 3 for usefulness and remaining 6 for usability. The distribution of these themes is shown in the bar chart of Fig. 7.5. It indicated that 19 out of 20 i.e. 95% of teachers perceived that IQuE is easy to use, 90% felt that it simple and self-explanatory, functions in IQuE were well integrated and IQuE has made checking alignment of AI with LOs easy. 55% users (teachers), who were only CS instructors expressed that some amount of prior knowledge regarding ontology, cognitive level, Bloom's taxonomy and alignment are needed to use the system. 60% users were positive about the usefulness of IQuE. User responses indicated that IQuE is easy to learn and operate, useful tool to check alignment and took away all the headache of checking the alignment of questions with the LOs.

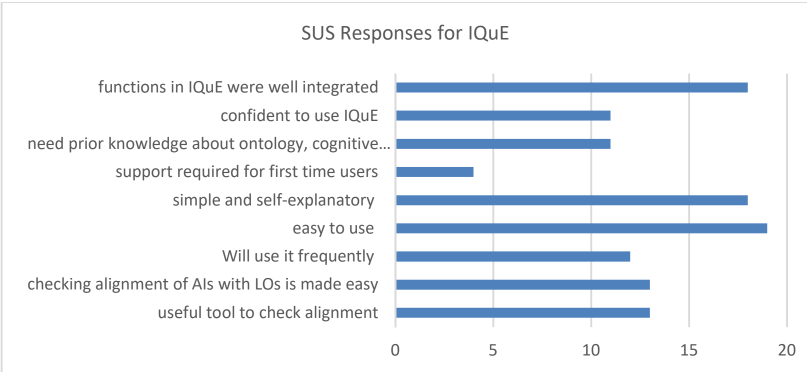


Figure 7.5. Distribution of SUS Responses for IQuE

7.4 Saturation of IQuE

After refining all the 3 stages, as described in Section 5.3, IQuE was subjected to summative evaluation and it was found that the accuracy for content alignment is 91.2% and accuracy of cognitive level alignment is 90.7%. Further, we wanted to confirm whether this process of refinement reach any saturation point or the accuracy will remain consistent even if the set of LOs and questions are different and there will not be any further modifications done in any of the stages of IQuE.

To establish this, the IQuE was subjected to one more round of evaluations with a different set of 100 questions and 10 LOs. The accuracy was found to be 91.2% for content alignment and 90.3% cognitive level alignment. This were further analyzed. Apart from the unsolvable problems discussed in Section 7.2.4. no further issues were found and therefore no modification was done to any of the stages. We consider that under the approach we followed, no further improvement is possible.

7.5 Summary

The design, development and implementation of IQuE was described in chapter 6. In this chapter, the process of formative evaluation, refinement and summative evaluation of IQuE is explained. The accuracy of IQuE was tested by comparing the system generated alignment score with the manually generated score by expert teachers and a confusion matrix was generated. We got an average accuracy of 91.2% for concepts and 93.23% agreement in cognitive level which is almost 10% higher than the accuracy obtained before refining IQuE. We did an in-depth analysis of individual LO wise performance. It was found that the performance of broader and higher cognitive level LOs, where an aligned question against them are supposed to test the students' ability to solve an open ended/real world problem, is poor. If such LOs are reused, the accuracy can become better. We administered a SUS survey to test the usability and usefulness of IQuE. The average SUS score was found to be 73.4% for ET researchers and 85.53% for non-ET and CS instructors. Thematic content analysis on the open-ended responses for each of the questions in the SUS survey indicated positive responses for both usability and usefulness of IQuE. Chapter 8 explains the design development and use of TTM which utilizes IQuE to measure the alignment of teacher entered questions with the system displayed LOs.

Chapter 8

The Teacher Training Module

The Teacher Training Module (TTM) is meant to be used to train teachers to write good assessment questions against a given set of LOs which in turn makes them better AI designers. TTM has been built on top of IQuE. The IQuE measures the alignment between the teacher entered questions and system given LOs and provides feedback. Following sections explain in detail the design and development of TTM.

8.1 Overview of TTM

The TTM utilizes IQuE to measure the alignment of teacher entered questions with the system displayed LOs. It uses this measure of alignment to formulate the feedback to be given back to the teacher. The feedback is given in simple English language and can be easily understood by the teacher (Shute, V. J., 2008). The teacher can modify the question as per the suggestions provided in the feedback and resubmit the same. This cycle of corrective feedback and question refinement is repeated till a satisfactory level of alignment is achieved.

The system architecture of TTM is shown in Fig. 8.1. TTM picks an LO/LOs from the file and presents it to the teacher. The teacher has to generate an assessment question/questions aligned to it. TTM forwards the teacher written questions and the system displayed LOs to IQuE. The IQuE then checks the alignment of the questions against the LOs and provides the measure of alignment. Using this, TTM gives corrective or advisory feedback. In case of corrective feedback, the teacher is told to modify the question and resubmit. Otherwise, he/she can move to the next LO. The advisory feedback is given to teacher to indicate that the alignment is in the acceptable range, but still there is scope for improvement. It makes the teacher more careful while generating questions for the subsequent LOs. Both advisory and corrective feedbacks are a type of formative feedback.

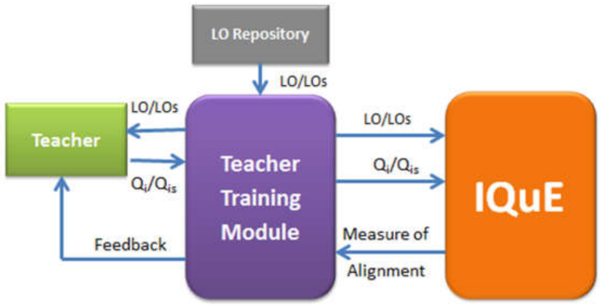


Figure 8.1 Overview of TTM

8.2 Design of TTM

Teachers face many challenges while framing questions for their AIs which are well aligned against the set of LOs of the course. Firstly, teachers have to clearly identify what are the different concepts covered by the LO. This will help them in framing the question within the boundary of content coverage of the LO. Secondly, they need to understand the cognitive level of the expected question from the corresponding LO. The cognitive levels are defined by the Bloom's taxonomy (Krathwohl, & Anderson, 2002). Difficulty increases because of the inherent ambiguity among the adjacent levels which is a common issue with Bloom's taxonomy. To make the situation worse, teachers in engineering discipline do not receive any formal training in this regard before they join teaching profession (Phatak, 2015). The design of TTM is influenced by techniques to solve this problem. Using the self-learning environment of TTM, teachers can practice writing assessment questions against a given set of LOs and improve them by following the automated formative feedback provided by it. Following subsections explain the critical decisions adopted to design TTM which includes the decisions of multiple stages and the implementation of the feedback mechanism.

8.2.1 Multiple stages of TTM

The tasks of generating aligned question(s) against given LO(s) can be categorized into three different scenarios, as follows

Case 1: Single LO and Single Question

There is a single LO against which the teacher has to write a single question. This is the simplest case as LOs will be more specific covering less concepts. For example, for the LO1 displayed by the system, the teacher frames a question as shown in Fig. 8.2

> *LO1: Students should be able to implement various operations on stacks.*
> *Q1: Write a program to demonstrate and implement 'push' operations on a stack.*

Figure 8.2. Example LO and Q with high alignment (case 1)

The above is an example of aligned question. The concept coverage is not that critical here because we have only one question to cover an entire LO but the question should be within the scope of LO. The cognitive level should exactly match. The LO may cover many concepts and covering all of them in a single question may not be necessary. LO1 is covering all the operations such as 'insert', 'delete', 'search', 'traverse', etc. and the question is framed only on one operation.

Case 2: Single LO and Multiple Questions

There is a single LO against which the teacher has to write multiple questions. In this case, the teacher's scope to create questions has increased. Multiple questions covering the various aspects of given LO can be written against it. But the teacher has to comprehend the content coverage of LO properly. An individual question may be covering the LO partially, but the teacher has to ensure that the overall content alignment is high. Similarly, the overall cognitive level alignment should also be high. For example, for the LO2 displayed by the system, the teacher frames a set of questions as shown in Fig. 8.3.

> *LO2: Students should be able to implement different methods for traversing trees and graphs*
>
> *Q1: Write a recursive program to implement inorder, preorder and postorder traversal of trees*
> *Q2: Demonstrate the working of DFS traversal on a given graph.*
> *Q3: Compare DFS and BFS traversal algorithms for a graph*

Figure 8.3. Example LOs and Qs (case 2)

The above questions are aligned with the given LO. All the three questions are together covering all the contents covered by the LO. Hence, there is 100% content alignment. Overall cognitive level alignment is also high. The LO is at 'Apply' level and Q2 and Q3 are at level difference of one level lower or higher than LO with 'Understand' and 'Analyse' levels respectively. Q1 has the same cognitive level as LO.

Case 3: Multiple LOs and Multiple Questions

In this case, a set of LOs are given against which the teacher has to write multiple questions. This scenario typically represents the designing of AI. Every question written by the teacher should be aligned with atleast one LO in terms of its content and the cognitive level, so that the overall alignment is high. For example, for the LO3, LO4 and LO5 displayed by the system, the teacher writes a set of questions as shown in Fig. 8.4.

Students should be able to -

LO3: Implement various sorting techniques and searching techniques

LO4: Compare the performances of different sorting algorithms

LO5: Demonstrate the working of Hash search including the steps of creating hash table, collision handling and collision resolution techniques.

Q1: Write a typical recursive implementation of Quick Sort for arrays. The implementation uses last element as pivot.

Q2: What are the advantages and disadvantages of indexed sequential search?

Q3: Distinguish between insertion sort and bubble sort

Q4: Implement hash search. Using linear probing and quadratic probing insert the following values into a hash table of size 10: 99 33 23 44 56 43. Show how many collisions occur in each technique

Figure 8.4. Example LOs and Qs (case 3)

In this example, questions are completely aligned with the LOs for both content and cognitive level. Even though individual pairs of LOs and questions are not aligned, the overall alignment is high. The content alignment is high as for every LO, there is at least one question that is completely aligned with it. For example, Q1, Q2 and Q3 are highly aligned with LO3. Similarly, Q3 and Q4 are completely aligned with LO4 and LO5 respectively in terms of both content and cognitive levels.

TTM uses these three scenarios as the three stages in the teacher training. The training will start in stage1 and will go through every stage sequentially. The level of complexity increases in each stage. Unless, the teacher performs well in the present stage, he/she cannot move to the next stage. When the teacher completes her/his training by going through all the stages, he/she is expected to be trained in writing aligned question/questions against a given LO/LOs. The implementation details of different stages are explained in Section 6.3.

8.2.2 Feedback Mechanism

For training teachers, Whenever they attempt questions, they have to be provided with feedback about its alignment to the given LO/LOs. The major challenge is that the feedback has to be designed in such a way that it should be framed in an easy and understandable language and should explicitly tell the teachers what needs to be fixed (Song & Keller, 2008). Such type of feedback can effectively reduce the cognitive load of a learner, especially novice (Paas, Renkl & Sweller, 2003). We adopted this strategy in formulating the feedback for TTM.

Teacher entered questions and system displayed LOs are given to IQuE to calculate the measure of alignment between them. In addition to the alignment scores, IQuE module also provides following types of outputs:

- The common concepts in LOs and questions
- The concepts covered by LOs but not in questions
- The concepts covered by questions but not in LOs
- The cognitive levels of LOs and questions

The TTM takes these inputs and formulates the feedback in natural language (English).

Content alignment is assessed in 3 levels for the purpose of feedback. If the measure of alignment provided by the IQuE is in the range of 0-30 %, then it is LOW alignment. If it is in the range of 30-70 %, then alignment is MEDIUM, otherwise it is HIGH.

TTM first gives the overall alignment percentage in terms of content and cognitive level and then gives individual alignment between a pair of question and LO. For each pair, it conveys the information about the concepts missing in question but are present in LOs and concepts extra in question but are not part of LOs (out of scope concepts). It also indicate cognitive levels of each of them. For example for the LO1 and Q1 in Fig. 8.2, which is highly aligned, the feedback that will be generated by TTM is shown in Fig. 8.5.

Suppose, the LO and question are as shown in Fig. 8.6 for case 1.

1)There is high level content alignment.
2)Cognitive Level of LO and question is matching
3)Question is completely aligned to the LO.

Figure 8.5. Feedback generated for Example in Fig. 8.2

LO6: Students should be able to implement various operations on stacks.
Q1: Explain stack

Figure 8.6. Example LO and Q with low alignment (case 1)

Here, there is very low alignment between the question and LO6. Hence the feedback generated in this case will be as shown in Fig. 8.7.

1)There is medium level content alignment.
2)Cognitive Level of LO:"Apply".You have entered "Understand" level question
3)There are concepts from LO that are not covered in your Question:[Operation]
4)Modify the Question.

Figure 8.7. Feedback generated for Example in Fig. 8.6

The feedback can be advisory or corrective. The teacher receives a corrective type of feedback if there is a major misalignment, either the content coverage is very low or there is more than one level difference in the cognitive level of LOs and questions. The corrective action is mandatory. The teacher has to modify the question and resubmit and can take multiple attempts to refine the question.

If the feedback indicates that the alignment is in an acceptable range even though it is not 100% for both content and cognitive level, it means that the content alignment is HIGH and there is only one level difference in cognitive level alignment.

8.3 Using the TTM

Teacher goes through each stage of TTM sequentially. The level of complexity increases in each stage for the teacher. Following subsections explain the working of TTM in each stage.

8.3.1 Stage 1 of TTM

Stage 1 is one-to-one, i.e., teacher has to frame a single question for a single LO. Stage 1 is the entry stage for any teacher. The system flashes an LO randomly selected from a repository to the teacher. Teacher submits the question against that. From the alignment information, TTM gets alignment information from IQuE from which it generates the feedback and displays it to the teacher. Fig. 8.8 shows the snapshot of stage 1 of the TTM.

If the feedback is corrective, then the question has to be modified by the teacher and resubmitted. Instead, if a teacher receives an advisory feedback, the teacher can go to the next LO. The above cycle is repeated again for this and for all subsequent LOs. The teacher can continue this till all the LOs of stage 1 are exhausted. Otherwise, if teacher performs continuously well, he/she can opt to skip the remaining LOs and go to stage 2. Currently, this is pre-decided in the system but can be easily automated by doing the statistical analysis of session logs maintained for each teacher.

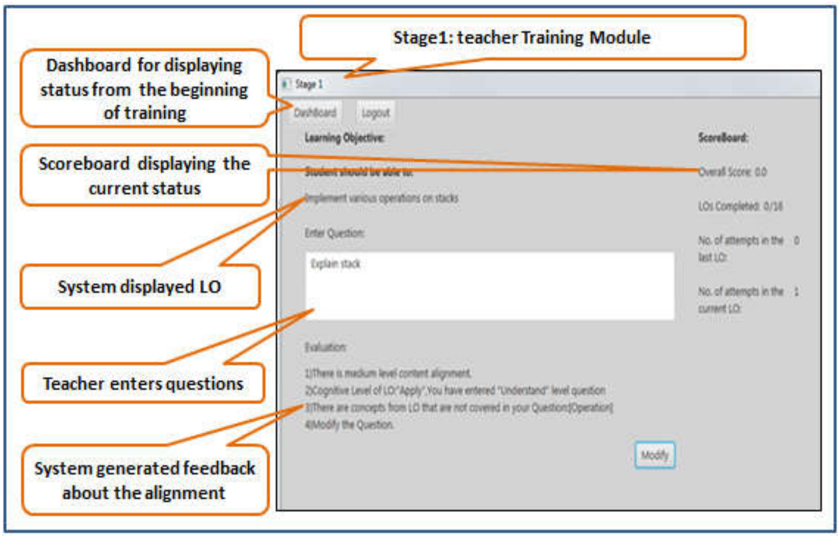


Figure 8.8. Snapshot of the stage1 interface of TTM

8.3.2 Stage 2 of TTM

Stage 2 is one-to-many, i.e., a single LO and multiple questions. The system displays a single LO to a teacher and the teacher is prompted to write a set of assessment questions aligned to that LO as shown in the example in Fig. 8.3. The feedback cycle is same as described in the previous section. However, TTM in this case will provide two levels of feedback. First, it will give the overall content and cognitive level alignment score. Then it will also give feedback about the alignment of individual questions against the displayed LO. Fig. 8.9 shows the snapshot of stage 2 interface of the TTM and Fig. 8.10 shows the snapshot of the feedback provided in stage 2 of TTM.

We have restricted the number of questions to be written by a teacher as 3 for implementation purpose. If more questions are to be written, it may kill the motivation of teachers to attempt more number of LOs.

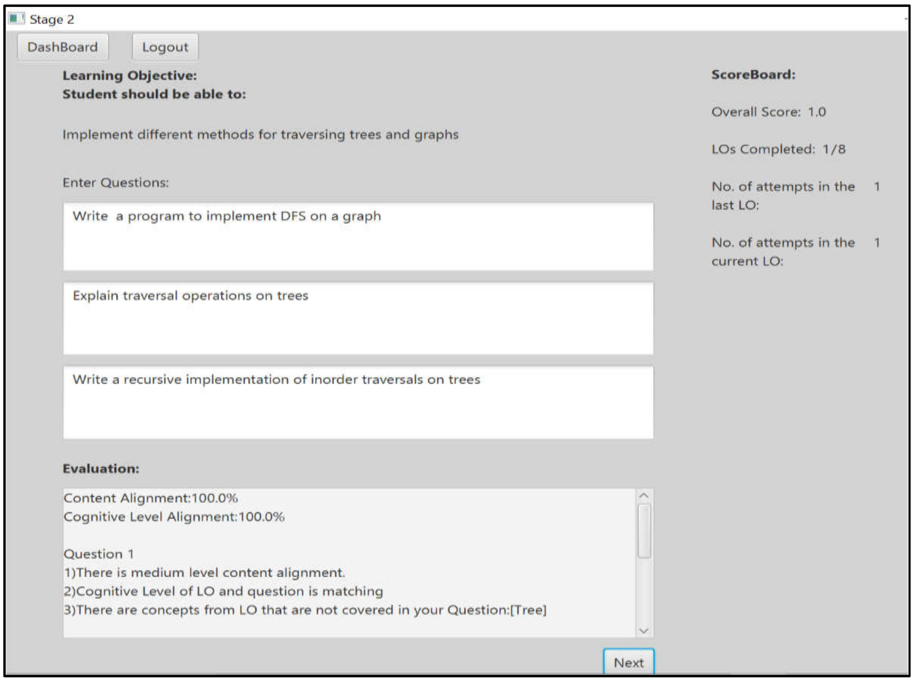


Figure 8.9. Snapshot of the stage 2 interface of TTM

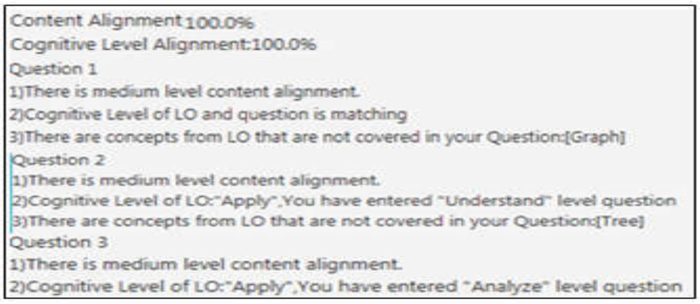


Figure 8.10. Feedback provided by TTM in stage 2

8.3.3 Stage 3 of TTM

Stage 3 is many to many, i.e., teacher has to frame multiple questions for multiple LOs. Then the alignment of the whole set of questions against the whole set of LOs is checked by IQuE and feedback is given. The IQuE provides the feedback about the overall alignment in terms of content and cognitive level and also for each pair of LO and question individually. Fig. 8.11 shows a snapshot of stage 3 of TTM.

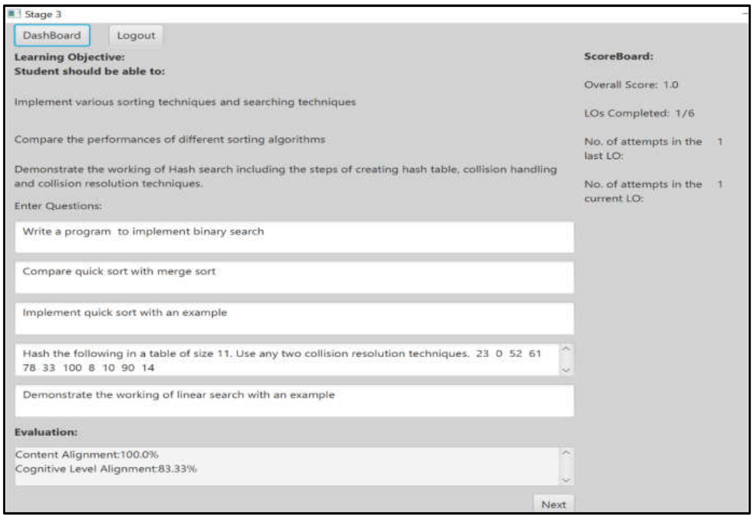


Figure 8.11 Snapshot of the stage 3 interface of TTM

Writing questions for stage 3 is like designing an AI for the course. Hence, when the teacher completes stage 3, s/he is expected to be a good AI designer in terms of LO alignment. For implementation, the system displays three LOs and the teacher has to frame five questions aligned to it.

8.4 Other Functionalities of TTM

The TTM provides various other functionalities such as dashboards and scoreboard. The scoreboard keeps track of the current status of the user (teacher). It displays information such as number of LOs completed, number of attempts taken for the last LO, number of attempts taken for the current LO and overall score representing the average attempts taken by the user for the attempted LOs. Such information motivates the teacher to perform well. It is also a step towards applying learning analytics to study the learning behaviour if some additional information is obtained from the teachers such as teaching experience, gender, qualification, etc. It may also help in gamification of the TTM in future.

It has a dashboard that displays the status from the beginning of the training. It stores information about the LOs that were displayed by the system, the questions that were written against it and the number of attempts taken for successful completion of each question along with the timestamp from the beginning for all stages. The system maintains the session for each user. If the user does logout before completing all the stages, the session is maintained and the user can continue at any other time from wherever he/she has left. Fig. 8.12 shows an instance of dashboard and scoreboard.

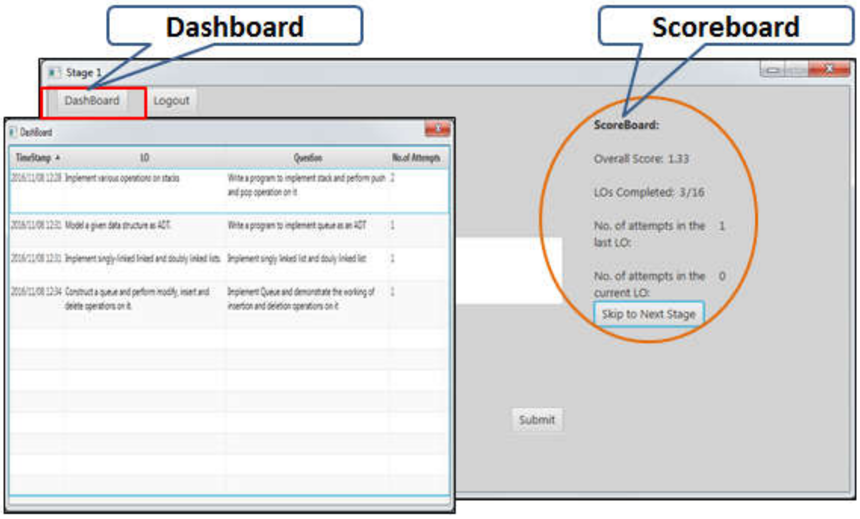


Figure 8.12 Scoreboard and Dashboard of TTM

8.5 Evaluation of TTM

TTM uses IQuE to check the measure of alignment and performance of IQuE has been rigorously tested with large number of samples (N =1000). It was found to be 91.2% accurate in terms of content alignment measure and 93.23% in terms of the measure of cognitive level alignment. Hence, we wanted to check whether teachers can be trained using IQuE. We were interested in knowing the teachers' perception about comprehension of the LO correctly and how much was the feedback useful in refining the question. After every successful attempt of writing aligned question in every stage, the teacher is asked to provide comments about the feedback generated by TTM. Figure 8.13 shows the snapshot of the feedback screen.

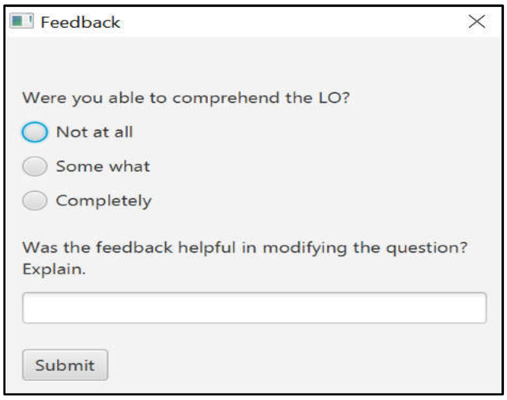


Figure 8.13 User comments interface

The first question in the form is a multiple-choice question in which they had to choose among the options indicating that they were not at all able to comprehend or somewhat able to or completely able to comprehend the LO. The second is an open-ended question about the usefulness of feedback. Whether they perceived it as useful or not, it is mandatory for teachers to provide comments about the feedback given by TTM for the alignment of their questions.

8.5.1 Pilot testing

Initially we gave the system to two teachers for pilot testing. They were told to explore all the functionalities of TTM followed by interview. Both of them were CS instructors who had an experience of more than 10 years and had taught *Data Structures* course multiple times. They were also education technology experts. According to them LOs were easy to comprehend. But they pointed out some issues in the wording of the feedback such as (i) Instead of reporting overall cognitive level alignment, individual cognitive levels of LO and question should be explicitly conveyed, so that teachers will know what is wrong with the cognitive levels of their questions (ii) Implementation of data structure involves the implementation of operations on them as well as their representation mechanism. The first issue required only the reframing the language of the feedback. But the second required some design changes. Whenever the LO included implementation of any data structure, concepts related to their representation and operations on them were added to list of concepts. Thus, if a teacher frames a question on representations and operations on data structures corresponding to such LO, it will not be considered as out of scope. For example,

LO: Students should be able to construct a queue for a given set of numbers

Q: Write a program to implement queue using linked list and perform modify, insert and delete operations on it.

Here, a representation mechanism such as linked list and operations such as insert, delete, etc. are part of implementation of queue. Hence, this question should be considered as aligned to the given LO.

8.5.2 Actual Testing

After incorporating all the changes suggested in the pilot testing, TTM was again given to (N=10) instructors for actual testing. The methodology followed was same but users were only CS teachers and not ET experts. This was done purposefully as the intended users of the system need not be ET experts. Teachers were told to explore all the three stages of TTM. It was compulsory to try atleast minimum required set of LOs in each stage i.e. five in stage 1, two in stage 2 and 1 in stage 3. They also have the option of trying as much as they want till the set of LOs are exhausted. For each successful attempt the teacher has to compulsorily write the comments on the feedback provided by the TTM. In stage 1, the 10 teachers together attempted 59 LOs. In stage 2 and stage 3, they attempted 21 and 18 LOs respectively. We got exactly same number of comments in each stage. The logs generated by the system for each user were analyzed including their open-ended explanations. The results are shown in Fig. 8.14 below. The values are normalized to percentages.

According to the teachers, there was no problem in comprehending the LO. 97% of the LOs in stage 1 were easy to comprehend followed by 86% in stage 2 and 83% in stage 3. As mentioned previously, the complexity of LOs increased in subsequent stages. Most of the comments indicated that the teachers were satisfied by the feedback and it helped them in refining the question. Some of the comments were explicit in mentioning that the teachers did not use the feedback as they successfully wrote the perfectly aligned question in one attempt itself and they just wanted to go to next LO. 22% and 24% of the times, multiple attempts were taken by the teachers to write aligned question in stage 1 and stage 2. But this was reduced to 11% in stage 3 as the teacher got more options to write questions as there are multiple LOs in this stage 3 and the teachers were trained more by the time they reach stage 3.

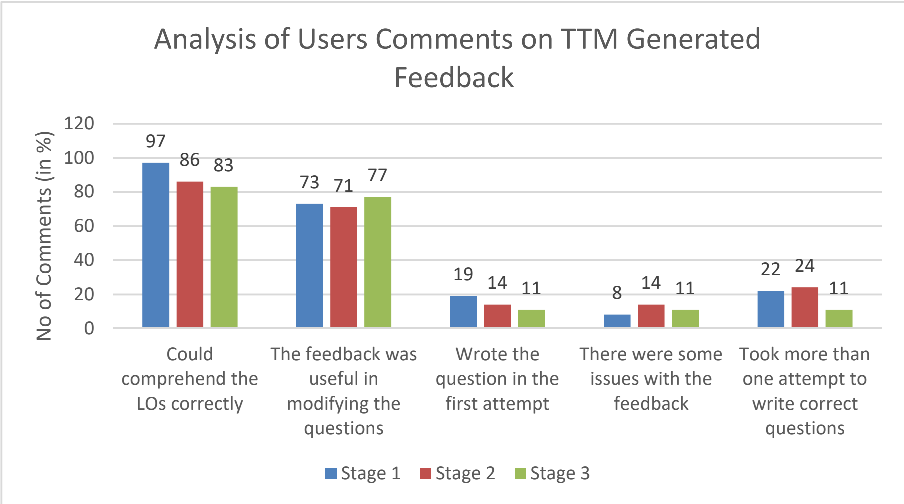


Figure 8.14 Analysis of user's comments on TTM Generated feedback

Some of the comments indicated that there were some issues with the feedback given by the TTM. They are (i) A correct concept was not recognized but it was because the concept was not present in the domain ontology and IQuE did not recognize it. (ii) The feedback given was "very low content alignment" for a fully aligned question. The reason was that IQuE was unable to recognize the implicit concept from a question incorporating real world problem. (iii) In one case, the user felt that the feedback was not clear. The user had to write question against the following LO: *Write programs that use different data structures.* IQuE recognized ***data structure*** and ***application*** (because of the word 'use') as concepts. This confused the teacher. (iv) Certain concepts were not recognized because of the problem which is discussed in detail in Section 6.2.4.1 of chapter 6.

One of the most encouraging oral comments we got from a teacher is "TTM is forcing me to write aligned question and motivating me to write more meaningful and better questions in every attempt".

8.6 Summary

TTM is a multistage training system that trains a user (teacher) to frame aligned assessment question against a given set of learning objectives. Three stages represent three levels of complexity in generating assessment questions aligned to LOs. Currently *Data Structures* course from second year engineering curriculum is chosen as the domain and all the examples are taken from that. TTM uses IQuE to get the measure of alignment and formulates a formative feedback and displays to the teacher. These formative feedbacks can be corrective or advisory. As per the feedback, the teacher will modify the question and resubmit. The teacher can take multiple attempts to refine the questions as per the feedback. To keep the interest and motivations, TTM gamifies the learning activities by using a scoreboard and dashboard. We have done a rigorous testing of TTM and the results are encouraging as the teachers found the feedback given by the system useful in refining the questions generated by them.

Chapter 9 describes the framework for automatic generation of AI and the corresponding AIGen tool.

Chapter 9

AI Generation (AIGen)

As mentioned in chapter 3, there are two broad approaches to better quality assessments. A framework for measuring AI quality is discussed in previous two chapters. In this chapter, the second way of improving the quality of AI is explained by providing a framework for automatic generation of AI. Fig. 1.2 of chapter 1 is reproduced here as Fig.9.1 for quick reference.

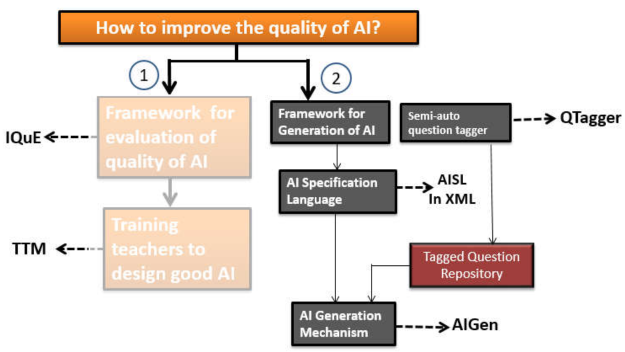


Figure 9.1 Different paths to improve the quality of AI

The framework has components such as, an interface where teachers can provide values for parameters of AIS, question selector which interprets the AIS and select the most appropriate questions from the repository to be put into AI and AI generator that generates AI from the selected questions. The characterization of AIS is described in Section 9.1. AI generation requires a pool from which it can select questions that adhere to AIS. A tagged question repository is needed for doing this. A semi-automatic question tagging system (QTagger) is built which can suggest appropriate tags for each question and then store in the repository. QTagger is explained in Section 9.2. As a proof of concept of AI generation framework, a prototype version of AIGen tool is built that takes a subset of AIS and uses a tagged question repository and generates an AI. The design and implementation of AIGen is discussed in Section 9.3.

9.1 AI Specification (AIS)

The selection of questions into an AI is guided by various parameters such as cognitive level, difficulty level, question type, subtopics to which the question belongs, marks, etc. These parameters are bound by a constraint of proper distribution of questions among different values of these parameters. These constraints are specified as per the requirement of the teacher. In order to automate the process of AI Generation, we need an interface where teacher's specification can be entered. AIS encompasses the structural as well as behavioural aspect of AI. It provides an overall outline or plan to guide the development of assessment tests, questionnaires, and procedures. It is crucial in determining the contents of an AI. An attempt is made to characterize AIS defining an XML based language for it.

9.1.1. Components of AI

In order to come up with a specification of AI, we need to understand structure of AI. A typical AI has a nested structure covering sections, questions, sub questions, etc. as shown in Fig. 9.2. Theoretically, the nesting is infinite. The sections are usually made based on some common properties of a set of questions in an AI. For example, all the questions in a section share a common LO or cognitive level or difficulty level. Some universities mandate the number of sections.

The structure of AI also incorporates information about the choice of options among questions or sub questions for the candidates. For example, a typical Mumbai university paper of 100 marks for engineering subjects says that "Question 1 is compulsory and out of remaining seven questions only five questions need to be solved". The proposed AIS framework should accommodate the specification of such patterns of options.

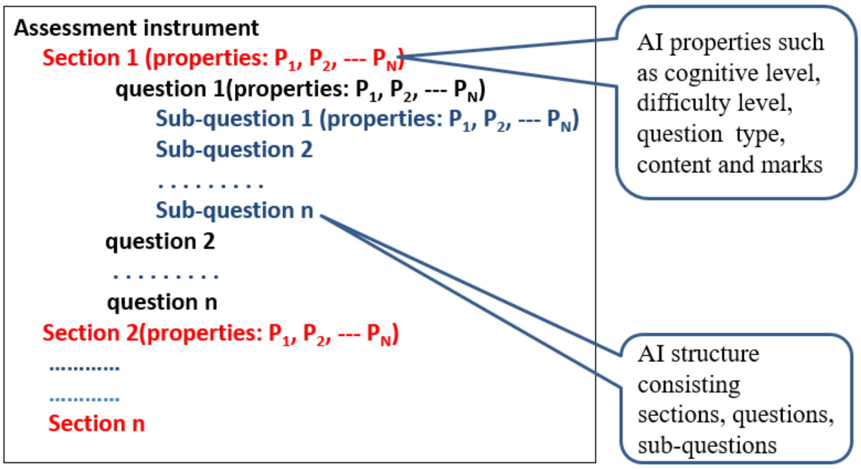


Figure 9.2 Components of AI

A number of implicit properties P_1, P_2, P_3, …, P_N are associated with an AI which include LOs, proportion of questions in different cognitive levels, proportion of questions in different difficulty levels, proportion of question types, distribution of questions among topics assigned for the assessment and marks associated with a question. All these properties can be associated with section, questions or sub questions in an AI. The AIS includes specification of structural as well as the property information of the AI.

One of the key properties is distribution of questions among topics and this is to be connected with syllabus of the course. Therefore, while designing the specification, one major concern is how to define the structure of syllabus. This aspect is described in detail in the next section.

9.1.2 The structure of Syllabus

Syllabus is used to provide a guideline for the instructor to teach a particular course. Syllabus for the same course may vary in different universities or when it is taught to students at different levels such as undergraduate, postgraduate, etc. The syllabus generally has a hierarchical structure. It is divided into different units, which contain major topics. Each chapter is further divided into topics and subtopics. A part of the syllabus for the *Data Structures* course of Computer Engineering branch of Mumbai University is shown in Table

9.1. The complete syllabus is given in Appendix B. Major topics in the syllabus are indicated with the number of lecture hours to be used to teach that topic. Intuitively this is also an indication of weightages in the examination.

There are dependencies and relationships between the different topics in the syllabus, which may not be explicit in this representation. Keeping this in mind, we explored the possibility of using an ontological structure for representing syllabus, like we did for AI quality evaluation framework. This helps in visualizing the syllabus as semantic network of concepts rather than a flat list of topics. The construction of ontology for *Data Structures* course is discussed in depth in Section 6.1.1. In addition to synonyms, the nodes in the ontology can be associated with the weight values indicating weightage of that topic.

Table 9.1. Syllabus format for *Data Structures* course

Data Structures		
S. No.	**Topic Description**	**Lecture Hours**
----	------	-----
5	**Queue:** ·The Queue as an ADT, Representation, Queue Operations, Circular and Priority Queues, Applications	03
6	**Linked List:** The Linked List as an ADT, Operation on Linked List, Linked Stacks and Queues, The Linked List as a Data Structure, Array implementation of Linked List, Linked List using Dynamic variable, Comparison of Dynamic and Array, Implementation of Linked List, Doubly Linked List, Circular Linked List	10
7	**Trees:** Basic tree concepts, Binary Tree Operations and Applications, Binary Tree, representations, Binary Tree Traversals, Threaded Binary Tree, The Huffman, Algorithm, Binary Search Tree Implementation, Expression Trees, Introduction of multiway tree (B-Tree, B+ Trees, AVL Tree)	12
-----	-------	------

Lecture hours mentioned in the previous paragraph can be transformed into suitable weights. These values are scaled in the range of 0-100 and are directly assigned as weights to the level 1 nodes in the ontology. We can say that 20% marks should be from the *Linked List* topic in the syllabus. Within this major topic, the examiners should have the flexibility of distributing the marks. Hence, instead of assigning a specific weight value to each node in the syllabus ontology, all the nodes below level 1 are given a weight range. Domain experts can provide weight ranges for all the subtopics as they have the knowledge about the difficulty and importance of each of the topics from the syllabus

A range of 2-4 implies that, in the AI, questions from this topic will account for minimum 2 marks and maximum 4 marks. The ranges may be associated with some properties. For example, the upper threshold of the individual child node does not exceed the upper threshold of parent node. If (x_1-x_2) is the range given in node 1 and (y_1, y_2) and (z_1, z_2) are the ranges given for its two child nodes, then maximum of values (y_2, z_2) at child nodes should not exceed the maximum value of the parent node (x_2).

9.1.3 Structure of AIS

In this section, the structure of AIS is detailed based on components of AIS and syllabus discussed in previous sections. AIS includes information about the structure of AI as well as the properties of questions in it. Structure part contains division into sections, sub sections, questions within them, level of nesting, information about choice of options, etc. The properties associated with any structural component include, LOs, proportion of questions in different cognitive levels, proportion of questions in different difficulty level, proportion of objective and subjective questions, distribution of questions among topics assigned for the assessment and marks associated with a question as discussed in Section 9.1.1. XML representation for each is introduced in the following sub-sections in a bottom up fashion.

The components of the AIS have an inherent hierarchical relationship among them such as sub-section is a subset of section or question type can be objective and subjective. They also share a common set of properties. Each property is specified with a range indicating that value should not be lower than minimum value and not exceed the maximum value of the range. We introduce a bound element to capture such value ranges which contains the upper bound (ub) and lower bound (lb) of its distribution. XML based representation was found to be suitable and useful for all the components of AIS. The hierarchy of AIS components is shown in Fig. 9.3.

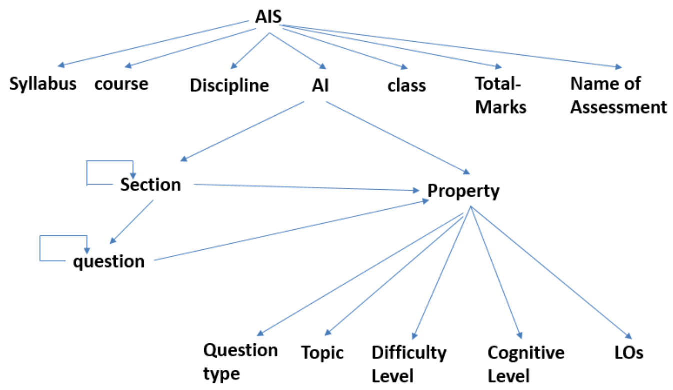


Figure 9.3 Hierarchy of AIS components

9.1.3.1 Bound Element

As mentioned before, most of the properties can be specified by using ‘ub’ and ‘lb’ that represent values in percentages. Consider a case where, in an instrument, questions of low difficulty should not be less than 20% and not more than 80%. Bound element contains two attributes ‘ub’ and ‘lb’ and this is used by all the property elements. If no values are specified then it is taken as <0-100>%.

```
<! ELEMENT bound>
   <! ATTLIST ub #CDATA>
   <! ATTLIST lb #CDATA>
```

9.1.3.2 The LO element

For every assessment, there are LOs that are taken from the set of LOs of the course or the module. The questions in the AI should match with the LOs of the portion assigned for the assessment (J. Biggs, 2003). There should be questions catering to every LO. The proportion is dependent on the importance of LO. The proposed AIS provide this information about the distribution of questions among a set of LOs. The notion of LO element is introduced which is a vector of n elements, each corresponding to one objective. For each element, there are a set of range values. Such a structure can be attached to anywhere in the hierarchy of section, subsection, question or sub questions. For example, the user may say that <20-30>% questions should meet the 2nd objective. DTD and the example XML code is given below. Table 9.2 shows the distribution of questions.

DTD:

```
<! ELEMENT LOs (LO)*>
```

```
<! ATTLIST LOs no_of_LOs CDATA #IMPLIED>
<! ELEMENT LO (bound)>
<! ATTLIST LO_statement CDATA #IMPLIED
```

XML Code:

```
<LOs>
    <LOs no_of_LOs =2>
     <LO>
        <LO_statement = “Students should be able to implement various searching algorithms”
/>
        <bound lower_bound="20" ub="50">
      </LO>
      <LO>
            <LO_statement = “Students should be able to implement various sorting
algorithms”>
            <bound lb="30" ub="80">
       </LO>
  </LOs>
```

Table 9.2: Distribution of questions among LOs

LOs	LO1	LO2
Lb	20	30
Ub	50	80

9.1.3.3 The Cognitive_Level element

The AI should contain the desired proportion of questions in different Cognitive levels. The AIS should have the mechanism to incorporate this information. The proportion is dependent on the purpose and objectives of assessment. Bloom’s taxonomy is the most widely used basis for educators to classify questions in different cognitive domains. In this, cognitive domain consists of six levels of cognitive skills: Recall, Understand, Apply, Analyse, Evaluate and Create (Krathwohl, & Anderson, 2002) (Junoh et, al., 2012). While designing an AI, the questions should have the cognitive level that matches with the cognitive level of LOs to which it is associated. For example, users can say that section A should contain only recall level questions and section B should have uniform distribution of questions in the remaining cognitive levels.

The cognitive_level is a composite element containing all the six levels of Bloom’s as elements. The constraints for all the cognitive levels are specified in terms of their bound values and should be within the limits of <0-100>%. For example, the user can put the restriction that instrument should have <30-50>% Apply level questions and others can be anything. Table 9.3 shows the distribution of cognitive level for the example.

DTD:

```
<! ELEMENT cognitive_level (create|evaluate|analyse|apply|understand|recall)*>
         <! ELEMENT recall (bound)>
        <! ELEMENT understand (bound)>
        <! ELEMENT apply (bound)>
        <! ELEMENT analyze (bound)>
        <! ELEMENT evaluate (bound)>
        <! ELEMENT create (bound)>
```

XML Code:

```
<cognitive_level>
        <recall>
                <bound lb="0" ub="10">
        </recall>
        <understand>
                <bound lb="20" ub="40">
        </understand>
        <apply>
                <bound lb="50" ub="50”>
         </apply>
        <analyse>
                <bound lb="30" ub="10">
        </analyse>
</cognitive_level>
```

Table 9.3: Distribution of questions in Cognitive level

C-Level	Recall	Understand	Apply	Analyse
Lb	0	0	30	0
Ub	100	100	50	100

9.1.3.4. The Difficulty_level element

Information catering to distribution of difficulty level of question across different parts of AI is part of proposed AIS. For example, we may say that at least 20% of questions should be of low difficulty level and not more than 30% should be of high difficulty level or section A should be predominantly of low difficulty level, etc. The framework should enable such specification to be represented.

A well-balanced instrument should cater to all type of students. The difficulty level of the instrument should generally be such that majority of the students can pass the test but it should also contain some challenging questions to inspire the high performers.

The constraints for all the difficulty levels are specified in terms of their bound values and should be within the limits of <0,100> %. The user can clearly state that “I want section 1 to have (0-20) % low and (20-70) % medium difficulty level for the questions”. Table 9.4 shows the distribution of difficulty levels. Limits for High difficulty level is not specified by the user and it is taken as <0-100> %. Internally, the system will validate the range for consistency.

The DTD below shows the definition for difficulty level with three possible values easy, medium and high with the bound for each denoting relative coverage of easy, medium and high difficulty questions in AI.

DTD:

```
<! ELEMENT difficulty_level (high | medium | easy)*>
    <! ELEMENT easy (bound)*>
    <! ELEMENT medium (bound)*>
    <! ELEMENT high (bound)*>
```

In the XML code, an example is given where lower and upper bounds for easy, medium and high difficulty levels

XML Code:

```
<difficulty_level>
    <easy>
        <bound lb="0" ub="10">
    </easy>
    <medium>
        <bound lb="20" ub="60">
    </medium>
    <high>
        <bound lb="20" ub="50”>
    </high>
</difficulty_level>
```

Table 9.4 Distribution of questions in Difficulty levels

D-Level	Low	Medium	High
Lb	0	20	20
Ub	10	60	50

9.1.3.5 The Question_type element

Questions are categorized as either objective or subjective. Objective assessment is a form of questioning which has a predefined set of correct answers. Objective question types include true/false answers, MCQs, and matching questions. Subjective questions include extended response questions, short answers and essays. There should be wide variety of questions in an AI, that caters to different types of students (Anderson & Morgan, 2008). The AIS should have the facility of representing the information about the desired distribution of questions into different question types. For example, users may state that 50% of the questions should be objective type. Question type can be basically divided into subjective and objective types.

DTD:

```
<! ELEMENT question_type (objective|subjective)*>
```

Each element of the objective type and subjective type in turn has a list of sub elements corresponding to the different type of questions within that. Against each of these, a range is attached. The DTD below shows the definition for objective and subjective question type with possible values for each.

DTD:

```
<! ELEMENT objective (MCQ|fill_in_the_blanks|match_the_pairs|true_false)*>
        <! ELEMENT MCQ (bound)>
        <! ELEMENT fill_in_the_blanks(bound)>
        <! ELEMENT match_the_pairs(#PCDATA)>
        <! ELEMENT true_false(#PCDATA)>
<! ELEMENT subjective (long_answer|short_answer|very_short_answer)*>
        <! ELEMENT very_short_answer (bound)>
        <! ELEMENT short_answer (bound)>
        <! ELEMENT long_answer (bound)>
```

At the highest level, the AIS provides the information that the AI should have <0-30> % objective and <0-70> % subjective questions. There can be further constraint that the 30% objectives should include 20% MCQs, 5% fill-in-the-blanks and 5% True/False questions. Similarly, in subjective part, there should be 20% essay type, 30% short answer and 20% long answer. Table 9.5 gives the distribution of questions in different question types. The XML code below shows the example where lower and upper bounds for subjective and objective question types.

XML code:

```
<question_type>
        <subjective>
                  <bound lb="0" ub="70">
            <very_short_answer>
                  <bound lb="0" ub="20">
           </very_short_answer>
            <short_answer>
                  <bound lb="0" ub="30">
           </short_answer>
            <long_answer>
                  <bound lb="0" ub="20">
            </long_answer>
      </subjective>
          <objective>
              <MCQ>
                  <bound lb="0" ub="100">
              </MCQ>
         </objective>
     </question_type>
```

Table 9.5 Distribution of questions among different Question_type

	Objective <0-30>%				Subjective <0-70>%		
	MCQ	FLB	Match	MR	VSA	SA	LA
Lb	0	0	0	0	0	0	0
Ub	20	5	5	30	20	30	100

9.1.3.6 The Content Element

Questions should be among the subtopics within the main topic decided for the assessment Agarwal. (2008) (CBSE Teachers' Manual, May 2017) (Chan Kan Kan, 2010) (Ahmed, Aziz-un-Nisa & Tayyaba, 2013). The proportion of distribution can be based on the weightage assigned to those topics in the syllabus. The topics usually have hierarchical relationship as discussed in Section 9.1.1 and are mapped to the nodes of the domain ontology. If a topic is selected for assessment, then the complete subclass hierarchy below it in the ontology tree is part of the portion selected for assessment. A topic contains an element Bound which is used to provide the proportion of questions catering to that topic. For example, if the content of assessment represents the topic stack, then user may say that <20-40>% questions should be from that topic. The DTD below shows the definition for content element with possible values for each topic and has attribute values for topic_name and bound.

DTD:

```
<! ELEMENT content(topic)*>
 <! ELEMENT topic (bound)>
<! ATTLIST topic topic_name CDATA #IMPLIED>
```

In the XML code, an example is given where there are 2 topics under the content element with topic names “stacks” and “queues’ with its lower and upper bound values.

XML code:

```
<content>
        <topic>
                <topic name="stacks">
                <bound lb="30" ub="50">
        </topic>
                <topic name="queues">
                <bound lb="50" ub="70">
        </topic>
</content>
```

Table 9.6 Distribution of questions among different Content type

Content	Stacks	Queues
Lb	30	50
Ub	50	70

9.1.3.7 Marks

Each question in the instrument is associated certain marks. The teacher can specify the marks for the complete instrument like “assessment should be conducted for a total of 50 marks”, or for each section “section A should be for 20 marks and section B should be of 30 marks”, or question wise or sub question wise. The marking scheme also has to take care of options in the AI.

```
<! ELEMENT marks #PCDATA>
```

9.1.3.8 The Property Element

The Property element is an aggregation of all the properties of question or questions in AIS. This element can be associated with a section, question or sub question depending on the user requirement.

```
<! ELEMENT property (question_type|content|difficulty_level|cognitive_level|LOs )*>
```

9.1.3.9 The Questions Element

The questions element can include an individual question or one or more sub-questions within it. Each question has attributes such as a problem statement, marks associated with that question, diagram or figure as part of the question. The property element can also be associated with every question. The DTD below shows the definition for questions element

DTD:

```
<! ELEMENT questions (property| (questions)*)>
<! ATTLIST questions questions_count CDATA #IMPLIED >
<! ELEMENT question>
        <! ATTLIST question problem_statement CDATA #IMPLIED>
        <! ATTLIST question marks CDATA #IMPLIED >
        <! ATTLIST question diagram CDATA #IMPLIED >
```

9.1.3.10 The Section Element

The section element can contain zero or more sections or a set of questions within it. Each section has attributes such as name of section, and the number of sub sections within a section. Sometimes section may not have sub sections but directly contain list of questions. The DTD below shows the definition for section element.

DTD:

```
<! ELEMENT section (property|(section|questions)*>
     <! ATTLIST section s_name CDATA #IMPLIED >
        <! ATTLIST subsection_count CDATA #IMPLIED>
```

9.1.3.11 The AI Element

AI element defines the structural components of AIS. It contains section element and property element which are defined earlier.

<! ELEMENT AI (section|property)>

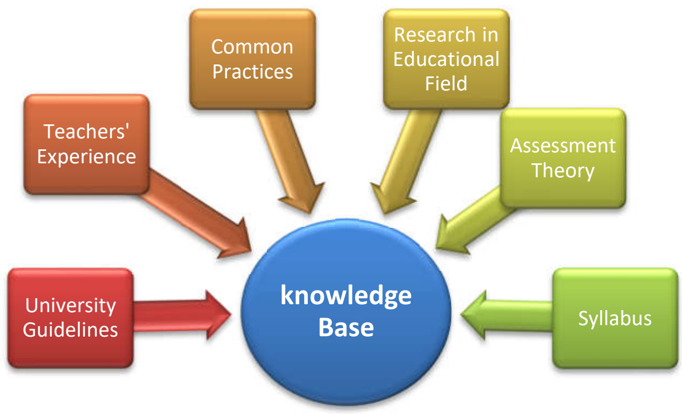
9.1.3.12 The AIS Element

The element AIS is the complete specification of the AI. It includes AI element which is defined earlier. It also has elements that provide the general information about the assessment

<! ELEMENT AIS (AI|syllabus|total_marks|discipline|class|course|name_of_assessment)>

9.1.4 The Knowledge Required to Generate AIS

Teachers may find creating AIS difficult and cumbersome. One solution for this is that teachers can provide minimal information needed to create the AIS. Then a system can be built that can automatically generate full AIS. The initial information such as Discipline, Class/Semester, Duration, Total Marks and Topic of Assessment are supplied by the teacher in the assessment request and is directly substituted in the AIS. The additional information such as proportion of objective and subjective questions, proportion of questions in different cognitive and difficulty levels, assessment objectives aligned with the LOs, distribution of contents across the AI may not be available directly. If the teacher provides everything, then the AI takes it as it is. However, if the teacher leaves them open, then how to generate adequately detailed AIS? In this situation, there is a need for additional support of various knowledge sources. Most of the knowledge comes from the



assessment theory, university Figure 9.4 Knowledge Sources guidelines, teachers' experience, common practices and research in educational field. This is shown in Fig. 9.4. These are briefly explained below. The knowledge extracted from these sources is converted into suitable representation such as rules, equations, functions, etc. and stored in the knowledge base. This can be then used to expand and fill in the partially complete specification.

9.1.4.1 University Guidelines

In most cases, the university guidelines mandate the structural information such as nesting in AI covering sections, questions and sub questions, total marks, section wise, or question wise marks and information about choice of options.

- A typical Mumbai university paper of 100 marks for engineering subjects says that "Question 1 is compulsory and out of remaining seven questions only 5 questions need to be solved".
- Pune University's guidelines for the Code of Conduct of the theory examination say that out of total 100 marks there should be online examination of 50 marks. It also says that there should be 1 mark, 2 marks and 3 marks questions for the online examination and they should be selected from a given question bank. It gives information about the marks distribution across various topics of the syllabus.
- The syllabus contains information about the weights assigned to topics and number of lecture hours allotted to teach that topic. These are then assigned as weight ranges to every node in the ontology. We are representing the syllabus in the form of ontology as explained in Section 6.1.1 of chapter 6. This can be used to determine the distribution of questions among different topics.

9.1.4.2 Teachers' Experience

Teachers' experience plays a major role in the assessment process. Teachers' knowledge also gets enriched through experience gathered over a period of time which is intuitively used to determine many aspects of assessment such as the students level of understanding, time required to answer a question, the complexity of a question, etc. For example, the teachers formulate AI with a variety of questions like descriptive, analytical, problems, etc. to cater to all types of students

9.1.4.3 Common Practices and Research in educational field

There are many studies in the field of assessment theory that result in best practice recommendations.

- The research in educational assessment says that the AIs need to be aligned with the LO of the course (J. Biggs, 2003) (Krathwohl, & Anderson, 2002). Cognitive level for an instrument is directly influenced by the cognitive level expectation on various parts of the syllabus. In a course scenario, this is partly reflected by the course syllabus and LOs.

- The choice of question type (essay, short answer, MCQ) is related to the nature of the content, the test objectives and the time available to write questions (Agarwal, 2008). According to LO, if students are required to demonstrate the depth of understanding of the course, the essay type questions are more suitable (Davis, 2002).

- Difficulty of the questions can be planned in conjunction with the objectives for the course. The harder questions have more influence on the total score. If these questions are concentrated in one content area, that area will unequally weigh the total score. Avoid tests that are too difficult, for even the best students; these tests destroy motivation.

- Difficulty level distribution of questions can be made a function of average level of students. Normally the level of students follows a bell curve with few students below average level of intelligence, more number of students in average level and again less number of students in high level. This can be used to find the distribution of questions among the difficulty levels. If the class average is significantly lower in one class than other, then use higher proportion of less difficulty questions. Similarly, if the class average is higher, then use higher proportion of high difficulty questions.

- Syllabus also facilitates to develop the LO of the course from which chapter wise and topic wise LOs can be derived. In an AI, questions can then be designed to meet each of the LOs.

9.1.5 Evaluation of AIS

Once the framework was finalized to some extent, a pilot study was conducted with two science teachers of Atomic Energy Central School (AECS) to test its usability and adequacy. AECS are managed by Atomic Energy Education Society, an Autonomous Body, under the Department of Atomic Energy, Govt. of India and are affiliated to CBSE. Teachers were teaching the science subject in ninth standard and had an experience of more than 10 years. School teachers were preferred as they are more trained and more comfortable with AIS terminologies and models. Teachers in higher education people do not necessarily follow that kind of methodology. CBSE has a Continuous and Comprehensive Evaluation (CCE) manual that provides teachers with rich and meaningful information about putting into practice an assessment model that is continuous and incorporates scholastic and non-scholastic aspects of learning (CCE, 2017). Under the tools and techniques of evaluation, it provides a detailed explanation of what constitutes a good question in an AI and different parameters to be considered.

CBSE also has a detailed assessment plan in the form of a Blueprint that is used to develop an AI. All the instruments adhere to this blueprint which is sent by CBSE across all schools in India. Blueprint involves dimensions such as content, type of question and number of questions.

A sample AIS for IX standard science subject was created for this study retaining all the existing dimensions and specifications of the blueprint and adding cognitive level, difficulty level, LOs and content dimension to a greater depth. In the AIS, it was stated that there should be two sections in the AI; A and B. Section B should contain only MCQs. Section A should be further divided into three sub sections A1, A2 and A3. A1 should only contain VSA (very short answer), A2 only SA (short answer) and A3 only LA (long answer) type of questions.

The AIS was given to two teachers selected from two different schools (Appendix F). AIS also had a supporting document that explained the educational technology terminologies used in it. Teachers were told to manually generate an AI within the given specifications of AIS. They were given a week’s time to complete this task. They were also told to note down all the difficulties that they faced while interpreting the contents of AIS and generating AI. Both teacher 1 and teacher 2 completed their work within the specified time. After this, they were interviewed separately about their experiences of using AIS. The interview was based on 8 open ended questions (Appendix G).

The responses of both the teachers were recorded and later analyzed. Both the teachers felt that *"AIS was not ambiguous and not difficult to understand. Only the unawareness of Bloom's cognitive level created some confusion. But the explanation given at the end with corresponding example for each cognitive level made it little simpler"*. Teacher 1, before putting each question in the assessment instrument, first checked in the AIS whether it meets all the constraints specified in it. Hence, once the question is put in the AIS it is final. Teacher 2 perceived the information in the AIS and then generated blueprint with all dimensions. Some of the important feedbacks provided by teachers are reported here.

"Linking of contents to type of questions, difficulty level and cognitive level is not directly provided in AIS as in the Blueprint. Only individual distribution was given in AIS". Teachers felt if this was provided, it would have been easy to put questions in an AI. When asked whether this will not restrict their freedom to select questions in instrument they were of the opinion that they already have the content wise freedom i.e. Given a topic at a broad level they have the freedom to select the questions from the subtopics.

"Science includes physics, chemistry and Biology which are normally taught by different teachers. Hence, if there is some ordering of questions and grouped accordingly in AIS, it is easy for teachers to correct".

The sub section within a section can be utilized to group the questions as each subsection can represent a group.

One teacher converted the information present in the AIS into a blueprint and then generated an AI. Hence, formally introducing an intermediate level called blueprint between AIS and generation of AI needs to be investigated. Blueprint is more user friendly and understandable than the machine generated AIS.

The next requirement is of a good repository with questions suitably tagged. This is explained in detail below in Section 9.2.

9.2 Generation of Tagged Question Repository

The AIGen selects questions from a repository to be placed into AI. In order to select, we require questions that are tagged with the values same as that of parameters in AIS.

Manually tagging questions is a cumbersome and laborious task. The person who tags needs to be a domain expert as well as educational technology expert. Fully automatic tagging system is also very challenging as questions are framed in natural language. All the inherent ambiguities and issues similar to that discussed in the design of IQuE are applicable here as well. Hence, a semi-automatic tagging system is built that takes a question from the user and identifies various types of tags associated with it and stores in QR. It is a semi-automatic system as it uses human intervention to increase the accuracy of tagging. We call this as QTagger. The architecture of Qtagger is shown in Fig. 9.5.

QTagger takes a question as input from the user and attempts to identify various tags such as Bloom's level, type of question, difficulty level, and the content or topic of the question. If the teachers are not satisfied with some of the suggested tags, manual editing facility is provided to modify the tags. Sometimes, nature of questions makes the automatic tagging impossible for the system. To solve this problem, it provides an option where teachers can reformat the questions at the entry level itself so as to facilitate more accurate tagging.

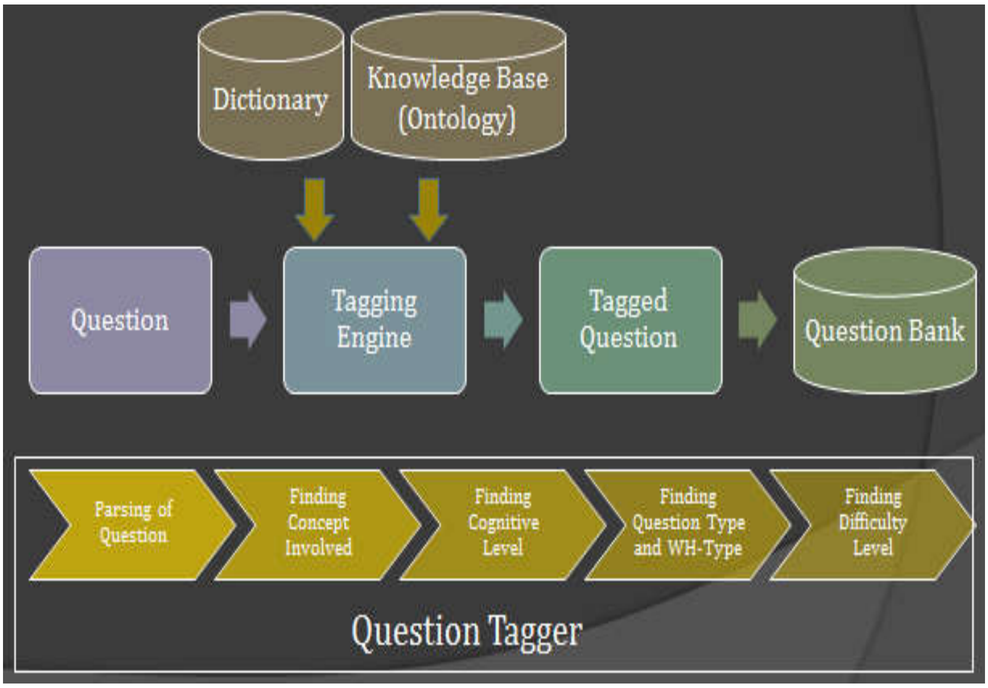


Figure 9.5 Architecture of QTagger

Four main tags are considered namely, cognitive level defined by the Bloom's taxonomy, difficulty level, type of question and the name of content or topic from the syllabus as explained before. The dictionary stores keywords that includes verbs associated with each cognitive level defined by Bloom's taxonomy, different question types described earlier, and phrases that are normally present in the questions specific to a particular domain. The phrases can be "Write a program", "Write an algorithm", "Write a short note on", "Provide an example", etc. Such phrases are stored in the dictionary along with its cognitive level. N-grams algorithm is used to identify token with multiple words. The dictionary is used to identify cognitive level and question type. For content identification, only ontology is used. For the various tags the value range is as shown in Table 9.7.

Table 9.7 Question tags and its values

Tags	Values
Cognitive Level	Six levels of Bloom's taxonomy: Recall, Understand, Apply, Analyze, Evaluate, Create
Question Type	Objective: Fill-in-the-blanks, Multiple-choice, Match-the-following, True-false, Answer-in-one-word
	Subjective: Short-note, Differentiate/Comparison, Program-implementation, Short-answer, Long-answer
	WH-type: Why, When, What, Who, Whom, How
Content	Topics and subtopics from the syllabus that forms the node names of the ontology.
Difficulty Level	Low, Medium, High

9.2.1 Cognitive Level Identification

Bloom's taxonomy forms the basis for cognitive level identification of a question. Every level of Bloom's taxonomy namely, Recall, Understand, Apply, Analyze, Evaluate and Create is associated with an elaborate set of keywords (Sakai LMS, 2017). These keywords are stored into a dictionary. The tokens are matched to the keywords in the dictionary and accordingly its cognitive level is identified. For example,

*Q1: Draw a binary tree for the given expression A*B – (C+D) * (P/Q)*

'Draw' is a keyword associated with Bloom's level 'Apply'. Hence, the cognitive level of question is identified as Apply. If two tokens match with the keywords of two different Bloom's level, then the higher level is chosen as cognitive level of the complete question.

For example,

Q2: State the difference between arrays and linked lists

The keyword '*State*' is at Recall level and '*Difference between*' is a phrase stored in the dictionary at Analyze level. Hence, the cognitive level of the question is identified as Analyze.

9.2.2 Question Type Identification

The dictionary stores many types of questions in it. Basically, they are broadly categorized as subjective type, objective type and WH-Type. Objective type is further classified as Fill-in-the-blanks, Multiple-choice, Match-the following, True-false, Answer-in-one-word, etc. (Handbook on Testing and Grading, 2017). Similarly, subjective questions are classified as short-note, differentiate/Comparison, Program-implementation, short answer, long-answer, etc. WH-Type questions are of type How, Why, When, What, Who and Whom. Question type is decided by matching the keywords extracted from the question with the keyword list in the dictionary. Consider a question

Q: Write a program to implement quick sort.

The keywords extracted will be '*write a program*' and '*implement*'. "*Write a program*" refers to Program Implementation. Question is of type "Program Implementation", Question category is "Subjective" and since there is no WH-Type words, it is "*Not a WH-Type*".

9.2.3 Content Identification

To identify the content/ topic of question, we have to map the concepts from a question to the contents of the syllabus. But there may not be direct matching of concepts in the syllabus. We have represented the syllabus using Ontology. The process of extracting concepts from question and mapping them to the nodes of the ontology is same as that is done in IQuE which is explained in Section 6.1.2.

9.2.4 Difficulty-Level Identification

From the literature, we found that the difficulty level of a question is based on using Cognitive Level, Concept Involved, Concept Difficulty and Question Type.

- Questions pertaining to higher cognitive level of Bloom's taxonomy are considered to be more difficult than the questions with lower cognitive level (Amer, 2006). The cognitive levels are coded to get numerical values which form the first parameter to calculate the difficulty level.

- Similarly, questions with multiple concepts are considered to be more difficult than questions with less number of concepts (Marbach & Sokolove, 2000) (Denny, Reilly & Simon, 2009). Hence, number of concepts form the second parameter.

- Moreover, concepts themselves are not of same difficulty levels. Some are more difficult than others. Hence, in our case, every concept is classified into one of the four levels of difficulty assigned by the domain expert as shown in Fig. 9.6. The lowest level of difficulty has the value 1 and the highest level has value 4. For example, the concepts such as *Arrays, Binary Tree, Minimum Spanning Tree and AVL Trees* are considered to be in increasing order of difficulty. If the question contains more than one concept, the highest value of difficulty level is taken among the difficulty level of all the concepts. The difficulty of the concepts forms the third parameter.

Figure 9.6 Concept Difficulty (given by domain experts)

The fourth parameter is question type. Subjective type of questions are considered to be more difficult than objective type questions. (Denny, Reilly & Simon, 2009).

Hence, the difficulty level of a question is the weighted addition of the values of all these parameters as shown in Fig.9.7. Higher the value, higher is the difficulty level. Based on these criteria, we have provided the estimate for the difficulty level.

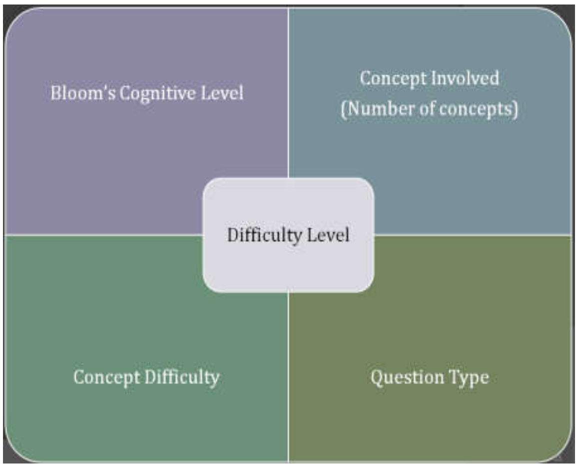


Figure 9.7 Formulation of Difficulty Levels

Consider a question

Q: Differentiate BFS and DFS

1) Cognitive level of the question: Analyze (Level 4 of 6 levels)

2) The number of concepts involved in the question: 2 (BFS, DFS). Maximum number of explicit concepts involved in a question is considered as 4

3) Question category: Subjective (2^{nd} out of 2 categories where objective is first category and subjective is the second category)

4) Concept difficulty: Both the concepts are in the level 3 of concept difficulty levels.

Hence, the calculated difficulty level =

$$Difficulty\ Level\ of\ question = \frac{4}{6} + \frac{2}{4} + \frac{3}{4} + \frac{2}{2}$$

= 2.92 (Moderate)

The difficulty values are considered as

- Easy ($0 \leq$ *Difficulty Level* < 1.5)
- Moderate ($1.5 \leq$ Difficulty *Level* < 3)
- High ($3 \leq$ Difficulty *Level* < 4)

The values of the difficulty levels are stored as annotations for each node in the ontology.

9.2.5 Implementation and Testing

The system is implemented using Java programming language. The ontology is created using protégé 4.3 application. The Protégé OWL file is parsed by the OwlParser class of Java. N-Grams algorithm is implemented to extract multi worded concepts from a question.

User Testing: To investigate the accuracy of the tagging, an accuracy-test is performed, where two CS education researchers tested the tags generated for a set of 50 randomly picked Data Structure questions. It was found that the accuracy with respect to the cognitive level annotations was 78%; with respect to question type annotations was 90%. The difficulty annotations were 93% accurate, and the content identification was 87.5% accurate. The inter-rater reliability was 100%.

In order to evaluate the usability and user friendliness, we gave the system to 11 users to explore and use the system. Each user is a CS instructor with an experience of at least 10 years. The user testing involved two phases of activities: In the first phase, users executed the Semi-Automatic Question Tagger on their computer. They were given simple set of instructions such as -

1) Explore each components of the system
2) Edit the existing Data Structures Questions"
3) Add your own Questions
4) Cross check the auto-generated annotations

In the Second phase, users were given a set of ten questions from the questionnaire prepared to test the System Usability (SUS) (Brooke, 1996) score of the Question Tagger. Each question of the SUS questionnaire is associated with an open-ended feedback question. For each of the SUS question, the users were asked to write an open-ended response to explain the justification of selecting a specific Likert's scale score for a question.

Only a preliminary evaluation is done with few users. While the N=11 for the SUS data is not sufficient for statistical significance, we have attempted to triangulate the scores using open-ended responses and analyzed them to validate our inferences. All the open-ended responses from all the candidates were qualitatively analyzed and coded to test the usability of the QTagger. The test revealed that 8 out of 11 users found the system to be useful. Users perceived the system to be user friendly. Some of the positive features frequently cited were:

- the easy GUI
- nil or least requirement of technical knowledge to use the system
- properly structured components of the system

Most prominent benefits of the system as reported by the users are:

- Setting questions as per student's level
- Saves time
- Coverage of important metadata associated with a question

Users also perceived that the system correctly annotates the questions consistently and unambiguously.

9.3 AI Generation system (AIGen)

AIGen takes teachers request of assessment using the interface provided for entering the values for parameters of AIS, interprets it, then selects questions from QR meeting those specifications. Structure of AIS and a semi-automated system to tag the repository are explained in the previous sections.

A prototype version of the AIGen is designed with restricted version of AIS structure. The objective was to determine the feasibility of such a system. The parameters in AIS are exactly matching with the tags associated with every question in tagged repository. Similarly, every parameter in AIS is associated with upper and lower bounds as discussed in Section 9.2. In this process, following assumptions are made.

- AI is considered as a flat list of questions. There is only one section with all the properties attached to it. No nesting of sections and questions.
- There are no internal options for sections and questions.
- Only total marks are provided by the teacher in AIS.

- No external knowledge base is available for providing intelligence to the system so that the system can complete the partially filled AIS by the teacher. Whatever is filled by the teacher is considered as final.

9.3.1 System Architecture

The system block diagram of AIGen system is shown in Fig. 9.8. It consists of three major components: (i) An interface to enter the values in the AIS (ii) Question Selector to take the inputs from AIS and searches the questions in a tagged repository (iii) AI Generator to take the questions provided by the Question selector, adds the header information and generates AI in XML and WORD format. The user is a teacher who wants to generate AI for a particular course (here Data Structures) either for part or whole of the course. AIGen uses the tagged repository generated by the QTagger.

9.3.2 AIS Interface

The AIS interface is used to enter the values for specification parameters by the teacher. The teacher will be asked to enter two types of specifications (i) the header or preamble; (ii) AI specification.

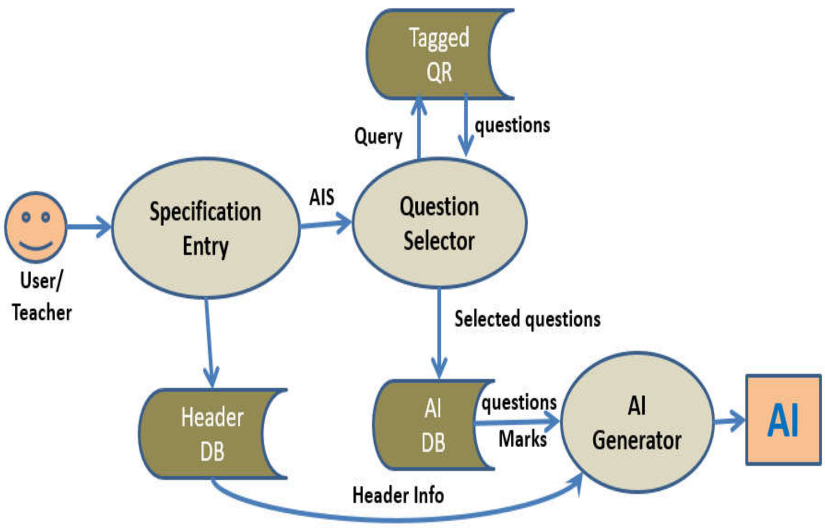


Figure 9.8 System Block Diagram

9.3.2.1 Header/Preamble

At the first level, teacher will be asked to enter the AI preamble specifications such as university, course, course_year, semester, subject, total marks, date of examination and notes if any as shown in Fig. 9.9. All the fields are validated for non-blank values. Only numeric values are accepted in marks field whereas rest of the fields accept alphanumeric values. The valid fields go to header table to print them further on AI

.

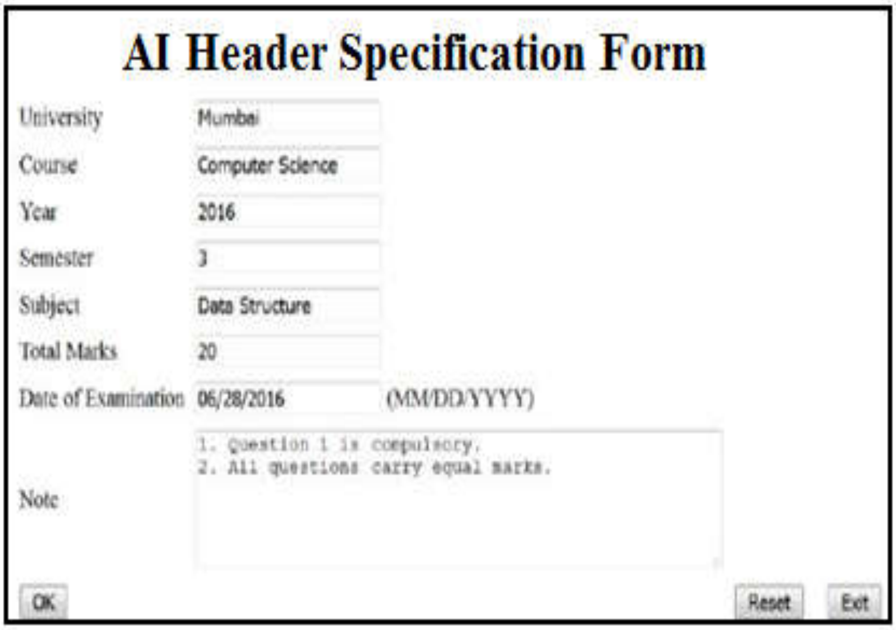


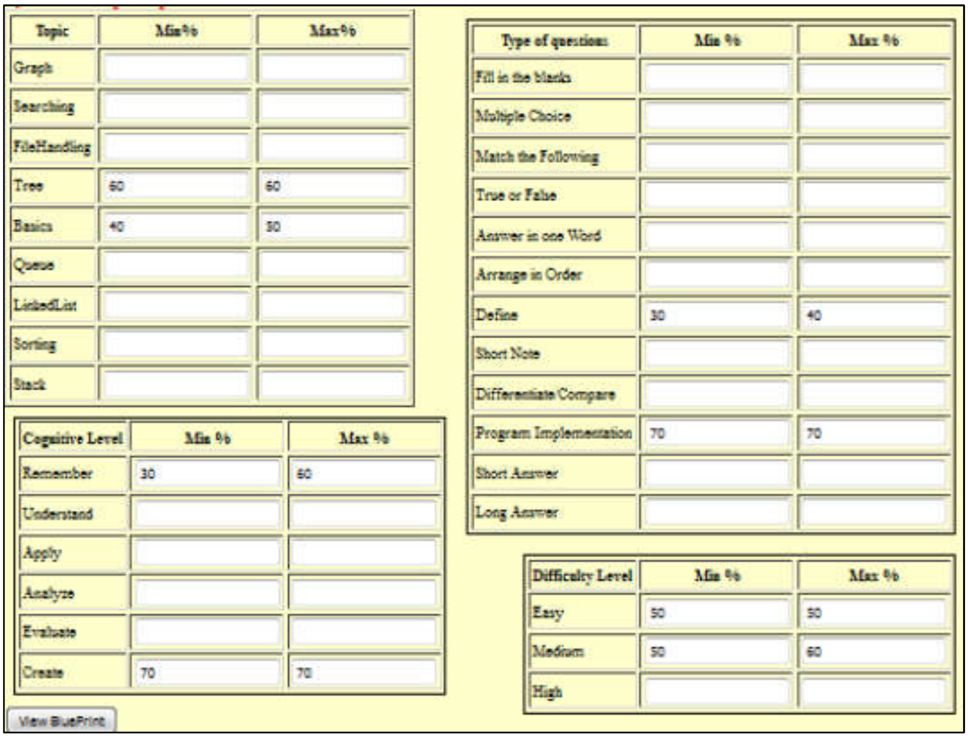
Figure 9.9 Header Specification Form

9.3.2.2 AI Specification

Teacher will be provided with the AIS form to enter the values for each property in the form of lower and upper bounds as shown in Fig. 9.10. There are many constraints on the values that are validated such as:

- only numeric values will be accepted in all the minimum and maximum range.
- The lower bound total of all values for a tag must always be equal to 100 and upper bound total for tag must be greater than or equal to 100.

- Any upper bound value of a tag attribute must always be greater than or equal to lower bound value of the respective tag attribute. These validations also hold for the min-max values of question type, cognitive level and difficulty level.



Topic	Min%	Max%
Graph		
Searching		
FileHandling		
Tree	60	60
Basics	40	50
Queue		
LinkedList		
Sorting		
Stack		

Cognitive Level	Min %	Max %
Remember	30	60
Understand		
Apply		
Analyze		
Evaluate		
Create	70	70

Type of questions	Min %	Max %
Fill in the blanks		
Multiple Choice		
Match the Following		
True or False		
Answer in one Word		
Arrange in Order		
Define	30	40
Short Note		
Differentiate/Compare		
Program Implementation	70	70
Short Answer		
Long Answer		

Difficulty Level	Min %	Max %
Easy	50	50
Medium	50	60
High		

View BluePrint

Figure 9.10 AIS Form

The user has the option for providing no specification, partial specification and full specification. If no specification is provided, then range values are considered as [0-100]. An example AIS (Blueprint) after the teacher enters required values is shown in Fig. 9.11. Teacher can generate the AI if he/she is satisfied with the AIS or he/she can go back and do some modifications. This process can be repeated till the teacher is satisfied with the generated AIS.

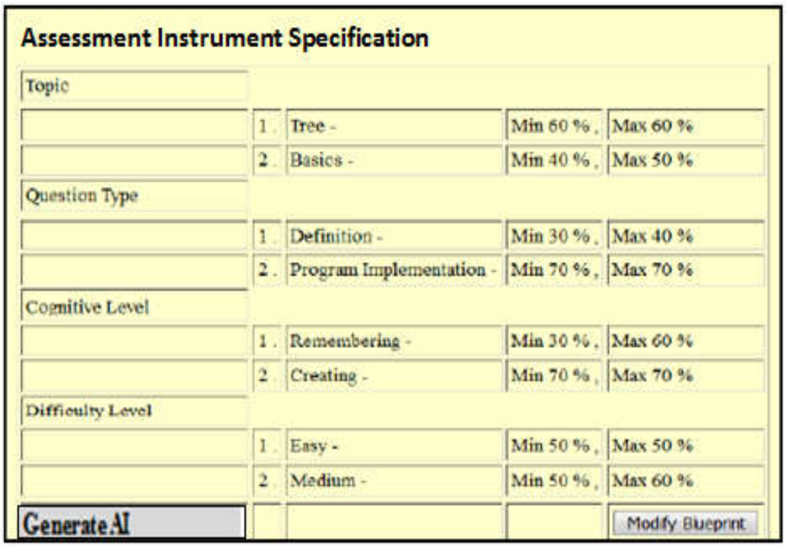


Figure 9.11 Example AIS generated by a teacher

9.3.2.3 Question Selector

After the valid specifications are received, the Question selector starts searching for the questions in a tagged repository which fit within the minimum and maximum marks range specified for a selected tag. First it starts with the tag topic, then question-type, then cognitive level and followed by difficulty level and selects the question whose tag values match with the parameter values in the AIS. Initially, it tries to satisfy the minimum requirements and then goes to maximum values. It does an exhaustive search for all the combination of parameter values in a *depth first order* for each tag in the AIS.

The questions are picked up randomly every time from the repository. Thus, the teacher will get new set of questions for the same specifications next time. If the questions are not found meeting the AIS, it will display the AIS as it is and print a warning message. The selected questions are then stored in AI database for AI generation. The algorithm followed by question selector is shown in Fig. 9.12.

ALGORITHM SEARCH ENGINE:

1. Get the input specification..
2. Marks_Left=Total_marks.
3. For all the values present in Topic, Question type, cognitive level and difficulty level, search the question in Min range column
4. If Marks_Left >0, then search in Max range columns.
5. For each question id from concept table, search the question in Question table using following query:
   select id, text, max_marks from question where id = concept_qid and max_marks <= Topic[j][i] and (cognitive = coglevel and (max_marks <= Coglevel[x][z])) and (q_type = Qtype and (max_marks <= QType[l][m])) and (difficulty = Difflevel and (max_marks <= DiffLevel[v][y]))
6. If question found, add it to Question_paper database
7. Decrement the values of the respective arrays and Marks_Left by marks of the question.
8. While Marks_Left>0, go to step 4.
9. If no question added to Question_paper database, issue a warning. else display the blueprint.

Figure 9.12 Question selector Algorithm

In the next step, the AI Generator takes the questions from the AI database and combines with the preamble information available in header table to produce AI in XML format. XML format is preferred as it is supported by all browsers. The XML output would look like as shown in Fig. 9.13. User will get an interface with an option to convert it into Word document. The AI in Word format is shown in Fig. 9.14.

```
  <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <questionset>
  - <Preabmle>
      <University>Mumbai</University>
      <Course>Computer Science</Course>
      <Course_Year>2016</Course_Year>
      <Semester>3</Semester>
      <Subject>Data Structure</Subject>
      <Total_marks>20</Total_marks>
      <examdate>2016-06-28</examdate>
      <Note>1. Question 1 is compulsory. 2. All questions carry equal marks.</Note>
    </Preabmle>
  - <questions>
    - <question qno="1">
        <text>Define AVL trees</text>
        <qmarks>2</qmarks>
      </question>
    - <question qno="2">
        <text>Discuss AVL trees. Insert the following elements into a AVL search tree: 27 23 25 29
        <qmarks>8</qmarks>
      </question>
```

Figure 9.13 AI in XML format

```
University :  Mumbai                              Year :        2016
Course :   Computer Science                       Semester :       3
Subject :   Data Structure                        Total Marks :   20
Date : 28-06-2016
----------------------------------------------------------------------
Note :
1. Question 1 is compulsory.
2. All questions carry equal marks.
----------------------------------------------------------------------
Q. 1    Define AVL trees                                          (2)
Q. 2    Discuss AVL trees. Insert the following elements into a AVL
        search tree: 27 23 25 29 35 33 39                         (8)
Q. 3    Define Data Structures                                    (2)
Q. 4    Write a Non-Recursive function for inorder traversal      (8)
```

Figure 9.14 AI in Word format

9.3.3 System Evaluation

We wanted to test the system for its validity i.e. to check whether the generated AI complies with the teacher entered AIS. The generated AI was analyzed to find the proportion of questions for each of the tag parameters. This was then compared with the teacher entered AIS. The difference between the teacher entered AIS and the system generated AI indicate that either the question selector algorithm needs improvement or there are not enough questions available for all combination of values for all the parameters of AIS in QR. This process was done for twenty AIs for different combinations of values for AIS. Ideally both should match. However, it was found that the generated AIs were 80% compliant with the corresponding teacher entered AIS. In majority of cases the problem was because of the non-availability of questions matching a specific combinations of parameter values.

System was evaluated for its usability and usefulness by five CS instructors with more than 8 years of teaching experience. They were given 20 minutes to do the following.

- Provide different combinations of values for AIS.
- Validate each entry on every screen with positive and negative inputs
- Check the correctness of generated AI in XML format and Word format

After that they attempted a SUS questionnaire. Each question of the SUS questionnaire was associated with an open-ended feedback question. For each of the SUS questions question, the users were to write an open-ended response to explain the justification of selecting a specific Likert's scale score for a question. All of them found the system to be useful. Most of them felt that the GUI is simple and easy to use, technical knowledge required to use the system is almost nil other than the knowledge of Bloom's taxonomy and question types and the flexibility of generating AI in XML and ready to print Word format provided an added advantage.

9.4 Summary

The framework for generation of AI was operationalized by developing AIGen tool that automatically generate the AI from teacher's request for assessment using a tagged QR. To design and develop such a system, we came up with structure of AIS to accept and represent the teachers request for assessment. Semi-automatically tagged QR was also built to facilitate the process of selecting the questions as per the specification. AIGen is only a prototype model as there are lot many improvements to be made. It takes only a subset of the original AIS. Only one particular strategy was considered here for design of Question Selector algorithm. Other strategies such as genetic algorithms or any other optimization algorithms need to be explored. Hence, the generated AI can be considered as the first version that can be submitted to IQuE where IQuE can evaluate its quality and give feedback to the teacher.

Chapter 10 connects the research objectives to the achieved solutions and discusses the generalizability of the solution and scope and limitation of thesis.

Chapter 10

Discussion

The research work started with the primary objective of improving the quality of AI. The solution approach identified two ways to achieve this objective. (i) Let teachers create AI and then do the automatic evaluation of its quality along different dimensions before it is given to students or (ii) Provide an AI Generation framework that will contain an interface for teachers to enter the AI requirements and the system will generate an AI compliant with that requirement and other best practices. The first approach is the major focus of the research work and resulted in the design of framework for evaluation of AI. For the second approach, framework is designed and a prototype version of the AI Generation is developed as a proof of concept.

10.1 Achievement of Research Objectives

The two solution approaches towards improving the quality of AI resulted in the formulation of three research objectives. In the process of planning the implementation of the solution approaches, the following research questions were formulated as shown in Table 10.1 (same as Fig. 4.1) against the research objectives.

Table 10.1 Research questions

Position	1) What role does a quality of AI play in achieving the intended outcome by learners? 2) What are the different parameters that determine the quality of AI? 3) What are the different problems usually found with the AIs of various universities in India? 4) What are the difficulties faced by the teachers in manually evaluating the quality of AI? 5) How can technology help in solving research problem of improving the quality of AI?
Design	<u>IQuE</u> 6) How to annotate the nodes of the domain ontology with the content and cognitive level information from LO and questions in AI? 7) How to devise a measure for alignment score of AI with the LOs of the course in terms of its content and cognitive level? 8) How to visually represent the alignment of AI with the LOs of a course? 9) What constraints should an ontology meet inorder to facilitate development of IQuE?
	<u>TTM</u> 10) What is an appropriate pedagogy model for online training of teachers about using IQuE? 11) How to design and develop training environment including suitable feedback mechanism?
	<u>AIGen</u> 12) What are the elements needed to specify requirements of the desirable characteristics of AI? 13) What tags are required to be associated with a question to facilitate AIGen? 14) How to design and develop a tool that automatically generates an AI from a teacher's specification using a question repository?
Evaluation	16) How usable is IQuE as perceived by the users? 17) How useful is IQuE as perceived by the users? 18) What is the users' perception about the effectiveness of feedback mechanism of TTM? 19) What is the accuracy of AIGen?

The questions from 1 to 5 are position type of questions and are answered from the literature, personal experience and analysis of previous years AIs. The answers to these questions established the need and context of our research work. From the literature, it was found that the quality of any educational assessment exercise depends on the quality of the instruments used. Hence, quality of AI needs to be improved. Alignment of AI with the LOs of the course is the most important quality measure of AI and the focus of the research work is on it. Further, it was seen from the literature that the AIs designed for examinations in engineering education in many of the Universities in India are of poor quality. This was confirmed by analyzing thirty AIs designed for end semester examinations of various courses from Mumbai University. Literature also says that teachers have to spend considerable amount of time and effort in manually ensuring the quality of AI and they face lot of difficulties in writing assessment questions against a given set of LOs. These findings from the literature led to a strong motivation towards building IQuE, TTM and AIGen tools.

Every question from 6 to 19 is associated with one of the objectives discussed below. They contribute towards the design and development phase and evaluation phase of IQuE, TTM and AIGen tools. The overall research process is shown in Fig. 10.1.

RO1: Designing a framework for evaluating the quality of AI in terms of its alignment with the LOs of the course.

The questions from 6-9 and 15-17 are contributing towards finding solution to RO1. The first research objective addressed the issue of evaluating the quality of AI by measuring its alignment with LOs of the course for which the AI is created. This was answered by designing the framework for AI quality evaluation. The framework was operationalized by building a software tool called Instrument Quality Evaluator (IQuE). It was designed with three major components (i) KR mechanism to represent the domain (ii) Annotating the information from AI and LOs into a KR and (iii) Quality Evaluator

Ontology was chosen for representing the domain. It was found that ontology is successful in capturing the hierarchical and dependency relationships among different concepts identified from the course content. The ontology was validated by three domain experts. Minor constraints were imposed on ontology structure; They did not change the basic nature of the ontology but facilitated the annotation mechanism of IQuE.

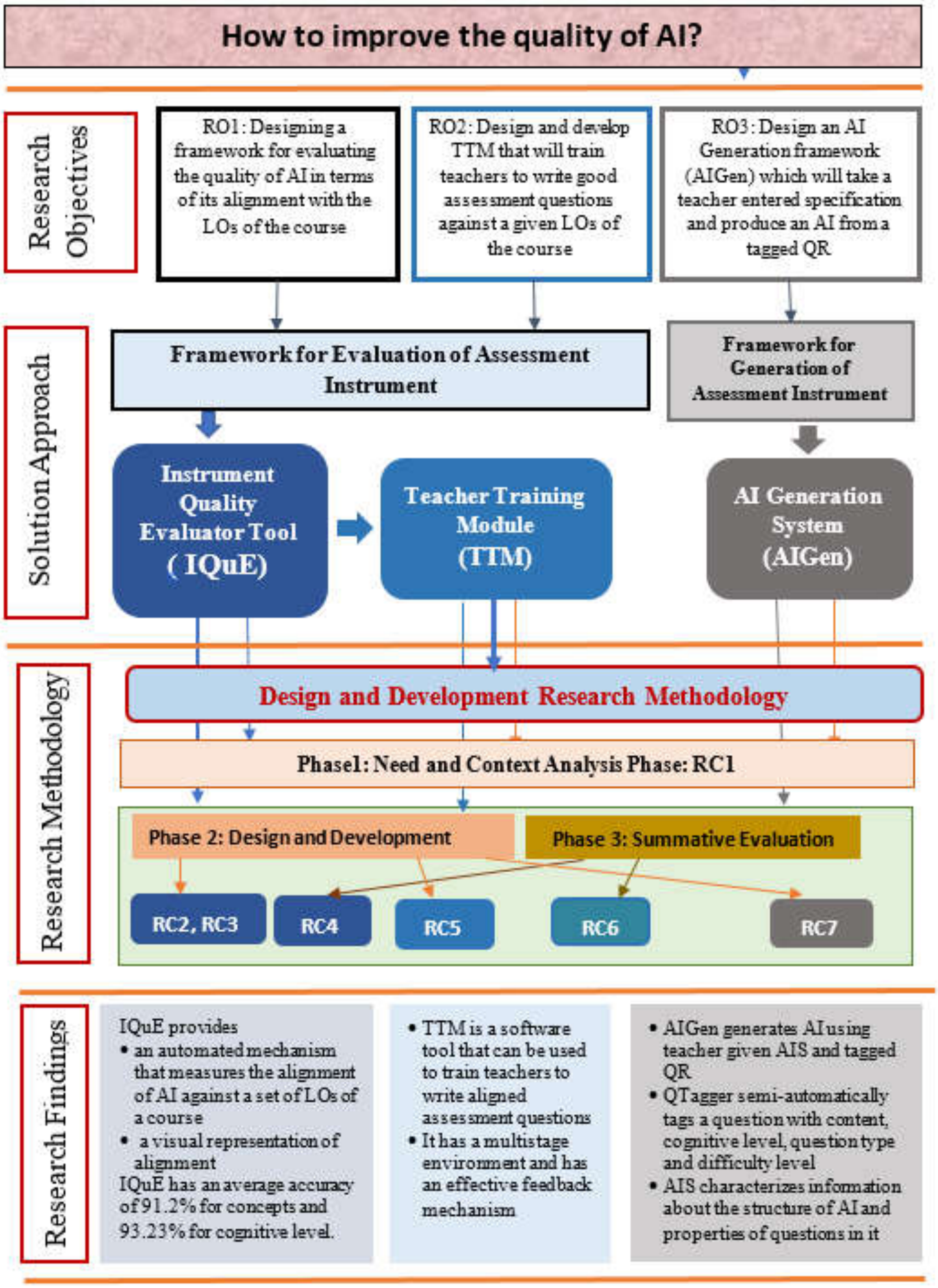


Figure 10.1 Overview of Research

These constraints are then formulated as transformation rules or guidelines that can be used to convert any given ontology into an ontology suitable for IQuE. If any teacher wants to use IQuE to evaluate the quality of AI of any other course, he/she can apply these rules to create an ontology compliant with IQuE

The content and cognitive level information are to be extracted from each question in AI and each LO from the set of LOs. The nature of the LO and question made the extraction process extremely challenging. The support of simple NLP techniques such as tokenization, lemmatization and Part-of-speech tagging were taken to identify the explicit concepts and cognitive level from LO and question. Implicit or hidden concepts were found by formulating the ontology traversal algorithm. The algorithm exploited the hierarchical and the dependency information of the concepts in the ontology. Similarly, cognitive level was identified by matching to the action verbs associated with Bloom's taxonomy, typical phrases that form question indicators and domain specific action verbs. The content and cognitive level information were mapped to the nodes of the ontology. The mapped nodes were then color coded with blue or red color as per their involvement in question and LO respectively. The color codes helped in the visual representation of alignment.

The quality evaluator needs a method to measure alignment by finding the commonalities and differences among the concepts and cognitive levels of questions and LOs. The alignment was separately calculated for content and cognitive level. Individual measures of content alignment between an LO and a question were calculated using an information theoretic formula. They are then aggregated to form the overall alignment measure. The aggregation process is not straightforward but takes into consideration the maximum contribution of all questions towards each of the LOs as well as the contribution of each question towards all the LOs. The later one decides the utility of a question. In case of cognitive level alignment of question and LO, the alignment measure is zero, if there is a difference of more than one in their cognitive levels. If there is only one level difference, the alignment measure is 50, considering the benefit of ambiguity among the adjacent levels in Bloom's taxonomy.

IQuE provides three types of output pertaining to alignment of teacher generated AI against the course LOs as shown in Fig 10.2. It provides a numerical measure of alignment for both content and cognitive level in terms of percentages and also provides information about non-utility questions. The non-utility questions do not contribute anything to the alignment score neither in terms of content nor in terms of cognitive level to any of the LOs. Hence, the system indicates that they can be discarded or replaced with more useful questions that may contribute to increase in the alignment score. It also gives a visual representation of alignment.

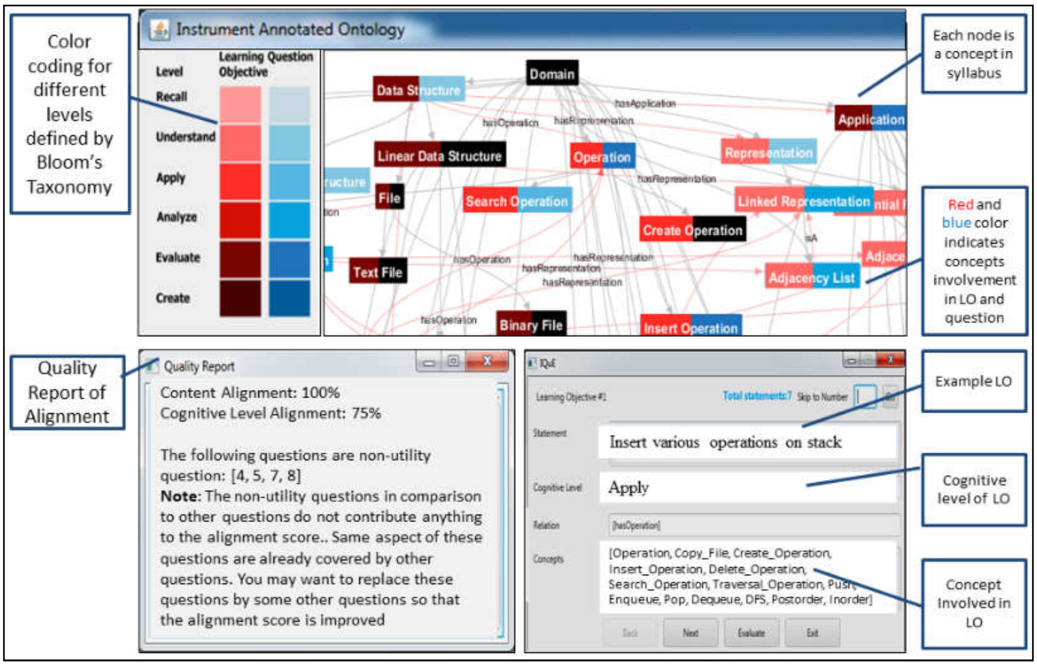


Figure 10.2 Different outputs provided by the IQuE

The repeated and thorough analysis of results during the formative evaluation using the real-world samples (questions and LOs taken from different Universities) and solving the detected issues one by one considerably improved the performance of IQuE. The formative evaluation was done by comparing the system generated results with the expert generated alignment values and generating the confusion matrix. The accuracy with respect to content alignment is 91.2% and for cognitive level alignment is 93.23%. There were some issues that could not be solved due to the current limitations of IQuE. They were appropriately classified into different classes with reasons for their failures.

Saturation in the performance of IQuE was established when the accuracy remained almost same when it was evaluated using a different set of samples of LOs and questions. The results of usability and usefulness tests for IQuE by teachers indicated a high rating for the utility of the system and ease of usage because of the simple and attractive GUI and the need of minimal training required for using the system.

RO2: Design and develop a Teacher Training Module (TTM) that will train teachers to write good assessment questions against a given LO

The questions 10, 11 and 18 are contributing towards finding solution to RO2. The second research objective addresses the issue of training the teachers to use IQuE efficiently. This was met by development of TTM that trains teachers to write aligned assessment questions against given LOs. TTM uses IQuE to find the measure of alignment of teacher written questions and the system displayed LOs. The TTM was designed with three levels which are called stages where progressive stages represent increasing levels of complexity. This concept is borrowed from principles of game design where each level in the game is a challenge to be completed. Stage 1 is simple where teachers has to write single question against a given LO followed by stage 2 where they have to write multiple questions against a given LO. In stage 3, they write multiple questions against given set of LOs.

A formative feedback mechanism has been incorporated where a teacher gets immediate feedback in every stage about the alignment measures in terms of its content and cognitive levels. If alignment measure is low, then the system tells the teacher to modify the question and resubmit. It also gives information about the unaddressed concepts from LOs, out of syllabus concepts and difference in cognitive levels of LOs and questions. This helps in generating hints to the teacher about the changes he needs to do if system recommends modification of questions.

The summative evaluation of the system indicated the feedback mechanism of TTM to be effective as it helped and motivated teachers in writing questions with better alignment every time. The functionalities such as scoreboard and dashboard also adopted from game design increased the motivation of teachers and also helped in maintaining the user logs containing valuable information such as number of attempts taken by the teacher to write the correct question for each LO/s displayed in every stage and the feedback provided by the system. General interface of TTM for stage 1 is shown in Fig. 10.3 which is reproduced from Fig. 6.8

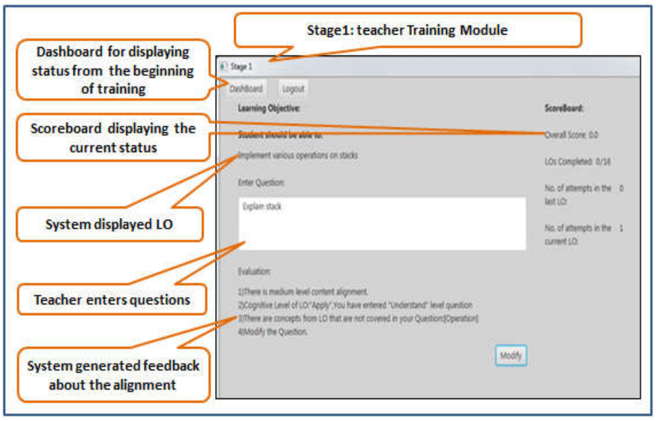


Figure 10.3. Stage1 interface of TTM

RO3: Design an AI Generation framework which will take a teacher entered specification and produce an AI from a tagged question repository.

The third research objective is towards the second approach of improving the quality of AI by designing of AI generation framework that will facilitate automatic generation of AI using a question repository. This objective was answered by developing a prototype version of AI generation tool (AIGen) that reads a given AI specification and selects the relevant questions from the repository to produce an AI. The characterization of AI using an XML specification language was the first stage towards this process. This AIS contained both structure and property information of AI. Structure of AI incorporates information about nested structure covering sections, sub-sections, questions, sub-questions and the choice of options among questions or sub questions for the candidates. Property information include learning objective of a question, proportion of questions in different cognitive levels, proportion of questions in different difficulty levels, proportion of question types, distribution of questions among topics assigned for the assessment and marks associated with a question. These properties can be associated with any of the structural component of AIS. The AIS is general enough to accommodate any type of AI of different universities.

In order to facilitate selection of relevant questions from a repository that matches with parameters of AIS, a semi-automatic tagging engine (QTagger) was developed and would tag the questions and store it in the repository. This would help to adapt existing collection of questions and question banks for use in the repository of AIGen. We considered four tags namely cognitive level defined by the Bloom's taxonomy, difficulty level, type of question and the name of content or topic from the syllabus. QTagger was tested for tagging accuracy, usability and usefulness. Tagging accuracy was independently evaluated for each tag. It was found that the accuracy with respect to the cognitive level annotations was 78%; question type annotations was 90%; difficulty level was 93% accurate and content was 87.5%. Users felt that the QTagger was useful and easy to use. If the teachers are not satisfied with some of the suggested tags, manual editing facility is provided to modify the tags. Sometimes, nature of questions makes the automatic tagging impossible for the system. To solve this problem, it provides an option where teachers can reformat the questions at the entry level itself to facilitate more accurate tagging.

The prototype version of AIGen is designed with a restricted version of AIS. The objective was to determine the feasibility of such a system. AIS can take flexible range values for all its parameters. AIGen has a rule based question selector which takes a combination of teacher specified parameter values and find questions that matches with that from the question repository. AIGen can generate AI in XML and Word format. Preliminary evaluation of AIGen with 20 different AIS is done and found that generated AI is 80% compliant with the teacher entered AIS. The system needs further testing and analysis to find the conditions under which it is failing to meet the given specification. Is it because of non-availability of questions in the repository or problem with the algorithm? What is the quality of AI generated? All these need to be investigated further. An example AIS and the generated AI are shown in Fig. 10.4.

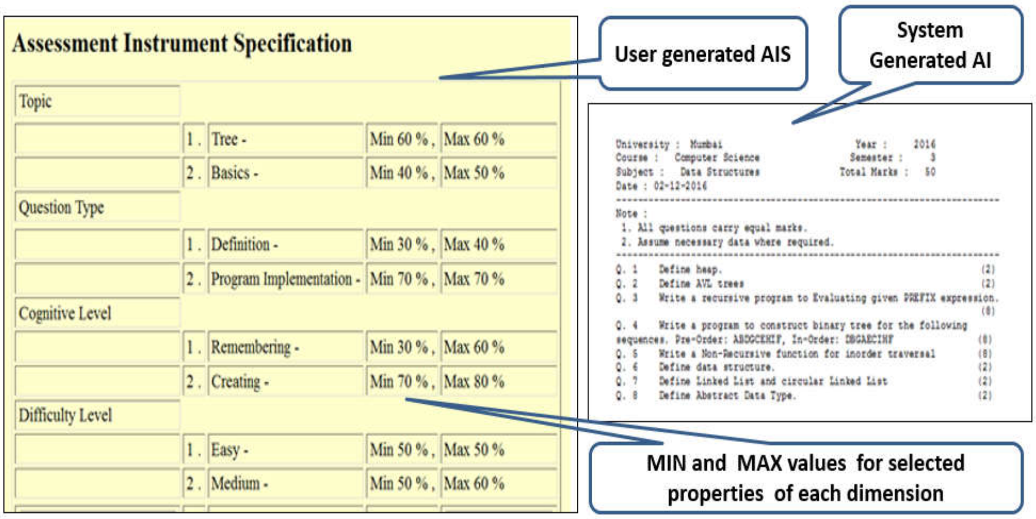


Figure 10.4. An example AIS and the generated AI

10.2 Establishing Generalizability

The research work is focused on improving the quality of AI and reducing the time and effort teachers spend on it. This is achieved by building tools such as IQuE, TTM and AIGen. The IQuE uses ontology as the knowledge representation mechanism. Ontology is used to represent the syllabus of the course. The computer engineering curriculum and specifically *Data Structures* course has been selected only because of the familiarity with that. An ontology was created for *Data Structures* domain and the LOs and questions from this course domain were utilized for all tests and studies. All the results and claims reported in this thesis are based on this. One issue that arises is that how will the system behave for other courses and programs. This is discussed along various dimensions below.

Establishing Generalizability with Respect to Course Domain

The *Data Structures* course was chosen only for convenience. While designing all the systems, we have kept this aspect in mind and made the design as domain independent as possible. To prove generalizability to other courses, we need to answer these two questions (i) Can ontology as per our requirements be created for any course domain other than *Data Structures*? (ii) Will the LOs and questions from any other course work for IQuE, TTM and AIGen?

For the first case, a set of domain independent guidelines were formulated for creating ontology for any course. Using these guidelines, ontologies for Data Mining and Image Processing course have been successfully created by teachers. Data Mining is a course offered in semester VI of computer engineering curriculum of Mumbai University. Image Processing is an inter-disciplinary elective course offered in Computer, Electronics and Telecommunication and Information Technology curriculum. Ontology required for our systems need set of concepts covering the whole domain, links connecting these concepts depending on its hierarchical or any other relations among them and synonyms for the nodes and links of the ontology. These requirements can be met by any course even for descriptive courses like History, Geography, English, etc. *Hence, ontology can be created for any other course in any discipline provided they satisfy the above requirements*.

One of the requirements of the system is that the LOs and questions should be machine parsable and machine analyzable. From the LOs and questions, the system must extract contents and cognitive level calculate the measure of alignment. The LOs and question should have indicators for these to facilitate automatic parsing and extraction.

Domains where assessment involves phenomenal focus on non-textual content such as mathematics, drawing, etc. will have a problem. Otherwise, the system is generalizable to other similar domains.

10.3 Limitation of our Research Work

Limitations of this thesis are with respect to quality parameters other than alignment, nature and type of LOs and questions and type of assessments other than the written assessments we are considering. These are explained below.

Limitations with Respect to Quality Parameters of AI

There are various quality parameters of AI such as proper proportions of cognitive level, difficulty level, type of question, distribution of questions among the different sub topics allotted for assessment, distribution of marks across the instrument, time allotted to students to solve the questions in AI, ambiguity in wording of questions, its alignment with the learning objectives of the course, etc. The other important quality measure is discrimination index of AI which tell whether the AI can discriminate between the good and bad students. *But, in our research study, the emphasis of framework is on alignment problem not on the other aspects of quality.*

Limitation with Respect to Type of LOs and Questions

Each question and LO from the AI are analyzed by IQuE to extract content and cognitive level information from it. For this, LOs and questions must be machine parsable. Further, the system will not be able to analyze certain types of questions in which there is no indication of any concepts involved. For example, in question such as "Solve y=f(x)" there is no way the system can automatically find what concept from mathematics course is involved in. Same problem occurs, if it is a video content question or a picture question such as "Given a picture, interpret it". In our view, currently the system is capable of handling domains where predominantly text based questions are present.

Similarly, there are constraints on the framing of LOs and questions. Certain natural language ambiguities cannot be handled by the system at present. Most of them are explained in Section 6.2.4 of thesis such as "The AND problem", "slot and multiple concepts problem", etc. These problems are encountered in QTagger system also while tagging the questions. Currently they are managed by a pre-processing step where such LOs and questions are manually reframed before giving it to the system. They could be addressed to a large extent by richer NLP techniques, and recommending rewrite.

Limitation with Respect to Type of Assessments

The focus of the research work is restricted only to written assessments in a typical university scenario. Here the teachers generate the AI following the prescribed syllabus and administer it to students at the designated time and venue. The only information available to build automated systems are syllabus, LOs and questions in AI. There is no human judgement involved in any of the systems. Hence, all other types of assessments such as oral examinations, laboratory tests, project, etc. are beyond the scope of this work.

Chapter 11

Contribution and Future Scope

This chapter lists the contributions from our thesis work towards the field of educational technology research and future research directions in this field.

11.1 Thesis Contribution

The thesis makes contributions in the field of educational assessment specifically in improving the quality of AI. The contributions are in terms of educational technology tools for teachers, guidelines and research knowledge.

- ***A framework to evaluate the quality of AI.*** The framework has components such as KR mechanism that represents the course domain, Integration mechanism that integrates the contents of LOs and AI into KR mechanism and formulation of measure of quality

- ***An Instrument Quality Evaluator (IQuE) tool that realizes this framework.*** IQuE measures the quality of AI in terms of its alignment with the LOs of the course. It uses ontology as a KR mechanism, integrates content and cognitive level information into it and uses the commonalities and differences between these parameters to formulate the measure of alignment.

- ***Teacher training module that can be used to train teachers to write good assessment questions against given LOs.*** TTM provides a self-learning environment with multiple stages and a formative feedback mechanism.

- ***A language to define assessment instrument specification (AIS) which encompasses its structural as well as behavioral aspect of AI.*** AIS is in its most general form and can be used to formally define any type of AI.

- ***A framework for the automated generation of AI.*** The framework has components such as, an interface where teachers can provide values for parameters of AIS, question selector interprets the AIS and selects the most appropriate questions from the repository to be put into AI and AI generator that generates AI from the selected questions.

- **A prototype model of a tool (AIGen) that realizes this framework.** AIGen has an interface where user enters values for various parameters, a rule based exhaustive question selector which takes a teacher entered specification and generates an AI from a tagged item repository in xml and Word format.

- ***Software tool (QTagger) which suggests metadata for a given question.*** The metadata currently covers to cognitive level, question type, content and difficulty level.

- ***Guidelines for building a domain ontology to represent the syllabus.*** Using these guidelines, the user can create ontology for any course in a form that is suitable for IQuE.

- ***Formulation of alignment measure of AI in terms of its content and cognitive level.*** It uses commonalities and differences in the number of concepts covered and the cognitive levels between the LOs and AI.

- ***Design of visual representation of alignment.*** Color coding scheme is used to represent the mapping of content and cognitive level information into a domain ontology so that the teacher can see the alignment of the generated AI at a glance and take corrective measures if needed.

In the backdrop of the academicians' concerns about the poor quality of AIs in engineering education, IQuE, TTM and AIGen are of great help to teachers. The design of framework and the resulting IQuE tool is based on the theory of constructive alignment (Biggs, 2003) that a good quality AI needs to be aligned with LOs of the course and the ontological engineering principle for representing course domain (Noy & McGuinness, 2001). The implementation is done based on the principle of (Krathwohl, 2004) that alignment is measured in terms of two dimensions: content and cognitive level. All the systems developed are tested and their validity is demonstrated.

Even though researchers have stressed the importance and benefits of aligning the AI to course LOs (Biggs, 2003) (Krathwohl, 2004) (Rowntree, 1977) (Nitko, 2001), to the best of our knowledge, no one has reported attempts to formulate and automate this task.

11.2 Future work

During the process of implementing both the frameworks, certain issues were underplayed, so that focus remains on the mainstream problem. These are explicitly listed in Section 9.3 and 9.4 in the form of assumptions and limitations such as the availability of comprehensive domain ontology and nature of LOs and questions. These can be taken up as a future work.

Extending to Other Domains

Currently, the scope is limited to engineering curriculum as all the samples for the study are taken from the *Data Structures* and similar courses in engineering. We have made the system as domain independent as possible. Hence, it is expected to work for other similar courses without much modification in the system other than some domain specific information added to its knowledge base. The domain ontology can be created for almost any course following our prescribed set of standard guidelines. We expect the system to work in domains where assessments contain predominantly text based questions. The near future scope would be to extend the system to other courses in engineering curriculum or outside it and investigate the performance on them.

Other Quality measures

We have looked at only alignment of AI with LOs as the measure of quality. There are other quality measures of AI such as ambiguity in wording of questions, marks distribution, the ratio of marks allotted against the time required to answer, distribution of difficulty levels, redundancy among questions, etc. These can be further explored and integrated into the existing system.

Nature of LOs and questions

The system makes some assumptions about the kind of language used in LO. Certain natural language issues cannot be handled by the system. They are explained in Section 6.2.4 of thesis such as “The AND problem”, “slot and multiple concepts problem”, etc. Currently, there is a manual pre-processing step before the LOs and questions are given to IQuE for further processing. This is done to ensure that LOs and questions are as per the requirements of IQuE system. As a future work, we would like to remove as much of these restrictions as possible with the help of technology intervention. This needs further investigation of more sophisticated NLP processing.

Improvement in AIGen

AIGen as only a prototype and there are major challenges in building a system like this. The first is how to generate AI which is guaranteed to be of good quality. AIS interface takes only a subset of the originally proposed AIS. It doesn’t handle external and internal options among the questions. It does not understand the meaning of quality AI. The assumption is that the teacher is providing all the required information in AIS. If the teacher fills the AIS partially, then AIGen needs to have the intelligence to complete it. The AI generator algorithm currently considers only one particular strategy which is blind exhaustive search for all the combinations of values in AIS. Other strategies such as genetic algorithms or any other optimization algorithms need to be explored. These all require significant modifications in algorithm and in the framework and we will be investigating this as a future scope.

Others

The other extensions of future work are related to the functional aspects of the systems like,

- Extending the TTM to a mobile platform with IQuE loaded on the server side. TTM is currently an add-on module attached to IQuE and is a standalone system. Having it on a mobile will increase the reachability of TTM and the teachers can use it at any time and place as per their convenience.

- Integrate the IQuE and AIGen systems into one whole system. Currently, they are working in an independent manner where IQuE is evaluating the quality of AI inputted to it and AIGen is automatically generating an AI.

11.3 Final Reflection

Overall, the research work is based on the principle that designing the AI aligned with the LOs of the course is very important in achieving the intended outcome in teaching and learning. The technology intervention in the form of tools like IQuE, TTM and AIGen will immensely help the teachers in this area. During the paper and poster presentation of our work, specifically IQuE and TTM, teachers have verbally expressed the excitement and eagerness to use the tool to evaluate the AI designed by them. As an educational technology researcher and a teacher, this has motivated me to follow this research work and exploit the full potential of developed tools further in my career.

Appendix

Appendix A Protégé implementation of Domain Ontology

Protégé includes machine-interpretable definitions of basic concepts in the domain and relations among them. Protégé is used to build domain ontology for *Data Structures*. The output is in the form of .owl file which can be utilized by the JAVA program. The ontology creation process is outlined below.

Start the Protégé and create a 'data_structures.owl' file.

Step 1: Add classes and create a class hierarchy

The main building blocks of ontology are classes. All the concepts finalized in Section 6.1.1 form the classes in ontology. The editing of classes is carried out using the 'Classes Tab'. The 'Classes Tab' has 3 options 1) Add a subclass 2) Add sibling class and 3) Delete an existing class. Using this, a complete class hierarchy is created as shown in Fig. 1.

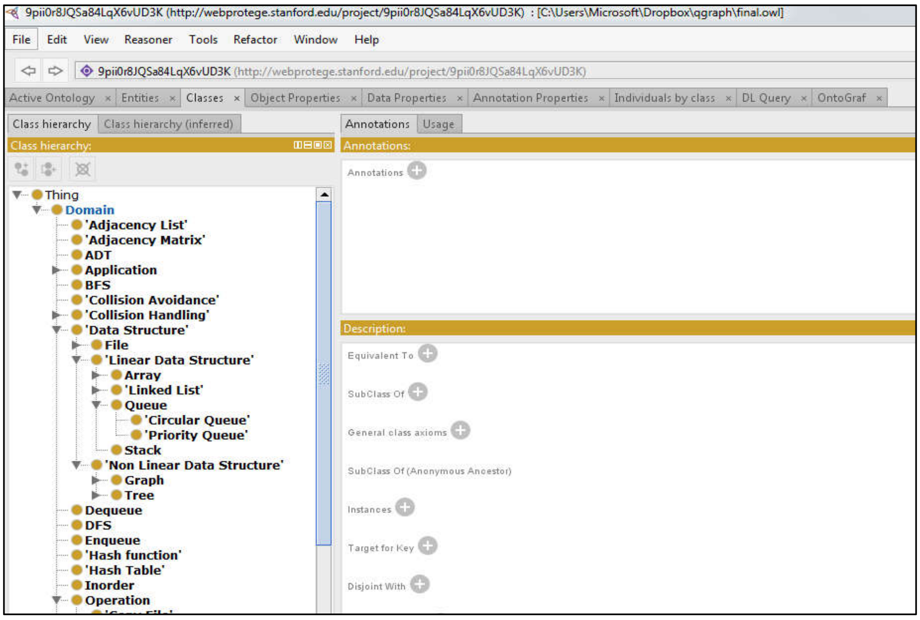


Figure 1 A Class Hierarchy in Protégé

Step 2: Add object properties

Object properties are relationships between two individuals. To create a link between two concepts, switch to the 'Object Properties' tab and use the 'Add Object Property' button to create a new Object property. For example, hasOperation property can be created using the 'Property Name Dialog'as shown in Fig 2. In this way, all the object properties are added

.

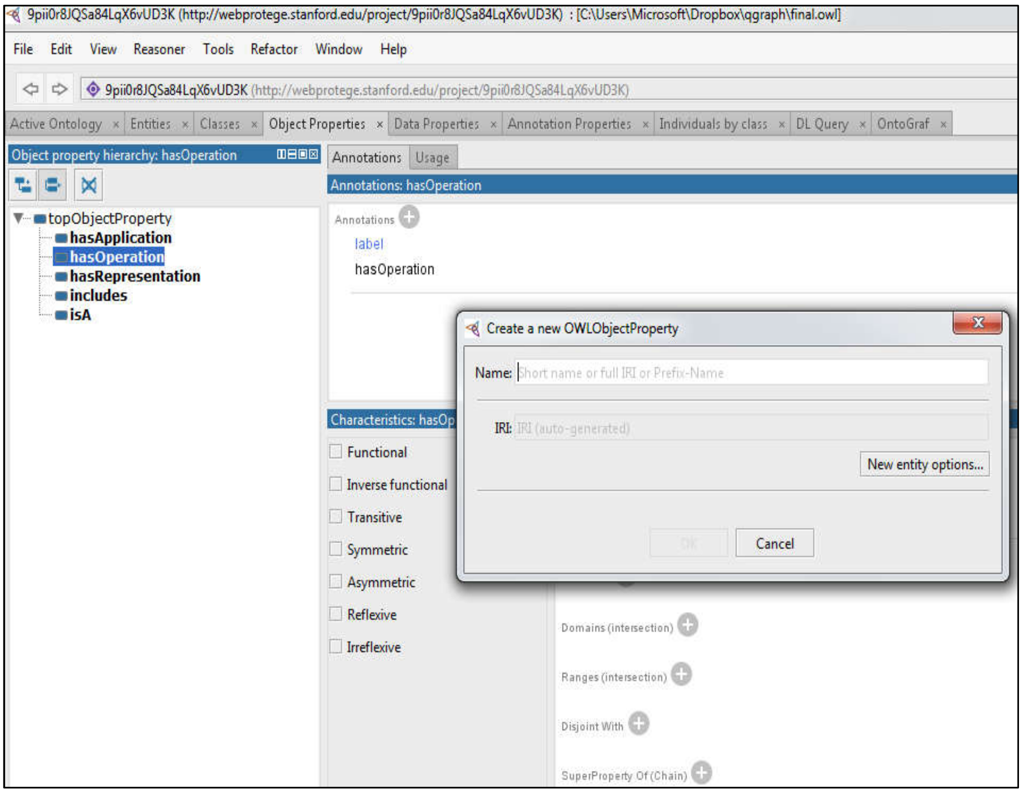


Figure 2 Object Property Window

In OWL relationships among classes is defined by using restrictions. A hasSubClass relation is built-in in OWL. For other relations, property restriction is used. The restrictions are added using an object restriction creator window as shown in Fig.3. Select the class from class hierarchy window, select the name of property that we want to restrict and then select the restriction filler which is again a destination class. For example, for a 'data structures' class, select the object property 'hasOperation' and 'Operation' as restriction filler. The same procedure is repeated for all the properties.

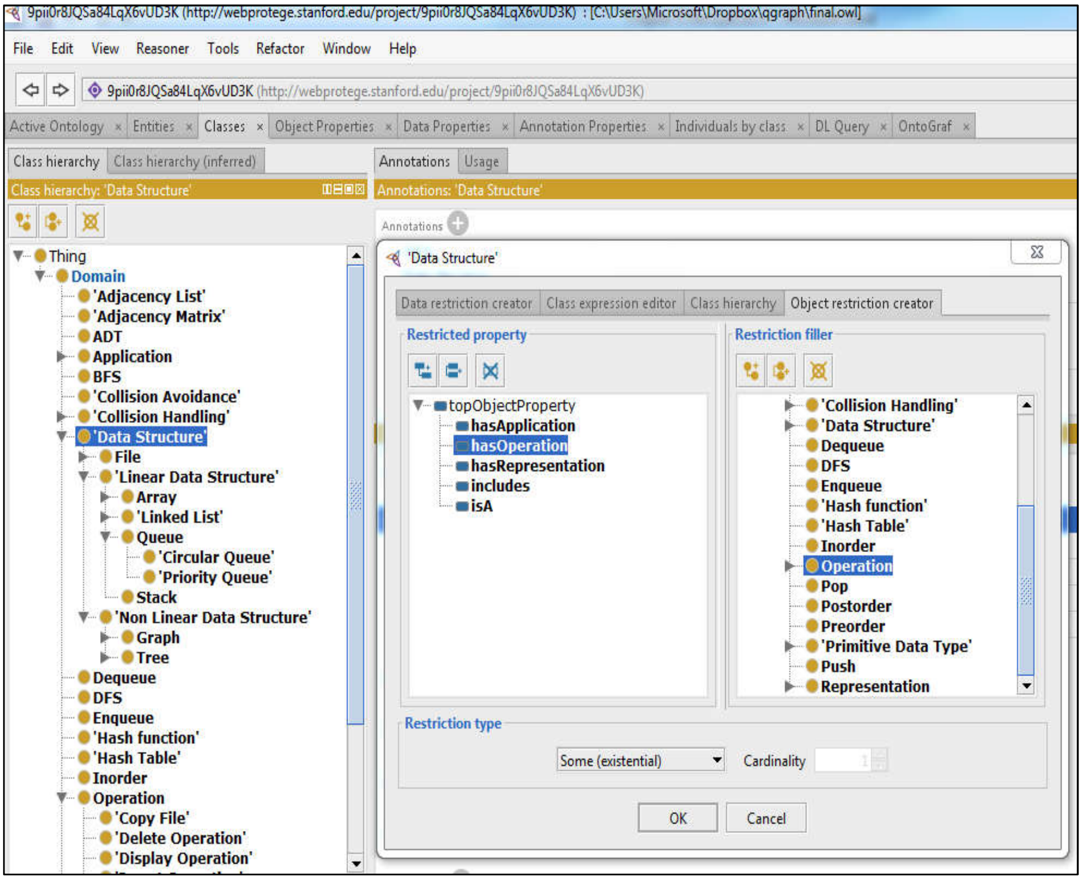


Figure 3 Object Restriction Creator Window

Step 3: Add Annotation properties

OWL allows classes, properties, individuals and the ontology itself to be annotated with various pieces of information/meta-data. We use annotations to provide synonyms to class labels and link/property names. Synonyms are given as alternate labels in annotation window as shown in Fig. 4. Enter the alternate labels/synonyms in the space provided. One or more synonyms can be provided.

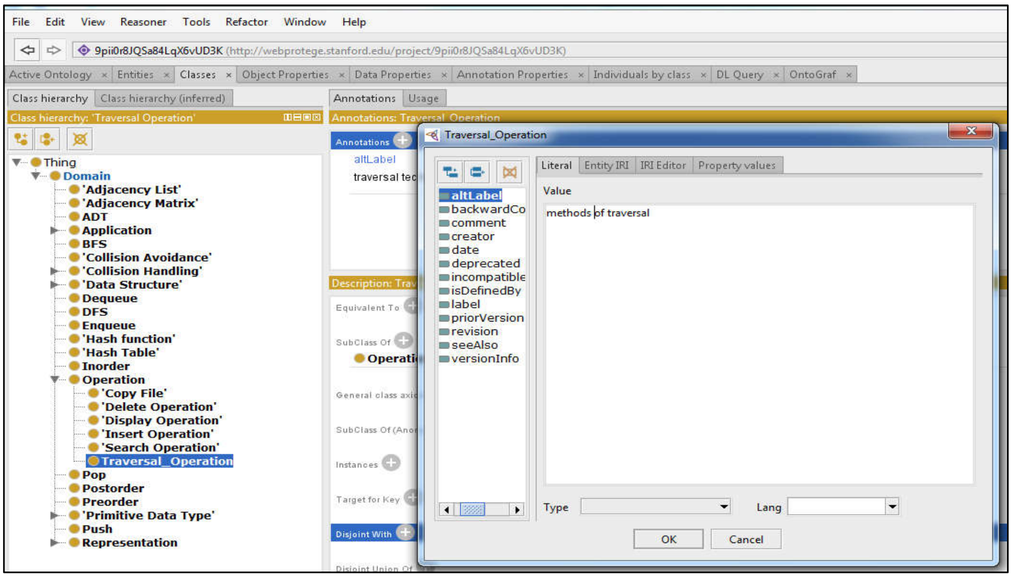


Figure 4. Adding Annotations

Step 4: Save the file as 'data structures.owl'.

Graphical view can be seen in the onto graph window. Partial view of the graph as a ontology is shown in Fig. 5.

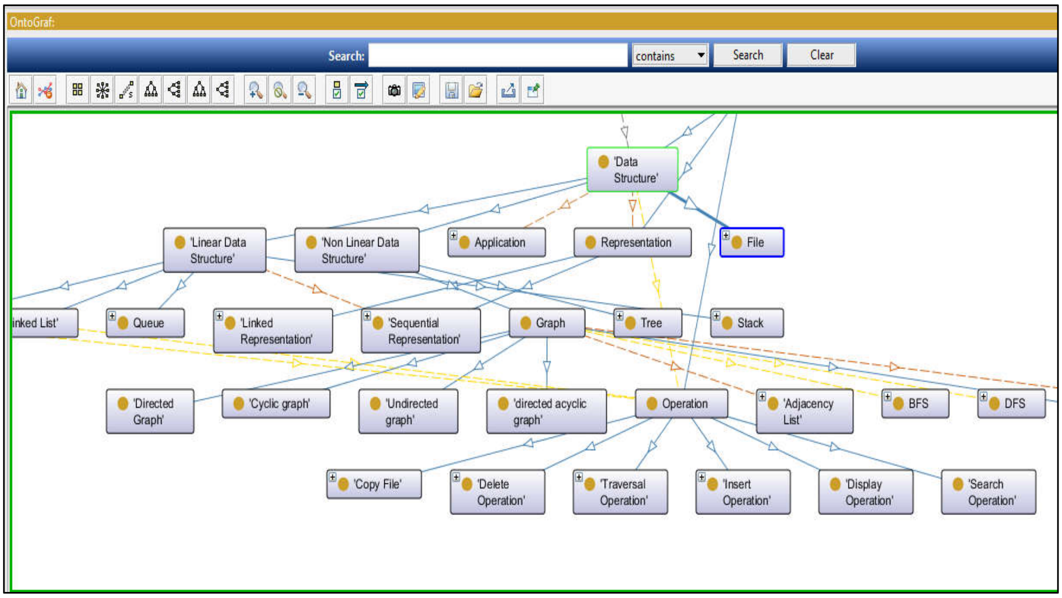


Figure 5 Partial Graphical View of Ontology

Appendix B Data Structure Course Syllabus

University of Mumbai			
Class: S.E.	**Branch:** Computer Engineering	**Semester:** III	
Subject: Data Structure and Files (Abbreviated as DSF)			
Periods per Week (each 60 min)	Lecture	04	
	Practical	02	
	Tutorial	--	
		Hours	Marks
Evaluation System	Theory	03	100
	Practical and Oral	02	25
	Oral	---	--
	Term Work	---	25
	Total	05	150

Pre-requisites: A Course in Object Oriented Programming Language such as **(JAVA)**

Module	Contents	Hours
1	**Introduction to Data Structures:** • Definition • The Abstract Data Type(ADT) • Arrays • Strings • Recursion	05
2	**File Handling:** • File Organization • Types of files • File operations	04
3	**Sorting and Searching:** **A. Sorting** • Insertion sort • Selection sort • Exchange sort (Bubble, Quick) • Merge sort • Heap sort **B. Searching:** • Linear Search • Binary Search • Hashing Technique and collision handling	07
4	**Stack:** • The Stack as an ADT • Representation	03

	• Stack Operations • Applications	
5	**Queue:** • The Queue as an ADT • Representation • Queue Operations • Circular and Priority Queues • Applications	03
6	**Linked List:** • The Linked List as an ADT • Operation on Linked List • Linked Stacks and Queues • The Linked List as a Data Structure • Array implementation of Linked List • Linked List using Dynamic variable • Comparison of Dynamic and Array implementation of Linked List • Doubly Linked List • Circular Linked List	10
7	**Trees:** • Basic tree concepts • Binary Tree Operations and Applications • Binary Tree representations • Binary Tree Traversals • Threaded Binary Tree • The Huffman Algorithm • Binary Search Tree Implementation • Expression Trees • Introduction of multiway tree (B-Tree, B+ Trees, AVL Tree)	12
8	**Graphs:** • Graph as an ADT • Graph Representation • Graph Traversal (Depth First Search, Breadth First Search)	04

TERM WORK

Term work should consist of graded answer papers of the test and 12 implementations using object oriented constructs & concepts. Students are expected to build their own classes and methods. Built-in classes are not to be used (preferably). Each student is to appear for atleast one written test during the Term. Each implementation must consist of Problem Statement, Brief Theory, Algorithm, Flowchart and Conclusion.

Topics for Implementation

- String functions, Recursion and Files
- Implementations of Stack & Queues (Circular & Priority)
- Implementation of Linked Lists (Singly & Doubly)
- Implementation of Searching & Sorting methods
- Implementation of Binary Tree
- Implementation of Graph

Text Books:

1. Y. Langsam, M.J. Augenstein and A.M. Tanenbaum, "Data Structures Using Java", Pearson Education .

2. R.F. Gilberg and Behrouz A. Forouzan, "Data Structure: A Pseudocode Approach with C", Thomson Edition .

3. Michael Goodrich & Roberto Tamassia, "Data structures and algorithms in Java^{JM}", Second Edition, Wiley India Edition.

Reference Books:

- John R. Hubbard and Hurry "Data structures with Java", Pearson Education.
- Mark Allen Weiss, "Data Structure & Algorithm Analysis in C++", Third Edition,Pearson Education.

- Sanjay Pahuja, “A Practical to Data Structure & Algorithms”, First Edition, New Age International Publisher.
- Alan L. Tharp “File organization and processing”, Amazon Publication.

Appendix C *Data Structures* Questions Considered for IQuE Analysis

The questions are taken from various universities from India and abroad. A sample of 10 questions from each set of 1000 samples are given below.

Set 1

1) Write a program to implement insertion sort. Show the passes of insertion sort for the following input.
2) Construct binary tree for the following preorder and inorder traversal sequences.
3) Convert given fully parenthesized infix expression to postfix expression.
4) Explain how a polynomial is represented using array with one example.
5) Given an array containing the digits 71808294, show how the order of the digits changes during each step of [a] insertion sort, [b] selection sort, [c] merge sort, [d] quick sort (using the array-based quick sort, and always choosing the last element of any subarray to be the pivot), and [e] heapsort (using the backward min-heap). Show the array after each swap, except in insertion sort. For insertion sort, show the array after each insertion.
6) How long does it take to determine if an undirected graph contains a vertex that is connected to no other vertex [i] if you use an adjacency matrix; [ii] if you use an adjacency list.
7) How many number of distinct minimum spanning trees is there for the weighted graph below?
8) Given an expression string *exp*, write a program to examine whether the pairs and the orders of "{","}"," (",")"," [","]" are correct in *exp* (stack).
9) Check whether a given binary tree is heap?
10) Given a doubly linked list, write a function to sort the doubly linked list in increasing order using merge sort.

Set 2

1) Given a string consisting of opening and closing parenthesis, find length of the longest valid parenthesis substring. (stack)

2) Explain the following with the help of a diagram: i) Insert a node at the beginning, at the end and at the specified position of the singly linked list. ii) Deleting the first node, last node and a node from a specified position in case of a doubly linked list. iii) Deleting the first node, last node and a node from a specified position in case of circular linked list.

3) Write an algorithm to convert valid infix expression to prefix expression and hence convert: (A t (((8- C) * (D- E) + F)/G) $ (H- J).

4) It is generally said that searching a node in a binary search tree is more efficient than that of a simple binary tree. Why?

5) Write an algorithm for inserting an item in a deque.

6) Write DFS algorithm to traverse a graph. Apply same algorithm for the graph given above (Figure I) by considering node 1 as starting node.

7) What do you mean by hashing and collision? Discuss the advantages and disadvantages of hashing over other searching techniques?

8) Write a program for a singly linked circular list which reverses the links

9) Write an algorithm to delete duplicate numbers from a linear array

10) Differentiate between B tree and B+ tree.

Appendix D Sample Assessment Instruments from Mumbai University

Data Structures (Semester III)

(3 Hours) [**Total Marks : 100**

N.B. (1) Question No. 1 is **compulsory**.
(2) Attempt any **four** questions out of remaining **six** questions.
(3) Assume **suitable** data whenever **required** but **justify**.
(4) **Illustrate** answers with **neat** sketches whenever **required**.

1. (a) Write a program in Java to implement circular Queue using array. **10**
   (b) Explain linear and non-linear data structure with example. **5**
   (c) Explain practical applications of trees. **5**

2. (a) Write a program in Java to copy content of a file to another file using command line argument. **10**
   (b) What are the advantages of linked list over array ? Write a program in Java to implement stack using linked list. **10**

3. (a) Write a program to implement insertion sort using Java. Show passes of insertion sort for the following input 15, 23, 22, 11, 44. **10**
   (b) Give different searching techniques. Write a program to implement binary search. **10**

4. (a) Explain different representations of graph. State advantages and disadvantages of each representation. **10**
   (b) Write a Java program to create a binary search tree. Show BST for the following input : 10, 05, 14, 22, 17, 01, 08. **10**

5. (a) Explain the method of Huffman Encoding. Apply Huffman encoding method for the sentence "MALAYALAM". Give Huffman code for each symbol. **10**
   (b) Hash the following in a table of size. 11 use any two collision resolution techniques. 23, 55, 0, 71, 67, 23, 100, 18, 10, 90, 44. **10**

6. (a) Write ADT for stack. Give applications of stack. **10**
   (b) Explain Priority Queue. **5**
   (c) Write a program in Java to create a linked list and perform the following operations : **5**
      (i) Insert into list
      (ii) Search for data
      (iii) Delete from list
      (iv) Display the list.

7. Write short notes on (any **two**) :— **20**
   (a) Tree Traversal Algorithms
   (b) Merge sort with example
   (c) AVL tree and multiway tree.

Operating Systems (Semester IV)

(3 Hours) **[Total Marks : 100**

N.B. : (1) Question No. **1** is **compulsory**.
(2) Attempt any **four** questions out of remaining **six** questions.
(3) Assume **suitable** data wherever **necessary** and mention it **clearly**.

1. (a) What is system call ? Explain any five system calls. **5**
(b) What is memory partitioning ? Explain different memory partitioning techniques. **5**
(c) Draw and explain five state process model. **5**
(d) Explain effect of page size on performance. **5**

2. (a) What is deadlock ? Explain necessary and sufficient conditions to occur deadlock. Explain deadlock avoidance, prevention and detection. **10**
(b) The requested tracks in the order received are – **10**
54, 57, 40, 20, 80, 120, 150, 45, 180
Apply the following disk scheduling algorithm starting track at 90.
(i) FCFS (ii) SSTC (iii) CSCAN.

3. (a) Consider the following set of processes with CPU burst time given in table. **10**

Process	Burst Time	Arrival Time
P_1	08	00
P_2	10	01
P_3	05	00
P_4	06	02

(i) Draw Gantt chart for preemptive SJF, non-preemptive SJF and Round Robin (Quantum = 02).
(ii) Calculate average waiting time and average turn around time.
(b) Explain different file access methods. **10**

4. (a) What is mutual exclusion ? Explain semaphore used for mutual exclusion. **10**
(b) Explain various I/O buffering techniques. **10**

5. (a) What is paging and segmentation ? Explain LRU and FIFO page replacement policis for given page frame sequences. Page frame size is 4. **10**
2, 3, 4, 2, 1, 3, 7, 5, 4, 3, 2, 3, 1
Calculate Page hit and Page miss.
(b) Explain critical section problem and its different solutions. **10**

6. (a) Explain LINUX concurrency control mechanism. **10**
(b) What are the characteristics of real-time operating system ? Explain in brief real time scheduling. **10**

7. Write short notes on :– **20**
(a) User Level and Kernel Level Threads
(b) Process Control Block (PCB)
(c) Unix File System
(d) Virtual Memory.

Computer Networks (Semester V)

(3 Hours) **[Total Marks : 80**

N.B. : (1) Question No. **1** is **compulsory**.
(2) Attempt any **four** questions out of remaining **six** questions.
(3) Assume **suitable** data wherever **required**.

1. (a) What is the need for layering ? Discuss the design issues for layers. **10**
   (b) Explain the ALOHA protocol. Compare the performance of Pure Aloha v/s Slatted Aloha at low load and high load. **10**

2. (a) Explain different framing methods. What are the advantages at variable length frames over fixed length frames ? **10**
   (b) Explain : FDMA, TDMA and CDMA. **10**

4. (a) What are transport service primitive ? **10**
   (b) How TCP controls the congestion, explain in detail. **10**

5. (a) Differentiate between the following :– **10**
   (i) Protocol and Interface
   (ii) Connectionless and connection oriented service.
   (b) What are different types of routing ? Explain any one in detail. **10**

6. (a) Explain the different factors associated with quality of service in inter network. **10**
   (b) Describe the IPV4 header format in detail. **10**

6. (a) Explain the different factors associated with quality of service in inter network. **10**
   (b) Describe the IPV4 header format in detail. **10**

7. Write short notes on (any **four**) :– **20**
   (a) SONET
   (b) Layer 2 v/s Layer 3 switching
   (c) Bluetooth
   (d) CIDR
   (e) Berkely Socket.

System Programming and Compiler Construction (Semester VI)

(3 Hours) **[Total Marks : 100**

N.B. (1) Question No. **1** is **compulsory**.
(2) Attempt any **four** questions out of remaining **six** questions.
(3) Assume **suitable** data if **necessary** and justify the **same**.
(4) **Figures** to **right** indicate **full** marks.

1. (a) Differentiate between Application program and system program. **5**
Indicate the order in which following system programs are used, from developing program upto its execution.
Assemblers, loaders, linker, macroprocessor, complier, editor.
(b) Eliminate left recursion present in following grammer (Remove Direct and Indirect recursion both) **5**
$S \longrightarrow Aa \mid b$
$A \longrightarrow Ac \mid Sd \mid \in$
(c) What is activation record ? Draw the diagram of General Activation record and explain the purpose of different fields of an activation record. **5**
(d) What are the different functions of loader explain in brief. **5**

2. (a) With reference to assembler explain the following tables with suitable examples : **10**
(i) POT
(ii) MOT
(iii) ST
(iv) LT
(b) Let L be the language consisting of strings of a's and b's having same number of a's and b's :
(i) Construct LL (1) grammer for L **4**
(ii) Construct a predictive parsing table for the grammer obtained in (i). **6**

3. (a) Explain different psudo-ops used for conditional macro expansion, along with example. **10**
(b) What are the different phases of compiler ? Illustrate compilers internal presentation of source program for following statement after each phase **10**
position = initial + rate * 60

4. (a) Explain working of a direct linking loader with a proper example. Clearly show the entries in different databases built by the direct linking loader. **10**
(b) Generate three address code for given expression **10**
```
while (a < b) do
    if (c < d) then
        x = y + z
    else
        x = y - z
```

5. (a) For the given grammer below, construct operator precedence relations matrix, assuming $*$, + are binary operators and id as terminal symbol and E as non terminal symbol. **10**

$E \longrightarrow E + E$
$E \longrightarrow E * E$
$E \longrightarrow id$

Apply operator precedence parsing algorithm to obtain skeletal syntax tree for the statement

id + id $*$ id

(b) Explain role of code optimization in compiler designing with suitable example. **10**

6. (a) For regular expression (a | b)* abb construct NFA and construct it into DFA. **10**
(b) With reference to stack allocation and heap allocation explain runtime storage organization. **10**

7. (a) Write a note on JAVA compiler environment. **5**
(b) Explain synthesized and Inherited attributes used in syntax directed definitions. **5**
(c) Explain DAG. **5**
(d) Find first and follow set for given gramer below : **5**

$E \longrightarrow TE'$ $\quad E' \longrightarrow + TE' \mid \epsilon$
$T \longrightarrow FT'$ $\quad T' \longrightarrow * FT' \mid \epsilon$
$F \longrightarrow (E)$ $\quad F \longrightarrow id$

System Security (Semester VII)

(3 Hours) [Total Marks : 100

N.B. (1) Question No. **1** is **compulsory**.
(2) Attempt any **four** questions from remaining **six** questions.
(3) Assume suitable **data** if **required.**

1. (a) Explain different Birthday problems. **5**
   (b) What are the key principles of security ? **5**
   (c) Compare and contrast SHA-T and MD-5 **5**
   (d) Explain the Honey Pots. **5**

2. (a) How flaw in TCP/IP can cause operating systems to become Vulnerable ? Also Explain how Kerberos are used for user authentication in windows. **10**
   (b) For the given values p = 19, q = 23 and e = 3 find n, $\phi(n)$ and d using RSA algorithm. **10**

3. (a) What is Buffer overflow and Incomplete mediation in Software Security ? **10**
   (b) Explain one-time initialization process and processes in each round of Advanced Encryption Standard. **10**

4. (a) What is a denial of service attack ? What are the way in which an attacker can mount a DOS attack on the system ? **10**
   (b) Compare Packet Sniffing and Packet Spoofing. Explain the session hijacking attack. **10**

5. (a) Explain Multiple level Security Model. Also explain Multilateral Security. **10**
   (b) What is Malware ? Explain Salami and Linearization attacks. **10**

6. (a) Explain software Reverse Engineering. Also Explain Digital Rights Management. **10**
   (b) Describe the different types of IDS and their limitations. **10**

7. Write short notes on (any **four**) :— **20**
   (a) CAPTCHA
   (b) Access Control Matrix
   (c) Covert Channel
   (d) Firewall
   (e) RC4.

Distributed Computing (Semester VIII)

(3 Hours) [Total Marks : 100

N.B. : (1) Attempt any **five** questions.
(2) **All** questions carry **equal** marks.

1. (a) Explain Absolute Ordering and Causal Ordering process with the help of example for many to many communication. **10**
   (b) Explain RPC system model in detail. **10**

2. (a) Discuss the need of the co-ordinator. Also give any one algorithm for co-ordinator selection. **10**
   (b) What is the difference between strict consistency model and sequential consistency model ? How sequential consistency model is implemented in DSM ? **10**

3. (a) What is physical clock synchronization ? Explain any one algorithm in detail. **10**
   (b) Explain deadlock avoidance algorithm in a distributed system. **10**

4. (a) How does light weight RPC work in cross domain architecture ? **10**
   (b) Discuss file caching for distributed system. **10**

5. (a) Discuss how processes and resources are bound together. How does process migration take place in Heterogeneous environment ? **10**
   (b) What are the good features of a Distributed File Systems ? Explain file sharing semantics of it. **10**

6. (a) Justify the need of stateful and stateless server in RPC management. **10**
   (b) How are failures handled in message passing system in distributed systems ? **10**

7. Write any **two** of the following :— **20**
   (a) Distributed Transaction Management
   (b) System Oriented Names for Distributed Computing
   (c) Load Balancing Issues
   (d) Distributed System Models.

Appendix E System Usability Scale

The System Usability Scale (SUS) is a simple, ten item scale giving a global view of subjective assessments of usability of product. The 5-point Likert's scale parameters represent the ranges from **Strongly Disagree** to **Strongly Agree**.

Strongly Disagree→Disagree →Neutral →Agree →Strongly Agree.

Each of the question is associated with a space where user can provide open ended answer as a justification to the option they have selected.

Following are the ten questions in the form.

- I think that I would like to use IQuE frequently
- I found IQuE unnecessarily complex
- I thought that IQuE was easy to use
- I think that I would need the support of a technical person to be able to use IQuE
- I found various functions in IQuE were well integrated
- I thought there was too much inconsistency in IQuE
- I would imagine that most people would learn to use IQuE very quickly.
- I found IQuE very cumbersome to use
- I found very confident using IQuE
- I needed to learn a lot of things before I could get going with IQuE

Note: Same survey form was used for QTagger and AIGen also just by replacing the tool name

Appendix F Sample AIS given to teachers

AIS for class IX science
Assessment: SA-I
Class: IX
Name of Course: Science
Duration: 3 hours

For the whole of Assessment Instrument

Total Marks: 90
Number of sections: 2

Cognitive Level Distribution:

Level	**Lower Bound (%)**	**Upper Bound (%)**
Recall	0	100
Understand	0	100
Apply	0	100
Analyze	0	100
Create	0	100

Difficulty Level Distribution:

Level	**Lower Bound (%)**	**Upper Bound (%)**
Easy	15	15
Medium	70	70
High	15	15

Content Distribution:

Topics	**Lower Bound (%)**	**Upper Bound(%)**
1. Matter its Nature & Behaviour • Matter in our surrounding • Is matter around us pure	15	15
2. Organisation in living world • The Fundamental Unit of Life • Tissues	25	25
3. Motion, Force & Work • Motion • Force and Law of Motion • Gravitation	35	35

4. Food • Improvement • Resources	15	15

For section A:

Number of subsections: 3

Number of items: 24

Section A-1

Number of items: 3

Item Type:

Type of Question	Lower Bound (%)	Upper Bound (%)
Very short Answer	100	100

Section A-2

Number of items: 16

Item Type:

Type of Question	Lower Bound (%)	Upper Bound (%)
Short Answer	100	100

Section A-3

Number of items: 5

Item Type:

Type of Question	Lower Bound (%)	Upper Bound (%)
Long Answer	100	100

For section B:

Number of items: 18

Number of sub-items in each item: NIL

Item Type:

Type of Question	Lower Bound (%)	Upper Bound (%)
Objective(MCQ)	100	100

Explanation:

1) **Assessment Instrument:** This refers to the question paper developed for a specific examination

2) **Item:** Items are the questions in the assessment instrument along with its associated information such as marks, cognitive level, difficulty level, etc.

3) **Cognitive level:**

   The cognitive domain involves knowledge and the development of intellectual skills. This includes the recall or recognition of specific facts, procedural patterns, and concepts that serve in the development of intellectual abilities and skills.

   Bloom's taxonomy can be used to define the cognitive level of an item in an assessment instrument. There are six major categories, which are listed in order below, starting from the simplest behaviour to the most complex. The categories can be thought of as degrees of difficulties. That is, the first one must be mastered before the next one can take place. Each item in an instrument can be associated with a corresponding Bloom's level that the student has to achieve in order to answer that particular item.

   - **Recall**: Recall data or information.

     ***E**.g. List four reasons to support that water is a compound and not a mixture*

   - **Understand**: Understand the meaning, translation, interpolation, and interpretation of instructions and problems. State a problem in one's own words

     ***E**.g. Derive the relation between force and acceleration. Define one unit of force*

   - **Apply**: Use a concept in a new situation or unprompted use of an abstraction. Applies what was learned in the classroom into novel situations in the work place.

     ***E**.g. A stone dropped from a window reaches the ground in 0.5 seconds.Calculate its speed just before it hits the ground.*

- **Analyze**: Separates material or concepts into component parts so that its organizational structure may be understood. Distinguishes between facts and inferences.

  E*.g. The velocity time graph for an object is shown in the following figure. State the kind of motion that the above graph represents. What does the slope of the graph represent?*

- **Evaluate**: Grasp meaning, explain, interpret, translate, paraphrase. Judge value based on criteria, make decisions. (assess, contrast, compare, evaluate, decide, etc)

  E.g. Distinguish among true solution, suspension and colloid under the following heads (i) stability (ii) Filterability (iii) Type of mixture

- **Create**: Generate new ideas, products or ways of looking at things

  E*.g. On sports day, Rohit fell on the ground and shrieked in pain as he had cramps in his right leg. Devise a method to treat this problem so that Rohit will be able to resume his sports.*

4) **Item Types:** There are 4 different types of items considered in this document. They are classified as:
   - **Very short answer (VSA)**: To be answered in one word or one sentence. Mostly one mark questions.
   - **Short answer (SA)**: To be answered in 30 to 50 words. Mostly 2 marks or 3 marks questions
   - **Long answer (LA)**: To be answered in 70 words. Mostly 5 marks questions
   - **Objective (MCQs)**: 1 mark multiple choice questions
5) **Difficulty Level:** There are three levels of difficulties which are defined as follows
   - **Low:** Questions are directly on the text book content
   - **Medium:** Questions are advanced questions on the text book content or from the reference books on the relevant topic.
   - **High:** Questions are challenging to solve

Appendix G: The Interview Questionnaire

What was the strategy followed? Did you put the questions first and then checked for conformity to AIS? or Vice Versa?

Q3. How do you decide the distribution if bound values are not specified for Difficulty Level?

Q4. How do you decide the distribution if bound values are not specified for Content?

Q5. How do you decide the distribution if bound values are not specified for Question Type?

Q6. Given the chapter has Learning objectives, how do you decide what kind of questions are appropriate for each?

Q7. How do you distribute the questions among the subtopics under the main topics?

Q8. Did you face any difficulty in understanding any part of AIS?

Q9. Do you feel that any component of AIS needs to be changed or represented in a different way?

Appendix H: How to use IQuE for other courses?

- A syllabus of the course is required.
- Create a domain ontology of the course from the syllabus. The ontology of a course for IQuE represents syllabus contents and its structure. Apart from identifying valid nodes, links etc., there are strong requirements regarding ontology structure. It should facilitate the mapping of identified concepts and relations from LOs and questions to the nodes and links in ontology. These are explained in section 6.1.1 of thesis and has to be followed for creating ontology suitable to IQuE. This is the most important and most cumbersome part of using IQuE for testing the quality of AI developed for any course. One of the factors determining the accuracy of AI is the completeness and correctness of domain ontology. Ontology can be created using Protégé tool (Refer to Appendix A for detailed explanation).
- Apart from the available action words in each level of the Bloom's Taxonomy, some domain specific action verbs are added into the dictionary in order to increase the accuracy of IQuE in identifying the cognitive level of LO or a question. For a Data Structures course, we have added action verbs such as 'write a program', 'devise an algorithm', 'show a stepwise execution' in 'Apply', 'Evaluate' and 'understand' etc.
- IQuE needs a set of LOs. If LOs are not available with the course content, then create LOs. These LOs can be changed for a given course. But for a given set of LOs, it must be ensured that it complies with our requirements. We need to get the LOs ready for IQuE.
    - The set of LOs are written in a simple text file saved in *filename.txt*.
    - LOs must be machine parsable. IQuE will not be able to analyze certain types of LOs in which there is no indication of any concepts involved.
    - They are written one per line and there are no restrictions on the number of LOs.
    - Any character other than text will be removed by IQuE at the time of preprocessing.
- Now you are ready to use IQuE. For testing an AI, put the list of questions in a file with a .txt extension. and they should also comply with the requirement of IQuE.
    - Questions must be machine parsable. IQuE will not be able to analyze certain types of questions in which there is no indication of any concepts involved.

- Questions are written one per line and there are no restrictions on the number of questions.
- The questions can be of any type such as short answer, long answer, essay type, MCQs etc.
- Any character other than text will be removed by IQuE at the time of preprocessing.

Getting ready to use IQuE to test AI:

It is advisable you evaluate the performance of IQuE when using first time by doing a trial with the expert teachers and look at the confusion matrix for some mistakes. For this, collect a few questions. Give these questions for expert evaluations and get their ratings for alignment values in the range of 0 to 1 for each pair of LO and question and store it in an excel file. Put the same set of questions in a file with a .txt extension and run the IQuE. Give the expert's file to form the confusion matrix and get the feedback (Refer to section 7.1.2 for details of generation of Confusion Matrix). Fine tune the system by revising the domain ontology and LO representations or by adding domain specific action words and rerun the IQuE. Once you are satisfied with results, IQuE is ready for use to test the AI.

References

Ahmed, A., Nisa, A., & Zarif, T. (2013). Table of specification development and usage: an overview. Interdisciplinary Journal of Contemporary Research Business, 4(12), 354-361.

Aldabe, I., de Lacalle, M., Maritxalar, M., Martinez, E., & Uria, L. (2006). Arikiturri: an automatic question generator based on corpora and nlp techniques. In Intelligent Tutoring Systems (pp. 584-594). Springer Berlin/Heidelberg.

Aldabe, I., de Lacalle, M., Maritxalar, M., Martinez, E., & Uria, L. (2006). Arikiturri: an automatic question generator based on corpora and nlp techniques. In Intelligent Tutoring Systems (pp. 584-594). Springer Berlin/Heidelberg.

Amer, A. (2006). Reflections on Bloom’s revised taxonomy. Electronic Journal of Research in Educational Psychology, 4(1), 213-230.

Anderson, P., & Morgan, G. (2008). Developing tests and questionnaires for a national assessment of educational achievement (Vol. 2). World Bank Publications. Chicago

Assessment Handbook, (2012), University of Ulster

Authority, S. Q. (2009). Guide to assessment.

Banerjee, R., & Muley, V. P. (2007). Engineering education in India. Report to Energy Systems Engineering, IIT Bombay, sponsored by Observer Research Foundation, September, 14.

Bechhofer, S., & Matentzoglu, N. (2014). The OWL API: An Introduction. COMP60421: Ontology Engineering for the The Semantic Web.

Biggs, J. (2003). Aligning teaching and assessing to course objectives. Teaching and learning in higher education: New trends and innovations, 2, 13-17.

Black, P., & William, D. (1998). Inside the black box: Raising standards through classroom assessment. Granada Learning.

Blackboard LMS: https://help.blackboard.com/en-us/Learn/9.1_SP_10_and_SP_11/Instructor/070_Tests_Surveys_Pools/018_Adding_Question_Metadata, Referred on 15/05/2017.

Brooke, J. (1996). SUS-A quick and dirty usability scale. Usability evaluation in industry, 189(194), 4-7.

Brown, S. & Glasner, A. (1999) Assessment Matters in Higher Education, Buckingham: Open University Press.

Brown, S. (2004). Assessment for learning. Learning and Teaching in Higher Education, 1(1), 81-89.

Brown, S., Rust, C., & Gibbs, G. (1994). Strategies for diversifying assessment in higher education. Oxford: Oxford Centre for Staff Development.

"CBSE Teachers' Manual, Continuous and Comprehensive Manual, Class IX and X. http://www.cbse.nic.in/cce/index.html", Referred on 15/05/2017

Chan Kan Kan. (2010). Using test blueprint in classroom assessments: why and how? International Association for Educational Assessment (IAEA).

Chi, Y. (2010). Developing curriculum sequencing for managing multiple texts in e-learning system. In Proceedings of International Conference on Engineering Education. (ICEE) July 18–22, 2010, Gliwice, Poland.

Chinn, D., Prins, P., & Tenenberg, J. (2003). The role of the data structures course in the computing curriculum. Journal of Computing Sciences in Colleges, 19(2), 91-93.

Chowdhury, G. G. (2003). Natural language processing. Annual review of information science and technology, 37(1), 51-89.

Chung, H., & Kim, J. (2016). An ontological approach for semantic modeling of curriculum and syllabus in higher education. International Journal of Information and Education Technology, 6(5), 365.

Cooper, H. M. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. Knowledge in Society, 1(1), 104-126.

Currier, S. (2007). Assessment item banks and repositories. JISC CETIS.

D. Kennedy, A. Hyland, and N. Ryan. (2006). Writing and Using Learning Outcomes: a Practical Guide. [Online]. In the Bologna Handbook. Available: http://www.bologna.msmt.cz/files/learning-outcomes.pdf.

Davis, B. G. (2002). Quizzes, tests, and exams. University of California, Berkeley. From Teaching Tips on Honolulu community College, University of Hawai (http://honolulu. hawaii. edu/intranet/).

Davis, B. G. (2009). Tools for teaching. John Wiley & Sons.

Deane, P., & Sheehan, K. (2003). Automatic Item Generation via Frame Semantics: Natural Language Generation of Math Word Problems.

Denny, P., Luxton-Reilly, A., & Simon, B. (2009, January). Quality of student contributed questions using PeerWise. In Proceedings of the Eleventh Australasian Conference on Computing Education-Volume 95 (pp. 55-63). Australian Computer Society, Inc..

Derek Rowntree (1977). Assessing students-How shall we know them. Harper & Row publications.

Educational Design Research Part A: An Introduction (2013): http://international.slo.nl/publications/edr/ , Referred on 15/05/2017

Fok, A. W. P., & Ip, H. H. S. (2007). Educational ontologies construction for personalized learning on the web. In Evolution of teaching and learning paradigms in intelligent environment (pp. 47-82). Springer Berlin Heidelberg.

Gibbs, G. & Rowntree, D. (1999) Teaching in Higher Education: theory and evidence. Chapter 3: Designing Assessment, Milton Keynes: The Open University.

Gibbs, G. (2010). Using assessment to support student learning. Leeds Met Press.

Gierl, M. J., Lai, H., & Turner, S. R. (2012). Using automatic item generation to create multiple-choice test items. Medical education, 46(8), 757-765.

Goto, T., Kojiri, T., Watanabe, T., Iwata, T., & Yamada, T. (2010). Automatic generation system of multiple-choice cloze questions and its evaluation. Knowledge Management & E-Learning: An International Journal (KM&EL), 2(3), 210.

Goto, T., Kojiri, T., Watanabe, T., Iwata, T., & Yamada, T. (2010). Automatic generation system of multiple-choice cloze questions and its evaluation. Knowledge Management & E-Learning: An International Journal (KM&EL), 2(3), 210.

Graesser, A. C., & Person, N. K. (1994). Question asking during tutoring. American educational research journal, 31(1), 104-137.

Handbook on Testing and Grading. (1991). University of Florida.

http://www.imsglobal.org/question/qtiv1p2/imsqti_oviewv1p2.htm, Referred on 15/05/2017.

John Heywood, (2003). Assessment in higher education, John Wiley & Sons, Ltd.

Junoh, A. K., Muhamad, W. Z., Abu, M. S., Jusoh, M. S., and Desa, A. M. (2012). Classification of Examination Marks according to Bloom's Taxonomy by using Binary Linear Programming. International Conference on Innovation and Information Management (ICIIM)

Kan, C. K. (2010). Using test blueprint in classroom assessments: why and how. International Association for Educational Assessment.

Karthwohl, D. R., & Anderson, W. (2002). A Revision of Bloom's Taxonomy: An Overview Theory into Practice. The Ohio State University.

Kiili, K. (2005). Digital game-based learning: Towards an experiential gaming model. The Internet and higher education, 8(1), 13-24.

Liddy, E. (1998). Enhanced text retrieval using natural language processing. Bulletin of the American Society for Information Science, 24(4), 14-16

Lin, D. (1998, July). An information-theoretic definition of similarity. In ICML (Vol. 98, No. 1998, pp. 296-304).

M. Agarwal. (2008) Guidelines for Setting a Good QP. Available:
http://www.egyankosh.ac.in/handle/123456789/26480, Referred on 15/05/2014

M. D. Gall, "The use of questions in teaching. Review of educational research," 707-721, 1970.

M. Ronchetti and J. Sant, J., "Curriculum Management and Review: an ontology-based solution", 2007

Mamta Agarwal, “Examination Reform Initiatives in India“, Journal of Indian Education, Vol. XXXI, No. 1,May’2005.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014, June). The stanford corenlp natural language processing toolkit. In ACL (System Demonstrations) (pp. 55-60).

Marbach-Ad, G., & Sokolove, P. G. (2000). Can undergraduate biology students learn to ask higher level questions?. Journal of Research in Science Teaching, 37(8), 854-870.

McClarty, K. L., Orr, A., Frey, P. M., Dolan, R. P., Vassileva, V., & McVay, A. (2012). A literature review of gaming in education. Gaming in education, 1-36.

McKenney, S. (2001). Computer-based support for science education materials developers in Africa: Exploring potentials. Doctoral dissertation. Enschede, the Netherlands: University of Twente.

McKenney, S. (2001c). The CASCADE-SEA research Website [Online]. Available: http://projects.edte.utwente.nl/cascade/seastudy/

Mizoguchi, R. (2004). Tutorial on ontological engineering Part 2: Ontology development, tools and languages. New Generation Computing, 22(1), 61-96.

Mizoguchi, R., & Bourdeau, J. (2016). Using Ontological Engineering to Overcome AI-ED Problems: Contribution, Impact and Perspectives. International Journal of Artificial Intelligence in Education, 26(1), 91-106.

Moodle LMS: http://docs.moodle.org/24/en/Question_bank, July 2014

Mumbai University Syllabus for computer Engineering
http://www.mu.ac.in/syllabus/4.61%20Computer%20Engg.pdf , Referred on 15/05/2017

Nalawade, G., & Ramesh, R. (2016, December). Automatic Generation of Question Paper from User Entered Specifications Using a Semantically Tagged Question Repository. In Technology for Education (T4E), 2016 IEEE Eighth International Conference on (pp. 148-151). IEEE.

Natalya F. Noy and Deborah L. McGuinness. (2000). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University

National Research Council. (2001). Knowing what students know: The science and design of educational assessment. National Academies Press.

National Research Council. (2001). Knowing what students know: The science and design of educational assessment. National Academies Press.

Nitko, A. J. (2001). Educational assessment of students. Prentice-Hall, Inc., PO Box 11071, Des Moines, IA 50336-1071.

Noy, N. F., Crubézy, M., Fergerson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., & Musen, M. A. (2003). Protege-2000: an open-source ontology-development and knowledge-acquisition environment. In AMIA Annu Symp Proc (Vol. 953, p. 953).

O'Rourke, M. (2011). The development of online assessment in the Moodle Virtual Learning Environment (VLE) as a replacement for traditional written assessment.

Paas, F., Renkl, A., & Sweller, J. (2003). Cognitive load theory and instructional design: Recent developments. Educational psychologist, 38(1), 1-4.

Parchure, M. (2006). Question Banking. In Seminar for Teachers on Demystifying e-learning accessible at.

Phatak, D. B. (2015). Adopting MOOCs for quality engineering education in India. In Proceedings of the International Conference on Transformations in Engineering Education (pp. 11-23). Springer India.

Plomp, T. (2007, November). Educational Design Research: an Introduction. IN PLOMP, T. & NIEVEEN, N.(Eds.) An Introduction to Educational Design Research. In Proceedings of the seminar conducted at the East China Normal University, Shanghai (PR China).

Porter, L., Bailey Lee, C., Simon, B., & Zingaro, D. (2011, August).Peer instruction: do students really learn from peer discussion in computing?. InProceedings of the seventh international workshop on Computing education research (pp. 45-52). ACM.

Prasad, G., Bhar, C., & Srivastav, M. V. Critical review of examination related problems in technical education in India.

Ramesh, R., Mishra, S., Sasikumar, M., & Iyer, S. (2014, December). Semi-Automatic Generation of Metadata for Items in a Question Repository. In Technology for Education (T4E), 2014 IEEE Sixth International Conference on (pp. 222-228). IEEE.

Ramesh, R., Sasikumar, M., & Iyer, S. (2016, July). A Software Tool to Measure the Alignment of Assessment Instrument with a Set of Learning Objectives of a Course. In Advanced Learning Technologies (ICALT), 2016 IEEE 16th International Conference on (pp. 194-198). IEEE.

Ramesh, R., Sasikumar, M., & Iyer, S. (2016, July). Integrating the Learning Objectives and Syllabus into a Domain Ontology for Data Structures Course. In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (pp. 266-271). ACM.

Ramsden, P. (1992). Learning to teach in higher education, London: Routledge.

Reddy, P. D., & Mahajan, A. (2016, July). Expert System for Generating Teaching Plan Based on Measurable Learning Objectives and Assessment. In Advanced Learning Technologies (ICALT), 2016 IEEE 16th International Conference on (pp. 207-208). IEEE.

Richard Frye, Gary R. McKinney and Joseph E. Trimble. (2007). Tools and Techniques for Course Improvement. Handbook for Course Review & Assessment of Student Learning, Western Washington University.

Richey, R. C., & Klein, J. D. (2005). Developmental research methods: Creating knowledge from instructional design and development practice. Journal of Computing in higher Education, 16(2), 23-38.

Sadler*, D. R. (2005). Interpretations of criteria-based assessment and grading in higher education. Assessment & Evaluation in Higher Education, 30(2), 175-194.

Sahami, M., Roach, S., Cuadros-Vargas, E., & Reed, D. (2012, February). Computer science curriculum 2013: reviewing the strawman report from the ACM/IEEE-CS task force. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (pp. 3-4). ACM.

sakai LMS: https://sakai.rutgers.edu/helpdocs/tests.html, Referred on January 2017.

Seels, B., & Richey, R. C. (1994). Instructional technology. Bloomington, IN: Association for Educational Communications and Technology.

Shaffer, D. W., Hatfield, D., Svarovsky, G., Nash, P. Nulty, A., Bagley, E., Franke, K., Rupp, A. A., & Mislevy, J. R. (2009). Epistemic network analysis: A prototype for 21st Century assessment of learning. The International Journal of Learning and Media, 1(2), 33–53.

Shute, V. J. (2008). Focus on formative feedback. Review of educational research, 78(1), 153-189.

Smythe, C., & Roberts, P. (2000). An overview of the IMS question & test interoperability specification.Chicago

Song, S. H., & Keller, J. M. (2001). Effectiveness of motivationally adaptive computer-assisted instruction on the dynamic aspects of motivation. Educational technology research and development, 49(2), 5-22.

SumTotal LMS: www.sumtotalsystems.com/enterprise/learning-management-system/ , Referred on January 2017.

Thorpe, M. (2000). Encouraging students to reflect as part of the assignment process Student responses and tutor feedback. Active learning in higher education, 1(1), 79-92.

Totara LMS: https://www.totaralms.com/sites/default/files/resources/totaralms_assessment_achievement_0.pdf

Van den Akker, J. (1999). Principles and methods of development research. In Design approaches and tools in education and training (pp. 1-14). Springer Netherlands.

Xia, w., Yihong, D., Yi, Z. (2006). Similarity Measurement about Ontology-based Semantic Web Services. Workshop on Semantics for Web Services (SemWS'06) in conjunction with 4th European Conference on Web Services (ECOWS'06).

Yi Zhao H., W. Xia W. (2008). A Rough Similarity Measure for Ontology Mapping. Internet and Web Applications and Services, ICIW '08, 136-141.

Publications

Nalawade, G., & Ramesh, R. (2016, December). Automatic Generation of Question Paper from User Entered Specifications Using a Semantically Tagged Question Repository. In Technology for Education (T4E), 2016 IEEE Eighth International Conference on (pp. 148-151). IEEE.

Ramesh, R., Sasikumar, M., & Iyer, S. (2016, July). Integrating the Learning Objectives and Syllabus into a Domain Ontology for Data Structures Course. In Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2016) (pp. 266-271). ACM.

Rekha R., Sasikumar M. Iyer S. (2016) "A Software Tool to Measure the Alignment of Assessment Instrument with a Set of Learning Objectives of a course", International Conference on Advanced Learning Technologies (ICALT).

Ramesh, R., Mishra, S., Sasikumar, M., & Iyer, S. (2014, December). Semi-Automatic Generation of Metadata for Items in a Question Repository. IEEE Sixth International Conference on Technology for Education (T4E), 2014 (pp. 222-228). IEEE

Rekha, R., Angadi, A., Pathak, A., Kapur, A., Gosar, H., Ramanathan, M., ... & Sasikumar, M. (2012, January). Ontology driven framework for assessing the syllabus fairness of a question paper. IEEE International Conference In Technology Enhanced Education (ICTEE), 2012 (pp. 1-5). IEEE.

Ramesh, R., Sasikumar, M. (2010). Ontology Based Approach for Finding the Similarity Among Questions in a Distributed Question Bank Scenario. IJCSA, 7(1), 79-93.

Acknowledgement

It indeed gives me great pleasure and a sense of achievement to have completed this thesis and I wish to express my sincere thanks to my guide **Dr. Sridhar Iyer** whose able guidance helped me steer through this journey. I am immensely grateful for the critical inputs he provided, during my research, without which, this work could not have been completed successfully. Apart from giving technical guidance, he has also been a constant source of support and motivation during this endeavor.

I express deep sense of gratitude to **Dr. Sasikumar M.** for his continuous guidance and support throughout these years of research work. He constantly monitored my progress and gave me valuable suggestions to proceed in the right direction. This thesis would not have been possible without his involvement and advice.

I express my gratitude and thanks to **Dr. Sahana Murthy** and **Dr. Pushpak Bhattacharya**, Research Progress Committee members for their valuable suggestions that helped me in improving my research approach. My special thanks are due to **Dr. Sahana Murthy** whose vital inputs during my progress seminars have made me think in the right directions and added clarity to my approach towards the research work.

During my research tenure, I have drawn help and support from many people at IDP ET department at IIT Bombay. A few names to mention are **Mr. Shitanshu Mishra, Mr. Rwitajit Majumdar, Mr. Jayakrishnan, Dr. Gargi Banerjee, Dr. Mrinal Patwardhan** and **Dr. Yogendra Pal**.

I thank trainee students **Mr. Niraj Palecha** from Shah and Anchor Kutchhi Engineering College and **Mr. Akshen Kadakia** from D J Sanghvi College of Engineering for their efforts put in towards the tool development work.

I remember with gratitude the support offered by my present institution, Shah and Anchor Kutchhi Engineering College (SAKEC). I am grateful to **Shri. Keshavji Umarshi Chhadva**, Chairman-SAKEC, **Shri. M. L. Shah**, Hon. Jt. Secretary-SAKEC, **Dr. Bhavesh Patel**, Principal-SAKEC, **Dr. V. C. Kotak**, Vice Principal-SAKEC, and **Prof. Uday Bhave**, HOD, Computer Department-SAKEC for their constant support and encouragement.

My special thanks are due to **Dr. Uma Rama Rao** who has been a constant source of inspiration and an immense help in all of my technical writings.

I am indeed indebted to all other **Research scholars** at IDP ET for all the warmth and affection they showered on me and made my PhD journey memorable.

I thank the **supporting staff** of IDP in Educational Technology and IIT Bombay for providing a friendly environment.

Lastly, I take this opportunity to thank my husband **Ramesh** and my sons **Ashwanth** and **Anirudh** and other members of my extended family for their unconditional support in this endeavor.

I once again express my gratitude towards each and every one who supported me directly or indirectly throughout my academic career.

I bow down and thank the Almighty for giving me this opportunity, the strength and the ability for undertaking this journey in the pursuit of knowledge.