Theses and Dissertations

Theses, Dissertations, and Senior Projects

January 2013

A Framework For The Integration Of Technology Into Curricula Development And Student Assessment

Washington Helps

A FRAMEWORK FOR THE INTEGRATION OF TECHNOLOGY INTO CURRICULA DEVELOPMENT AND STUDENT ASSESSMENT

by

Washington Isaac Helps

Bachelor of Science, University of Technology, Jamaica, 2008

A Thesis

Submitted to the Graduate Faculty

of the

University of North Dakota

In partial fulfillment of the requirements

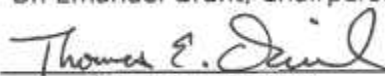for the degree of

Master of Science

Grand Forks, North Dakota

May 2013
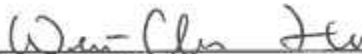
This thesis, submitted by Washington Helps in partial fulfillment of the requirements for the Degree of Master of Science from the University of North Dakota, has been read by the Faculty Advisory Committee under whom the work has been done, and is hereby approved.

_____

Dr. Emanuel Grant, Chairperson

_____

Dr. Thomas O'Neil,

_____

Dr. Wenchen Hu,

_____

_____

This thesis is being submitted by the appointed advisory committee as having met all of the requirements of the Graduate School at the University of North Dakota and is hereby approved.

_____

Dr. Wayne Swisher,
Dean of the Graduate School

april 30, 2013
_____
Date

PERMISSION

Title   A Framework for the Integration of Technology into Curricula Development and Student Assessment

Department   Computer Science

Degree   Master of Science

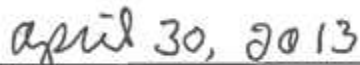In presenting this thesis in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis work or, in her/his absence, by the Chairperson of the department or the dean of the Graduate School. It is understood that any copying or publication or other use of this thesis or part thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of North Dakota in any scholarly use which may be made of any material in my thesis.

Signature _____

Date   _April 27, 2013._____

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The purpose of this work is to amalgamate technology and education in a manner which will prove to be beneficial to all stakeholders involved and which will allow for an improvement in the ubiquitous process of curriculum development in the teaching and learning sphere. The primary motivation surrounds the issue of limited human resources in terms of teachers and expert knowledge as well as available physical resources such as computer equipment or other classroom artifacts. It should be duly noted that it is in many developing countries where the student-teacher ratio is very high and where such a framework will be most useful. However, this work has the potential to benefit not only developing countries, but also developed countries where the available technology is more advanced and where its integration in student learning is more pronounced.

The major focus will be on the creation of a framework which will allow for systematic, structured and seamless curriculum development and learning outcome assessment. Therefore, a major component will be the modeling of a curriculum in a structured and qualitative way to include goals and objectives which will then lend it to adaptation and use downstream within the framework. Another key component is Assessment which will encapsulate various ways in which this paradigm can leverage the assessment aspects of a curriculum. The benefits from this work may prove to be quite

immense as the proposed framework may help to standardize the design, delivery and

assessment of any curriculum regardless of location and resource.

CHAPTER I

INTRODUCTION

1.1 Research Definition

Despite the advances in technology, there still seems to be limited utilization of such technology in the field of education. While there is a belief that technology could improve the educational productivity and help schools to teach more efficiently, evidence to support this belief is scarce [1]. Indeed, while the business landscape has seen a dramatic transformation due to the integration of technology, this sort of impact in educational institutions has been modest. There remains a very broad spectrum in which the harnessing, utilization and integration of technology would prove beneficial to the development, delivery and assessment of education. This work is not merely trying to address an existing problem, but rather will strive to chart a path towards a new paradigm in teaching and assessment from the ground up.

1.1.1   Amalgamating Technology with curriculum design

The fusing of technology with curriculum design will result in benefits being derived long after the actual curriculum has been created. Incorporating technology into the design process will allow educators and other stakeholders to capitalize on a framework which relies on the systematic and methodical modeling and presentation of

related artifacts. The basic idea is to integrate technology from the ground up, that is, from the beginning point of the entire process all the way through to assessing what was learned.

Traditionally, technology in education has been viewed in two ways; as a transmission device and as a learning device [1]. According to [1], too much emphasis has been placed on learning from technology (e.g. viewing educational television, computer drills etc.), rather than learning with technology. It has also been argued that traditional teaching involved the dissemination of information from the front of the room, assigning chapters from text books, and grading worksheets and exams rather than helping each student search for personal understanding [2]. Figure 1 illustrates this traditional view schematically.

The aim of this research is not to adopt these existing views, but rather to go a step further in an attempt to integrate technology in one of the foundational pillars of the educational process; that of the development of curricula. Remember, that without any set goals (a 'roadmap' of sorts), how can one know what it is that ought to be achieved and hence how and what it is that ought to be taught. Therefore, the view of technology in education will be changed from the standpoint of this research to reflect a model that more closely resembles the one shown in figure 2 below. The idea here is to not just continue integrating or improving the use of technology in curriculum, but to also utilize technology in the development and refinement of said curriculum. Refinement is mentioned here to emphasize the use of technology in the continued

improvement and positive modification of a curriculum through feedback, assessment analysis and student learning outcomes (psychomotor, cognitive, etc.).
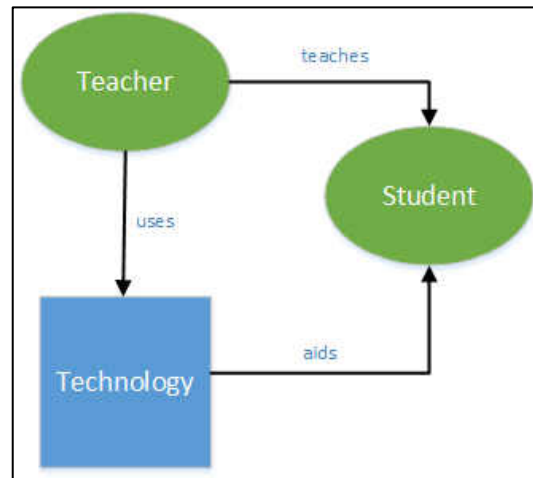
Figure 1 – Traditional view of technology in education
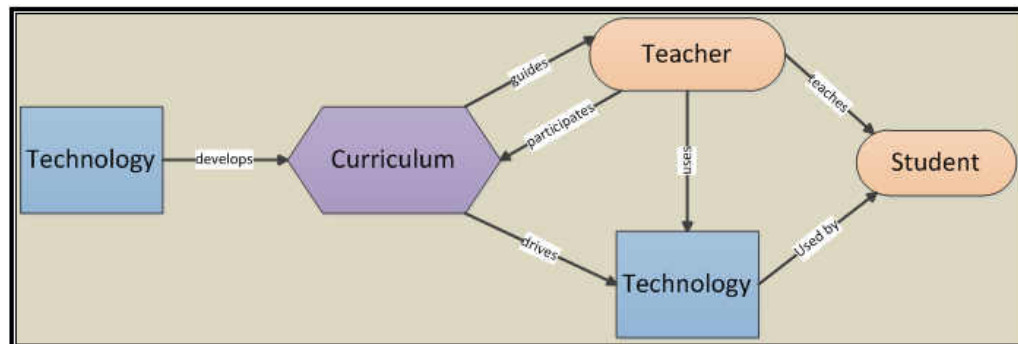
Figure 2 – Future of view of technology integration in education

One of the focal points of this research is to look at the integration of technology into curriculum development holistically. That is, there is a need to go beyond the traditional view of simply using technology in the classroom to aid in the delivery or teaching of a subject for instance. To this end, there needs to be an effort to get to a point where technology is much more involved in the overall process from the actual

selection of content for a curriculum to even the assessment of student performance (learning outcome) and the feedback of such assessment to inspire positive changes in a given curriculum.

1.1.2   The need for systematic curriculum design

A curriculum can be described as the embodiment of a program of learning which is the aggregate of courses of study given in a school and includes philosophy, content, approach and assessment. Given this view, a curriculum is a very important aspect of any educational program. Therefore, if this process can be formalized in a more rigid way, the benefits to be derived will be quite immense. Also, because of the fact that this process is one that virtually every established educational institution and program must partake in, then it stands to reason that greater care and emphasis must be taken. Curriculum design and development are core functions of institutions which occupy substantial human resources [10]. Hence there is an inherent need for the systematic designing of curricula.

1.1.3   Leveraging technology in a new paradigm for teaching and assessment

This research effort will essentially leverage the use of technology in a new paradigm for teaching and assessment. The framework that will be created will present a collection of artifacts that will help educators to more easily manage a curriculum and to more easily manage assessment of students. This model is a push toward greater

coupling of content and assessment and is an innovative way of modeling and representing such educational artifacts.

1.2 Motivation

1.2.1 Scarcity of classroom human resources and Limited physical resources for teaching

The scarcity in human resources in the way of lecturers, faculty and generally experts in the related field has been a barrier to the effective delivery of standardized world class curriculum in developing countries. Many higher education institutions there do not have faculty with PhDs (not to say that a lecturer with a PhD is the 'be all and end all', but a certain level of credibility and authority comes with it). A cursory comparison between the computer science departments at the University of North Dakota (UND, located in the United States – a developed country) and the University of Technology, Jamaica (U-Tech, located in Jamaica – a developing country) is summed up in table 1 below. It is clear to see that the percentage of faculty with PhDs is far greater in the developed country.

Table 1: Comparison of Faculty in the Computer Science Department at two Universities

	University of Technology, Jamaica	University of North Dakota
Faculty with MSc	23	2
Faculty with PhD	5	8
Total Faculty	28	10
Percentage Faculty with PhD	17%	80%

In many developing countries, there are far more community colleges than universities and students from lower income households tend to gravitate towards community colleges primarily because of financial constraints and ease of access. Community colleges typically offer 2-year programs, after which students wishing to obtain a bachelor's degree will have to transfer to a university. Although community colleges are not the target of this work, there are implications to be had such as the fact that a wider pool of students in developing countries may eventually benefit from the proposed framework. A solid foundation is very important for students of these institutions who intend to go on to pursue bachelor's degrees in various STEM disciplines. Generally, faculties with PhDs tend to shy away from these two year colleges and so most faculty members there have only master's degrees. The argument of whether a faculty with a PhD is a better teacher than another with a master's degree or vice versa is one that will not be entertained in this work, suffice it to say that the best of both worlds would be the ideal situation to have. In an article in the Chronicle of Higher Education, it is noted that "professors prefer four year universities where there are fewer classes and where research can be pursued" [9]. In developing countries where there may not be as many faculties with PhDs or experienced faculty with master's degrees, it is expected that the proposed paradigm will be very beneficial. This framework will ease the burden that such underserved institutions and territories bear by making available a common pool of resources necessary for teaching and assessment.

Sub-standard and limited physical resources such as communications equipment (broadband internet, wired or wireless phone infrastructure) in varying degrees has also contributed to the educational gap and lack of coherency which exists between developed and developing countries. Furthermore, within the borders of a developing country, one may find that the expertise and available resources are only available in the urban centers, thus limiting accessibility and compounding macro-economic issues such as 'brain drain'. This gap can be expounded by examining the scenario in which a student with an undergraduate degree from Jamaica (a developing country) is not viewed in the same way as a student from the United States who has a similar undergraduate degree in the same discipline. It is for this reason that many colleges require a Graduate Comprehensive examination for students wishing to matriculate in a graduate program. This exam attempts to ensure that all students in their graduate program have successfully mastered the undergraduate level programs regardless of where such programs where taken. Most US universities also require a Graduate Record Examination (GRE) whether it is a general or subject test for much the same reasons. This incongruent view is not necessarily a matter of culture, but one for which there is some merit because of the fact that as things stand currently, two courses having the same name and taught in two different countries or different universities does not mean that the content or delivery is the same.

1.2.2   High teacher to student ratio

It has been found that there is a high teacher to student ratio in developed countries virtually at all the levels in the education system. This again, is another marked difference between what obtains in the classroom in a developed country versus a developing country.

In conditions where one teacher is responsible for too many students, the individual attention that each student needs may be lacking in terms of post assessment, feedback and overall progression evaluation. A framework such as that proposed in this research will significantly reduce the negative impacts that such high ratios have on both the teaching and learning process.

1.2.3   The need for a minimum quality threshold

A program of study taught in different regions of the world often has differing outcomes and qualities. Especially in disciplines which are young (like Software Engineering) and still developing, the disparity may be quite significant. Therefore it is very important to have a minimum standard which defines the quality of a curriculum. In this way, there can be the establishment of certain standards which direct and guide the overall effort. Thus maximizing the expert knowledge in the particular field in which the system is utilized. For instance, a student pursuing a Bachelor of Science in Computer Science with a Software Engineering major in Jamaica (a developing country) and being taught by faculty with M.Sc. degrees should have a similar learning outcome

as a student in the USA (a developed country) doing a similar program of study and taught by faculty with PhDs. Therefore, regardless of where the curriculum is utilized, there should be a minimum standard that is achieved, below which the quality cannot be guaranteed. Having used the previous example is however, not an indictment on the teachers/faculty. Thus the example is used here to make the distinction that the quality of a curriculum varies tremendously from institution to institution and this is the greater problem that needs to be addressed.

1.2.4 International collaboration and a common pool of resources (Repository)

This new paradigm will work well in an international setting wherein the internet will act as an enabler in this regard. Once the repository is set up, the common pool of resources will be available internationally, thus strengthening the collaborative efforts between higher learning institutions. This will also bring into closer alignment the notion of a standardization of curriculum and minimum quality thresholds.

1.2.5 The need to establish measurable goals (learning objectives) tightly coupled with student learning outcomes

A famous quote by Fitzhugh Dodson reads "Without goals, and plans to reach them, you are like a ship that has set sail with no destination." In a similar light, it is very important to have well-established goals in any program of study. Hence, there is a need to have measurable learning goals (objectives), and further to tightly couple such objectives with learning outcomes. Doing so will facilitate the seamless assessment of

students in that it will be easier to compare student learning outcome to learning objectives. There are curriculum information management systems which do exist, but these take a perspective which is more targeted towards administration such as accreditation issues, skill-set mapping and quality control. The new paradigm being suggested in this work will go much deeper in taking the perspective of teaching and learning in the classroom and packaging all the relevant aspects in a systematic way.

1.2.6 Meaningful and automatic evaluation of students

One of the distinct benefits of this new paradigm is that it will allow for meaningful and automatic assessment of students. This will be especially useful in developing countries where the high student to teacher ratio means that educators do not have the time to give individual attention to all students. Automatic evaluation means that the system can take a student's raw score and based on certain characteristics of the assessed content along with certain parameters/metrics, an automated evaluation of that student can be made. This evaluation may include recommendations and other qualitative analyses.

1.2.7 Automatic generation of assessments

Educators need only specify the parameters they want the students to be assessed on and the system will automatically generate the questions. The success of such a scenario will depend on a Question Bank that is supported by contributors and

authorities in the given domain such that every question is tagged in a manner which will link it to a specific content or set of contents. The idea here is that based on the fact that there will be a rigid content to assessment mapping, every question can be tied to some content and in so doing the system can easily generate for instance a test containing a set of questions given the criteria of a content or set of contents. See Figure below:
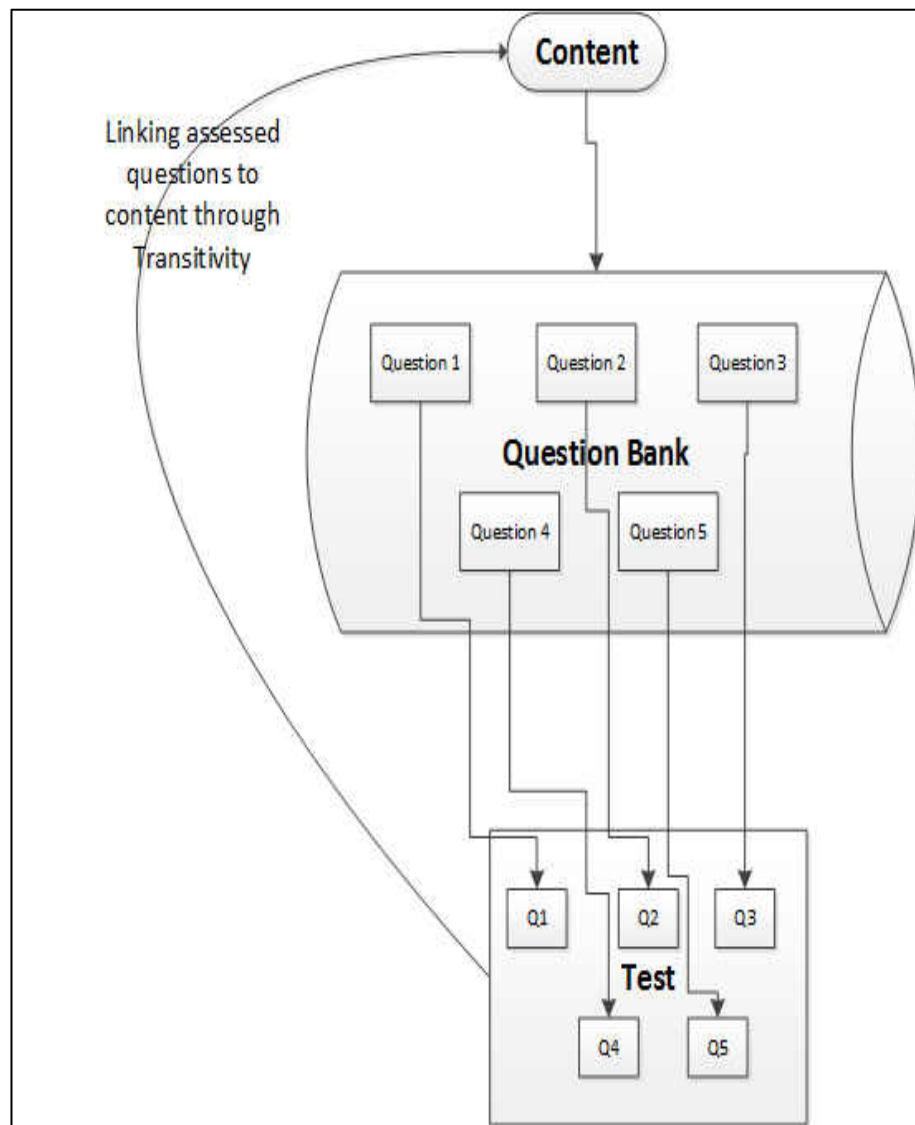
Figure 3 – Mapping Content to assessment

1.3 Research Methodologies (Approach)

1.3.1   Software Engineering model used as a case study in this research

This framework is potentially useful across many domains and utilizing it as a template in any one domain theoretically means it will be useful in others. However, for the purposes of this paper, the Software Engineering discipline will be used as a model to demonstrate this framework. In other words, a Software Engineering Model will be used as a case study in this research. Software Engineering is a STEM (Science, Engineering, Technology, and Mathematics) discipline and as mentioned before, it is hoped that such a paradigm as that proposed will be applicable to any STEM discipline.

1.3.2   Mapping content to delivery and assessment

In the previous section the importance in mapping objectives to content was mentioned. In this section, a similar approach obtains, however, the focus here is the mapping of content to delivery and assessment. Tightly coupling content to assessment especially will provide many benefits such as that of automated assessment generation. But more importantly, the hierarchical approach mentioned earlier will be maintained which will lead to many other possible benefits and use in this framework.

1.3.3   Rigid objectives to content mapping

The objectives of any curriculum are a critical component and can be viewed as a starting point or a beacon which acts as a guiding light for the entire effort. In the same light, the objectives themselves cannot stand alone. The objectives guide the selection

of content, and it is the delivery of said content which helps to fulfill the objectives. Therefore, if one were to abstractly look at this process from a top-down approach, it can be seen that it is fairly easy to rigidly map objectives to content. A given objective may map to one or more content or content areas, while a given content may be mapped to one or objectives.

A foundational principle of this system is the notion of hierarchy or better yet, granularity. The levels of abstraction are very important and so the artifacts of the system can be viewed from top-down to bottom-up. The two examples given in figure 4 sums up this notion. Another major feature of this framework is the notion of scalability. This is an important feature because the system must provide a mechanism for the systematic design of curriculum which reduces the complexity of an otherwise complex task. Scalability in this context will allow for consistent usability throughout, from early in the design process to the end as well as from a size aspect, that is, whether the curriculum is small or large.

Stakeholders who will benefit	Level of Generality
Students	
Teachers	
Administration	
Institutions	
Educational Sector	
Governments / Countries	
The World	

Curriculum	Level of Generality
Domain e.g. Aerospace, Computer Science	
Program Curriculum	
Subject Area	
Course	
Section	
Topic / Sub-Topic	
Atomic Unit	
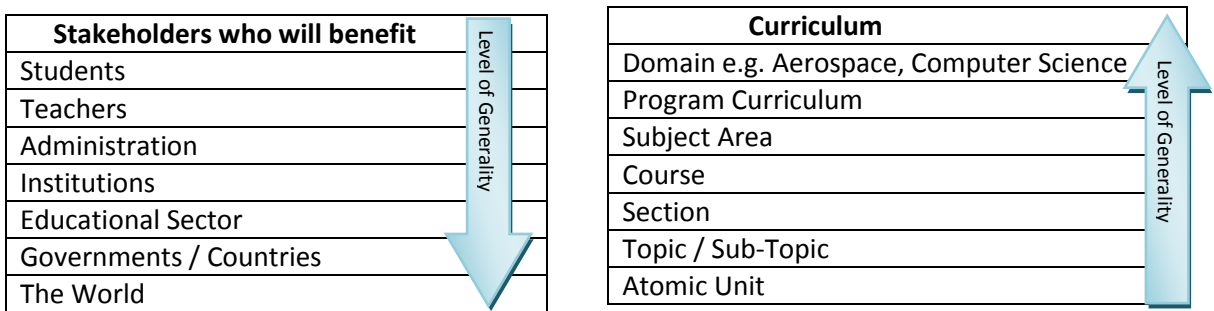
Figure 4 - Increasing and decreasing levels of generality

1.3.4 *Remodeling/Reshaping the Curriculum Design Process*

This work serves as a proposal for a solution/paradigm that will significantly improve the way in which curricula are designed and administered. Also, there will be many benefits to be garnered from taking the proposed approach. A comparison between the traditional curriculum design and the proposed paradigm will be done.
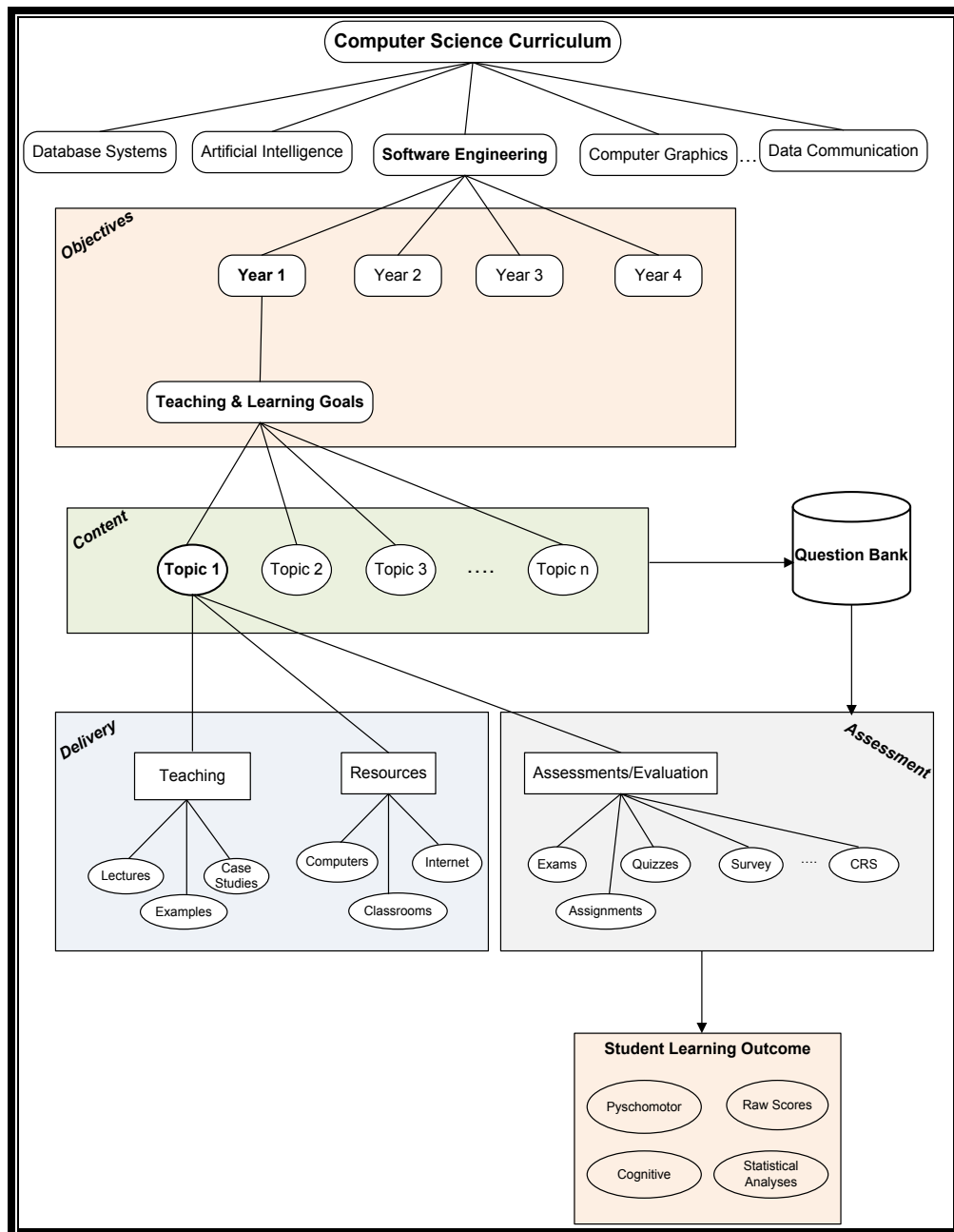
Figure 5 - Top Level Schema expanded

1.4 Scope

1.4.1 The paradigm will be applicable to any STEM discipline

This paradigm will be applicable to any STEM discipline and the scope of this work will be limited to a case study of its use with Software Engineering. An abstract view of the framework is depicted in figure 5 in which a hierarchical outline is given. In this work, the Software Engineering curriculum in the field of Computer Science will be used as the working prototype.

Thus for a given four year undergraduate degree program, a system derived from this framework will be able to for each year:

- Present all the teaching and learning goals

- Provide access to all relevant content in terms of a set of topics

- For each topic, show the recommended delivery (teaching) methods and required resources. For topics which are not selected, alternatives may be suggested, but regardless of whichever topics constitutes the content, the coverage metric (a measure of how much of the curriculum is covered or selected) will be given.

- Recommend relevant assessment methods and provide automated assessments which are mapped directly or indirectly to the content.

- Provide analysis of student learning outcome which allows for a comparison between performance and expectations (feedback between outcome and objectives).

In terms of the scope of this work, there will not be an implementation of this system in a manner in which one will see it in operation. But rather, as outlined in the schema in figure 5, an attempt will be made to lay the ground work for this new paradigm and to as succinctly as possible make the case for this research. A fully functional implemented system of this nature may take years, and hence the scope of this work will be limited to the case study of the SE discipline. Administrative issues and concerns will not be taken into consideration for this work and so the focus will be on teaching and learning.

1.5 Expected Results

1.5.1   A model/paradigm for curriculum development in STEM areas

It is anticipated that this research work will result in a paradigm/model for curriculum development in STEM areas. Essentially, this work will present a roadmap outlining the reshaping of curriculum design from concept to assessment through technology driven methodologies.

At the conclusion of this work, questions as to whether technology can be used to enhance and significantly improve the curriculum design process should be answered. It will be demonstrated whether or not educators will be able to use artifacts from a repository to enhance their teaching programs in a technologically driven way, effectively using technology as a driving force.

It should be shown that a model in which educators can utilize a centralized repository built up by experts in a given domain and consisting of artifacts relevant to that domain is both feasible and sustainable. Such a repository should be amenable to any faculty who wishes to benefit from it as an aid and guide in the design, delivery and assessment of customized curricula whose end result is in keeping with international standards and quality.

CHAPTER II

BACKGROUND

2.1 What is a Curriculum and what constitutes it?

The term curriculum is often times misunderstood, misrepresented and takes on different meanings in different contexts. It is therefore very difficult to find a common definition for the term. In simple terms, however, a curriculum is the embodiment of a program of learning which is the aggregate of courses of study given in a school and includes philosophy, content, approach and assessment. But what is this embodiment? In some sense, a curriculum can be referred to as any document that exists in a school that defines the work of teachers by identifying the content to be taught and the methods to be used [3]. In most of today's educational settings, such a document even in a digital form is quite static, and not flexible to the needs of the beneficiaries.

From the simple definition above, philosophy would include the implicit and explicit standards and expectations that ought to be captured and in the entire teaching and learning process in a given institution or department or setting.

At this juncture, it would be useful to mention a few of the alternative views on a curriculum in schools. Educational theorist Larry Cuban [4] suggests that there are at least four different curricula utilized in schools:

- The official curriculum – what the policy makers or other authoritative body sets forth and is precisely what teachers are expected to teach.

- The taught curriculum – what teachers actually end up teaching or what they choose to teach in their individual settings and circumstances

- The learned curriculum – unspecified lessons that students learn which are embedded in the classroom environment

- The tested curriculum - a subset of the official, taught and learned curricula for which students are evaluated on.

In the brief description outlining the four different curricula above, it is clear to see the conundrum which results with the implementation of a curriculum (what is actually taught) and in assessing students (tested what has been taught and learned). Assessment is really two-fold because it helps to determine what was taught and what was learned. But in fairness to the educators, the results of a test does not conclusively suggest that a given topic was not taught or taught properly because a student may not have learnt that which was taught, which may not be the fault of the teacher. This is one of the issues that the proposed paradigm will address because a system designed using this framework will readily create an assessment given a set of teacher-defined parameters and determine how much of a course's content is covered in a given test. Larry Cuban in [4] also suggests that what is tested is a limited part of what is intended by policy makers, taught by teachers and learned by students and further that standardized tests often represent the poorest assessment of the other curriculums. A major problem with standardized tests is that teachers are far removed from the actual

construction of the test which results in a greater disparity between what is being tested and what was taught and learned. The proposed paradigm would fit nicely in a solution for this problem because a framework would be in place for teachers to be able to dynamically design an assessment tailored to their specific needs and environments. In different institutions and across national borders, "one size does not fit all." Hence there is the need to put the control of assessments in the hands of those who are delivering the curriculum and that curricula must be designed with the users and the environment for which it will be utilized in mind. The main idea here is that this task can be done easily, automatically and correctly.

Another perspective of curriculum is that of the Null curriculum. This notion is put forward by Elliot Eisner in [5] which suggests that what curriculum designers and/or teachers choose to leave out of the curriculum is no less important than what they choose to include and that those choices are based on a number of factors. Some of these factors include personal beliefs, knowledge and skill level of the educators, and cultural nuances of the curriculum designers. This is an important point because the proposed paradigm will expressly model such factors both in terms of the rationale for choosing certain topics while omitting others. Consider a teacher who gives a test which includes questions that were not covered by the teacher. The ways in which the students answer such question are important indicators about the students' cognitive ability. Such a marker can easily be captured and expounded upon by the proposed framework, because for instance, it will indicate based on how the students handled the questions that the particular content may be removed because it is covered in some

other content area (overlapping), or that it must be covered. So over time, the curriculum can be improved and enhanced.

2.1.1   Definition of Terms

Software Engineering Education Knowledge (SEEK): the body of knowledge that is appropriate for an undergraduate program in software engineering.

Knowledge Area: a particular sub-discipline of software engineering such as Software Design or Software Management.

Course: a subset of a knowledge area. This term is used interchangeably with the word unit.

Unit: a subset or module of a knowledge area. This term is used interchangeably with the word course throughout this paper.

Topic: the lowest level of the SEEK hierarchy. A topic is a subset of a unit.

2.1.2   The Parts of a Curriculum

A curriculum is not simply a document that contains a list of objectives and the topics that would fulfill each objective. But it is much more involved that this, in that it embodies the educational process in a holistic way. The Duke Centre for Instructional

Technology [7] postulates that there are six distinct parts to any curriculum. These parts are concisely presented as follows:

1. Needs Assessment – evaluates the need for such a curriculum,

2. Rationale – is based on the needs assessment

3. Goals and Objectives – the core of the curriculum which presents specific skills, knowledge and attitudes that learners ought to achieve through the program,

4. Teaching and Learning Strategies – these are essentially the "how" of the curriculum and include methods that will be utilized in the delivery of the material such as lectures and projects,

5. Evaluation Strategies – methods of measuring the objectives achieved from the perspectives of the learners, the educational methods and the overall program,

6. Management Plan – this is the implementation which takes the curriculum from design to use.

All the parts of a curriculum are clearly important and must be addressed at some point throughout the design process by various stakeholders. For instance, the needs assessment may be important to faculty members who actually teach a given course. So that need may be initially raised by some faculty member who will then need to justify such a need to perhaps the school board or the university's president. Therefore, certain aspects of the design process are inherently administrative, and so for the purposes of this work, such aspects will be omitted. This will be done in an attempt to keep the focus on the scope of this work which assumes that parts one and

two listed above have already been done. The focus is therefore on parts 3 through to 6 and each of those four individual parts will be addressed and incorporated in the proposed paradigm in some way.

2.2 The Curriculum Design Process

According to V. Kaprielian [8], a curriculum is always a work in progress. So it is important to keep in mind the fact that as the environment in which the curriculum is used changes or the market that is being satisfied changes, so too must the curriculum. If not, then the institution or stakeholders involved risk descending into oblivion or obsolescence. A curriculum must be dynamic and fluid, changing appropriately with changing needs and adapting as necessary.

Figure 5 depicts a top level schema of the framework for the proposed paradigm. The parts of a curriculum (parts 3-6 in keeping within the scope of this research) as outlined in section 2.1.1 are very much congruent with the view depicted in figure 5. In fact, it encapsulates both implicitly and explicitly the four parts in question. Part three which is the Goals and Objectives is clearly shown in the *Objectives* box. The objectives drive the makeup of the content and hence the selection of a set of topics as depicted in the *Content* box. Part four which is the Teaching and Learning Strategies is depicted in the *Delivery* box which covers both teaching methods and the resources needed to satisfy those methods. Part five which is the Evaluation Strategies is depicted in the *Assessment* box which shows various assessment methods mainly focused on evaluating

students. Part six which is the Management Plan is encapsulated in the *Student Learning Outcome* component shown in figure 5, which focuses on analyzing student performance to serve as one of the main feedback mechanisms. This feedback is important, because in addition to actually implementing a curriculum, ways must be provided to determine what areas were deficient, what areas need improvement or more time in delivering and so on. This feedback also serves to paint an overall picture from design to use to determine the effectiveness and worth of said curriculum.

| Curriculum design Process, Duke University [8] | Curriculum Design Process Cycle |
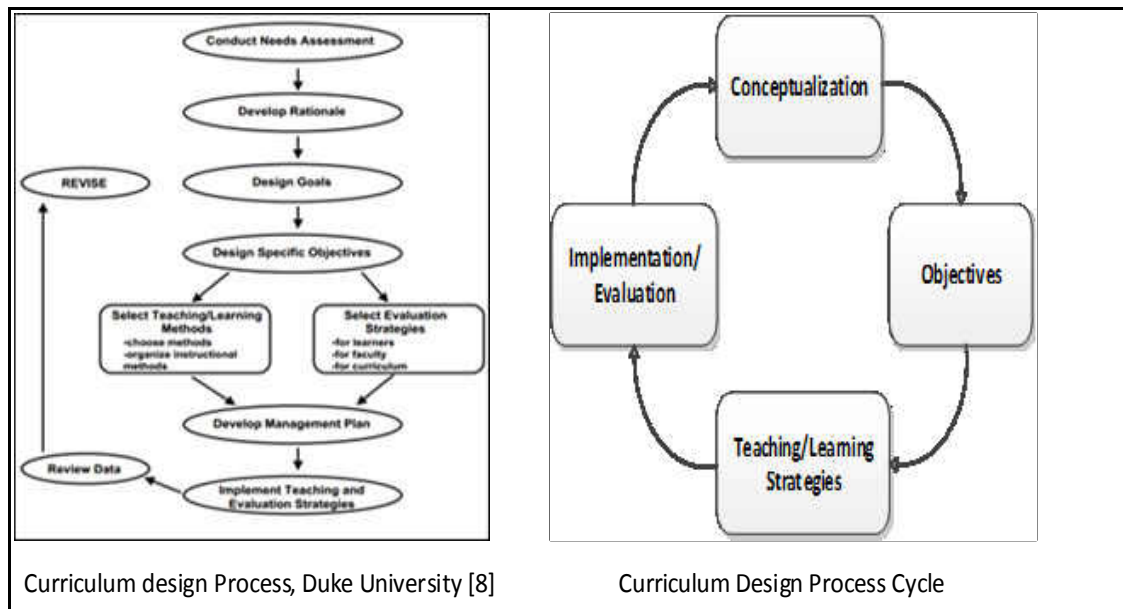
Figure 6 – Curriculum design Process and Cycle

The diagram shown in figure 6 is a generic one which illustrates the curriculum design process. The general flow and components are very similar to that shown in Figure 5 which is equivalent to this diagram but specific to the proposed paradigm. A curriculum needs to be dynamic in order to accommodate changing needs and changing environments. This is even more apt in the computing fields and other STEM areas where technology is constantly improving and evolving and in much the same way a

curriculum must evolve. This is why the process depicted in figure 6 is really a cycle where following evaluation there is review which may lead to changes in the curriculum to make it current. The diagram on the right in figure 6 clearly highlights the cyclic nature of curriculum design in which it becomes apparent that a curriculum is a dynamic artifact which is constantly evolving. This is a characteristic which ought to be harnessed especially in the STEM disciplines.

The linear approach to curriculum, which includes objectives feeding into content, content feeding into teaching and learning which in turn is followed by assessment and evaluation is a very simple and easy one to follow. However, it is lacking in a number of ways and limits the full potential of having a curriculum in the first place. Such limitations include feed-back mechanisms, and customization caveats. These caveats would allow for *context awareness* to be incorporated into the design process as well as allow for consistent and constant updates where necessary to ensure that the curriculum is both relevant and adaptable to changing needs.

2.3 How are curricula designed?

Jan Miller in his paper entitled "Computer Science Innovation in Thailand" [6] describes an empirical qualitative study of Computer Science education in Thailand. The focus of the study was to determine the diffusion and extent of adoption of the presented technological and educational innovations and to evaluate the Thailand-Australia Science and Engineering Assistance Project (TASEAP) success from the Thai perspective. Interviews from ten computer science departments were analyzed in relation to computer science technologies, teaching methods, innovation diffusion and adoption, organizational culture, systems success and

national cultural behavior [6]. Much of this study paid close attention to the higher educational developments, economic and technological environments and conditions that were taking place in Thailand at the time. The outcomes and the evaluations of the project were also done from a Thai perspective.

TASEAP focused on teaching and research methods, curriculum development, the use of technology such as the internet for teaching, specialist discipline skills, and laboratory, school and faculty management. The reason for highlighting this research here is to bring across the point that curriculum development and innovation is influenced by many factors including cultural context and sensitivity. For instance, a culture where internet usage and adoption is very high will work well for a curriculum that is heavily biased towards distance learning, but would not be well suited for a country that has limited internet availability. Hence there is the need for a framework in which the development of a curriculum can seamlessly accommodate such cultural differences and diversities without negatively affecting the standard and quality of said curriculum. A curriculum developed and utilized in Thailand may work well in a Thailand setting, but may not work well in the United States for instance. So even though certain features may be different such as the delivery, resources and so on, the quality, standard and overall effect must be similar so that regardless of which ever territory the curriculum is, its uniformity can be guaranteed. So the bottom line is that in designing a curriculum, context sensitivity such as culture, the stakeholders and needs must in some way be taken into consideration. Figure 7 illustrates this point schematically. Again, it must be highlighted that the illustration in figure 7 is really the essence of this new paradigm. A paradigm which allows a curriculum to seamlessly and easily capture the needs of its audience, the culture of the target population and the resources that are available to deliver said curriculum
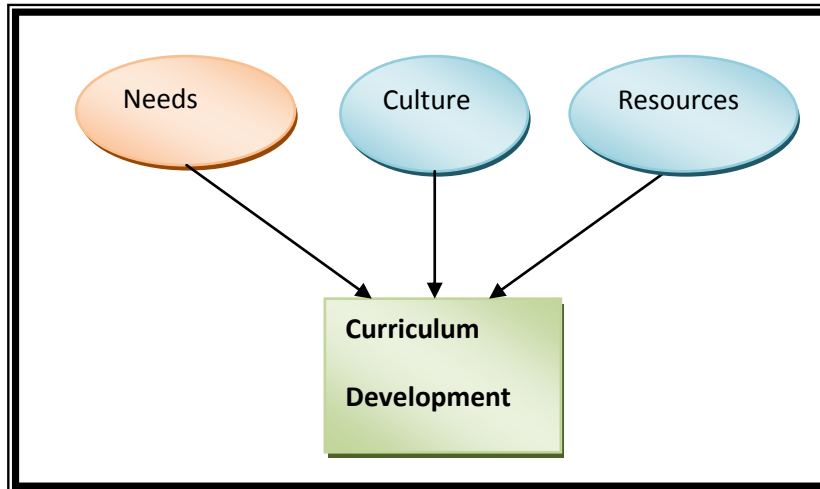
Figure 7 – Considerations that feed into a curriculum design.

2.3.1 Bloom's Attributes & Topic Relevance

Bloom's Taxonomy is a classification of learning objectives within an education setting which was proposed in 1956 by a committee of educators chaired by Benjamin Bloom. Bloom was the editor of the first volume of the book "*Taxonomy of educational objectives: the classification of educational goals.*" This taxonomy followed a series of conferences which took place from 1949 to 1953 which were designed to improve communication between educators on the design of curricula and examinations. The culmination of these efforts saw a consensus being reached where it was felt that educational objectives provide the basis for building curricula and tests and represent the starting point for much of our educational research [30].

2.4 Related Work

In this section of the thesis, several related works will be presented which are of significance in some way or another to this research. The Joint Information Systems Committee [11] in the United Kingdom is in charge of several projects at different universities that are looking into transforming curriculum design and delivery through technology. The Higher Education Funding Council for England [17], also based in the UK is doing a similar sort of work. Richard Gluga et al. has done significant work skills mapping and competence level tracking as well as developing a curriculum information tracking system [19][20]. The work being done by Grant et al. in the way of an international collaborative effort in enhancing the teaching and learning of software engineering [25][27][28], is also very significant and will be considered in detail in this chapter as well as in chapter 3. Finally, the IEEE/ACM SE 2004 document will also be examined to understand the steps taken to develop the document, especially since it will be used as a major source of input in the methodology and case study.

2.4.1   JISC - Transforming curriculum design and delivery through technology

A committee based in the United Kingdom (UK) has embarked on an ambitious program to tackle the problem of managing curriculum change. This committee which is called JISC (Joint Information Systems Committee) is branded as the UK's expert on information and digital technologies for education and research. JISC initiated a four-year program which started in 2009 to investigate how processes involved in the design of programs of study can be made more agile and responsive through the use of

technology. This program comprises 12 projects led by teams in UK universities. The over-arching theme of their work is to manage curriculum change and transform curriculum design and delivery through technology. JISCs vision is of a world where learners, teachers, researchers and wider institutional stakeholders use technology to enhance the overall educational experience by improving flexibility and creativity and by encouraging comprehensive and diverse personal, high quality learning, teaching and research [11].

2.4.1.1 Integrating Technology into curriculum Design

JISC has found that the importance of curriculum design has prompted many institutions to rethink the processes and products involved in the planning, designing and administering of programs of study. Their publication [12] also notes that in the 21st century, institutions aim to be increasingly demand-led and responsive to cultural and economic change. This is an issue that will be pointed out later in this paper and that the proposed paradigm will attempt to address. Various institutional products and services support the design of a curriculum. These systems include quality assurance and validation processes, student record systems, virtual and managed learning environments, assessment systems, repositories of learning resources, timetabling systems and physical space allocation (such as classroom scheduling).

One of the things that various JISC projects are attempting to do is to test process modeling tools to achieve more agile and adaptive working procedures. To this end, they are exploring various ways of integrating a wide range of stakeholder views

and enabling learners to benefit from more personalized curriculum designs. It is noted in [12] that curriculum delivery presents many complex challenges such as responding to changing student needs, ensuring availability of high-quality learning resources and environments and delivering a more flexible and engaging learning experience. Such challenges are only a part of the myriad issues that the proposed paradigm in this work will address. This is just another reason why this work is important and why it is important to exploit technology in order to achieve more innovative, context-centric and learner-centered approaches to curriculum design and delivery.

2.4.1.2 Integrating Technology into Curriculum Lifecycle

JISC breaks down curriculum development into the two distinct processes of design and delivery and highlight the fact that in reality, there is overlap and interplay between them. For example, feedback from student assessment which takes place in the delivery process should inform future modifications in the curriculum which takes place in the design process. Figure 8 shows the curriculum lifecycle according to JISC.

The JISC program vision is that the strategic use of technology can enhance a wide range of educational systems with significant benefits for students and other stakeholders through their engagement with a more flexible and relevant curricula. In conjunction with the lifecycle shown in figure 8, an initial attempt was made by [13] to map elements of the curriculum lifecycle to the potential enhancements available through strategic use of enabling systems. One key point to make regarding this

mapping is that the most demonstrable benefits are achieved through an efficient flow

of information across all the elements of the lifecycle. This notion is congruent with the

proposed framework regarding the rigid objective to content mapping and content to

assessment mapping, allowing for streamlined information flow, tracking and feedback.

Table 2 gives detailed descriptions of the key activities and enabling technologies for

each phase.
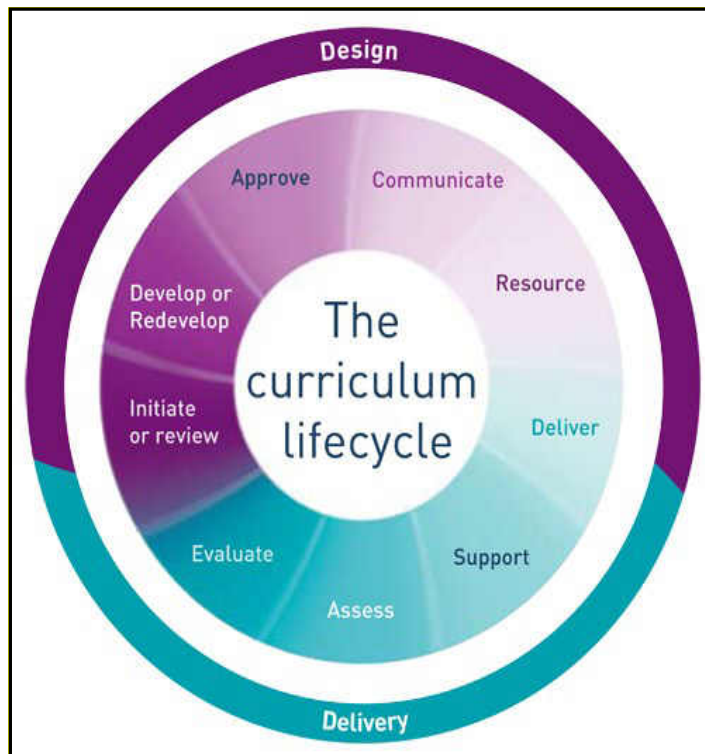
Figure 8 – JISC Curriculum Lifecycle [13].

Key visions for curriculum design as outlined by JISC include the ability to

respond creatively and flexibly to changing cultural and economic climates as well as

learning resources that are searchable, accessible and sharable.

Table 2 – JISC integrating technology into the curriculum lifecycle [13]

Level	Activity	Description	Enabling Technology
Design	Initiate or review	A new course is conceptualized to meet an identified educational need, or to fill a gap in the market. Or an existing course is re-developed in light of evaluation and market analysis.	Market research, enrolment and course evaluation data
	Develop or redevelop	As a result of a review, new elements of learning are developed or existing ones redeveloped. Course teams work to design or redesign how learning should be delivered, resourced, supported and assessed.	Learning design and pedagogic planning tools help teams to explore and share new concepts and designs. Market research, enrolment and course evaluation data – are used to support educational design. Learning design systems help define the relationships between, for example, learning outcomes, learning activities and assignments.
	Approve	Internal approval and validation is sought for new course, module or unit designs. This typically involves a committee-based process including one or more external members (such as regulatory and accreditation bodies).	Committee processes and workflows can be enhanced through e-admin systems. Information captured in definitive course documentation is managed efficiently and transparently
	Resource	Learning opportunities demand resources, human, academic and technical. Sessions are planned to determine the detailed timing of activities and the logistics of delivery, support and assessment. Physical and virtual learning spaces and learning content are prepared, whether designed from new or repurposed/instantiated for this course and cohort. Staffing issues are addressed.	Digital resources are made open, adaptive, accessible and available. Timetabling information is synchronized with information such as staff and student availability. Course related information is synchronized with library systems and learning repositories. Pedagogic planning tools can support logistical planning of sessions and optimal design in light of logistical or resource constraints.
Deliver	Deliver	Practitioners initiate learning activities in line with course purposes and learning designs, and in accordance with learners' prior experience and expertise. Learning resources and tools are introduced where appropriate, and made available to learners.	Technology-enhanced learning can engage learners and widen participation, increase choice and entitlement.
	Support	The diverse abilities of learners demand different types and levels of support; different modes of delivery may be considered to suit preferred patterns of attendance and approaches to learning. Curriculum is made responsive and adaptive to the requirements of different types of learners.	Learning designs accommodate learners' preferred tools and software, and assist learners in developing appropriate digital literacies and skills.
	Assess	Formative feedback assessment planning and delivery grading summative feedback reflection (learners)	Technology-enabled formative and summative can ensure prompt feedback and promote active learning. Technology can also record learning processes for reflection and review
	Evaluate	Learner achievement and feedback data, and evidence from course evaluation and review, feed into the lifecycle. Lessons learnt are shared and disseminated.	Data from virtual learning systems can be recorded; data can be aggregated and shared between systems to provide a more rapid, accurate and comprehensive overview

On the curriculum delivery front, JISC envisions scenarios for two distinct group of stakeholders; learners (who receive the curriculum) and teachers (who deliver it). For learners, they should be able to show evidence of their skills and achievements against the requirements of employers and professional bodies. For teachers, their practice will be informed by current research and evidence, they will be able to obtain timely access to learner information, and be able to give prompt, supportive feedback to learners.

2.4.1.3 Principles in Patterns - University of Strathclyde

The University of Strathclyde in Scotland is one of the universities that have a project that has been partially funded under the JISC Institutional Approaches to Curriculum Design Program. This project which is called Principles in Patterns (PIP) focuses on developing new ways of documenting and describing modules and courses so that students, academic staff and university managers and administrators can benefit from better information. Their ultimate goal is to improve the sharing and flow of information in order to enhance quality assurance processes and related university planning and monitoring activities (activities that span various points in the curriculum lifecycle). Here again it is seen where the theme of sharing and information flow is very prominent, a theme which no doubt will be explored in the proposed framework. A second goal is to enhance the educational impact of learning task, modules and programs by producing learning design 'patterns' which are based on some core pedagogical principles. The project aims to explore methods of representing these

design patterns so that academic staff members in departments and faculties can draw on them when engaged in design or redesign activities. Across clusters of modules or programs these patterns will help academic staff develop programs which offer a consistent student experience centered around some core principles (e.g. defined in relation to student engagement and empowerment) [14].

Out of all the projects which fall under the JISC portfolio, this PIP project by the University of Strathclyde is perhaps the one which most closely aligns with some of the efforts of this research, not necessarily in a broad sense, but specifically as it relates to modeling a curriculum. One of the problems that the JISC found is how to represent the curriculum in a way that can be reused to meet the needs of various stakeholders because of the tendency for different faculties within and between different institutions to store information in different forms and media. This of course makes the curriculum difficult to use or re-use, difficult to interact with and makes the tracking and flow of information more cumbersome. This goes back to the notion of standardization of curricula and the whole development process which the proposed framework attempts to do. The PIP project at the University of Strathclyde tackles this issue in some respect in that their project has seen the creation of a *demonstrator tool* to replace paper based class and course descriptors. This is similar to a course management system. The demonstrator version has achieved considerable standardization of the range of forms in use in different departments. It is already showing its potential to improve the efficiency and effectiveness of the approval process and will be further enhanced to

provide information about learning designs and a repository of pedagogical support materials [15].

2.4.1.4 How the Proposed Paradigm Differs from JISC's Projects

From surveying various papers relating to the work being done by JISC and related projects, it is apparent that their focus is an end-to-end integration of technology in the curriculum processes by utilizing various tools and pre-existing systems. It seems that the level of abstraction through which they address the problem is a level or two above what this proposed framework addresses the same problem. For example, take the assessment activity in which [12] suggests that technology can record assessment outcomes for internal evaluation and review, but what they do not say is how that will be achieved. What is the underlying architecture and model that will allow for that? How are the outcomes verified (did the student achieve the goals of the course/topic) and validated (did the assessment test the objectives and is it accurate and did it test what it is supposed to test)? These are questions that are not addressed by JISC but questions which are easily accounted for in the proposed framework. JISC suggests solutions to bring together various and sometimes disparate tools and technologies in a seamless workflow (lifecycle), but the proposed framework delves much deeper than that in order to put in place an infrastructure that is built from the ground up to achieve all of what JISC has put forward as benefits and much more. It

therefore means that the work done in the proposed framework can be used as an underlying architecture and framework that can be used in any setting.

2.5 Enhancing learning and teaching through the use of technology: HEFCE

The Higher Education Funding Council for England (HEFCE), in 2005 launched a strategy for technology enhanced learning. However, in 2009, that body published a revised approach [16] which follows an independent review of the strategy and was designed to provide further support to higher education institutions as they develop their own e-learning strategies. Even though the focus in [16] is on e-learning, there is certainly enough mention of the use of technology to enhance teaching and learning to make it relevant to this work. Curriculum development is just a subset of their work and that is where the emphasis will be placed in this review.

According to HEFCE, the term 'e-learning' is sometimes too narrowly defined to fully describe the widespread use of learning technology in institutions. It is with this in mind that HEFCE believes it is more important to consider ways in which institutions can enhance teaching, learning and assessment by using appropriate technology. Therefore for them, the focus is on the benefits and the outcomes from using technology to support learning and related processes which will be different in each institution [16]. This last point here is very critical because it points to the fact that there are inherent benefits to be garnered from the use of technology in not just the learning process but other related processes such as curriculum design and assessment, coupled with the

fact that these processes will be different in each institution. HEFCE's emphasis is on recognizing that technology has a fundamental part to play in higher education, but that institutional contexts and strategies are key; bringing back the point that curriculum design should be context sensitive (an attribute the proposed framework addresses).

In the same publication, a 2008 survey of technology-enhanced learning for higher education in the UK [17] by the Universities and Colleges Information Systems Association (UCISA) is also mentioned. It is noted that the survey is useful in demonstrating how far the educational sector had come in its use of learning technologies. The survey found that a wide range of centrally supported software is used, but institutions are aware of students using many others and that technology is used for a variety of purposes. Some of the tools that were pointed out as supporting teaching and learning include e-assessment, e-portfolios, podcasting, blogs and wikis. The reason for highlighting this survey is that it is important for an institution to know what resources are available to not only the faculty, but also to the students. This may be information which is critical during the curriculum design process.

It is inarguable that assessment is a central facet in any teaching and learning endeavor. In making the case for the use of technology in any educational setting, there must be provisions for the effective use of electronic multiple choice. According to [17], e-assessment is now widely used for summative assessment such as end of module tests and most of its current use employs computer-marked, objective questions. It is also noted that the main benefits of the technology is the immediacy of feedback to students and the reduction of marking for tutors. These are benefits which have been mentioned

in chapter one of this paper and it will be shown in the proposed framework how integrating assessment into the whole curriculum development process, in a very finely granular way using technology will greatly amplify such benefits. There is evidence that indicates that well-designed and well deployed diagnostic can foster more effective learning for a wider diversity of students [18]. E-assessment enhances the process of reporting, storing and transferring data associated with public and internal assessments.

Table 3 shows a snippet of the HEFCE framework to assist institutions in maximizing the strategic benefits of technology. It highlights areas where institutions may see benefits from investing in technology. The idea behind the framework is that it should help institutions identify priorities for development as well as for it to be flexible enough so that institutions can adapt it to suit their own needs. So once again the theme of adaptability is coming out here and as one correlates this HEFCE framework to that of the proposed paradigm, the importance of a curricula being context sensitive is emphasized.

Some of the points raised in Table 3 are quite relevant to the work being done in this research. In the third column (harnessing technology for strategic gain), examples of development goals are given and many of these goals are congruent with the motivation for this work outlined in chapter 1. For instance, one of the goals mentioned in the area of *Pedagogy, Curriculum design and development* is "Technology is used to help identify learners with specific aptitudes or needs." This is one benefit which the proposed framework will be able to easily harness. Another goal worth mentioning falls in the area of *Learning resources and environments* where [17] gives an example of

tutors collaborating in subject communities to produce high quality, reusable learning

resources.

Table 3 - Enhancing learning and teaching through the use of technology: a suggested framework for institutions [17]

Activity area	Strategic priorities	Harnessing technology for strategic gain
Pedagogy, Curriculum design and development	• Enhancing excellence and innovation in teaching and learning • Engaging employers (or other stakeholders) in curriculum design and delivery • Improving efficiency of curriculum design and delivery processes	• Tutors have access to a wide range of tools to support teaching, and a wide range of high-quality resources to engage students • Innovative uses of technology for learning are supported by the curriculum design process • Technology is used to enhance the responsiveness and flexibility of curriculum offerings • Technology is used to help identify learners with specific aptitudes or needs • E-assessment technologies are used to support innovative practices such as just-in-time assessment and peer review
Learning resources and environments	• Widening participation and improving access • Effective management of learning resources	• Tutors are collaborating in subject communities to produce high-quality, reusable learning resources • Tutors have access to relevant learning resources, and support for adapting, integrating and enhancing them • There is continuity across learning, teaching, research and administrative environments to support joined-up processes
Quality	Institutional quality processes can support objectives and enhance benefits in all the other areas	Institutional quality processes are agile enough to respond quickly to learner and employer needs
Research and evaluation	• Enhancing excellence in learning and teaching • Enhancing institutional processes	• Staff have access to research, evidence and scholarship to inform curriculum development and research-based teaching • Institutions have effective mechanisms for evaluating learners' experiences of learning, including learning with technology

Again, the nature of the proposed framework speaks to the need and implementation of collaboration, not just internally (within institutions), but also internationally (across borders). This will result in the sharing and utilization of teaching and learning artifacts, compiled and collated in some repository by experts and authorized persons in a given field and accessible to all who need it. In table 3 there are many other instances that highlight the parallel between what HEFCE is doing and the proposed framework and a simple perusal of the table should provide a quick overview.

2.5.1 The Proposed Paradigm and HEFCE Compared

Similarities

1. HEFECE like the proposed paradigm aims to enhance teaching, learning and assessment through the use of technology. Their approach is a little more in line with the proposed paradigms notion of the holistic integration of technology in the entire process.

2. HEFCE also recognizes the fact that each institution is different and therefore has a unique set of needs which must be met individually through technological tools and methodologies customized for their specific needs and context.

Differences

1. The proposed paradigm differs from HEFCE in a number of ways. Firstly, HEFCE focuses on e-learning which is only one component in myriad of delivery and

assimilation options available. The proposed paradigm could easily encompass e-learning since the holistic integration of technology in the curriculum design process is not restricted to the mode of delivery and in fact, e-learning can be improved through use of the proposed paradigm.

2. Secondly, HEFCE takes an approach which assumes that various infrastructures and technologies are already in place. They have taken the approach of trying to fit together disparate pieces of technologies into one as if trying to fit together pieces of a jigsaw puzzle in which the pieces do not perfectly align (that do not have perfect matches). HEFCE talks about the technological tools that may be available to students but not about how a curriculum can be designed to incorporate such tools. The proposed paradigm on the other hand, ensures that the resource available to both faculty and to students is taken into consideration during the design process.

2.6 Improving University Curricula - Richard Gluga

It is safe to say that the review of literature for this research would be incomplete without looking at the work done by Richard Gluga, who at the time of this writing is a PhD student at the University of Sydney. His research interests include large-scale, long-term learner models supporting flexible curriculum definition and he has done extensive work in improving university curricula. Gluga notes that there is a growing need for the design of higher quality university curricula that better prepare

students for employment in their chosen discipline. Immediately one can infer that the focus of Gluga's work is not necessarily on integrating technology in the curriculum design process, but the thinking is that technology may be a means to his work's end. What you will find is that his work makes use of various proprietary tools, methodologies and architectures to arrive at possible solutions to the problem at hand. To this end, the approach of Gluga et al has been to design, engineer and evaluate advanced curriculum information management systems, which at its heart enables the systematic mapping of relevant skills and mapping of competence levels as well as the reporting and visualization of skills developed throughout an entire degree. The mapping of skills is across all subjects and assessments of a degree, while the mapping of competence levels is associated with the skills and assessments. One thing to quickly point out here is the various mappings which are so integral to the approach taken by Gluga, but whereas Gluga talks about the mapping of skills and competence levels, the proposed framework is more concentrated on the mapping of goals to content and content to assessment. So from a high level, Gluga's work is more concerned with systematically tracking the skills that students develop, intertwining that with requirements from various stakeholders (like accreditation bodies and standards bodies) and having the ultimate aim of enhancing the caliber of future graduates.

In [19], Gluga speaks to the complexity involved in curriculum design and the problems which arise from having flexible degree programs that must meet multiple accreditations, professional and institutional requirements. Gluga, in his works, attempts to satisfy all stakeholders involved. In keeping with this, a number of questions

42

are posed from each stakeholder's perspective, for which the design of every degree should satisfactorily answer. These stakeholders include standards and accreditation bodies, employers, faculty and university administration and students. Recall that the scope of this work overlooks administrative issues and concerns with the focus on teaching and learning. Figure 9 is a snippet taken from [19] which covers three of the more important stakeholders relevant to the proposed framework. It shows side by side questions that are relevant to each.

Degree Designer

1. What are all the relevant accreditation/national/professional requirements that Degree X must satisfy?
2. Which units can best satisfy these requirements?
3. Do all elective paths taken satisfy the minimum requirements?
4. How are all the requirements being taught and assessed? Is anything being left out?
5. Does the degree structure make sense in terms of pre-requisite knowledge and skill progression?

Subject Lecturer

1. How does my unit fit into the big picture of the degree(s) in which it is being offered?
2. What are the relevant skills, competencies and knowledge that my unit needs to teach?
3. What skills and knowledge can I expect students to have prior to commencing my unit?
4. What is the most effective way for me to teach and assess these skills and knowledge?

Enrolled Student

1. What units must I enrol in to complete Degree X?
2. What skills and knowledge do employers in Discipline X value?
3. How do the units I am studying relate to a job in Discipline X?
4. Which elective units will best prepare me for a job in Discipline X?
5. Which elective units will allow me to qualify for accreditation in Discipline X?
6. If I transfer from degree Z, which competencies and skills do I already have that are relevant and can be credited to Discipline/Degree X?
7. What's the most efficient selection of electives that allows me to meet all competency/accreditation requirements for both Discipline X and Discipline Y?

Figure 9 – Snippet from [19] showing questions posed by degree designers and subject lecturers.

From the questions posed, one can see the various issues that are important to each stakeholder and how those issues span all the stakeholders at varying levels of granularity. For instance, a degree designer may be interested in knowing if the degree structure makes sense in terms of pre-requisite knowledge of both the faculty who will be delivering the given unit(s) and of the students who will be taking the unit(s). The subject lecturer (faculty) in turn will want to know what skills and knowledge students ought to have prior to commencing a unit. Subsequently, the enrolled student will need to know what units they can enroll in based on certain pre-requisites (such as prior knowledge, skills and pre-requisite courses) and what units they must enroll in to complete Degree X.

A key point that is brought about in figure 9 is the flexibility of degree programs to allow for electives. Recall that a degree program can comprise of electives which may not necessarily be a part of the core courses for said program. These electives can be drawn from intra-disciplinary (within the same discipline) as well as inter-disciplinary (from other disciplines) subjects/courses. So the degree designer needs to ask the question "Do all elective paths taken satisfy the minimum requirements?" The notion of minimum quality threshold is an idea that the proposed framework will address because the framework will allow curriculum designers customization abilities based on their individual (institutional needs, students' needs, available resources and so on).

To further elaborate on Gluga's work, one can look at the questions which apply to enrolled students. For instance, a student may ask "Which elective units will best prepare me for a job in discipline X?" The system should be able to easily answer those

questions based on the requirements of the job in the given discipline and the skill-sets mapped to the individual subjects.

2.6.1 Course and Unit of Study Portal

Richard Gluga et al. have developed an architecture and a tool called CUSP (Course and Unit of Study Portal) that serves a curriculum information tracking system which facilitates the systematic tracking of skill and competence level progression in a Computer Science context. The work entitled "An architecture for systematic tracking of skill and competence level progression in Computer Science" by Gluga et al. [20] is felt to be an important pre-cursor to the proposed framework. The work surrounding CUSP touches on some issues that are important to the proposed framework and some of the methods will certainly be useful at various stages of this effort. The issue of effectively modeling curriculum skills, mapping them to assessment tasks across subjects of a degree, and measuring the progression in learner competence level is seen as an unsolved problem for the most part. So the key contributions in [20] are in the exploration of principled approaches to formulating curricula so that the long term learning over a full degree program is more effectively planned and monitored; which will mean that students can be assured a coherent series of learning experiences that build to achieve the key learning goals.

CUSP is a tool that has been engineered and deployed which is used to map graduate attributes and discipline competencies across over 200 degrees and over 2000 units of study. CUSP provides live big-picture visualizations and reports that are vital for

accreditation and curriculum quality control [19]. It allows degree coordinators and subject lecturers to map and visualize transferable generic skills and accreditation competencies across whole degree programs [20][21]. CUSP captures the representation of multiple sets of graduate attributes and accreditation competencies (named curriculum goals) and maps these to the relevant degrees. Each degree structure is then modeled into the system as a collection of core subjects plus the rules governing the selection of elective subjects. A high level design of CUSP is shown in figure 10. As shown, a primary skill set is selected for a given degree and then the skill levels from this primary set are mapped to subject learning outcomes which are then mapped to assessments. Such a design enables the CUSP system to generate reports that visually depict the curriculum coverage for entire degrees against some goal or attribute.
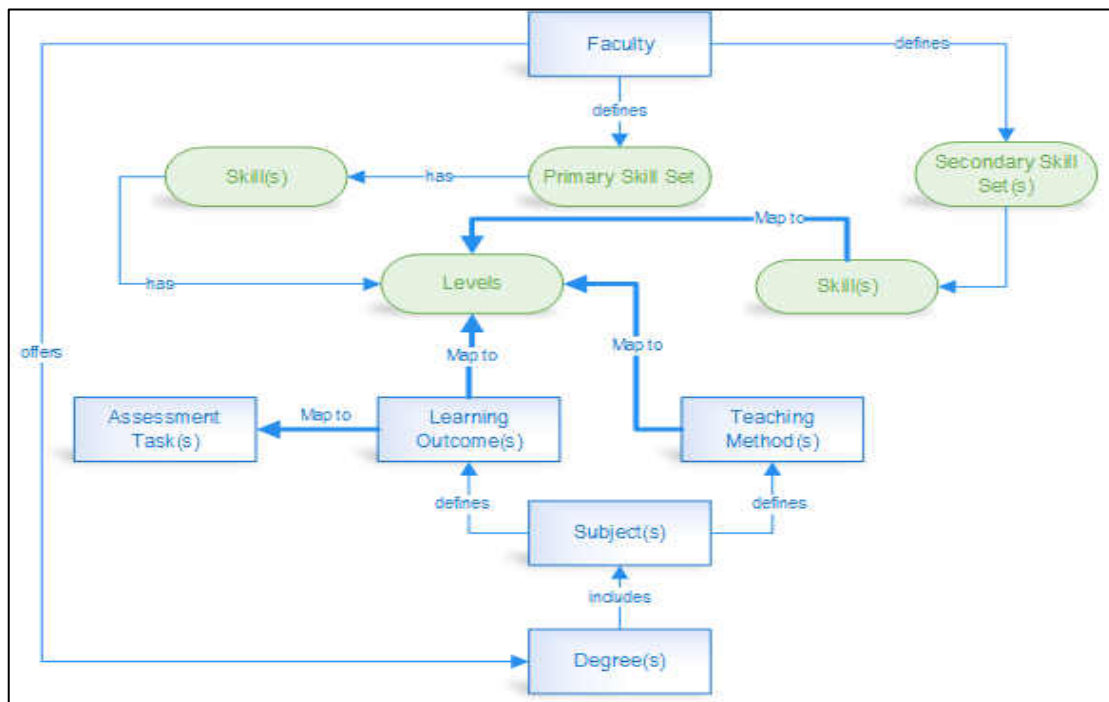
Figure 10 – CUSP High Level Design [20]

This system is also useful for quality control audits, formal accreditation and ensuring that content is taught and assessed in an appropriate and effective progressive sequence. It is widely known that the aim of university degrees should include instilling both generic transferable skills (such as communication and team work) and discipline specific skills in students. Gluga contends that learners need to develop these skills progressively over the course of a degree program (typically 4 years), aided by a suitable sequence of learning experiences. The mapping and tracking of such skills to individual assessment tasks throughout the subjects that comprise university degrees is at the core of their architecture and design for implementing a skill and competence level mapping solution.

The problem of managing the flexibility in programs of study such as is the case with electives has been addressed fairly well by the system in [20]. CUSP generates charts that show the assessment weight associated with skill sets for individual degrees. An example is given in [21] and shown here in figure 11, where the x-axis has seven engineering graduate attributes and the y-axis gives the percentage of assessment weight for the degree as a whole.

The authors postulate that the *Information Skills* attribute is the most under-assessed for the given degree program and that being able to quickly and readily visualize this is quite valuable and can therefore lead to optimization of the curriculum. The system algorithmically calculates the minimum set of skills that a student can be assessed on based on the elective subject options that they make. So if one elective subject has a 40% assessment task associated with a certain skill, while another elective

subject has a 20% assessment task associated with that same skill, then CUSP will

generate the report based on the latter (which has the lesser weight). According to the

designers of the system, this ensures that the reports show the 'worst case' scenario for

the assessment weight of each skill. This has implications for both students who may try

to 'game' the systems in terms of the electives they choose as well as for curriculum

designers who need to ensure that there is adequate assessment coverage for every

subject. The authors also mention that the system should be able to show reports that

differentiate between the skills levels of a top student versus that of a student who is

barely passing. The importance of this is that it will help to determine the minimum set

of learning objectives from a given curriculum that a student must satisfy and at what

level of competence, in order to be a graduate in the given discipline.
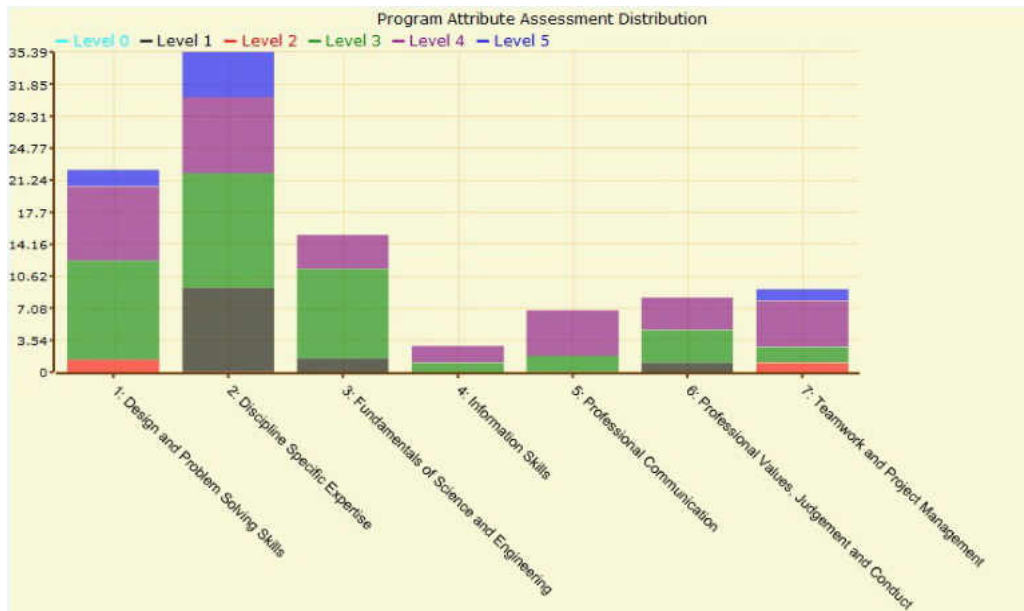
Figure 11 – A stack column chart showing sample output chart generated by CUSP [21]

The authors in [21] postulate that learning requirements drive the design of a degree structure in each institution and can be represented as a set of curriculum goals that the degree aims to enable students to achieve. As of such, they view a degree structure as a collection of Core and Elective subjects, each of which are broken down into Entry Requirements, Assessments and Exit Conditions, all of which are mapped against curriculum goals and competence levels. It is important to highlight the fact here again that the authors keep mentioning the word mapping, which is a very key concept in the proposed framework.

Another tool mentioned in [20] is ProGoSs (Program Goal Progression), which is under active development and focuses on systematic methods to measure learning progression. ProGoSs engrains Bloom's Taxonomy which is particularly important since Bloom plays a key role in defining curricula like the current ACM/IEEE-CS curriculum guidelines.

2.6.2 A Synergy worth Exploring

As mentioned at the beginning of section *2.6*, the work done by Gluga et al. is one of the more important ones to have been reviewed for this research. It is worth mentioning that at the CSEIT 2011 conference in Singapore, the author of this work presented [27] and also had the pleasure of meeting Gluga. One of the comments that Gluga made following the author's presentation was that transforming the curriculum design process is a very complex, labor intensive and time intensive undertaking,

especially when one considers the many factors that must be considered (both internal to the institutions and external as well).

2.6.3 How the proposed paradigm differs from Richard Gluga's work

Gluga's work incorporates requirements from external stakeholders such as standards bodies and employers as well as internal requirements such as those from school administrators. Recall from the scope outlined in section *1.4* that the proposed paradigm is limited to focus on teaching and learning from the level of the faculty through to students, and so the direct stakeholders are limited to faculty, degree designers and students.

Another contrasting point is that Gluga focuses on mapping skills and competence levels and the visualization (reporting using bar charts) of such data, while this work looks into much more than that and takes a more in-depth approach. Gluga focuses on the mapping of skills derived from the subjects taken and then attributes competence levels to those skills which are further derived from assessments. The approach in the proposed paradigm is more end-to-end in that it seeks to be more holistic by going from goals to content and from content to assessment. In this way, the derivation of the skills attained, the competence levels of such skills and myriad other metrics such as course coverage and assessment coverage can be more easily achieved. As a side note, Gluga speaks of skills, which in the proposed paradigm are really the goals (learning outcomes), since a skill comprises one or more goals.

The granularity of Gluga's work seems to be a level above that of the proposed paradigm in that the content is viewed from the course level and above. For the proposed paradigm, the granularity is much finer in that the content goes all the way down to the level of topics.

CUSP, which is summarized in section 2.6.1, facilitates the systematic tracking of skill and competence level progression in a computer science context and is more focused on an individual. The proposed paradigm on the other hand, goes further in that it tracks the quality of a program of study from inception to completion and can be viewed as both individualistic (per student) as well as generalist (per degree program).

One thing that is similar in both works is the concept of minimum coverage, whether it be for the minimum set of skills that a student can be assessed on (in the case of Gluga) or the minimum quality threshold that a curriculum must meet (in the case of this work). There are other areas in which the coverage concept is applied in this work, such as how much does a given assessment cover a certain content or how much of the content for a course was covered by the sum of assessments or actual delivery. These issues will be covered later in the paper.

2.7 Towards an Internet Based Education Model for Caribbean Countries

In 2000, Grant et al. published a paper entitled "Towards an Internet-based Education Model for Caribbean Countries" which looked into leveraging the usage of low-cost technology to offer *Virtual Classroom* environments to teaching institutions in remote areas via the Internet. The work in [23] is being mentioned here because it was

important to the initial motivation that spawned this research. In developing countries there is a high student-teacher ratio and further still within those countries, the majority of high-quality teachers and teaching facilities are located in the urban areas.

There is no question that in recent years there has been significant improvement in the adoption and availability of broadband Internet and other Internet-related technology infrastructure in many developing countries. For instance, Jamaica now boasts a robust fiber optic network providing wired broadband internet service to most of Jamaica's major towns and urban centers, as well as mobile broadband internet through cellular providers which provides coverage to the majority of the island [24]. At the time of the writing of [23], such advances had not taken place, and so the potential is even greater now for such technologies to be used within developing countries to enhance the delivery of high quality education. The delivery of educational materials, according to Grant et al. over the internet is almost commonplace in more affluent developed countries. The sharing of teaching resources and the implementation of the concept of 'classrooms without borders' are quickly becoming standard in many major institutions of higher learning. For instance, the students of a busy professor who travels often to conferences and other engagements overseas need not worry that they will miss valuable face-time and teaching experiences from the professor. This is because that same professor can still deliver a quality teaching experience to his/her students regardless of time and space through the use of various technologies. For example, the professor may prepare a recorded lecture while waiting at the airport terminal (or in a hotel room), that can then be played to students (in a future lecture), or conduct a real-

time lecture via Skype (or some other video-conferencing means), or interact with students via SMS, chartrooms and emails.

This segues nicely into one of the main motivating factor in [23] which is to ensure that there is the capability to provide consistent, high-quality educational materials to students across a wide geographical area, in a timely manner and with limited human and physical resources. Some of the benefits highlighted in the paper include:

- The availability of high quality teaching materials and teaching methodologies to geographically remote schools in developing countries.

- Rural schools can now obtain the same material and pedagogical skills which are available to their counterparts in the best urban schools.

- The negative effects of a large student/teacher ratio can be suppressed and not extend to students utilizing Virtual Classrooms in a great way, because of the required concentration and attention demand on such students.

- Virtual Classroom students will transparently develop and expand their computing skills as a side effect from using the system.

The end-users in the Virtual Classroom concept are ultimately the students who would be the clients, while the end-users in the proposed framework are the teachers, faculty, the institution and transiently the students who interact with the model at various levels (whether it be in receiving content through the teaching methods or participating in assessments). Figure 12 depicts this contrast beautifully where the

Virtual Classroom Over the Internet (VCOIN) on the right side of the diagram shows at

a low level, the physical setup of the infrastructure. If one were to abstract away the

details of the architecture of VCOIN it would be possible to replace it with the high

level view shown on the left, with the only difference being the users on the left which

are not directly students.
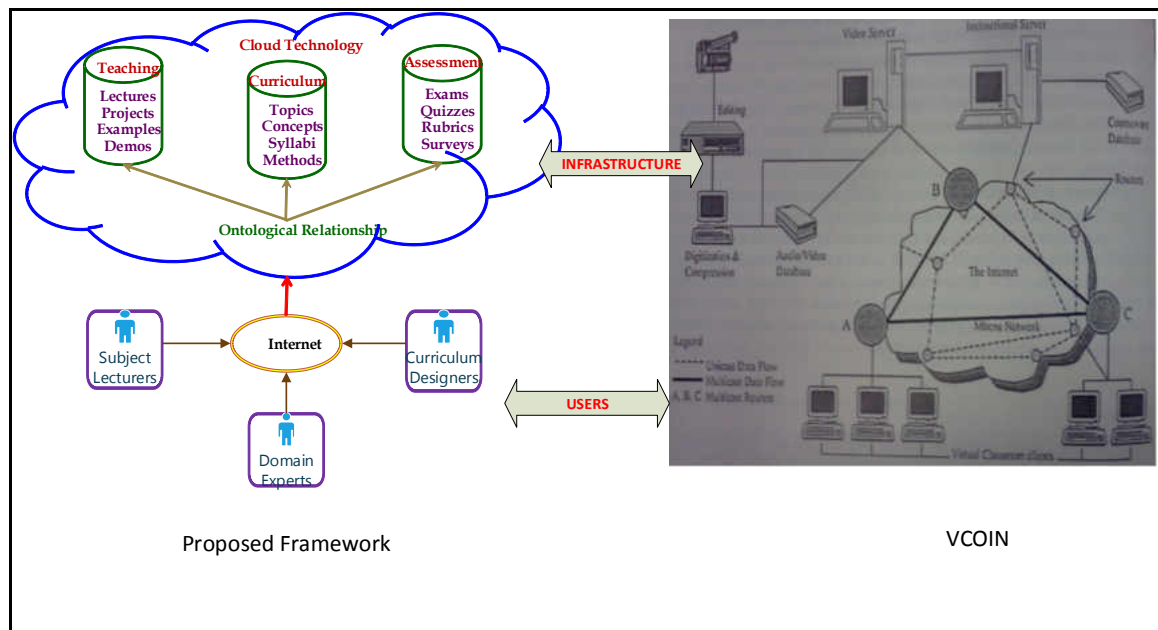
Figure 12 – Diagram showing User and Infrastructure View of the proposed
framework being contrasted with the architecture of VCOIN as seen in [23]

The work in [23] was geared towards primary (elementary) and secondary (high)

schools, but one can see clearly how such a model could easily be extended to higher

learning institutions such as community colleges and universities. The proposed

framework in this paper shares with [23], the theme of reaching a wider cross section

of students especially in developing countries through the seamless sharing of

teaching and learning resources and the design of curricula that is sensitive to the needs and resources available to the target institutions.

2.8 Online Collaborative Teaching of Software Engineering

In 2010, Emanuel Grant along with other faculty from institutions in South-East Asia embarked on an ambitious international collaborative Software Engineering teaching research project. The project got going with a series of international workshops where the goal was to identify a core set of topics for teaching software development over a four year degree program, which is relevant to the growing Information Technology needs of South-East Asia. Software Engineering is widely recognized as a cornerstone of computer science. However, there is room for improvement and directed maturity and various problems exist:

1. Teaching of the subject is not standardized,

2. There is no sharing of teaching resources across departments nor institutions

3. There is a wide diversity of topics

4. No proper tracking of deprecated and outdated topics as technology evolves and improves

5. Wide choice of text, methodologies and techniques

There are many issues that are concerned with the teaching of Software Engineering, some of which are specific to the domain and others which are more generalized. Questions such as "what is the best way to assess the learning of the

fundamental concepts of software engineering?" A more in-depth look into such issues among others will be presented in subsequent chapters of this paper. In [25], Grant suggests a list of desired solutions that would in part address the problems mentioned above and go a long way in effectively tackling the plethora of issues surrounding the pedagogy of the discipline:

1. Establishment of an open framework,

2. Common teaching material (lectures, assignments, course projects, etc.),

3. Common teaching assessment process, and delivery platforms,

4. Environment for sharing and structuring curricula in an international context.

A cursory look at the four items listed above immediately paints a picture of the standardization and maturation of the Software Engineering discipline in an environment which fosters collaboration and seamless sharing. This is the essence of just one facet of the proposed framework, and the ideas raised here will blend nicely into said work. Software Engineering is 'ripe' for continued improvement and development as a core discipline in computer science. It must be dynamic enough to keep pace with evolving technologies and to be as current as possible. This is one of the reasons why Software Engineering is being used as a case study for the proposed framework.

In keeping with the nature of the project in [25], there are now participating institutions from several regions all over the world. The institutions in the research project include, but not limited to:

- Baylor University, Waco, Texas, USA

- HELP University College, Kuala Lumpur, Malaysia

- International Institute of Information Technology, Hyderabad, India

- Holy Angel University, Angeles City, Pampanga, Philippines

- Jimma University, Ethiopia

- Montclair State University, Montclair, New Jersey, USA

- University of North Dakota, Grand Forks, North Dakota, USA

- University of the West Indies, St. Augustine, Trinidad

As mentioned earlier, the project began with a series of pre- and post-repository development workshops that were planned to gather necessary input data for the research on defining and developing the project repository and repository artifacts. Figure 13 shows a glimpse into what the inputs and outputs of the workshops entailed and what the progression of the work would look like.
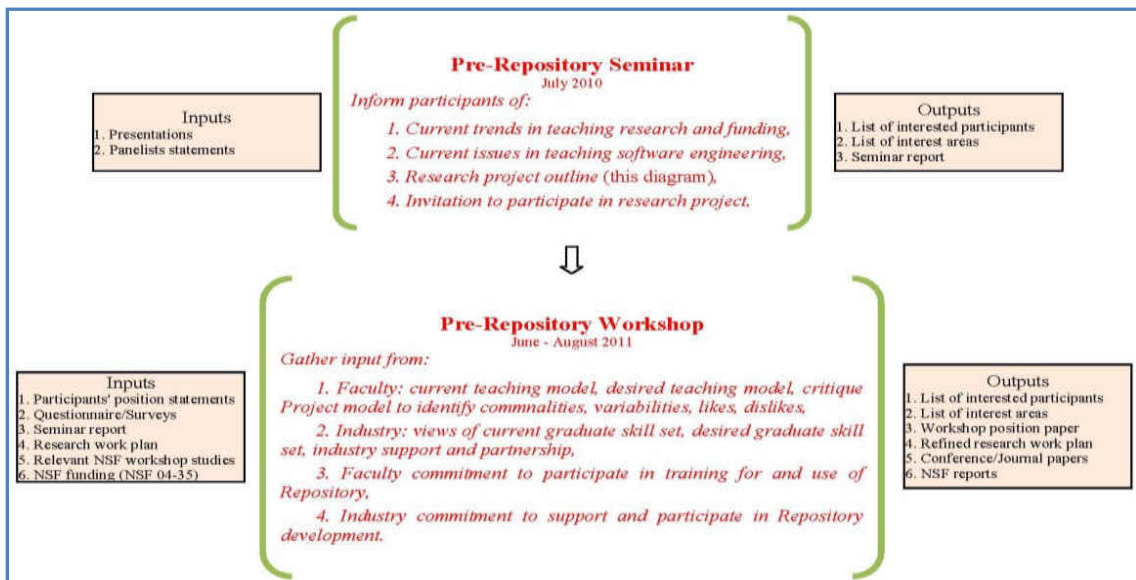
Figure 13 - The activities of the first phase of the project [25].

The goal of the project is to formulate a framework for structuring undergraduate software engineering curricula. The core of the framework will be a repository of essential software engineering teaching modules, assessment artifacts, course projects and assignments. The repository will be searchable collection of teaching material that will be available online for open use. However, one of the key aspects of this repository is that it will enable the assembling of such material to fit the requirements of a particular curriculum for teaching software engineering. This is just another way of saying that the repository will allow for a curriculum to be context aware and dynamic in design (a feature that the proposed framework will cover).

2.8.1 Workshop on Enhancing Teaching & Learning of Software Engineering in an International Environment

The Institute of Electrical and Electronic Engineering Computer Society (IEEE-CS) and Association of Computing Machinery (ACM) Software Engineering 2004 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (IEEE-CS/ACM SE 2004) [26] was used as the main source of input data for the workshop in [25]. This document is the second version of the original document produced in 1991, and is currently under review for the release of a more updated version. The IEEE-CS/ACM SE 2004 primary purpose is to provide guidance to academic institutions about what should constitute an undergraduate SE education. The IEEE-CS/ACM SE 2004 defines ten subject areas, which are the Software Engineering Education Knowledge (SEEK) areas. More will be said about this IEEE-CS/ACM SE 2004 document in section 3.3.

Table 4 is an example of one of the output tables that was produced at the workshop and it shows the results of the deliberations of one of the panel groups on Professional Practice and Software Management (two of the SEEK Knowledge Areas). It shows the recommendations for the year/s in which each topic should be taught, the depth at which they should be taught (the sample table shows depth instead of hours as is the case in [26]), as well as a rationale for each of the attributes ascribed to each topic. The depth is measured as a percentage of time that should be dedicated to the given topic.

Table 4 – Sample workshop output table [28]

Professional Practice			
TOPIC	YEARS	DEPTH	RATIONALE
PRF.psy	1 to 4	15%	# 5 & #6 be change from K to C->"DEALING" is not merely KNOWING. Dealing needs to provide course of actions to certain issues e.g. forecasting, predicting, identifying, presenting alternative solutions. All items are ESSENTIAL
PRF.com	1 to 4	40%	All items are ESSENTIAL
PRF.pr	1 & 4	45%	All items are ESSENTIAL
Software Management			
TOPIC	YEARS	DEPTH	RATIONALE
MGT.con	2 to 4	10%	All items are ESSENTIAL
MGT.pp	2 to 4	31%	All items are ESSENTIAL
MGT.per	2 to 4	11%	All items are ESSENTIAL
MGT.ctl	2 to 4	21%	All items are ESSENTIAL #5 & #6 be change from O to E and classified as K->to achieve quality performance/output would require effective supervision/management
MGT.cm	2 to 4	27%	All items are ESSENTIAL #7 be change from D to E and classified as A->Security is an essential part of software configuration management. Topic #7 must be renamed as Distribution, Back-up, and Security. #4 be change from C to K->BUILDS deals more on conceptual/theoretical rather than actual application in the classroom set-up

It is expected that the work done in [27] will serve as a foundation in this work with the guidelines in [26] being an important contributing component. A discussion

into the relevance of this work to the proposed framework will ensue in the methodology in chapter 3. The 4 panel groups were in high agreement with the original SEEK topics and a set of topics for teaching software engineering across the four years of an undergraduate program was identified. Much work has already gone into the IEEE-CS/ACM SE document, and so the aim of the workshop was not to re-invent' the wheel and unnecessarily redo work, but rather to tailor the curriculum to meet the needs of the target audience.

2.9   IEEE-CS/ACM SE Curriculum Guidelines – A Framework Input Source

It is anticipated that the major source of input for the content and the initial framework is the Institute of Electrical and Electronic Engineering Computer Society (IEEE-CS) and Association of Computing Machinery (ACM) Software Engineering 2004 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (IEEE-CS/ACM SE 2004). The purpose of this document is to provide guidance about what should constitute and undergraduate Software Engineering education. The recommendation found therein have been developed by a broad, internationally based group of volunteer participants which took into account much of the work that has been done in software engineering education over the last quarter of a century. Such a document as this is of great importance given the current surge in the creation of software engineering degree programs.

The idea is not to re-invent the wheel by trying to redo the work that has gone into the IEEE/ACM SE document, but rather to take that document and enhance it and use it as a source of input for this framework.

2.9.1   Summary of IEEE-CS/ACM SE document

The body of knowledge that is deemed as appropriate for an undergraduate program in software engineering is designated as the SEEK (Software Engineering Education Knowledge). According to [26], knowledge is a term used to describe the whole spectrum of content of the discipline to include information, terminology, artifacts, data, roles, methods, models, procedures, and so on. SEEK is organized hierarchically into three levels. The highest level is the education *Knowledge Area* which represents a particular sub-discipline of software engineering that is generally recognized as a significant part of SE knowledge that an undergraduate should know. Knowledge Areas are like a high level way of organizing and describing software engineering knowledge. The second level is called *units* where each knowledge area is broken down into smaller divisions or modules. Each unit is then subdivided to form the lowest level which is a set of *topics*. See figure 14 for an illustration of this hierarchy. There are ten knowledge areas that make up the SEEK.
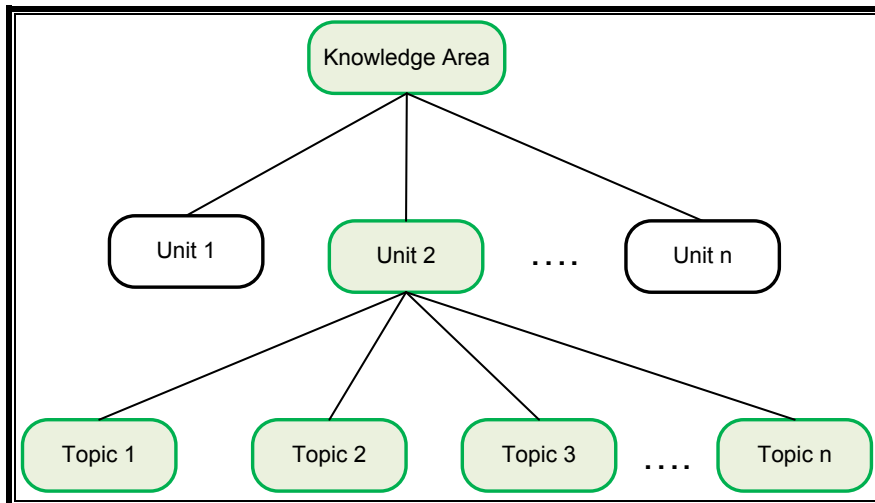
Figure 14 - SEEK hierarchical organization showing Knowledge Areas, Units and Topics

In [27], the IEEE-CS/ACM SE document was used as the main input source for a workshop to identify a set of topics for teaching software engineering across the four years of an undergraduate program. This workshop was a step towards the establishment of a proposed repository that will serve as a tool to enhance the teaching and learning of Software Engineering in an international environment.

2.9.2 Development Process of SE 2004 Volume

The recommendations presented in [26] have been developed by an international cohort of volunteers. This is somewhat akin to what will be required for the development of the repository in the proposed framework and which has been demonstrated already with the participants in the workshops in [25]. So the incorporation of international collaboration is vital to the success of such a major undertaking. There are three major efforts that engaged the volunteers and a Steering Committee in the construction of the volume:

- The development of a set of learning outcomes

- The determination and specification of what every SE graduate must know (SEEK)

- The construction of a set of curriculum recommendations describing how a SE curriculum incorporating the SEEK could be constructed in various contexts (eg Delivery, Assessment).

Development of the SE 2004 volume was a spin-off from the Computing Curricula 2001 task force which recognized the huge breadth of the computer science domain and hence the need to have a more narrow focus on sub-disciplines like software engineering. Initial aims of the task force included having curricular guidelines that were current (by matching the latest developments of computing technologies) and being future proof (by having curricula that would endure through the next decade). These are all ideas that are important and considered in the proposed framework.

The initial body of SEEK Knowledge Areas was developed by Education Knowledge Area volunteers, leaders in software engineering education and Pedagogy Focus group members, and later refined by a Steering Committee. A resulting SEEK document was then reviewed by a set of internationally recognized software engineering experts, followed by several rounds of open reviews by external entities. The approach taken by ACM/IEEE here, once again highlights the collaborative efforts necessary from educators and experts in the given domain. Also, development of a curriculum model is not an 'overnight' effort as seen with the SE 2004 volume, in which

there were incremental improvements over time following reviews and contributions by experts in the domain. A somewhat similar approach will be taken in developing the Repository in the proposed framework, but with added robustness and fluidity to ensure that the result is not a document with a 'release-cycle' of eight years, but rather a system that is dynamic and adaptable to changing needs on the fly.

CHAPTER III

METHODOLOGY

3.1 Description of Methodology

In describing the methodology for this work, it is important to draw attention to the scope as outlined in section 1.4 of this paper. The scope places focus on the teaching and learning aspects of a curriculum and assume that there is already a set of units and accompanying topics that will constitute the content. The Repository lies at the heart of the proposed framework. Recall that in [27] the focus is on Software Engineering and so the Repository was put forward to be the major part of the framework to contain essential software engineering teaching modules, assessment artifacts such as exam questions, course projects and so on. Essentially, SE educators will be able to access and use the Repository to develop course curriculum and syllabus. The methodology will be broken down into three distinct phases:

1. A model that captures the building of the repository

2. The modeling of a curriculum

3. The utilization of the curriculum

Figure 15 gives a broad overview of what the methodology will entail. Phase 1 will detail the framework for the initial building of the repository and its future maintenance. This will build on the work already done by Grant et al. in [25][27] in order

to clearly outline a path to the realization of the repository. Phase 2 is the most involved of the three phases and this is where an attempt will be made to model a curriculum given the input from Phase 1. Finally, phase three will detail the utilization of the repository and the modeled curriculum, such as a faculty designing a sequence of courses and accessing various teaching artifacts, the delivery of the content, student assessments as well as automated feedback to the repository. Such feedback can be private in nature where various statistical and anonymous data are uploaded to the repository. Of course, for countries like the US, this approach would steer clear of any potential legislative issues like FERPA (Family Educational Rights and Privacy Act)[1].
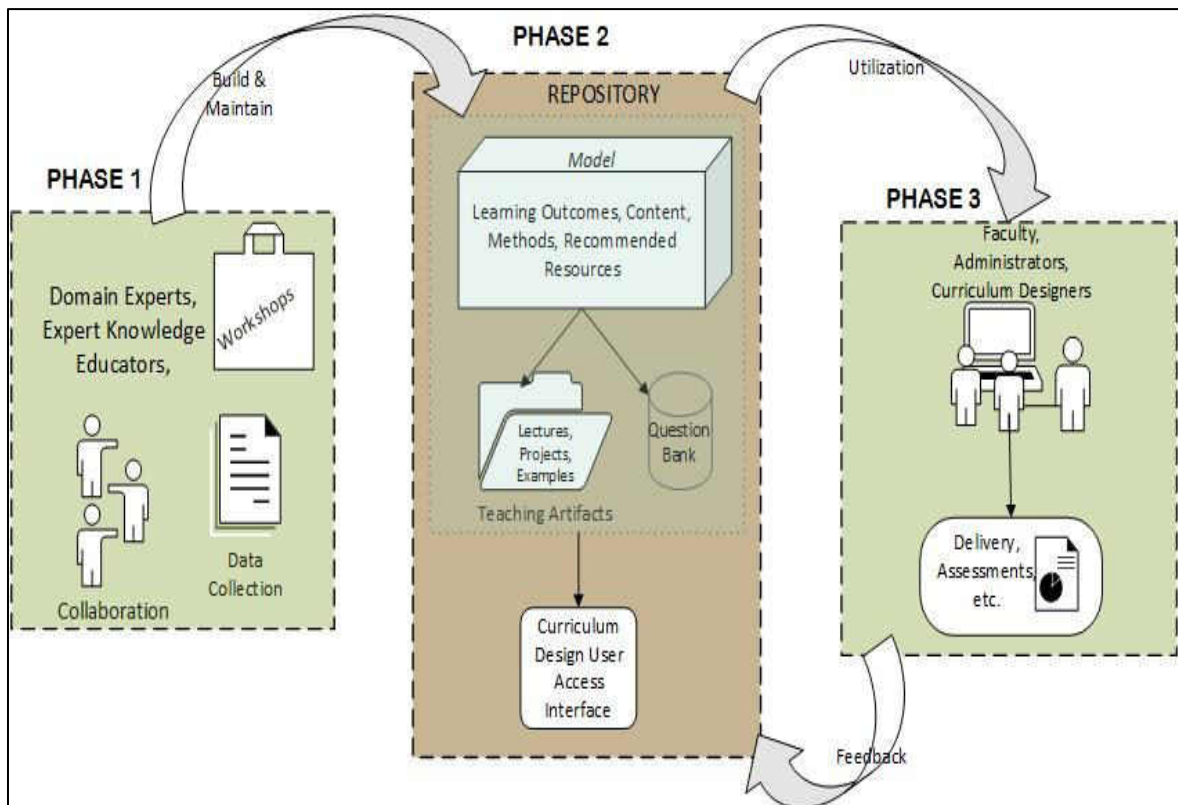
Figure 15 – Overview of the three phases of the Methodology

---

1 http://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html

3.2 The Repository Development

Phase 1 of this methodology involves the description of a model that captures the building and maintenance of the repository and feeds into the Repository contained within Phase 2 as Figure 15 depicts. The major aspect of phase one is the collaboration of various domain experts and educators in the given field. Such collaboration is crucial to the successful implementation of such a framework and similar approaches as outlined in Section 2.9.2 (the IEEE/ACM 2004 Development Process) have proven to be very useful. This collaboration has already started as outlined in [27] with the workshops at HAU in the Philippines and the ongoing widening of that pool of contributors since then. This collaborative work will be looked at in greater detail in this section where the output of those workshops will be presented and detailing how those outputs can be used as the data input source for Phase 2. The fact that the workshops are still ongoing, this will be a limitation as there will not be a complete set of output that is accessible for this thesis. Therefore, the available data will still be presented in this chapter, but for the purposes of the Case Study in chapter 4, the IEEE/ACM SE 2004 document in its original form will have to be used in its place.

The main source of input will be the IEEE/ACM SE document. The end result of the workshops will produce something similar to this document (a set of topics considered the best and most relevant to the most beneficiaries). So in the absence of that final product, this work will go off the premise that the IEEE/ACM SE document is the actual document.

3.3 Workshop Description

The workshops which have been conducted in south-east Asia [25] as well as future planned workshops are an important component to the proposed paradigm. This is because they facilitate a great degree of collaboration which allows for a community of educators in a given domain to have meaningful discussions and contributions surrounding this future paradigm which has great potential and promise.

The first data collection workshop consisted of research data capture activities and was the first in a series of data collection workshops. It took place at the Holy Angel University (HAU) in the Philippines in August of 2011 and had thirty participants comprising of SE educators and a few recent graduates of the HAU IT program. Recent graduates were a part of the workshop because they facilitated the assessment of the appropriateness of each topic from the students' point of view. The participants were divided into four panel groups for breakout sessions which were followed by a presentation and wrap-up plenary sessions.

3.3.1 Workshop Preparation

Before the workshop took place, there was pre-workshop planning which was carried out by a small cohort of the research project team. The purpose of this planning was to identify suitable material to be used by the participants, to design instruments for the collection of data, assigning duties to participating researchers, finalize administrative functions and so on. Perhaps, the most challenging aspect of the planning involved the selection of material to be used in the workshop which would

serve as a guide in identifying an appropriate set of topics for teaching SE across the four years of an undergraduate program. The final consensus was that the IEEE-CS/ACM SE 2004 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering would be the main data input source for the workshop.

3.3.2   Workshop Activities

The main workshop activities were the breakout panel sessions and the wrap-up plenary session. For the breakout panel sessions, the participants conducted discussions in the following SEEK areas:

- Computing Essentials,

- Software Modeling & Analysis Design,

- Software Verifications & Validation, and Evolution,

- Professional Practice and Software Management.

The SEEK areas of Mathematical & Engineering Fundamentals, Software Process, and Software Quality were not included in the panels, as it was deemed to be excessive work on the groups given the limited duration of the sessions. These areas have therefore been set aside to be covered in future workshops where a larger number of participants are expected.

Each panel had a moderator who was tasked with efficiently guiding the discussions, recording the decisions and votes taken by the panel as well as seeking clarification for

unresolved issues. Each panel group discussed one of the SEEK areas and had the responsibility to answer the following questions:

- Is this topic relevant to teaching the fundamentals of SE?

- In which year(s) of the program should it be taught?

- How many hours should be taught in each year?

- What is the rational for the selection of this topic?

In keeping with the collaborative theme even within the workshop itself, there was a wrap-up plenary session in which the moderators presented reports on their group's activities so that each participant was privy to the discussions that took place in the other groups, along with the decisions that were made.

3.3.3   Workshop Input and Output

3.3.3.1 Sample Seek Tables

As was mentioned earlier, the single source of input for the breakout panel sessions was the IEEE-CS/ACM SE 2004 document. A subset of this document relevant to the discussions to be had was made available to the participants. This portion of the document comprised introductory information, a description of the SEEK and tables of the SEEK areas. Table 5 shows a summary of the ten knowledge areas that make up the SEEK. The first column in the table lists the Knowledge Area/Knowledge Unit (**KA/KU**), which is coded for easy reference in the documentation. The **Title** column lists the topic areas under each unit and the third column (**hrs**) specifies the recommended contact hours or lecture hours for the topic.

Table 5 – SEEK Knowledge Areas and Knowledge Units [26]

KA/KU	Title	hrs	KA/KU	Title	hrs
CMP	Computing Essentials	172	VAV	Software V & V	42
CMP.cf	Computer Science foundations	140	VAV.fnd	V&V terminology and foundations	5
CMP.ct	Construction technologies	20	VAV.rev	Reviews	6
CMP.tl	Construction tools	4	VAV.tst	Testing	21
CMP.fm	Formal construction methods	8	VAV.hct	Human computer UI testing and evaluation	6
			VAV.par	Problem analysis and reporting	4
FND	Mathematical & Engineering Fundamentals	89	EVL	Software Evolution	10
FND.mf	Mathematical foundations	56	EVO.pro	Evolution processes	6
FND.ef	Engineering foundations for software	23	EVO.ac	Evolution activities	4
FND.ec	Engineering economics for software	10			
PRF	Professional Practice	35	PRO	Software Process	13
PRF.psy	Group dynamics / psychology	5	PRO.con	Process concepts	3
PRF.com	Communications skills (specific to SE)	10	PRO.imp	Process implementation	10
PRF.pr	Professionalism	20			
MAA	Software Modeling & Analysis	53	QUA	Software Quality	16
MAA.md	Modeling foundations	19	QUA.cc	Software quality concepts and culture	2
MAA.tm	Types of models	12	QUA.std	Software quality standards	2
MAA.af	Analysis fundamentals	6	QUA.pro	Software quality processes	4
MAA.rfd	Requirements fundamentals	3	QUA.pca	Process assurance	4
MAA.er	Eliciting requirements	4	QUA.pda	Product assurance	4
MAA.rsd	Requirements specification & documentation	6			
MAA.rv	Requirements validation	3			
DES	Software Design	45	MGT	Software Management	19
DES.con	Design concepts	3	MGT.con	Management concepts	2
DES.str	Design strategies	6	MGT.pp	Project planning	6
DES.ar	Architectural design	9	MGT.per	Project personnel and organization	2
DES.hci	Human computer interface design	12	MGT.ctl	Project control	4
DES.dd	Detailed design	12	MGT.cm	Software configuration management	5
DES.ste	Design support tools and evaluation	3			

Table 6 presents the details of the Software Design Knowledge Area with the units of Design Concepts, Design Strategies and Architectural Design. For each of these three units, the topics that comprise that unit are listed. Column 1 is the coded identifier of the topic. Column 2 gives the topic title. Column 3 gives the Bloom taxonomy level (indicating what capability a graduate should possess; 'k' for knowledge, 'c' for

71

comprehension and 'a' for application). Column 4 gives the topics relevance (indicating whether the topic is essential, desirable, or optional to the core). Column 5 lists the recommended lecture hours, and column 6 denotes the related topics.

Table 6 – Sample SEEK Knowledge Area: Software Design Detail [26]

Reference		k,c,a		E,D,O	Hours	Related Topics
DES	**Software Design**				45	
DES.con	*Design concepts*				3	
DES.con.1	Definition of design	c		E		
DES.con.2	Fundamental design issues (e.g. persistent data, storage management, exceptions, etc.)	c		E		CMP.ct.6,VAV.tst.2,CMP.cf.11
DES.con.3	Context of design within multiple software development life cycles	k		E		
DES.con.4	Design principles (information hiding, cohesion and coupling)	a		E		
DES.con.5	Interactions between design and requirements	c		E		DES.ar.4
DES.con.6	Design for quality attributes (e.g. reliability, usability, maintainability, performance, testability, security, fault tolerance, etc.)	k		E		FND.ef.4,MAA.tm.4,DES.ar.2,CMP.ct.14,VAV.fnd.4
DES.con.7	Design trade-offs	k		E		FND.ec.3,DES.ar.2,DES.ev
DES.con.8	Architectural styles, patterns, reuse	c		E		DES.ar,DES.dd.2,CMP.ct.3
DES.str	*Design strategies*				6	
DES.str.1	Function-oriented design	a	c	E		
DES.str.2	Object-oriented design	c	a	E		CMP.cf.9,DES.dd.5,CMP.ct.4
DES.str.3	Data-structure centered design			D		
DES.str.4	Aspect oriented design			O		
DES.ar	*Architectural design*				9	
DES.ar.1	Architectural styles (e.g. pipe-and-filter, layered, transaction-centered, peer-to-peer, publish-subscribe, event-based, client-server, etc.)	a		E		DES.con.8
DES.ar.2	Architectural trade-offs between various attributes	a		E		FND.ec.3
DES.ar.3	Hardware issues in software architecture	k		E		CMP.ct.13
DES.ar.4	Requirements traceability in architecture	k		E		MAA.af.5,DES.con.5,EVO.pro.2
DES.ar.5	Domain-specific architectures and product-lines	k		E		
DES.ar.6	Architectural notations (e.g. architectural structure viewpoints & representations, component diagrams, etc.)	c		E		MAA.tm

3.3.3.2 Sample Workshop Output Tables

In table 7, a sample of one of the moderator's panel discussion report is given. In section 2.8.1, Table 4 is an example of one of the Workshop Output tables showing the results of the deliberations of one of the panel groups on Professional Practice. Table 7 here more or less conveys a similar message, but in a more 'raw' form from a moderator's perspective. It captures some of the thinking that went into the final decisions made, and so for some of the rationale given in column 4, one can see an explanation along with the changes that were made.

Table 7 – Sample Panel Discussion Moderator Report [25]

Topic	Year(s)	Depth	Rationale
Software Verification and Validation		**42 hours**	
V&V terminology and foundations	3	**5 hours (1/8)**	**ALL voted YES**
Objectives and constraints of V&V	3	**E**	**ALL voted YES**
Planning the V&V effort	3	**E**	**ALL voted YES; This happens during software development. Testing is done by module and components.**
Documenting V&V strategy, including tests and other artifacts	3	**E**	**ALL voted YES; Part of project management. Everything should be documented.**
Metrics & Measurement (e.g. reliability, usability, performance,etc.)	3	**E**	**ALL voted YES; 5 hours is enough for the introduction. But for an application, it should be discussed longer.**
V&V involvement at different points in the lifecycle	3	**E**	**ALL voted YES; This should be discussed before Planning.**
Human computer user interface testing and evaluation			
The variety of aspects of usefulness and usability	3	**E**	**User-friendliness, effectiveness.**
Heuristic evaluation	3	**E**	**Investigative, statistics (mean time before failure), controlled enrolment, continous process of evaluation.**
Cognitive walkthroughs	3	**E**	**Training the user/client how to use the software.**
User testing approaches (observation sessions etc.)	3	**E**	**Obeserving the user/client during the first system usage.**
Web usability; testing techniques for web sites	3	**D**	**Not all systems are web-based. (from Essential to Desired)**
Formal experiments to test hypotheses about specific HCI controls	3	**D**	

Table 7 cont. – Sample Panel Discussion Moderator Report [25]

Topic	Year(s)	Depth	Rationale
Problem analysis and reporting			
Analyzing failure reports	*3*	**D**	**Not for under-graduate students because they lack experience.** **(from Essential to Desired)**
Fault isolation techniques & Debugging	*3*	**E**	**Debugging/Tracing tools; good for under-graduate students.** (From Debugging/Fault isolation techniques & to Fault isolation techniques & Debugging)
Defect analysis	*3*	**E**	
Problem tracking	*3*	**k E**	**What's the cause of error? Use break points available from compilers. (From comprehension to knowledge)**
Software Evolution			
Evolution processes			
Basic concepts of evolution and maintenance		**E**	
Relationship between evolving entities (e.g. assumptions, requirements, architecture, design, code, etc.)		**a E**	**Implications when you changed PL. (Egs. .Net platform)** **(From knowledge to application)**
Models of software evolution (e.g. theories, laws, etc.)		**C E**	**What is the appropriate approach when revisions are needed?** **(From Essential to comprehension)**
Cost models of evolution	*3*	**a E**	**New software/maintenance package depending on the needs of the client. Migration.**
Planning for evolution (e.g. outsourcing, in-house, etc.)	*3*	**E**	**New software to be developed in-house or outsource. (From Desired to Essential)**
Evolution activities			
Working with legacy systems (e.g. use of wrappers, etc.)		**E**	**Existing software system that are still functional. Compatibility.**
System and process re-engineering (technical and business)		**E**	**To re-train developers to adopt to new programming technologies.**
Impact analysis		**E**	**Experienced analyst is needed.**
Data reverse engineering	*3*	**a E**	**Custom-fit the existing data to another/new software to reduce cost and time. (From Desired to application/Essential)**

Legend

Topic – Topic code as taken from the IEEE SE 2004 document. Example, **MAA.md.4** *Properties of modeling languages*.

Year(s) – The year that the group decides the topic should be taught; there may be multiple years. Example, **1, 2** *For the topic being taught in years 1 and 2*.

Depth – The number of hours the topic should be taught, using the guidelines from the IEEE SE 2004 document. Example, **5, 10** *For hours of lecture in two years.*

Rationale – The reason for selecting this topic, and the vote if it was not unanimous. Example, **This topic is included because it is viewed as fundamental to SD**.

3.3.4   Bloom's Attributes & Topic Relevance

Blooms Taxonomy divides educational objectives into three main domains: Cognitive, Affective, and Psychomotor. Skills such as knowledge, comprehension and critical thinking are usually associated with the cognitive domain. Within this domain, there are six levels in the taxonomy: knowledge, comprehension, application, analysis, synthesis and evaluation (see figure 16). Topics in the IEEE-CS/ACM SE document as shown in column 3 of sample table 6 are labeled with one of three levels from the Cognitive domain; namely, knowledge, comprehension or application. It was suggested in [26] that only these three levels of learning were chosen from Bloom's taxonomy because they represent what knowledge may be reasonably learned during an undergraduate education.

The three levels of Bloom's cognitive domain which are used in [26] are summarized as follows:

- Knowledge (k) – Remembering previously learned material.

- Comprehension (c) – Understanding information and the meaning of material presented.

- Application (a) – Ability to use learned material in new and concrete situations.
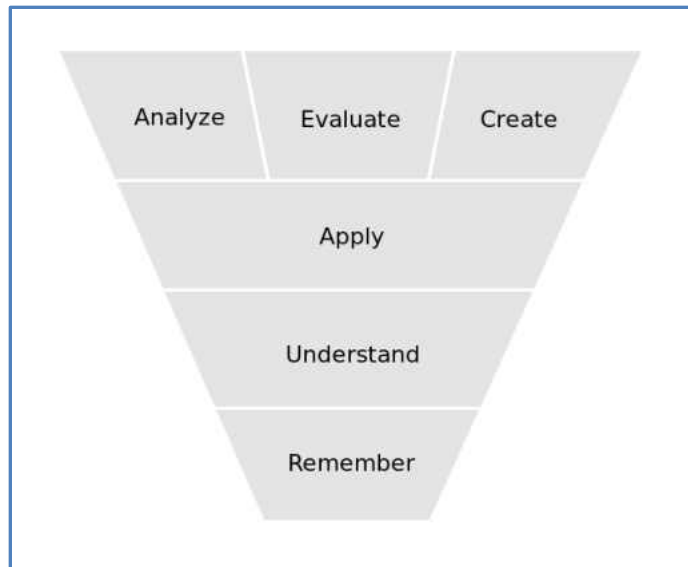
Figure 16 – Levels in the cognitive domain of Bloom's Taxonomy [31]

At this juncture, it would be apt to also mention the topic relevance assigned to each topic as shown in column 4 of sample table 6. Therefore, a topic's relevance as it relates to the core is summarized as follows:

- Essential (E) – The topic is part of the core.

- Desirable (D) – The topic is not part of the core, but it should be included in the core of a particular program if possible; otherwise, it should be considered as part of elective materials.

- Optional (O) – The topic should be considered as elective only.

3.3.5 Discussion of the output and how they were arrived at

The panel groups went through the detailed SEEK tables similar to the one shown in table 6 and deliberated over each topic. The results of these deliberations saw

adjustments being made to various entries in the table, but mainly to columns 3, 4, 5 and 6. The tables were also open to additions and enhancements, and so for instance, one of the panels added a column to the original table to identify the year(s) in which a given topic should be taught.

Table 4 was produced by one of the panels as a result of their deliberations on the SEEK areas of **Professional Practice** and **Software Management**. In comparison to the original SEEK tables, a few changes are apparent. They changed the **hrs** column to **Depth** and assigned percentage of teaching time instead of numeric hour values. The rationale for this decision was that different institutions ascribe different quantities of lecture hours to each topic. A **Rationale** column was also added which provides reasoning behind the decisions taken for each topic

One of the main goals of the panel's deliberations was to identify a most appropriate set of topics (based on the group's experience) using the IEE-CS/ACM SE 2004 SEEK areas format. This task could be super-imposed by the phrase "defining the perfect SE curriculum." The panel's presentations at the closing plenary session of the workshop highlighted a few significant points. One key point was that even though each panel had separate and private deliberations, they were all conducted in a similar manner, ensuring uniformity and some level of consistency. All the participants expressed an interest in being involved in future related research activities, and some of these participants even expressed a desire to incorporate some of the ideas and topics introduced at the workshop in their own Software Engineering courses and programs.

This is indeed very encouraging early results and reinforces the need and practicality of the proposed paradigm.

3.3.6   *The Importance of the Workshops*

It is important to discuss the workshop because it shows the approach that should be taken to develop curriculum. It shows the beginnings (foundation) of a platform that other domains can follow. It lays the groundwork required for the building and future maintenance of the repository which lies at the heart of this new paradigm. At the one end, (the beginning end) you have the collaboration and data collection (similar to even what the IEEE/ACM have done) that lays the groundwork for curricula that is not only region specific but that can also be institution specific (context-aware). On the other end (the consumption or interface end) you have the utilization of the repository. But the workshops and workshop type efforts is where it really begins.

3.4   A Model for the Repository Initial Development and Future Maintenance

In order for the repository to be maintained and continuously developed, a structure must be put in place to ensure this. One approach being explored is one that is similar to the Wikipedia[2] model. Wikipedia is a web based encyclopedia made up of a large number of interconnected web pages. The idea behind this is that it is a community of users who develop and maintain these pages; adding, updating and editing the content. Wikipedia's content is written by volunteers - writers are not paid

---

2 http://en.wikipedia.org/wiki/Wikipedia:About

to contribute the entries. According to the site, "anyone with Internet access can write and make changes to Wikipedia articles (except in certain cases where editing is restricted to prevent disruption or vandalism)" [29]. This open concept has both its pros and cons. Perhaps, the positives of this Wikipedia model can be enhanced and adapted to be used in this new paradigm where experts across the world can contribute to the repository and participate in its continued maintenance and development.

3.4.1   Nupedia

A great deal can be learned from the history of Wikipedia in terms of the cogent aspects of its platform that this work may capitalize on. Wikipedia was founded as an offshoot of Nupedia, a project to produce a free, open source, collaborative online encyclopedia which lasted from March 2000 until September 2003. Nupedia had an elaborate peer review system, encompassing an arduous seven-step approval process to control content of articles and which required highly qualified contributors. The fact that Nupedia was characterized by an extensive peer-review process meant that its intrinsic design was to make its articles of a quality comparable to that of professional encyclopedias. The keyword in the previous sentence is 'quality', and as it was with Nupedia, quality is an attribute that will be strived for in the proposed paradigm. There is however a very fine line to be trod between ensuring quality and stifling growth, because as noted in [32], the strict control and fastidiousness in Nupedia limited the posting of articles and may have led to its ultimate demise.
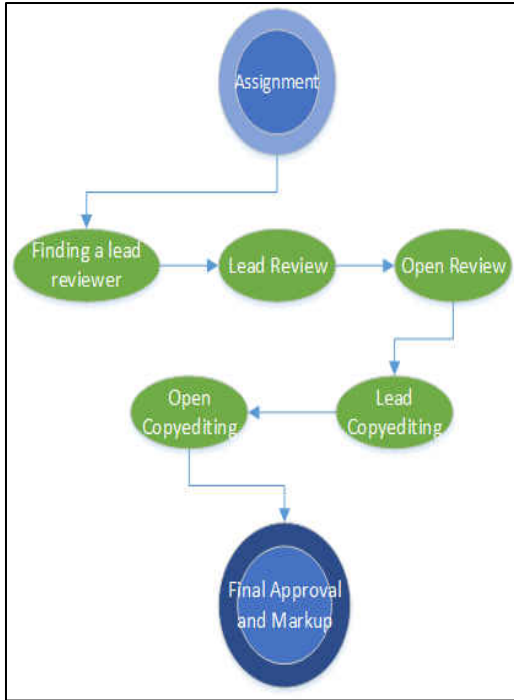
Figure 17 – Nupedia Editorial Process   Figure 18 – Wikipedia Publishing Flow
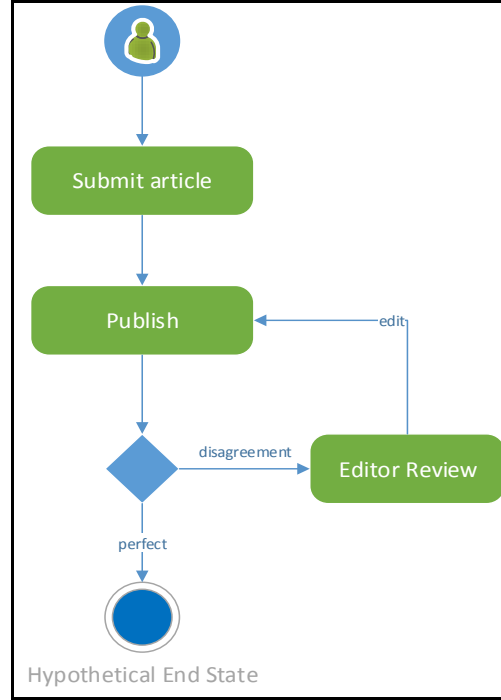
A fundamental difference between Nupedia and the current Wikipedia is that articles do not have to be reviewed before being posted on Wikipedia. No implicit or explicit expectation is placed on Wikipedia authors to be of some repute, that is, be some sort of expert in whatever they are writing about. With Nupedia, however, authors were explicitly expected to be experts in their fields, and editors were expected to be bona fide experts, possibly possessing PhDs. For good measure, figure 17 above shows the seven-step editorial process, while figure 18 shows the simplicity in the Wikipedia publishing flow with reactive editorial review.

3.4.2 *Wikipedia*

Wikipedia overcame the flaws of Nupedia to quickly become one of the world's most visited websites. Most people visit the site to acquire knowledge, but there are also many who visit to build this knowledge base – the contributors. There are mechanisms in place to help contributors create high quality articles. For example, if there are disagreements on how certain facts are presented, editors work together to arrive at an article that as close as possible represents current expert opinion on the given subject. For good reason, published articles are never considered complete as they may be edited at any time and are subject to constant debate, discussion and ultimately revision. Often times the quality of a new article will be very low with incomplete and maybe erroneous information, but overtime, such quirks are 'ironed out' and the article becomes more mature with less bias and greater quality through the consensus of the contributors (or users of Wikipedia). At the writing of this work, there were over 77,000 editors from scholars to average readers who regularly edit Wikipedia, helping to create a consistent style throughout its content.

The open nature of Wikipedia means that not much can be done to prevent the publishing of inaccurate, incomplete or biased information in the first place. Therefore, much of Wikipedia's quality control can be viewed as reactive stemming from the constant editing that all of Wikipedia's content is subject to. In the open web, this is an

accepted limitation, provided that consumers of this content know what they are getting (in terms of Wikipedia's disclaimers[3]).

3.4.2.1 Wikipedia Strengths and Weaknesses

- Open to a large contributor base – the large number of editors from diverse backgrounds significantly reduces cultural and regional biases.

- Openness encourages the inclusion of a tremendous amount of content – it can be argued that this is a case of quantity trumping quality.

- Is very current – articles covering newsworthy events can be published within a very short time of their occurrence.

- Anyone can edit – which means that vandalism and unchecked information are a constant threat.

- Contributors with expert credentials have the same weight as amateurs.

- No formal peer review – which means that the authenticity of scientific, medical or engineering articles will always be called into question.

A significant problem that Wikipedia has run into in recent times is that there has been a steep decline in administrator applicants and acceptances. Administrators play an important role and they have various powers such as blocking a user or IP address, dealing with abusive editors, deleting a page and blacklisting spam. Having too few administrators will result in the quality of the Wikipedia platform degrading over time.

---

3 http://en.wikipedia.org/wiki/Wikipedia:Disclaimers

The problem is that less people are interested in this role, less people are being accepted into this role and this is all voluntary work. This is an issue that the proposed paradigm must seek to avoid granting incentives to repository moderators and administrators to be active participants. Such incentives are not necessarily financial since participants would most likely have a passion for such roles and are not doing it for monetary gains.

3.4.3 Tightly Guarded vs. Unrestricted Publishing

It may also be useful to briefly examine two contrasting platforms to further extrapolate meaningful patterns that the proposed paradigm can use. The Google Play Store and Apple's App Store (iTunes) provide such a contrast. Google's Play Store is the marketplace for Android device applications, and with Android being an open source operating system is it not surprising that the model Google has employed for publishing apps is open distribution which is unrestricted. To get an app published in the Play Store, one only needs to have a Google account and as long as all the basic requirements are met such as having the correct image sizes and a valid Android Application Package file (APK)[4], the app once submitted is available in the store in a matter of minutes. In the case of Apple on the other hand, iOS app submissions are subject to approval by Apple and can take weeks for the approval process to be completed.

There are obvious pros and cons to the approaches used by either company. For Google the main disadvantage is app quality and the possibility of malicious apps that

---

4 http://developer.android.com/distribute/open.html

may contain malware or other nefarious code. The openness has the benefits that developers do not have to feel restricted and confined to the ideals and interests of Google. For Apple, the reliability testing and analysis that is done in there strict approval process ensure the approval of only high quality apps which enhance user experience and does not degrade the platform. The downside for Apple's approach is that developers' creativity may be stifled somewhat and they are left at the 'mercy' of Apple. There have been highly documented cases where Apple took steps to deliberately delay approval for apps that either competed against Apple or that were not in the best interest of Apple's 'bottom-line'. Apple is often seen to be hypocritical and seen to be an authority with "censorship of convenience."

3.4.4   Repository Maintenance Model

The strengths of the publishing quality of Nupedia can be coupled with the openness of the Wikipedia model to create a more streamlined model suited to the proposed repository. Looking back at the app stores comparison, a fundamental difference is that Google's Play Store openness leads to significantly more security and quality issues which can only be remedied after the fact (reactive), while Apple takes a more proactive approach ensuring that bad apps never make it into the marketplace. The proposed framework would take advantage of Google's open approach by welcoming collaboration and contribution from a wide community, and take advantage of Apple's closed ecosystem by ensuring that all contribution are properly and efficiently vetted (reviewed) before they have any impact on the repository.
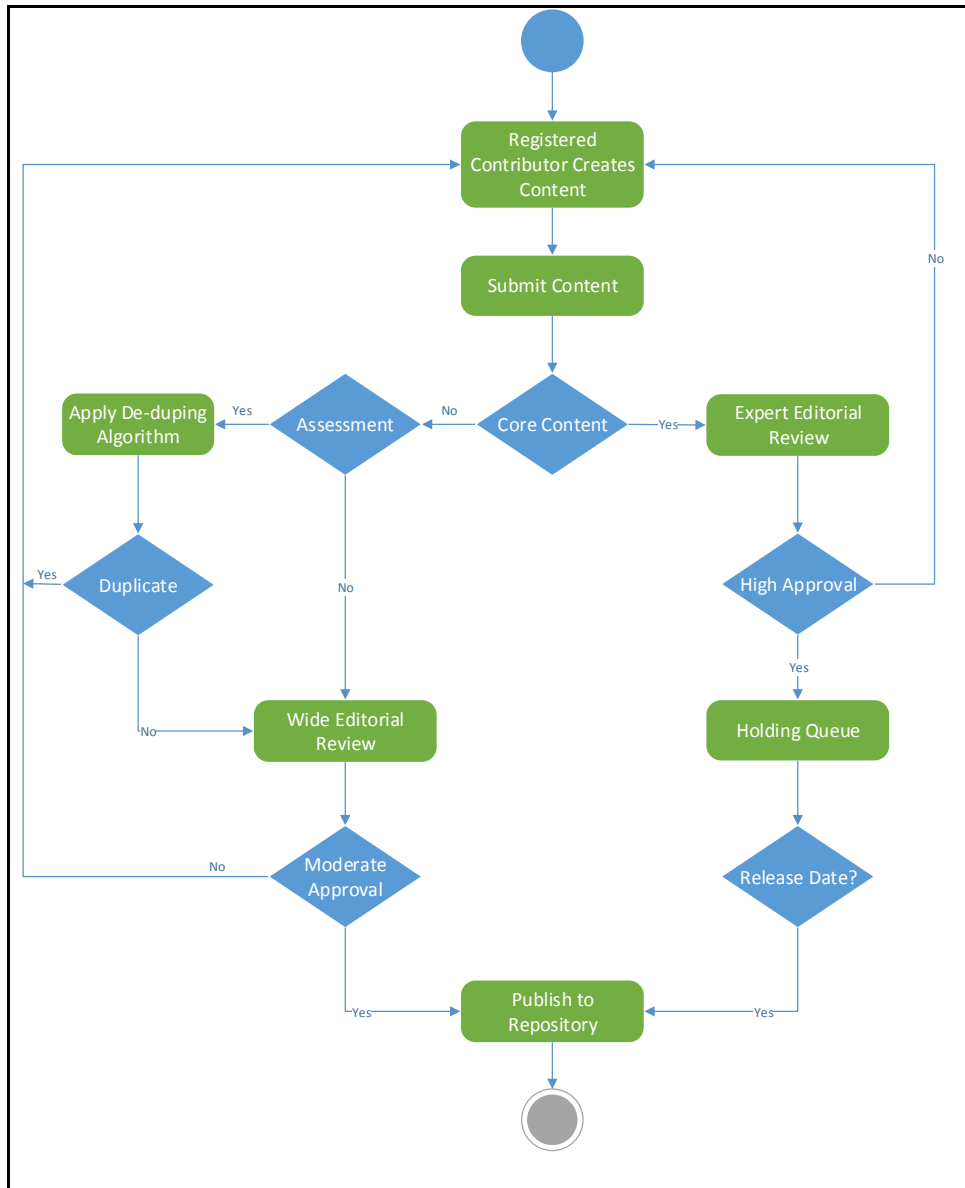
Figure 19 – Proposed Model for Repository maintenance

There will be a need to have stricter control over content that is core to the curricula such as courses and topics. On the other hand, content that has less precedence (supporting content) such as assessment artifacts (questions for the Question bank) will not need to be as tightly guarded and so lower levels of restriction would suffice. Figure 19 is a proposed model for maintenance of the Repository through active collaboration and contribution.

3.4.4.1 Description of Maintenance Model

Only registered contributors can create content for the Repository. It is expected that these contributors will be Software Engineering faculty with PhDs and Masters level degrees, with possible consideration for experienced faculty who may not possess advanced degrees. Contributors can submit content that has one of four objectives: Creating, Replacing, Editing and Deleting (CRED). Content is considered to be either supporting content or core content.

Expert Editorial Review: Core content must be evaluated by experts (that is, contributors with PhDs only). This step is similar to the review process of Nupedia or Apple's app store, but far less cumbersome. This review step goes off the simple premise that there is strength in numbers; where the greater the number of experts that agree on something, the greater its quality and value. A set threshold of expert approval must be surpassed in order for a submission to be accepted. This is much like a Peer Review where all the editors have equal weighting and if for instance, there is high consensus among the reviewers, then the submission is accepted. If this threshold is not met, the submission fails and is sent back to the submitter. If there is a valid way to scientifically quantify what a "high approval" or "high consensus" is, then that will definitely be a more rigid and robust way to determine these thresholds. Just to point out how streamlined this review step is, a submission that needs to be edited for whatever reason can be denied by an editor and if enough editors deny this submission, it will not meet the threshold and will eventually end up back with the submitter to make the corrections. So the theme of collaboration and general consensus is what makes this approach practical and quality-controlled. Compared with the Wikipedia model, this approach ensures that only quality content gets published since the review is done before the fact and not after.

One possible limitation of this approach is that it may not scale well. This is because too many core content submissions may result in backlog and longer approval times. A possible fix for this limitation would be to require a fixed number of expert editors for threshold approval, rather than requiring a fixed percentage or a hypothetical "high approval". But this can only be done if there are enough experts available, such as setting an acceptance threshold of 20 expert approvals if there are 50 total experts instead of a fixed percentage like 80% which would require approval of 40 out of the 50 experts.

Wide Editorial Review: Supporting Content is content which has lower precedence than Core content and as such can be reviewed by any registered contributor. The acceptance threshold for acceptance can therefore be more relaxed and flexible. A possible acceptance threshold could be to require moderate approval rates (such as hypothetically having at least 25% of all registered contributors approving it - this would be a strict 25% approval and does not mean 75% denial). Fast approvals would be suited as it would lead to rapid growth of the repository from a supporting content perspective. Accepted submissions are immediately published to the Repository.

Holding Queue: Accepted Core content submissions are placed in a holding bucket called the Holding Queue. This is to ensure that changes to the Core are not made too frequently, thus ensuring stability. Content in this queue are published at fixed times throughout the year such as every 6 months or at the end of every semester. This is akin to already released software that has a fixed release cycle to provide updates or patches.

3.5 The Repository – Modeling of a Curriculum

As was diagramed in figure 15, the modeling of a curriculum comprises Phase 2 of this methodology. Modeling a curriculum is a very pivotal part of this work and as such every attempt will be made to make it as concise and as lucid as possible, beginning with a very abstract high level layout and going down to very low level detailed structures. A high level schema will be presented with a dissection of the relationships (logical connections) that exist between blocks in this schema. Proposed content tagging and tracking will be presented with a focus of establishing proper Metadata for all content. In order for a curriculum to be designed to meet required quality standards (such as ensuring there is proper Essential content coverage) with varying contextual needs (such as lack of certain resources), Threshold, Ranking and Recommendation algorithms and methodologies will be presented.

3.5.1   Top Level Schema

Recall figure 5 which was a diagram depicting a hierarchical abstract view of the framework. In section 2.2, this view was compared to that of traditional ways of designing curriculum and the similarities and superiority of the proposed framework was highlighted. In this section, a closer look will be taken into how this model fits into the overall framework.

The objectives component is one in which the learning objectives are defined for the four years of a Software Engineering undergraduate degree. Each year has its own objectives, but objectives can themselves span multiple years. For example, an objective

"Design software so that it can be changed easily" is one that could be stated for both an introductory Software Engineering course in Year 1 as well as a Software Quality course in Year 3. The Content component is one where all the core content that satisfies all the objectives is defined. This content includes the complete set of courses and topics across all four years of the degree. The content can be further explored by the fact that a course is comprised of a set of topics and topics can be comprised of a set of sub-topics. So there can be *n-ary* levels of topics and sub-topics. There is a direct relationship between the objectives and the content in that an objective can be satisfied by one or more courses, and conversely, a course can satisfy one or more objectives. Hence, the relationship between these two entities is many-to-many. The course is like a placeholder or an abstract class, in that it really does not exist or is of little use until the topics that comprise that course are defined. Therefore, as topics are added to a course, the objectives that are satisfied by that course are updated to match the objectives that the given topics of a course satisfy. This can be viewed as a tree where all the children (topics) determine what the parent (course) maps to. This is akin to a logical consequence (or implication) in which for instance if topic A satisfies objective 1, and topic A is a child of course B, then this implies that course B satisfies objective A.

In terms of coverage metric, one can quickly see that modeling this part of the curriculum in the way described above provides two equitable ways of measuring how much of a degree has been covered. One way is by looking at how many of the objectives have been satisfied and the other way is by looking at how many of the topics have been covered.

The Delivery component is made up of supporting content such as teaching artifacts (lectures, notes, case studies, examples and soon), as well as resources required for the delivery of content (both core and supporting). Delivery has a direct mapping to topics and the relationship is that a given topic can be satisfied by one or more supporting content. Practically speaking, a topic can be delivered by lectures, case studies and examples.

A closer look at figure 5 will reveal a direct link between the Content component and the Question Bank. The Question Bank is a special type of supporting content, where for a given topic, there can be zero or more questions directly mapped to it. The Metadata attributed to each topic will ensure that questions can be properly tagged and credited to the proper topic. Every question submitted to the Question Bank must be tagged with metadata. The more information tagged or the lower in the tree that the tagging is linked to, the more useful the question becomes. For example, tagging a question with only the objective (linking it only to an objective) means that an auto-generated assessment for a specific topic will not be able to include such a question because the question's context is too ambiguous (not specific and directed enough). In such a case, only an objective-wide assessment could make use of the question. The lower in the tree the mapping is, the more accurate the system is. That is, the more ancestors a question mapping has, the more information we can derive from it.

3.5.1.1 The Component Mappings (Relationships)

Figure 20 visually describes the mappings or relationships between the components outlined in the top level schema. The beauty of this diagram and by

extension, describing these logical connections, is that one can easily extrapolate the implications that arise from within the hierarchy. What does this mean exactly? To explain this, take for example a given topic X. Topic X can be taught in both years 1 and 2. By simply looking at the diagram and focusing only on Topic X in isolation, one will not be able to readily know that Topic X can be taught in both years 1 and 2. But by examining the relationship of Topic X's parent (Course A for instance) and grandparent (Objective 1 for instance), basically following the tree from the given node (Topic X) all the way to the root, numerous metadata and implications can be revealed about the node in question. So essentially, because Topic X is a child of Course A which is a child of objective 1 which in turn is a child of Knowledge Area 1 which is a child of both Years 1 and 2, this all implies that topic X can be taught in both Years 1 and 2 (logical consequence; if A=B and A=C, then B=C).

This sort of implicit information can be embedded and extracted by simply mapping the various components in the way described above. Basic information about a piece of content in the structure can be embedded (tagged) along with the content itself, but the power of this model, however, lies in the logical connectedness of all the components. The model is really a tree, and of such all content is connected to some other content whether implicitly or directly.
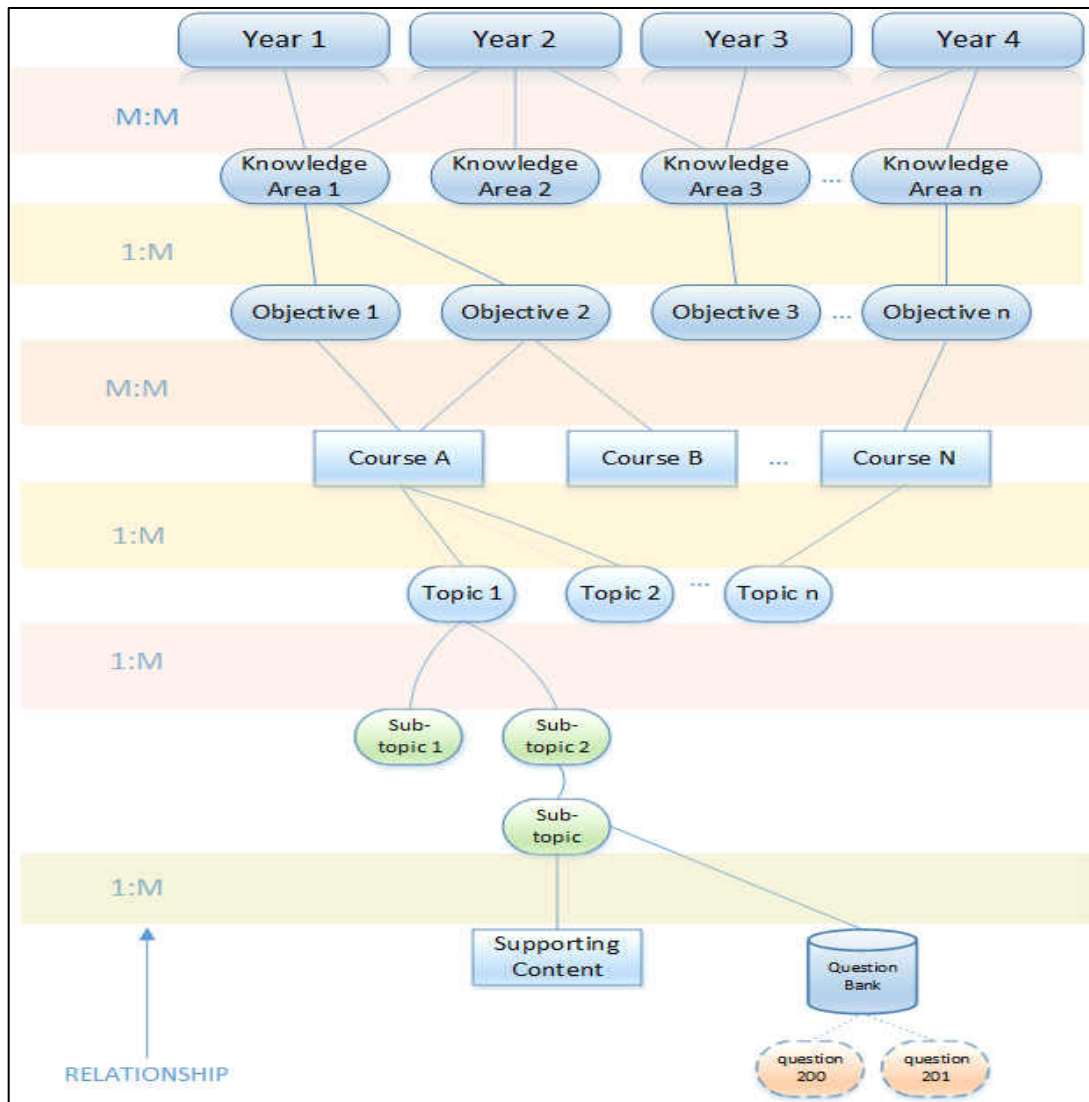
Figure 20 – Diagrammatic representation of the logical connections (relationships) between components.

3.5.1.2 Modeling Using a Tree Data Structure

The hierarchical structure of the model above can be represented more precisely as a tree. A general tree is a nonlinear data structure in computing in which each node may have zero or more children. It turns out that it is more suitable to model the curriculum as an ordered tree because there is implicit ordering in the relationships both

horizontally (siblings) and vertically (parents-children). This ordering exits because for instance, it makes sense to teach a certain topic before another, or teaching all year 1 courses before year 2 and so on. Trees provide a natural organization for data and as such have become ubiquitous structures in file systems, databases and other computer systems. The relationships in a tree are hierarchical, with the generic parent-child (family tree) layout, a theme that fits perfectly in the model outlined in figures 5 and 20. There are properties that every tree subscribes to; however, there are other properties that are specific but not unique to the curriculum model. These properties are outlined below:

1. A general tree T is a finite set of one or more nodes with one designated root node r. This root node is the top element and is the only node in the tree that does not have a parent (ancestor).

    a. In the curriculum model, this could be the particular domain or field such as Computer Science or the program of study such as Software Engineering. It depends on how broad of a scope the implementers of the tree decide to define.

2. All descendants or r (all other nodes) are partitioned into n>=0 disjoint subsets $T_1, T_2, ..., T_n$, each of which is a tree, and whose roots $r_1, r_2, ... r_n$, respectively, are children of r.

3. Each node with the exception of the root node has a unique parent node w; every node that has the parent w is a child of w.

a. The exception for the curriculum model with this property is that a node other than the root node can have one or more unique parents. For example, an objective (child node) can be delivered in Years 1 and 2, and so Year 1(a parent) and year 2 (another parent) will share the same child.

4. Nodes that share the same parent are siblings, meaning that they are at the same level in the tree. Useful information can be gleaned from deciphering this property such as ranking the importance of a piece of content in comparison with other content.

5. A node v is external if v has no children (a leaf node). For a completed tree or completely modeled curriculum, a leaf node could be a topic or sub-topic, or some other atomic unit of data that cannot be reasonably decomposed any further such as a question in the Question Bank.

6. A node v is internal if it has one or more children. Knowledge of this property for a given node can be useful for algorithms that parse the model (such as knowing when traversal should stop or continue), and for ranking content.

7. The tree is an ordered tree if there is a linear ordering defined for the children of each node; that is, the children of a node can be identified as being first, second, third, and so on. In figure 20, this ordering can be seen with the Years (which are siblings) be ordered from left to right according to their ordering. This is an important property, because it enforces the ordering in which content is delivered.

a. As an example to reinforce this concept, take the components of a structured document such as a book which are organized hierarchically as a tree. The root of the tree is the book itself, chapters contain paragraphs and paragraphs contain sentences. In order to get the proper understanding of the book, this logical sequence must be followed. It would be ridiculous to read the last sentence of a paragraph and not reading form the first in order to get to the last.

One complexity added in modeling a curriculum as an ordered tree is the notion that a child can have several parents. Even though this adds complexity, it is important in that it adds flexibility and captures the essence that a curriculum should be dynamic and having multiple paths allows for this. Figure 21 depicts the tree structure with some of the properties which are outlined in the listed above annotated on the diagram. It should be pointed out that figure 21 is only an example (or a snippet) of how a curriculum can be structured as an ordered tree. So even though it would seem like a there is only one path to any given topic, let it be known that a topic can implicitly satisfy several objectives and hence there can be several paths to a topic. The word implicit is used here to denote the fact that in the hierarchy, topics are not linked directly to objectives but to courses, and it is the courses that are directly linked to objectives.
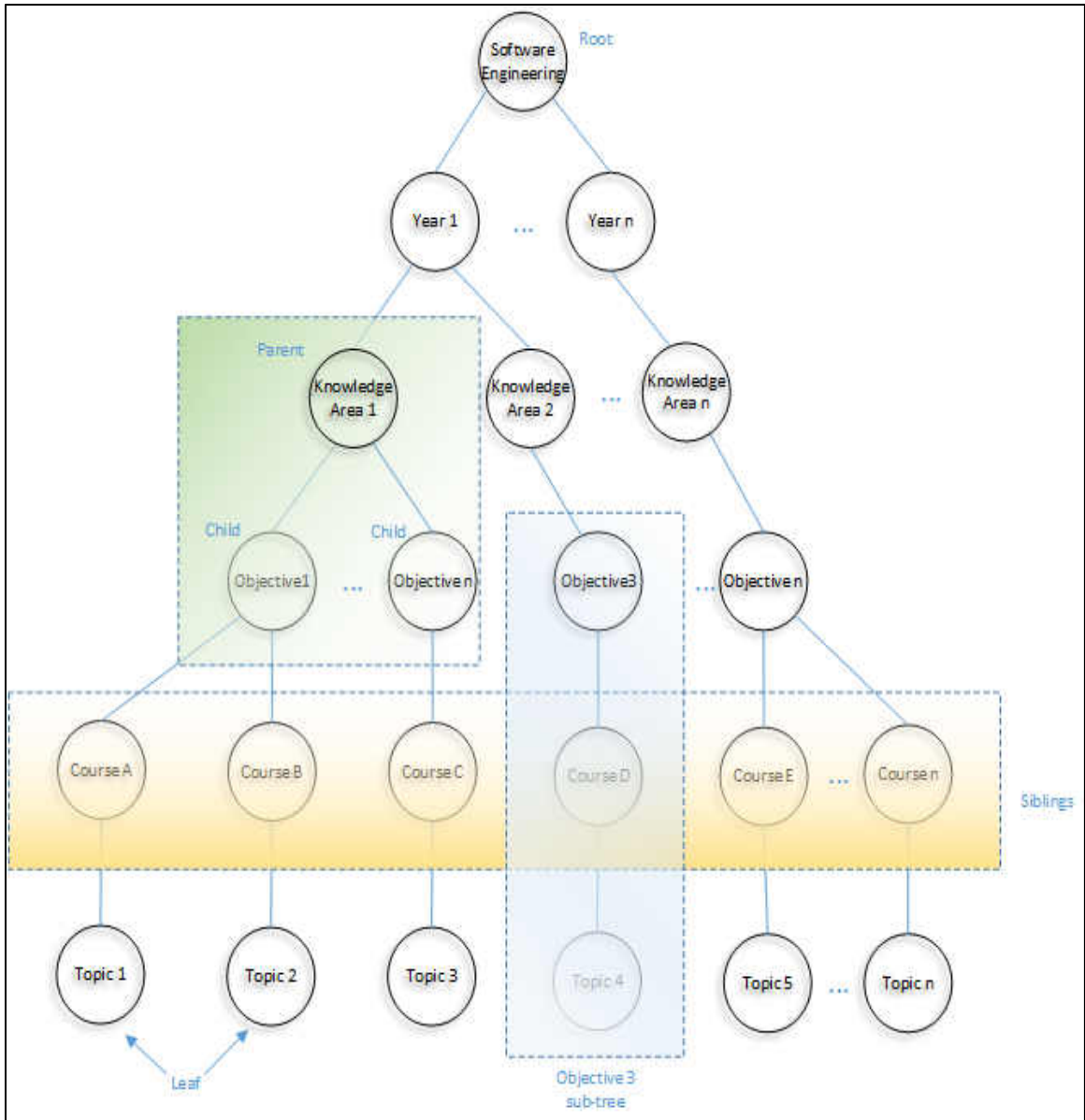
Figure 21 – Ordered Tree structure annotated with some of the properties of the Tree Data Structure

Imagine a scenario where two distinct topics satisfy the same objective. There are two ways in which this could be represented in the ordered tree. The first way depends on if both topics fall under the same course, in which case there is a straight-forward link between the one course and the objective in question. The second way which is less trivial occurs if each topic falls under separate courses, which suggests that

the two courses satisfy that same objective. So if one were to try and optimize topic

selections based on objective fulfillment, one can see how duplicate topics can be easily

found and treated appropriately. This can be a useful tool to aid faculty who may be

time or resource restricted in optimizing the curriculum that they deliver. In essence, an

objective can be met by delivering less topics for instance (a classic case of quality over

quantity) while still maintaining a high level of coverage from an Objective stand point.

3.5.2   The Repository Metadata

According to the National Information Standards Organization, Metadata is

structured information that describes, explains, locates, or otherwise makes it easier to

retrieve, use, or manage an information resource [33]. The term is often referred to as

data about data. There are three types of metadata – descriptive, structural and

administrative. For the purposes of this work, the descriptive metadata is the type most

relevant, as it describes a resource for purposes such as discovery and identification;

two activities which are integral to efficient operation of the proposed modeling

scheme. The idea behind including metadata as a part of the model is that every single

piece of content in the repository must be identifiable in isolation. That is, if one were to

access one unit of data, whatever it is, no matter how small or large, there must be a

mechanism in place to describe or to reveal all there is to know about that data, as well

as to be able to decipher the context of that data. The exact implementation of this

mechanism may be too low-level to describe at this point, but the general idea can be

expressed here.
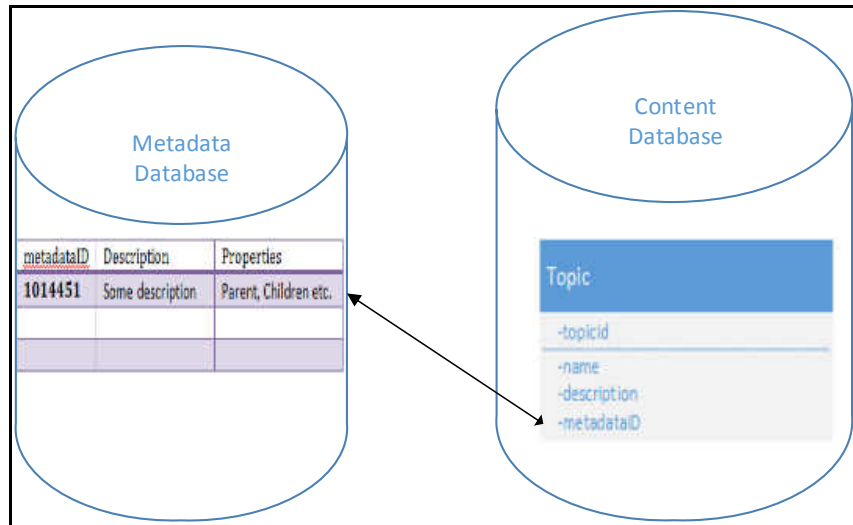
Figure 22 – Possible metadata solution where metadata is abstracted from the actual content

One solution would be to store the metadata with the object it describes, thus ensuring that the metadata is never lost and prevents any problems that may arise from linking them if they were separate entities. Another solution would be to store the metadata separately (see figure 22) which would simplify metadata management as well as decoupling the actual data and the metadata. This adds flexibility to the metadata itself making it more future proof such as if better methods of marking-up metadata arise, these can be easily applied without worrying about changing the data to fit the new methods. Searching and retrieval will still be efficient as linking an object to its metadata would simply require an index look-up.

This idea of metadata synchs nicely with the need to have sound structure to every piece of content that ends up in the repository. For this reason, it is expected that all input will have some form of a template that moderators come up with to ensure conformance with expectations. So for instance, multimedia such as audio lecture files

98

needing to be of the type mp3 and video lectures needing to be of a certain minimum quality and size. Table 8 provides a template for what a course as an entity in the repository should look like while table 9 represents the course metadata model.

Descriptive metadata will provide many benefits to the repository model. These include:

1. Resource Discovery

    a. Enabling content to be found quickly and easily based on provided criteria. For example, Topic X may have the following metadata:

    { *Date Added: 10/17/2012*

    Parent: Course A;

    Children: None;

    Topic Relevance: Essential;

    Bloom's Attribute: Comprehension;

    Required Resources: Computer, Internet}

    The system will be able to determine a wealth of information about this topic by simply inspecting the metadata. For instance, the Parent tag will indicate which parent the topic belongs to, so if the parent is a course, the path can be followed towards the root to determine which objectives Topic X satisfies and which year or years it should be taught in. Tables 8 and 9 shows course elements and course metadata model respectively.

    b. Identifying content and the context of said content.

    c. Distinguishing rank and relevance of content.

2. Organizing Electronic Resources.

   a. We are moving into an increasingly digital age where almost all physical content has a digital counterpart and more and more digital artifacts no longer have a physical counterpart. Books are a perfect example, where some authors no longer provide hard copies, opting instead to go strictly withe-books. So in the repository, descriptive metadata will help to organize the vast array of artifacts and help to maintain the logical relationships between them.

3. Archiving and Preservation

   a. Metadata is crucial to ensuring that resources will survive and continue to be accessible to stakeholders well into the future. As the repository will be an international collaborative asset, metadata will provide a means to track the history of the digital content (such as its origin and how it has changed over time).

As mentioned before, the low-level details of the metadata implementation does not need to be discussed in this paper as there are many readily available tools and methods that can be utilized to achieve its integration on the repository model. For the repository, metadata templates specific to the curriculum model can be created to ensure users populate pre-set fields with validated input, then there are existing tools that can be used to Mark-up this information, extract it or convert it to whatever form needed.

Table 8 - Course Elements

Element	Definition
COURSE ABBREVIATION	The abbreviation for the course. Example: CMP for Computing Essentials
COURSE NUMBER	The unique number associated with the course. May be alphanumeric as well.
COURSE LONG	The un-abbreviated name of a course. Example Computing where the abbreviation is CMP
COURSE TITLE	The short description of the course. E.g. Computing Essentials
COURSE DESCRIPTION	Describes the course in greater detail than the title.
DEPTH	The amount of time attributed to this course as a subset of the Knowledge Area. This can be expressed as a percentage or as a unit of time.
YEARS	A collection of one or more years for which this course is to delivered

Table 9 - Course Metadata model

Element	Definition
PARENT	A collection of zero or more immediate parents (objectives) of this course.
CHILDREN	A collection of zero or more references to linked topics.
OBJECTIVES	A collection of zero or more references to linked objectives
REQUIRED RESOURCES	A collection of recommended resources for the effective delivery of this course. Example: Broadband Internet, Computers.

3.5.3 *Repository Content Ranking, Threshold and Recommendations*

A unique feature within the proposed paradigm is the notion of content ranking. This feature stems from the need to allow for flexible curriculum designs while not compromising on quality. But in order for this to be possible, there must be mechanisms in place to prioritize content by providing qualitative ranking coupled with quantitative metrics. What this really means all content such as topics must be ranked based on their importance or relevance to the curriculum and certain pre-determined threshold is set to determine not just the quality of the selected curriculum but also the coverage achieved.

In section 3.3.4, topic relevance was discussed, and according to the IEEE-CS/ACM document, a topic can be rated as essential (E), desirable (D) or optional (O). A topic rated as E is more important than one rated D, and a topic rated D is more important that one rated O. Put another way, an essential topic is one which must be included in every curriculum, while a desired topic should be included if possible. Once the input source is completely modeled, all topics would have been appropriately tagged with topic relevance.

3.5.3.1 Dynamic Adaptive Selection Algorithm

An important feature of the proposed paradigm is the ability for a user to design a curriculum that suits his/her needs, is context aware and still maintain some minimum quality threshold. This is where the Dynamic Adaptive Selection Algorithm (DASA) comes into play. DASA as the name suggests is dynamic, as it allows the user the flexibility of customizing their curriculum, and it is adaptive in that the algorithm actively

'crawls' the repository to find alternatives and recommendations for any action that may diminish the quality of the curriculum. So if a user declines an essential topic because of a lack of resources (such as no faculty with required expertise) to deliver such topic, then the algorithm tries to find a suitable alternative from within the content base and make recommendations. The algorithm is outlined below in pseudo code and time sequence diagram.

The full set of topics and courses is made available to the user to choose from. The user must select the courses for each knowledge area followed by the topics for each course. After each selection is made, the entire tree is traversed checking for coverage and minimum quality threshold based on pre-determined parameters, and the coverage/quality statistics along with recommendations are updated and displayed to the user. Recommendations are being shown in real-time for gaps that exist in the coverage (which is dependent on the coverage achieved to date). Resources required can be shown when a course is selected or when a topic is selected.

To the user, the core content is just a set of topics and courses grouped under knowledge areas available for each of the 4 years. But on the back-end, there is a treasure trove of information linking these sets together and incorporating important attributes such as the objectives that are satisfied by a given piece of content.

Coverage at the higher levels is dependent on coverage at the lower levels. That is, the coverage of a Course node is dependent on the coverage of its descendants (selected topics), and so on. As the curriculum is being designed (that is, as the courses

and topics are being selected or deselected), the curriculum tree for that design instance is actually being generated.
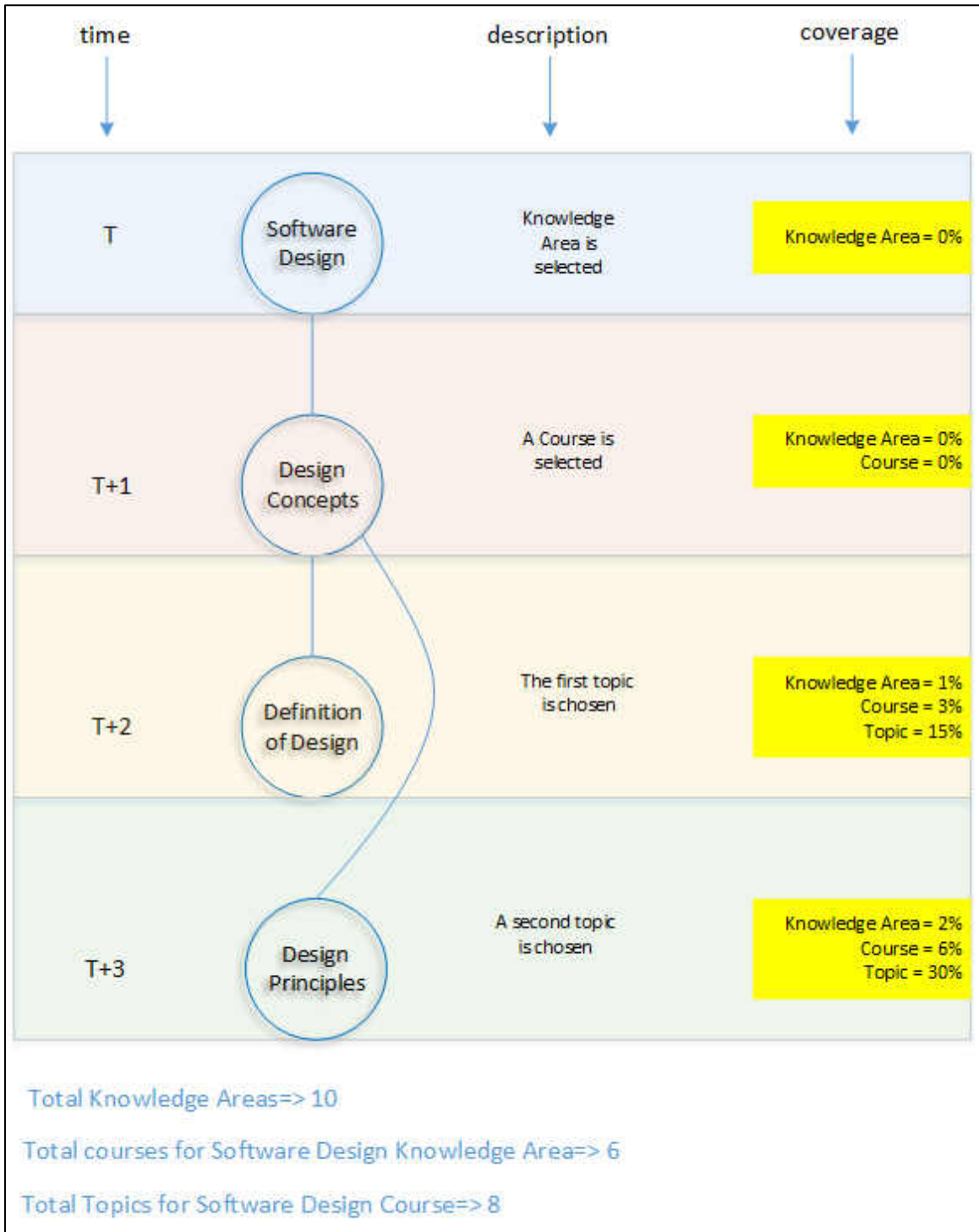
Figure 23 – Illustration of tree generation with the progression of time and actions taken.

```
public void Coverage_Check()
{
        foreach Knowledge Area
        {
                list nodesToVisit = currentKnowledgeArea  //take the KA to be the root of this sub-tree
                while(nodesToVisit != null){
                        currentNode = nodesToVisit.first();
                        TOTAL_COVERAGE += currentNode.Coverage;
                        nodesToVisit.prepend(currentNode.children);
                }

                if(TOTAL_COVERAGE < minimumRequired){
                        foreach item not selected{

                                Search_Within({
                                        traverse user tree comparing metadata of each item with
                                        unselected items to see if the coverage can be increased by
                                        current selections without the need to make additional
                                        selections
                                })

                                Search_Without({
                                        traverse the unselected portion of the tree model to find
                                        alternatives and recommendations to increase the coverage.
                                })

                                Search_External_Domains({
                                        //This is for future use when other CS sub-disciplines and
                                        domains are integrated with this platform
                                })
                        }
                }
        }
}
```

Figure 24 – Partial pseudocode for the DASA algortithm

This idea is illustrated in figure 23, where the action taken at each time interval is shown along with the progression of the tree. The coverage values are also shown for a typical scenario when selections are being made for a given Knowledge Area. In the diagram,

the Knowledge Area coverage shown is in relation to the entire curriculum, the coverage for the Course is in the context of the given Knowledge Area only, while the topic coverage is in the context of the given course only. What this means abstractly speaking, is that the curriculum core content has already been modeled as an ordered tree, and so when a user accesses the repository to design a curriculum from this model, the user is really creating his/her own customized version of this model (or of this ordered tree).

3.5.3.2 Algorithm explanation

Figure 24 is a snippet of the pseudo code that DASA entails. After each selection is made, DASA determines the coverage and hence the quality of the current selections. So if at a given point in time the designer decides to make no more selections, then the algorithm would show the coverage up to that point. So if for instance, the minimum quality threshold was set to be 70%, and the current selections achieve only 60%, then DASA will aid the designer to improve that percentage by:

1. Cross-referencing the selected courses and topics with unselected ones to find duplicates that can be weeded out which would result in an increase in coverage. This cross-referencing is done be comparing metadata and comparing the objectives that each topic satisfies.

2. Crawling the unselected items to find gaps that have diminished the coverage. This crawl (search) starts first with the KA followed by the Courses followed by the topics. At the KA level, it first looks to see if there is any KA that is missing, and so missing ones are crawled first. At the Course level, missing courses are

crawled first followed by courses with the lowest coverage. At the topic level, recommendations are made based on the level of importance where the Essentials are looked at first followed by Desired and then Optional topics. So if an Essential topic is missing and it will increase the coverage to acceptable levels then DASA will not need to look any further within the Course for recommendations. See figure 25 which illustrates these concepts.
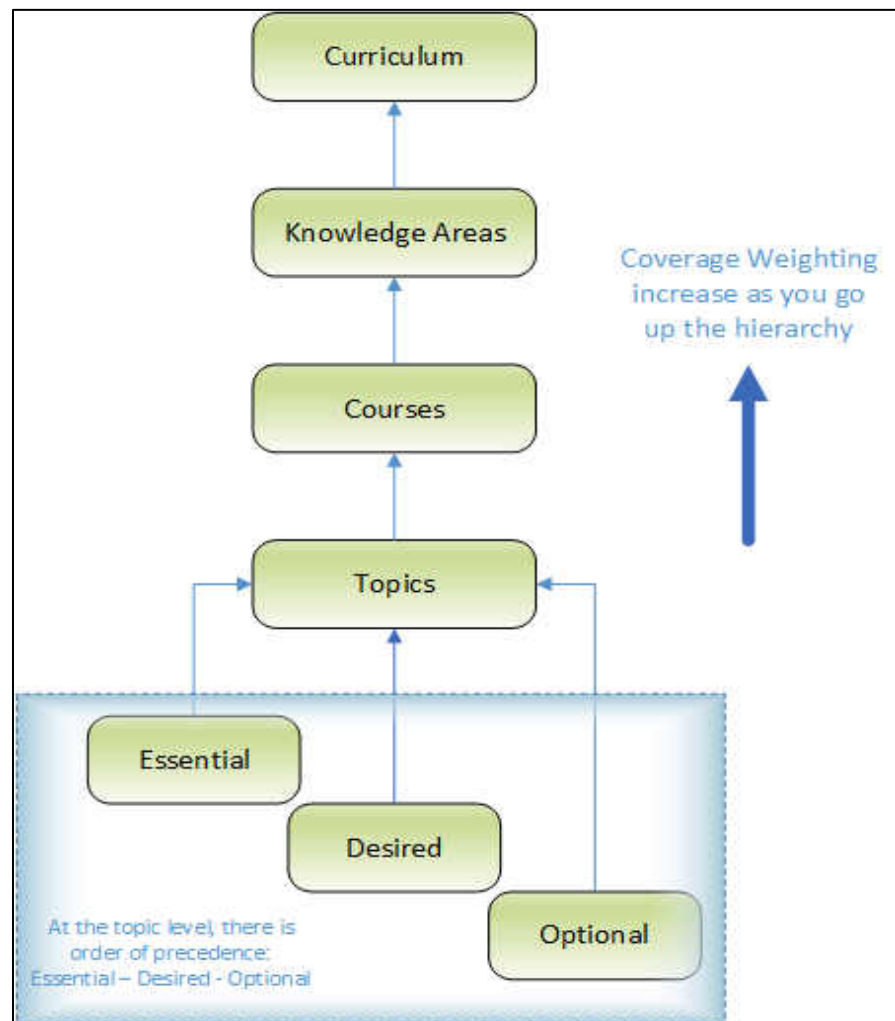
Figure 25 – Diagram showing Coverage Hierarchy and the direction of the Dependencies

3. Crawling external domains. This is a future aspect of the algorithm that will become useful when other computer science sub disciplines and other domains are integrated with this platform.

4. Enforcing topic relevance metrics. Topic precedence is an important issue because of the fact that topics can be designated as essential, desired or optional. So parameters can be preset to ensure that essentials topics must have coverage of 90%, desired topics must have coverage of 70% and optional topics can be 0% for instance.

3.5.3.3 How the running coverage values are calculated and updated

Consider the scenario where a Knowledge Area contains 5 courses and there are 10 topics for one of the courses, Course X. if only 5 of these topics have been selected, then Course X has achieved 50% coverage (5 out of 10), the Knowledge Area will then have 10% coverage (given that only this one course has been selected to date). Course X represents 20% of the total number of courses (which is 5), and so 50% of Course X's 20% share is 10%. The course weightings may or may not be equal since one course may have more topics than another course in the same Knowledge Area.

3.5.4 Summary of Methodology

In this methodology, the major aspects were divided into three phases; the workshops and maintenance model, the representation of a curriculum in the repository, and the utilization of the repository. The importance of the workshops as the

starting point for a repository of the type proposed was presented along with a model that will ensure its continued development and maintenance. The internal structure and representation of a modeled curriculum was presented which included the relationships between various components, the representation of the curriculum as an ordered tree and an algorithm that compliments the structure called DASA. Chapter will be a case study in which the phases outlined in the methodology will be applied in theory as a proof of concept.

Chapter IV

CASE STUDY

In this chapter, a case study will be presented using a 4 year undergraduate Software Engineering curriculum as a prototype. This prototype will be used to exercise the proof of concept of the three phases outlined in the methodology in chapter 3 of this paper. An end-to-end example will be presented that will include sample data from the IEEE-CS/ASM SE document as well as sample content that represents reasonable repository supporting content. A logical sequence of actions and usage scenarios expected for the development, maintenance and use of the repository will be presented in a manner that will allow for a non-trivial illustration of the framework.

4.1   Use Case – Workshops

It was elucidated in chapter 3 that the workshops play a pivotal role in this new paradigm. Such workshops represent the over-arching theme about collaboration and the sharing of expert knowledge and content across borders (in an international setting). The workshop reports that are summarized in the methodology arose from the efforts of Grant et.al. in [25] and is a most fitting example of what it takes to build the foundation and core input necessary for a repository and framework for any educational domain. Therefore, those cited workshops will constitute the workshop use-case for this

case study with a few minor additions. These additions stem from the fact that the workshops in [25] are currently ongoing which means that the complete output and results of those efforts are not available for this paper.

The expectations for the workshops upon completion are as follows:

State what the goals of the workshops are and annotate with tangible data figures

- A full set of courses and topics that represents the collective views of the participants of what should be in a SE curriculum.

- The learning objectives and goals for each course

- The attributes for each course such as the years in which they should be taught, the amount of time allocated for each whether it be in hours or a percentage of total course time.

- The output content from the workshops will also help in the organization and modeling of such content in the Repository.

4.2 Repository Contribution Via Maintenance Model

The Wikipedia Hybrid Model is an interesting approach to address the need for continued development and maintenance of the Repository. In this section, detailed examples of how this model will work will be outlined. These examples will include the contribution of supporting content such as the submission of questions to the Question Bank and the submission of core content such as the addition of new topics to a course. Recall that the Question Bank is a component of the Repository which stores and associates question to curriculum content.

4.2.1 Submission of questions to the Question Bank

It is expected that the implementation of a Question Bank and its integration within the framework will allow for a number of unique capabilities and value-added benefits. Imagine being able to generate an effective, well-targeted assessment that is tailored to the content (or portion) of the curriculum that has been delivered and that is generated automatically! Of course the user would have to specify the parameters and boundaries of such an assessment and at the click of a button such an assessment is auto-generated and ready for consumption. After students take such an assessment, the learning outcomes along with any number of useful coverage metrics can then be easily analyzed and traced back to the content. This is only but tip of the iceberg in terms of what is envisioned and the possibilities of having the Question Bank and its integration into the framework.

4.2.1.1 Question metadata model

It is important to point out the fact that a question is not the only artifact that constitutes an assessment or that makes up an assessment. An assessment could include projects, articles to review, research based work and myriad others. But for the purposes of this paper, questions are looked at as the main or simplest artifact for an assessment.

Essentially, the Question Bank is a data warehouse of questions that relate to the content of the domain for which it is a part of. So in this case study, the domain is Software Engineering, and so questions in the Question Bank will pertain to Software

Engineering. One of the key features of the Question Bank is that the questions contained within will have both syntax and semantics. So rather than the Question Bank being a static container of ad hoc questions, there will be inherent intelligence in the questions. This means that a question that exists in the Question Bank does not exist in isolation, but will take on both context and meaning. A question derives context from its association with curriculum content in the repository and it derives meaning from the fact that this association allows for its quantification as part of a whole. This quantification could mean for instance, deriving the coverage value or worth of a given question as part of the set of questions on an assessment. Questions in the Question Bank have syntax which stems from the metadata attributed to them.

The more precise the metadata is for a given question, the more useful that question is. In section 3.5.1, the usefulness of a question was discussed in relation to its metadata and the level in the hierarchy at which the relationship exists. If a question is linked only to a topic (that is, only the topic element was provided for the question metadata), then all the other derivative properties for that question can be garnered because a topic is the lowest level in the curriculum hierarchy and so tracing its ancestors will yield information like the courses and objectives that the question is attributed to. On the other hand, if the question was linked only to an objective, then the only other link that can be derived is the year in which such as question should be administered in an assessment.

Questions should not be ambiguous because the more accurate a question is in terms of the content it is targeted towards, the more useful the question will be. So while it is perfectly possible for a question to be targeted to several topics or courses, it is recommended that the question have a one-to-one relationship with curriculum content at any level in the hierarchy. That is, a question can only be linked to one Knowledge Area, one Course and or one Topic. Table 10 is a representation of what the question metadata should look like, while Table 11 provides a template for what a course as an entity in the Question Bank should look like.

Table 10 – Question Metadata Model

Element	Definition
METADATA ID	A unique system-generated id which links this metadata to the content it describes.
DESCRIPTION	A description of the rationale for the question.
KNOWLEDGE AREA	The knowledge Area that the question is linked to
OBJECTIVE	The Objective that the question is linked to
COURSE	The Course that the question is linked to
TOPIC	The Topic that the question is linked to
DATE SUBMITTED	The date the question was submitted to the Repository
SUBMITTER	The user who submitted the question
DATE PUBLISHED	The date the question was published to the Repository (made available to the community)

Table 11 – Question Elements

Element	Definition
QUESTION ID	A unique value attributed to each question; auto-generated by the system
QUESTION	The actual content of the question
ANSWER	A sample answer for the question
WRONG ANSWERS	A collection of incorrect answers to facilitate multiple choice assessments.
METADATA ID	A unique value which links the question to its metadata

The system would have to implement logic to remove duplicate (de-dupe) questions based on content and its metadata. When a new question is submitted, it first goes through an algorithm that compares the actual question and its metadata to what already exists in the database. The algorithm first analyses the questions metadata, and only compares questions whose metadata for KA, Objective, Course and Topic matches exactly. Only then will the algorithm compare the actual question and if the wording is the same or where there is a high degree of matching, the question will be rejected as a duplicate. Even if the exact same question (in terms of the wording) exists already in the Question Bank, it may exist in a different context, because the metadata would have been different and it is the metadata which gives a question its semantics. For example, the question "What is your immigration status?" could exist twice in the database, but what it is asking for is different depending on the context in which it is asked; wherein if it is asked in the context of tax filing, the correct answer could be *resident*, but if asked in the context of voting rights, the correct answer could be *non-resident*. So in such a

scenario, the question is not a duplicate. This is how you can have a question be linked to several topics or courses (even though the questions exists as separate entities in the database, the actual wording may be the same or similar), the metadata is what distinguishes them because each would be linked to a different course and or topic. In this case, the questions are not ambiguous in terms of their targeted content, but the end result of having a question attributed to more than one piece of content is achieved. If the de-duping algorithm is effective, then the concern of having duplicate questions on an assessment for auto-generated assessments will be eliminated.

4.2.1.2 End to End Flow for Submission of Assessment Supporting Content

An end-to-end flow for the submission of assessment supporting content such as that of questions can be seen in figure 26. The sample question elements and sample metadata model tables shown in the figure can be found in appendix A. In figure 26, the input is provided by a contributor who does the actions of creating and submitting a question to the platform. The submitter is considered a producer of the content and will not be allowed to be a moderator or editor (with the ability to review) for the instance of content that he/she submitted. The content submitted is analyzed by a De-duping algorithm and if it successfully passes this phase (meaning it is not found to be a duplicate), it is made available for wider editorial review. If the question receives moderate approval from contributing editors it is immediately published to the repository and made available for consumption.

Input Process

Contributor

Level: Expert
Credential: PhD
UserType: Producer

Create a question

Submit question

An expert contributor creates a question and submits it to the system for approval.

Data

Element	Definition
QUESTION ID	1025547492
QUESTION	What is a Data Flow Diagram?
ANSWER	The primary output of the system design phase.
WRONG ANSWERS	[The modern version of flowchart.] [Mainly used at systems specification stage.] [All of the above.]
METADATA ID	4444

Sample Question elements

Element	Definition
METADATA ID	4444
DESCRIPTION	A description of the rationale for the question.
KNOWLEDGE AREA	Software Modeling and Analysis
OBJECTIVE	Effectively create data flow diagrams and understand DFDs as a core modeling activity in structured analysis.
COURSE	Types of Models
TOPIC	Structure Modeling
DATE SUBMITTED	March 1, 2013.
SUBMITTER	Emanuel Grant
DATE PUBLISHED	March 10, 2013.

Sample Metadata model

Approval Process

It can be seen from figure 19 that a question is not core content and so because its type is assessment, it goes through a de-duping process where the De-duping algorithm is applied to it. Assuming that this is a new question in that it was not found to be a duplicate, it is then made available for wide editorial review, after which a moderate approval rate will see the question being published to the repository.

Result/Output

The question is published to the Question Bank component of the repository and made available immediately to users (consumers).

Figure 26 - End to End Flow for Submission of Assessment Supporting Content

117

4.2.3 Example of auto-generated assessment based on certain parameters/criteria

The framework outlined in the proposed paradigm can be leveraged to provide a plethora of benefits. One such benefit will be the ability to generate automatic assessment based on certain criteria. Once the Question Bank is sufficiently developed and populated with a significant number of questions, assessments that are well targeted and well spread will be a realistic expectation. The metadata attributed to each question allows for a wide variety and number of possible criteria for any given assessment. For example, the number of possible unique set of criteria to generate an assessment for a curriculum with 10 knowledge areas and 50 courses and 500 topics is 10*50*500 = 250,000. Coupled with such a wide set of criteria, the question bank will be able to produce an exponentially increasing number of unique assessments as the number of questions in the Question Bank increases. For example, a Question Bank with only 50 questions will be able to yield 10,272,278,170 unique 10 question assessments. Still, 50 questions is a very small number of questions that students will be able to memorize, so it is very important to have a much larger pool of questions. This value was derived using the formula:

nCr = n!/[r! (n-r)!]

where n is the number of question in the question bank

and r is the number of questions on an assessment.

Input Process

Consumer

Credential: Lecturer
UserType: Consumer

Sign in to portal

Submit criteria

A lecturer at a university logs into the repository portal and submits his criteria for an auto-generated assessment

Input Data

Parameters	Definition
NUMBER OF STUDENTS	30
NUMBER OF QUESTIONS	25
ASSESSMENT TYPE	{ [multiple choice], [open-ended] }
KNOWLEDGE AREA	Computing Essentials
OBJECTIVE	-
COURSE	{ [Computer Science Foundations], [Construction Technologies], [Construction Tools], [Formal Construction Methods] }
TOPIC	-
BLOOM'S ATTRIBUTES	{ [Knowledge], [Application] }
TOPIC RELEVANCE	Essential

Sample Criteria

Selection Process

- An assessment generation algorithm takes the parameters from the criteria the user selected and applies it across the Question Bank. The algorithm tries to match the provided parameters with the metadata of questions in the Question Bank.
- Since no topic parameter is supplied, all questions targeted at the specified courses in the Computing Essentials KA are selected,
- The secondary set of parameters such as the Bloom's Attributes, Topic Relevance and Assessment Type are applied to this selection to eliminate any questions not matching these secondary parameters.
- The requested numbers of questions are then randomly selected for the remaining set of questions.

Result/Output

- A set of questions matching the criteria for assessment
- Percentage assessment coverage for the specified KA
- Percentage assessment coverage for the specified Courses
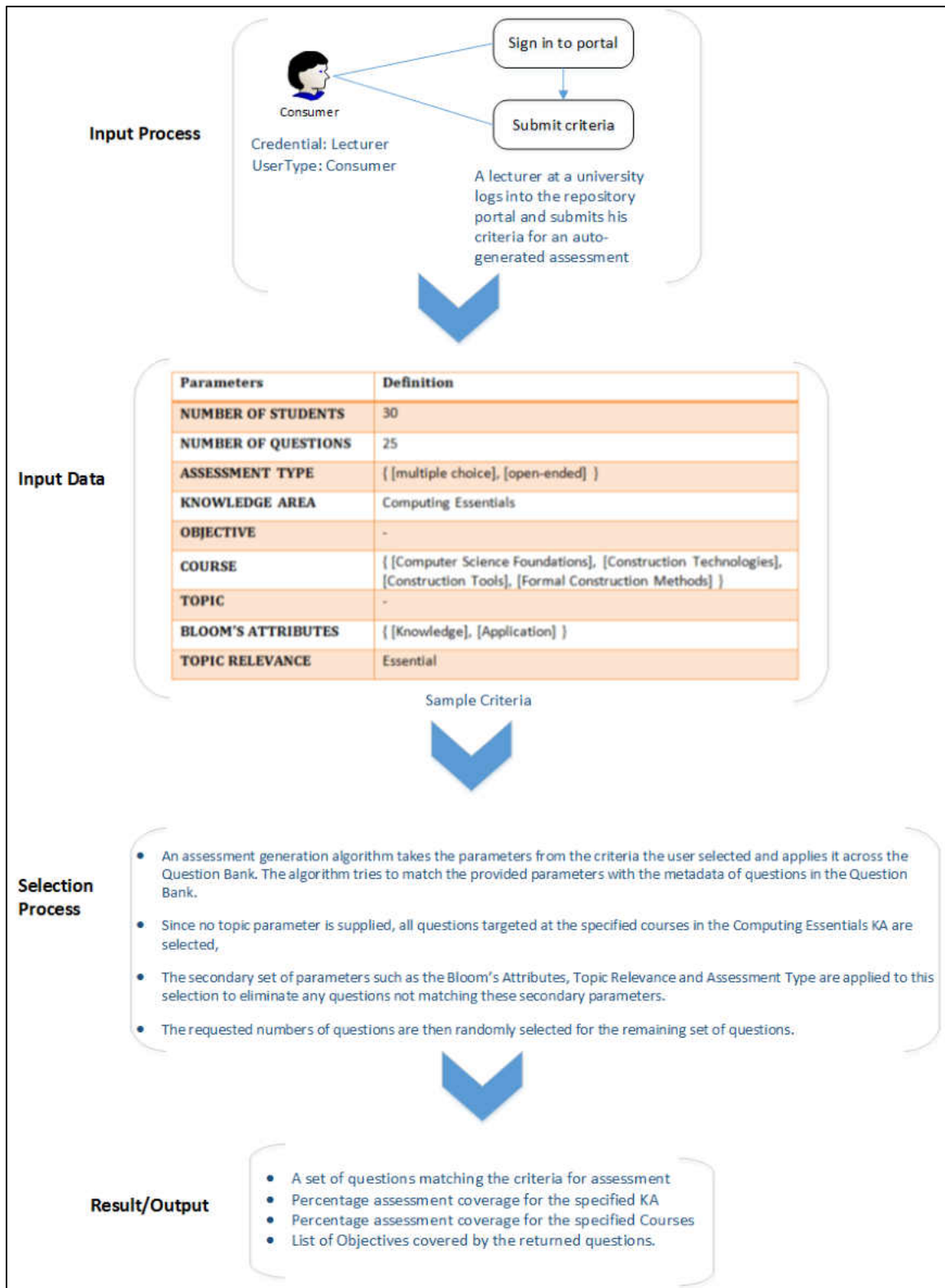- List of Objectives covered by the returned questions.

Figure 27 - End to End Flow for Auto-Generation of Assessment

Assessments can also be generated against the designer of a curriculum. Based on the curriculum that was selected, the questions will be generated within those boundaries. So if the curriculum did not include the *Construction Tools* course from the *Computing Essentials* knowledge area, then an assessment with the explicit criteria of only the *Computing Essentials* knowledge area will not include questions from the *Construction Tools* course because the system knows that that course was not included in the curriculum and therefore not delivered and should perhaps not be assessed upon.

Consider the scenario where a Professor wishes to give the students taking his *Computing Essentials* Knowledge Area class an assessment at the end of the semester. The semester has been a very busy one for the Professor and he does not have time to create all the questions for the assessment. The solution for the Professor is to access the Repository portal and let it do all the work for him to create this assessment. The Professor know exactly what units and topics were covered (taught) during the semester, and so it will be an easy task for him to just enter the criteria that best matches what he wants the students assessed on. This scenario is depicted in figure 27. The input data table showing the Assessment Criteria model and sample criteria can be found in the Appendix B.

4.2.4 Submission of Core Content

The previous section showed use cases for the submission of supporting content such as assessment questions and the utilization of such supporting content such as the generation of assessments. In this section, two examples will be given to demonstrate

the flow that will take place for the submission of core content. One example will show the path of rejection and the other example will show the path of acceptance to the repository.

Figure 28 depicts the scenario where a core content submission is rejected. There can be any of a number of reasons why a submission is rejected. But overall, if enough reviewers reject the submission, the high approval rate required for acceptance will not be met and the submission will be rejected. A few hypothetical reasons are given in the review process phase depicted in the diagram.

Figure 29 depicts the scenario where a core content submission is accepted (the tables used in both figure 28 and 29 can be found in Appendix C). Even though some reviewers may find faults or issues with the submission, this does not mean that they will reject it. In fact, there may be reviewers who reject the submission, but as long as enough accept it so that the overall approval rate is high, the submission will be accepted.
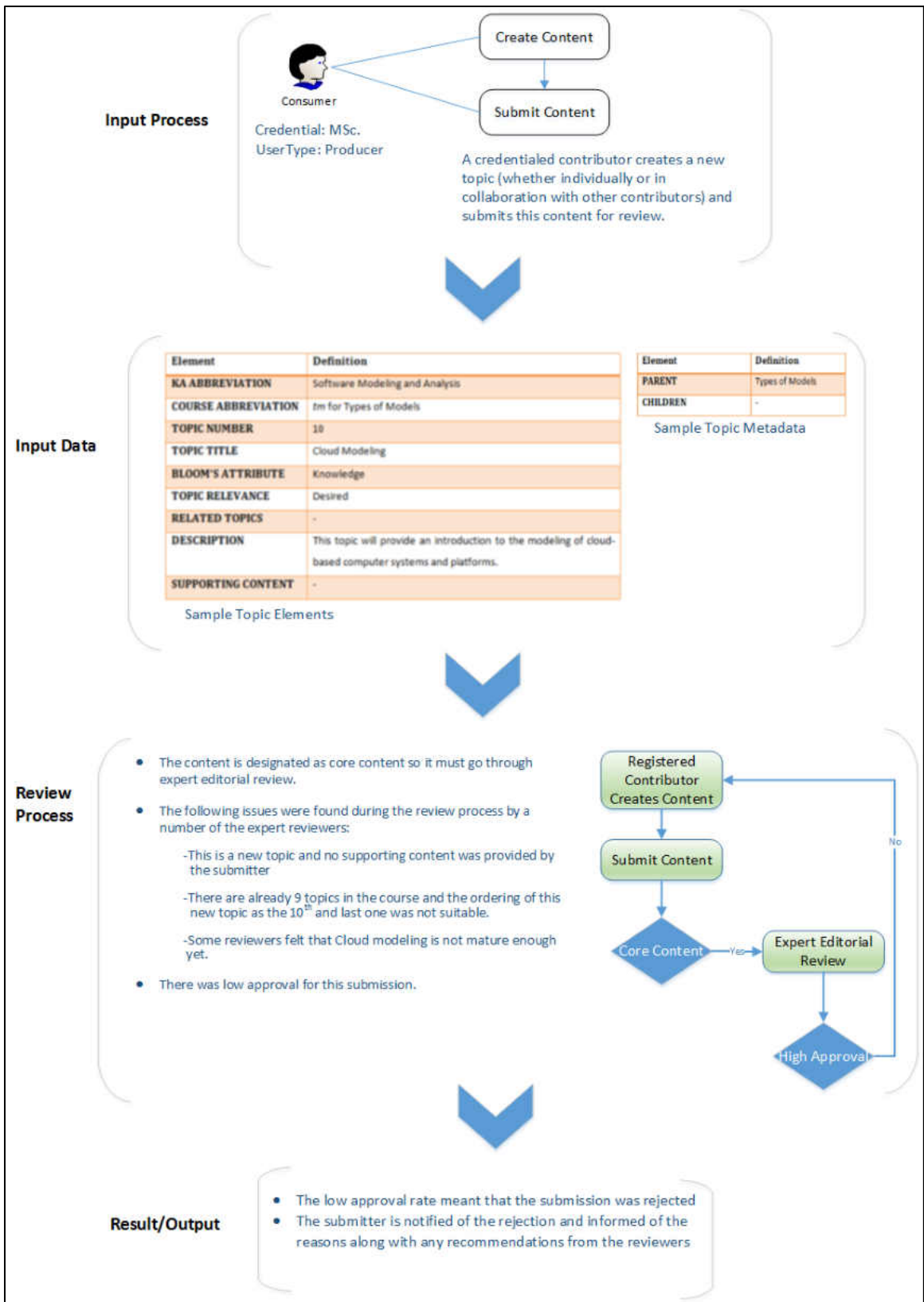
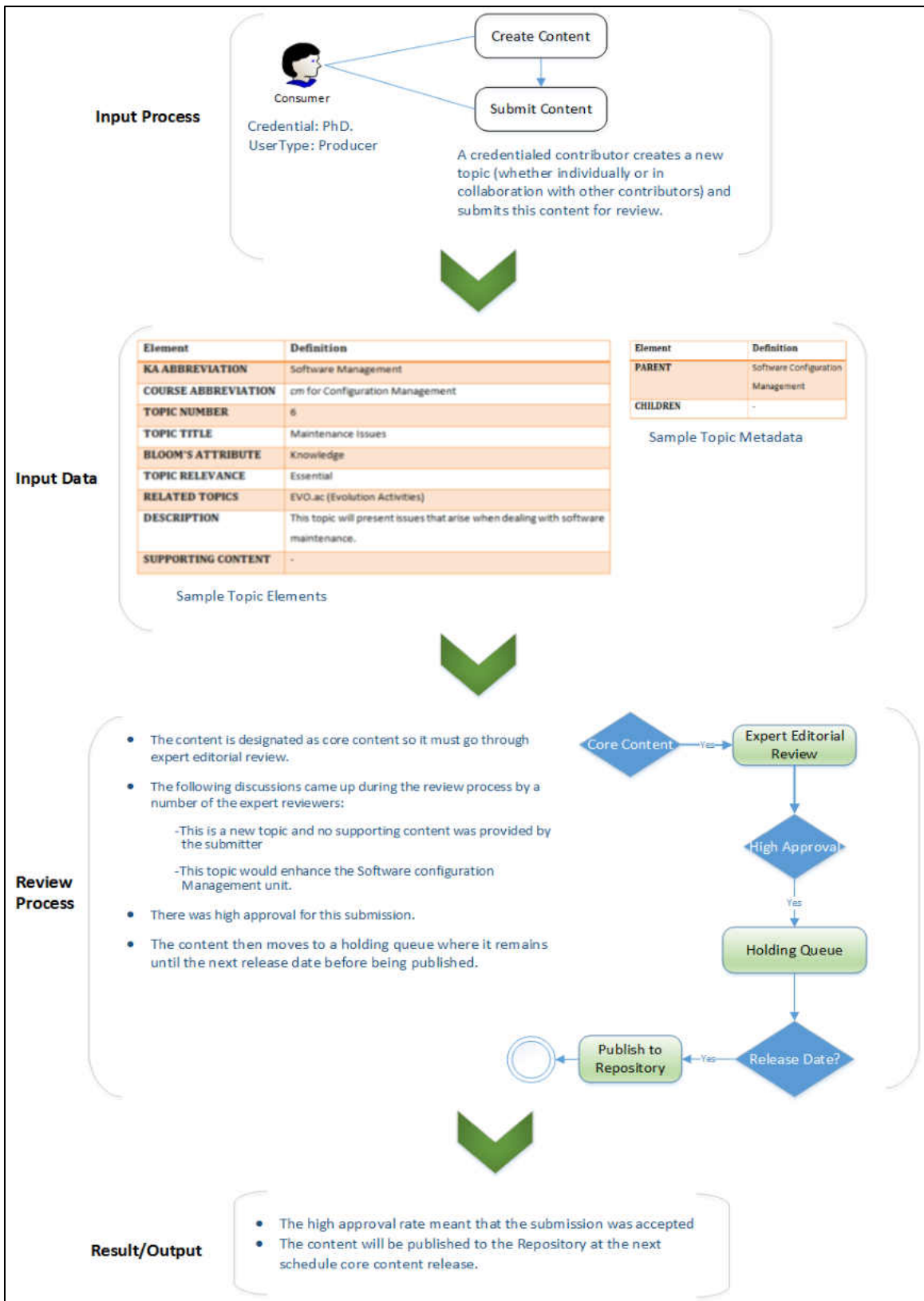Figure 28 - End to End Flow for submission of core content that is rejected

Figure 29 - End to End Flow for submission of core content that is accepted

4.3 Modeling and Representing Core Content in the Repository

As was mentioned earlier, the main input source for this case study is the IEEE-CS/ASM-SE 2004 document. Therefore, in this section, sample content form said document will be used to demonstrate the modeling of content using the proposed highly connected tree structure. A small subset of the content that constitutes the core of a software engineering curriculum will be used to demonstrate this modeling.

Table 12 – Professional Practice Knowledge Area learning objectives

Reference	Description
Objt.PRF.1	Make ethical decisions when faced with ethical dilemmas, with reference to general principles of ethics as well as codes of ethics for engineering, computing, and software engineering.
Objt.PRF.2	Apply concern for safety, security, and human rights to engineering and management decision-making.
Objt.PRF.3	Understand basics of the history of engineering, computing, and software engineering.
Objt.PRF.4	Describe and apply the laws that affect software engineers, including laws regarding copyright, patents, and other intellectual property.
Objt.PRF.5	Describe the effect of software engineering decisions on society, the economy, the environment, their customers, their management, their peers, and themselves.
Objt.PRF.6	Describe the importance of the various different professional societies relevant to software engineering in the state, province or country, as well as internationally.
Objt.PRF.7	Understand the role of standards and standards-making bodies in engineering and software engineering.
Objt.PRF.8	Understand the need for continual professional development as an engineer and a software engineer.
Objt.PRF.9	Understand the importance of working in teams/groups.

Table 13 – Professional Practice Knowledge Area with courses, topics and references [26]

Reference		k,c,a	E,D,O	Hours	Related Topics
PRF	**Professional Practice**			35	
PRF.psy	*Group dynamics / psychology*			5	
PRF.psy.1	Dynamics of working in teams/groups	a	E		
PRF.psy.2	Individual cognition (e.g. limits)	k	E		DES.hci.10
PRF.psy.3	Cognitive problem complexity	k	E		MAA.rfd.8
PRF.psy.4	Interacting with stakeholders	c	E		FND.ec.2
PRF.psy.5	Dealing with uncertainty and ambiguity	k	E		
PRF.psy.6	Dealing with multicultural environments	k	E		
PRF.com	*Communications skills (specific to SE)*			10	
PRF.com.1	Reading, understanding and summarizing reading (e.g. source code, documentation)	a	E		MAA.rsd.1
PRF.com.2	Writing (assignments, reports, evaluations, justifications, etc.)	a	E		
PRF.com.3	Team and group communication (both oral and written, email, etc.)	a	E		MGT.per
PRF.com.4	Presentation skills	a	E		
PRF.pr	*Professionalism*			20	
PRF.pr.1	Accreditation, certification, and licensing	k	E		
PRF.pr.2	Codes of ethics and professional conduct	c	E		
PRF.pr.3	Social, legal, historical, and professional issues and concerns	c	E		
PRF.pr.4	The nature and role of professional societies	k	E		
PRF.pr.5	The nature and role of software engineering standards	k	E		MAA.rsd.1,CMP.c t.14,PRO.imp.3,7, QUA.std
PRF.pr.6	The economic impact of software	c	E		FND.ec
PRF.pr.7	Employment contracts	k	E		

Figure 30 shows the Professional Practice knowledge area modeled as an ordered tree. There are 9 learning objectives attributed to this knowledge area as shown in table 12 along with the references (code) that are used in figure 30. Table 13 shows the content taken from the source document in [26] that is also used to create figure 30. These objectives are satisfied by 3 courses. The links between the objectives and the courses indicate which courses satisfy which objective.
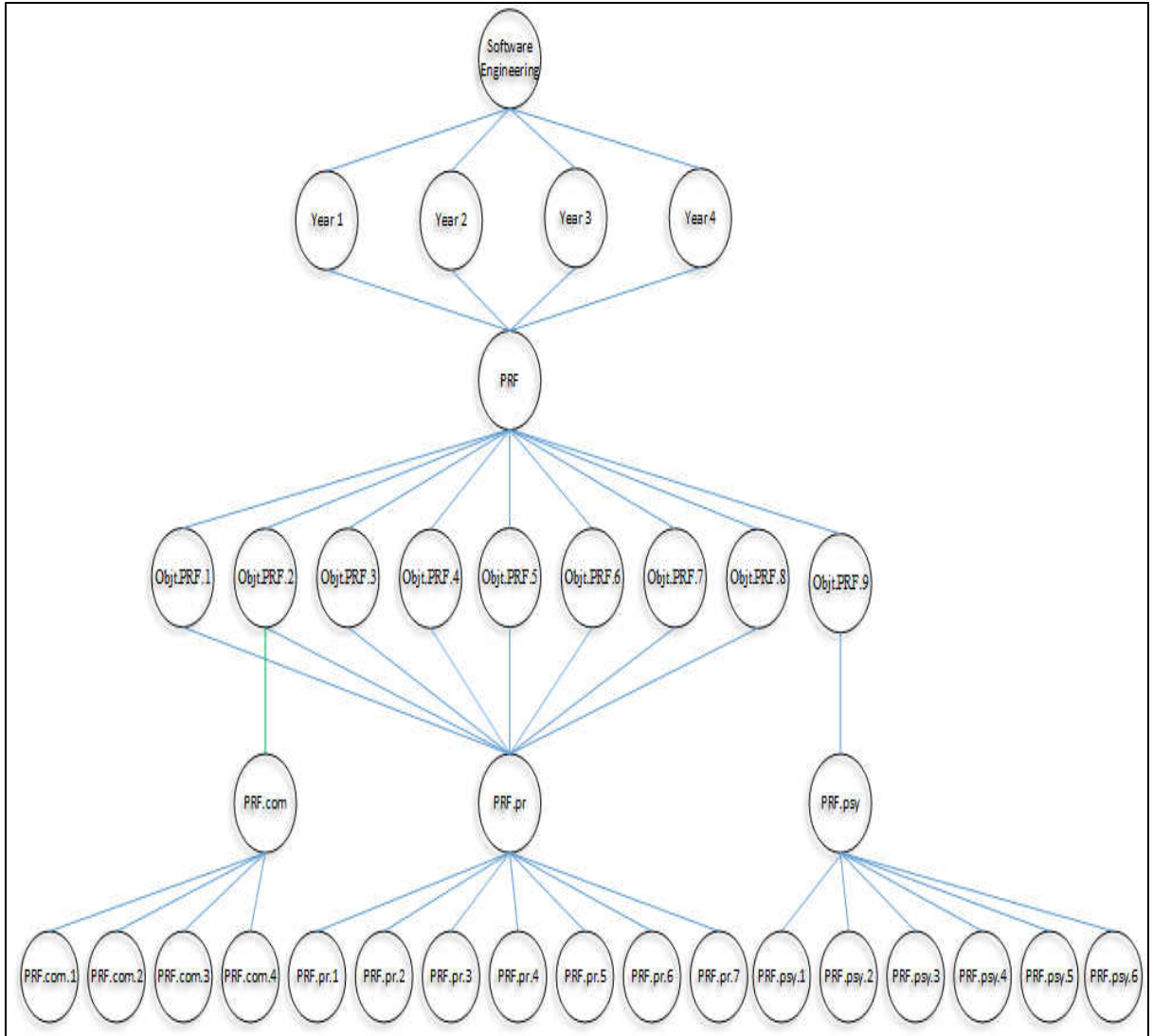
Figure 30 – Professional Practice Knowledge Area modeled in Tree Structure.

For a given leaf node (topic) in the tree, a number of valuable information can be learned by tracing its ancestry. Take *Prf.pr.4,* by querying on this topic alone, the following can be inferred about this topic:

- It is a part of the Professional Practice course

- It possibly satisfies objectives 1 through 9. Even though this one topic in reality does not necessarily satisfy all eight objectives, based on the logical

connectedness in the tree, this is a reasonable inference even though it may not

be very useful or well-targeted.

- This topic can be delivered in years 1 to 4. The metadata model for this topic has

    a property for the years in which it is most appropriate to deliver the topic. But if

    this property is not specified, it can be inferred (derived) from the tree (which is

    obviously less accurate than if it was taken directly from the metadata.

- It can also be seen that two of the courses (*PRF.com* and *PRF.pr*) satisfy the same

    objective (*Objt.PRF.2*). So if one were to query the Repository to find all courses

    that satisfy objective *Objt.PRF.2*, both *PRF.com* and *PRF.pr* would be returned.

    Similarly, a query to find all the topics that satisfy this same objective would

    return all the children (topics) of both *PRF.com* and *PRF.pr*.

It would be very difficult to manually represent the entire software engineering

curriculum core content in this paper, and so only a sample of this content has been

modeled. Figure 31 is a snippet of what the tree would look like down to the level of the

knowledge areas and including only two of the knowledge areas – Professional Practice

and Software Management.

Figure 32 shows the Software Management knowledge area modeled as an ordered

tree. Table 14 lists the ten learning objectives attribute to this knowledge area along

with the reference codes that are used in figure 13. Table 14 lists the courses and topics

along with their reference codes taken from the input source document. These codes

are used in figure 32 to make it more readable. The trees represented in figures 30 and

32 are only a part of the complete tree that would exist if all the Knowledge Areas were
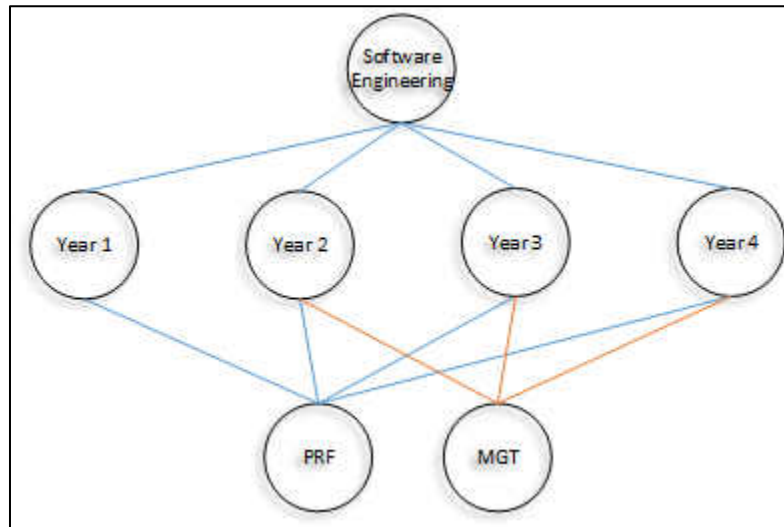
modeled in a similar way.

Figure 31 – Snippet of Tree highlighting the years in which Professional Practice and
Software Management are to be taught.

Table 14 – Software Management Knowledge Area learning objectives

Reference	Description
Objt.MGT.1	Develop a comprehensive project plan for a significant development effort
Objt.MGT.2	Apply management techniques to projects that follow agile methodologies, as well as methodologies involve larger-scale iterations or releases
Objt.MGT.3	Effectively estimate costs for a project using several different techniques.
Objt.MGT.4	Apply function point measurement techniques
Objt.MGT.5	Measure project progress, productivity and other aspects of the software process
Objt.MGT.6	Apply earned-value analysis techniques
Objt.MGT.7	Perform risk management, dynamically adjusting project plans
Objt.MGT.8	Use configuration management tools effectively, and apply change management processes properly
Objt.MGT.9	Draft and evaluate basic software licenses, contracts, and intellectual property agreements, while recognizing the necessity of involving legal expertise
Objt.MGT.10	Use standards in project management, including ISO 10006 (project management quality) and ISO 12207 (software development process) along with the SEI's CMM model

Table 15 – Professional Software Management Area with courses, topics and references [26]

| Reference		k,c,a	E,D,O	Hours	Related Topics
MGT	**Software Management**			19	
MGT.con	*Management concepts*			2	
MGT.con.1	General project management	k	E		
MGT.con.2	Classic management models	k	E		
MGT.con.3	Project management roles	k	E		
MGT.con.4	Enterprise/Organizational management structure	k	E		
MGT.con.5	Software management types (e.g. acquisition, project, development, maintenance, risk, etc.)	k	E		FND.ec.4,MGT.p p.6,EVO
MGT.pp	*Project planning*			6	VAV.fnd.2,QUA.p ca.2
MGT.pp.1	Evaluation and planning	c	E		
MGT.pp.2	Work breakdown structure	a	E		
MGT.pp.3	Task scheduling	a	E		
MGT.pp.4	Effort estimation	a	E		FND.ec.3,QUA.cd .4
MGT.pp.5	Resource allocation	c	E		
MGT.pp.6	Risk management	a	E		FND.ec.4
MGT.per	*Project personnel and organization*			2	PRF.com.3
MGT.per.1	Organizational structures, positions, responsibilities, and authority	k	E		PRF.psy.1
MGT.per.2	Formal/informal communication	k	E		PRF.com.1, PRF.com.2, PRF.com.3
MGT.per.3	Project staffing	k	E		
MGT.per.4	Personnel training, career development, and evaluation	k	E		
MGT.per.5	Meeting management	a	E		
MGT.per.6	Building and motivating teams	a	E		
MGT.per.7	Conflict resolution	a	E		
MGT.ctl	*Project control*			4	
MGT.ctl.1	Change control	k	E		MAA.rfd.5,MGT.c m.1,2
MGT.ctl.2	Monitoring and reporting	c	E		
MGT.ctl.3	Measurement and analysis of results	c	E		PRO.con.4
MGT.ctl.4	Correction and recovery	k	E		
MGT.ctl.5	Reward and discipline		O		
MGT.ctl.6	Standards of performance		O		
MGT.cm	*Software configuration management*			5	
MGT.cm.1	Revision control	a	E		MGT.ctl.1
MGT.cm.2	Release management	c	E		MGT.ctl.1
MGT.cm.3	Tool support	c	E		
MGT.cm.4	Builds	c	E		
MGT.cm.5	Software configuration management processes	k	E		
MGT.cm.6	Maintenance issues	k	E		EVO.ac
MGT.cm.7	Distribution and backup		D		
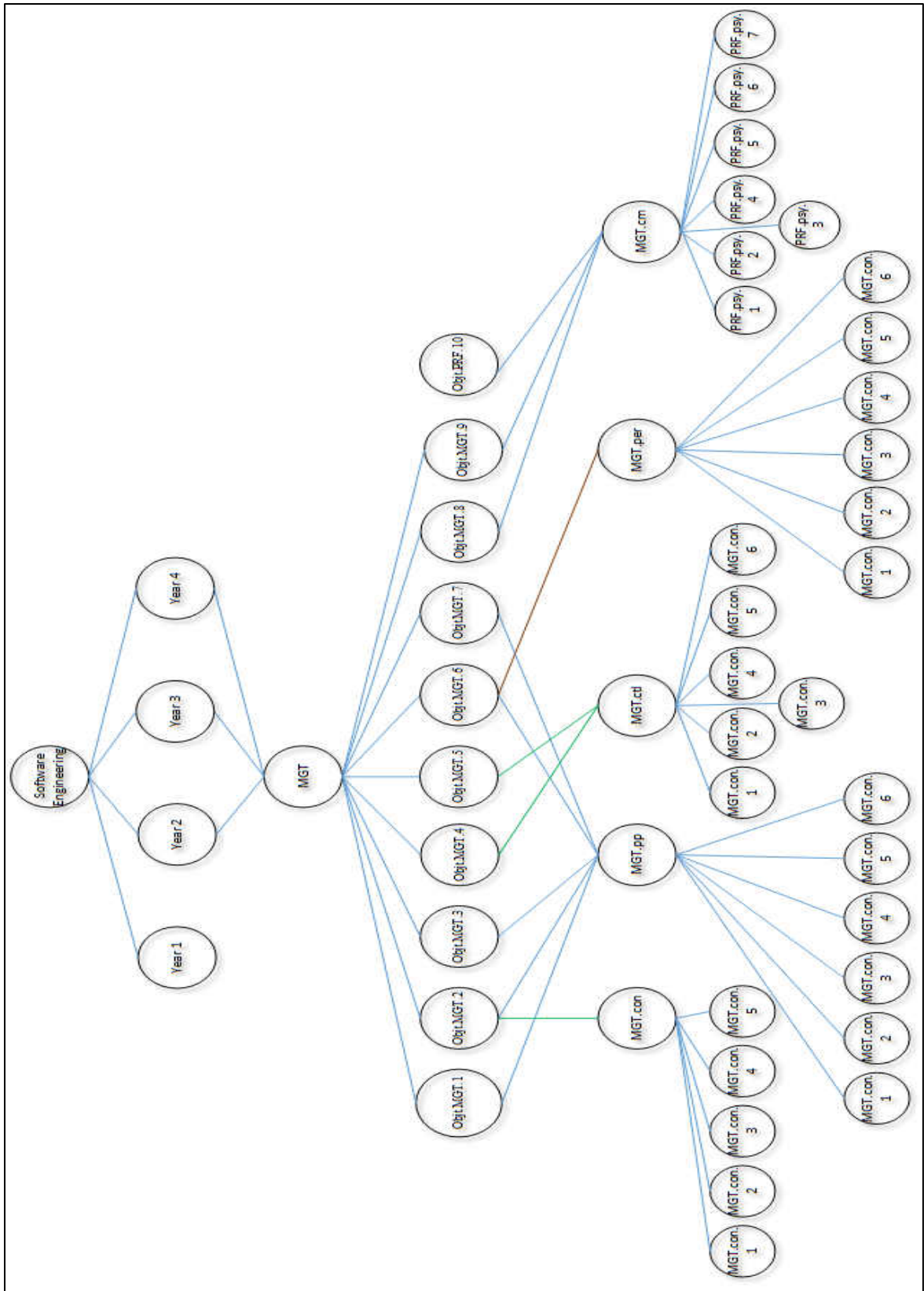
Figure 32 – Software Management Knowledge Area modeled in Tree Structure.

4.4 Repository Utilization for Curriculum Design

In this section, several hypothetical scenarios will be presented in order to demonstrate various use cases. These use cases will exercise the theory presented in chapter three which surrounds accessing the Repository to design a curriculum and how the proposed algorithm DASA helps to guide that process.

4.4.1 The Big Picture

The names and information given here are fictional, but serve to represent a realistic scenario to illustrate the utilization of the proposed paradigm. Mr. Campbell is a lecturer and Head of the Computer Science department at the University of the West Indies in Jamaica (UWI). Mr. Campbell has a Master of Science degree in Computer Science and has been teaching at the tertiary level for over 10 years. Currently, his department offers an undergraduate degree program in Management Information Systems, but he now wants to add an undergraduate degree program for Software Engineering. As a certified lecturer and teacher, he has been granted access to the Repository to use it to design a Software Engineering curriculum suited to the needs of his department. He will access the Repository portal via the internet.

The human resources available to Mr. Campbell and his CS department are:

- The level of expertise of the faculty in the department is limited to MSc and BSc with teaching certifications.

- There is no faculty on staff with a Mathematics degree or background

The physical resources available are:

- Broadband internet access

- Desktop computers with both Windows OS and Linux OS, but none with Mac OS

- Classrooms with overhead projectors

The cultural setting is one where students generally have excellent communication skills, but are not as strong on technical aspects of a curriculum. The culture at the university is one where there is great emphasis on theory with less reliance on practical approaches and demonstrations. These cultural issues and context will be important for Mr. Campbell in designing an SE curriculum that is context-aware, but one which threads a fine line between sticking to the norms while adapting to a new curriculum with strictly enforced quality thresholds. Essentially, Mr. Campbell's goal in this curriculum design process is to end up with a curriculum that meets the minimum quality requirements, while ensuring that the curriculum is a best-fit for his unique circumstance and context.

4.4.1 The Design Process

Now that the context of the scenario has been laid out, it is time to show an end to end flow. An attempt will be made to show the sequence of actions taken by a user and the subsequent behind the scenes machinations that would accompany such actions. The sequence diagram shown in figure 33 illustrates the transitions between different core content and metadata entities in the Repository which is triggered by the user (Mr. Campbell) selecting a course. Firstly, a list of courses is presented to the user.

When the user selects a course, the system fetches the actual content such as its description, depth and years (drawn from the course element model) for that course and displays it to the user. The other event that occurs asynchronously is that the metadata model for that course is fetched in order to display additional information to the user such as required resources, but more importantly, to retrieve data that will allow the system to continue modeling the user's selections.
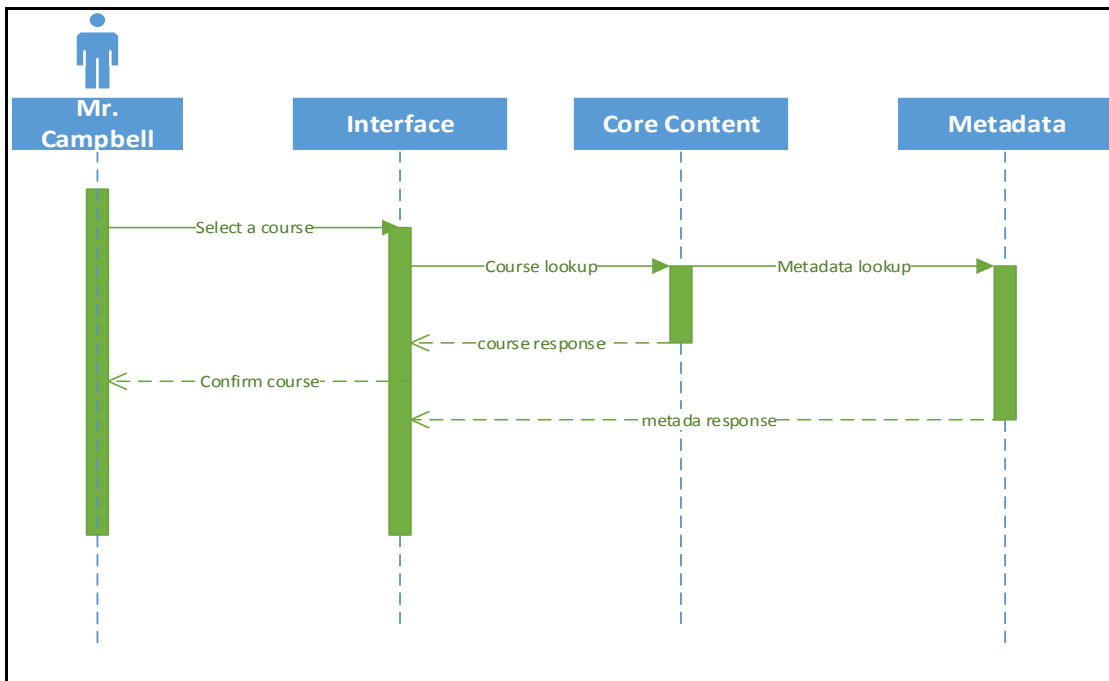

Figure 33 – Sequence diagram illustrating typical course selection interaction.

This segues nicely into figure 34 which encapsulates these actions and results pictorially. The first third of the diagram shows the user selecting a course, the fetching of the metadata model event that is triggered and the accompanying data in the response. The second third of the diagram is similar to the first, but this time it involves the selection of a topic. The final third shows the resulting Software Management (KA) sub-tree that is made possible by the metadata for selected the entities. Essentially, as

actions are completed, the tree representing the users selections (curriculum) to date is being constructed.
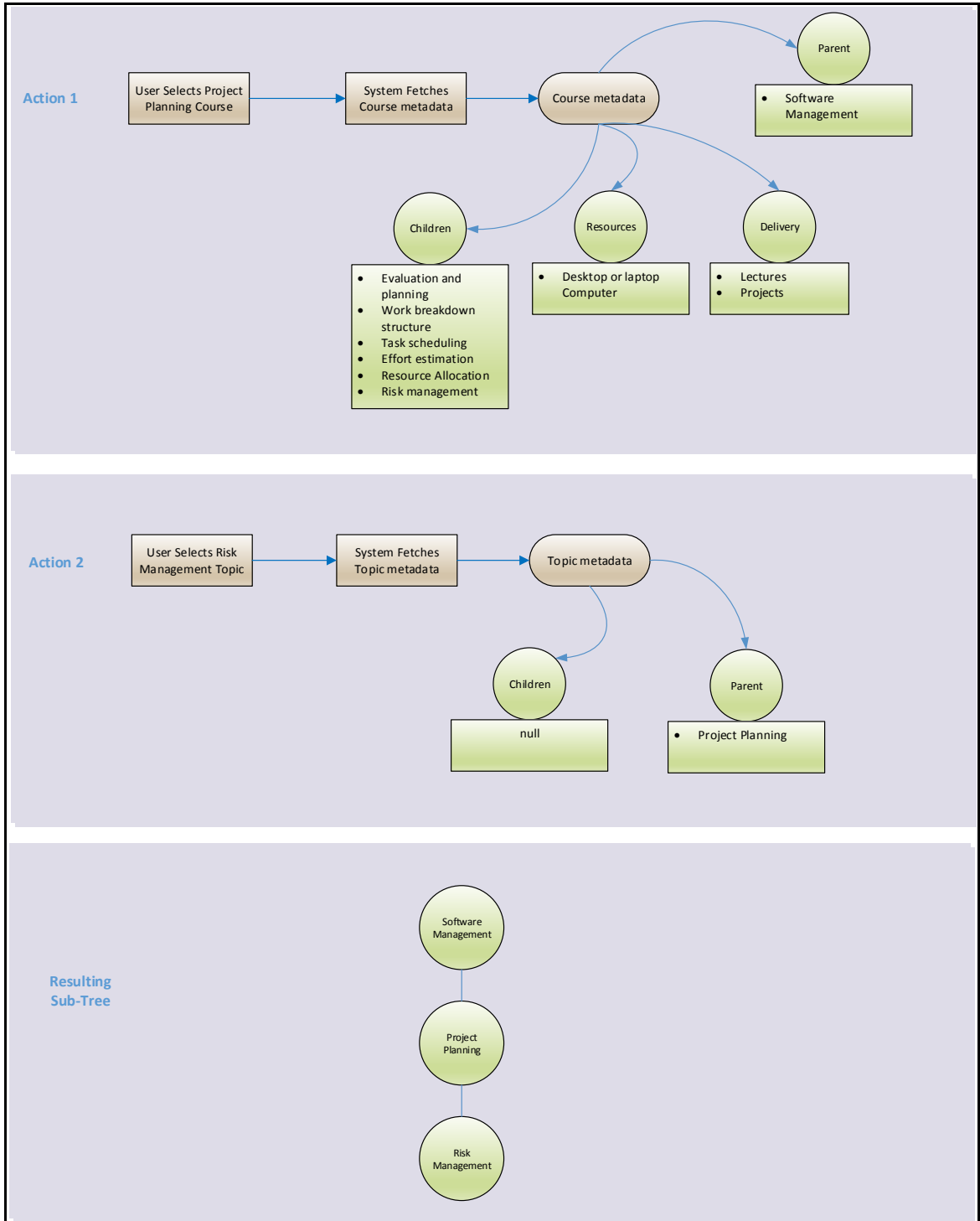

Figure 34 – Selection sequence showing content retrieval via metadata model for a selected course

One thing that may delineate various SE curricula is the coverage that is achieved

since ensuring that the quality of each meets certain minimum standard is one of the

main goals of this paradigm. Coverage can be measured at the overall curriculum level,

knowledge area, course and topic levels. DASA actively crawls the user's selection and

determines in real time the coverage to date. Figure 35 shows the coverage in parallel

with the time sequence for the Software Management sub-tree in Mr. Campbell's

curriculum. The table for the content codes used can be found in table 15. The following

is the explanation:

1. At time *T*, the KA is selected and its coverage at that point is 0% since nothing for

   that KA has been added yet.

2. At time *T+1*, the first course for this KA is selected (this KA has 5 courses). Both

   the KA and course coverage are still 0% since no topics have been added yet.

   Recall from figure 25 (the coverage hierarchy), how the coverage bubbles up

   from the lowest level in the hierarchy (topics).

3. At time *T+2*, the first topic is selected (this course has 5 topics):

   a. Topic coverage = 20% since 1 topic selected from 5 is $^1/_5$ which is 20%

   b. Essential Topic coverage is also 20% since all 5 topics in this course are

      essential topics

   c. Course coverage increases to 20% since there are 5 topics in the course

      and 1 has been selected.

   d. KA coverage increase to 4% since there are 5 courses in this KA and if

      equal weighting is applied to each course, each course would represent

      20% of the total. Since the course coverage to date for this one course is

20%, the calculation is 20% of this course's 20% stake in the KA which yields 4%.

4. Finally, at time *T+n*, it is seen that all 5 topics for the Software Management course in the Management Knowledge Area have been selected. Therefore, the course coverage is 100% (5 out of 5 topics selected), the essential topics coverage is 100%, and the KA coverage increases to 20% (since this course which represented 20% of the KA's coverage has achieved 100% coverage).
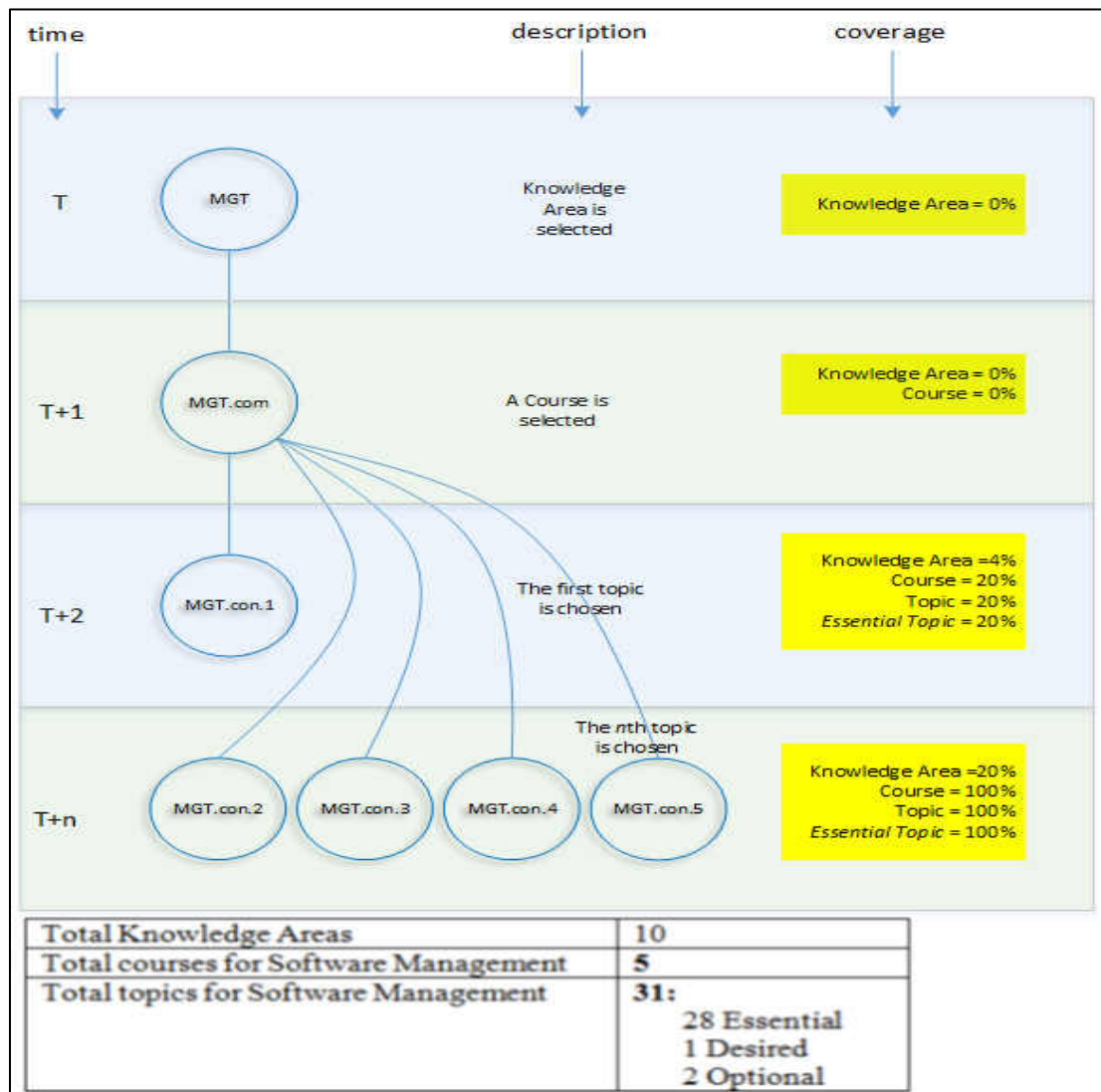

Figure 35 – Selection sequence with coverage progression calculations

If one were to attribute equal weighting to all 10 KAs, each would have a 10% stake in the overall curriculum coverage. Thus, taking the example in figure 35, the 20% coverage achieved for the Software Management KA would translate to a 2% curriculum coverage (20%*10%).

4.4.1.1 Scenario 1: Mathematics expertise limited

Mr. Campbell tried his best to limit the number of topics that he had to select under the Mathematical and Engineering Fundamentals knowledge area. This was due to the fact that the level of expertise in his department in this area is limited and so he feels that a student in this SE program can take a Mathematics elective from the Mathematics department. This is a very legitimate suggestion, but the curricula in other domains are not a part of the Repository (maybe in the future other domains will be integrated), therefore the system cannot verify that the content of a Math elective from another department is a good enough alternative for the core content in this KA. This is where DASA becomes useful, in that it suggests alternatives or adjusts the coverage based on current selections and possible duplicates.

Given that DASA found the Mathematical and Engineering Fundamentals coverage to be low after Mr. Campbell completed his selections, the following recommendations are made:

1. For the *Engineering Design* topic (FND.ef.5) that was not selected, the *Search_Within* routine of DASA found that Mr. Campbell has already selected the *Analyzing well-formedness* topic from the Software Modeling and Analysis KA, as

well as the *Evaluating cost-effective solutions* topic from the same Mathematical and Engineering Fundamentals KA. Given the fact that these two topics are already in the selected set and that they are related to this unselected topic, DASA determines that these already selected topics are suitable alternatives. So *FND.ef.5* is removed from the required topics list, effectively increasing the attained coverage by whatever percentage the removed topic was.

2. The *Search_Without* routine of DASA searched for all topics in the Mathematical and Engineering Fundamentals KA that have not been selected and displays an appropriate set of the essential topics to Mr. Campbell that would improve the coverage for this KA to acceptable levels if selected. Only the essentials are displayed as these have a higher order of precedence than the other types of topics.

4.4.1.2 Scenario 2: Recommended Resources

Given some of the limitations facing developing nations like the one where Mr. Campbell is, it is very useful to have the system display the recommended resources for each course before he makes the selection. So when he made the selection for the Testing course in the Software Verification and Validation KA, is was useful for him to know that some of the recommended resources included Intranet and Internet access, Computers and various operating system inclusive of Windows and Linux.

4.5 Analysis of Case Study

Based on the use cases and examples presented in this chapter, it is apparent that the proposed paradigm has the potential to radically improve the way curricula are designed and will allow for the leveraging of many derived benefits. For instance, at the click of a button one should be able to get a quantitatively and qualitatively generated assessment that can be trusted. The Wikipedia Hybrid model will allow the repository to quickly grow and include teaching artifacts like recorded lectures (multimedia), written lectures, research papers, questions and many more. A curriculum can be designed in a finite number of unique ways, but only a subset of that number will actually meet the pre-determined quality requirements.

This paradigm is in its infancy, and as is the case with novel approaches, there are "teething pains". Limitations found include:

1. The need to have wide consensus on what constitutes minimum quality threshold. This can be just a quantitatively defined value based on the raw figures of included versus excluded content. This may or may not be the best approach since topics, courses and KAs may be interpreted as having order of importance and carry different weightings.

2. The Wikipedia Hybrid Model for the Repository maintenance will depend on the contribution and time of the participants. If the level of submissions to the repository becomes too over-whelming, time constraints may begin to negatively impact the whole workflow.

3. The framework outlined for the Repository and modeling of the content along with the algorithms described are only a part of the whole story. Getting users of the system, especially those who will use it to design their curricula to accept the recommendations and guidance provided will take much effort and understanding.

4. The scope of this work did not go beyond the academia to include administration and accreditation bodies. So having such authorities "buy into" this new paradigm will also take effort. For example, having a Repository with core content that may change the next year after it was accredited by university administrators or boards may prove problematic.

CHAPTER V

CONCLUSION

A new paradigm for the integration of technology into curriculum design and student assessment was presented in this paper. In doing so, a proof of concept for the development, maintenance, modeling, representation and utilization of a repository for this purpose was outlined. A case study of a Software Engineering undergraduate curriculum was subsequently used to exercise aspects of this proof of concept. The findings thus far seem very promising as much of the theory presented are feasible and the implementation of this framework using any number of available programming language, web technologies and architectures is very practicable.

This work has indeed presented a roadmap for the reshaping of curriculum design from concept to assessment through technology driven methodologies. It has been shown that technology can be used to enhance and streamline the curriculum design process, capitalizing on expertise from international contributors in the given domain and making this knowledge and content available in a structured and cogent way. It was shown that having a centralized repository consisting of content and artifacts relevant to a given domain is both feasible and sustainable, with the added bonus that the quality of curricula birthed from such a repository will improve the quality of said curricula along with the resultant degrees and graduates.

5.1 Future Work

This work laid the foundations for greater things to come. The limitations that were found will have to be further investigated and addressed. There will need to be continued international collaboration and input gathering to solidify the set of content that will constitute the initial Repository core content. Cultivating a wider pool of contributors and experts will also be important to building the credibility, authenticity and prowess of this new paradigm and framework.

The proposed paradigm is a long term effort which will take a substantial amount of time to get "off the ground" and then there will be continued work to keep it going. Implementation of the proposed models is definitely one of the near term sequels to this work. Individual research works to tackle each of the major components of the framework, such as the Wikipedia Hybrid Maintenance model, the DASA-like algorithm, and an Analytics Framework that integrates the Repository with student learning outcomes from future participating institutions. Om that last note, more substantive work will need to be done to incorporate and integrate more of the student assessment aspect of the framework.

APPENDICES

Appendix A

Sample Question Elements

Element	Definition
QUESTION ID	1025547492
QUESTION	What is a Data Flow Diagram?
ANSWER	The primary output of the system design phase.
WRONG ANSWERS	[The modern version of flowchart.] [Mainly used at systems specification stage.] [All of the above.]
METADATA ID	4444

Sample Question Metadata Model

Element	Definition
METADATA ID	4444
DESCRIPTION	A description of the rationale for the question.
KNOWLEDGE AREA	Software Modeling and Analysis
OBJECTIVE	Effectively create data flow diagrams and understand DFDs as a core modeling activity in structured analysis.
COURSE	Types of Models
TOPIC	Structure Modeling
DATE SUBMITTED	March 1, 2013.
SUBMITTER	Emanuel Grant
DATE PUBLISHED	March 10, 2013.

Appendix B

Auto-generation of Assessment Criteria model

Parameters	Definition
NUMBER OF STUDENTS	The number of students expected to take the assessment
NUMBER OF QUESTIONS	The number of questions to be returned.
ASSESSMENT TYPE	The type of assessment; such as multiple choice, open-ended, subjective, etc. Users can specify multiple assessment types.
KNOWLEDGE AREA	A collection of one or more knowledge Areas that the questions are targeted at.
OBJECTIVE	A collection of one or more objectives that the questions are targeted at.
COURSE	A collection of one or more courses that the questions are targeted at.
TOPIC	A collection of one or more topics that the questions are targeted at.
BLOOM'S ATTRIBUTES	Return questions targeted at topics who's assigned Bloom's attribute matches those specified. User can specify one or more of Knowledge, Comprehension or Application.
TOPIC RELEVANCE	Return questions targeted at topics who's assigned topic relevance matches those specified. User can specify one or more of Essential, Desirable or Optional.

Sample Auto-generation of Assessment Criteria parameters

Parameters	Definition
NUMBER OF STUDENTS	30
NUMBER OF QUESTIONS	25
ASSESSMENT TYPE	{ [multiple choice], [open-ended] }
KNOWLEDGE AREA	Computing Essentials
OBJECTIVE	-
COURSE	{ [Computer Science Foundations], [Construction Technologies], [Construction Tools], [Formal Construction Methods] }
TOPIC	-
BLOOM'S ATTRIBUTES	{ [Knowledge], [Application] }
TOPIC RELEVANCE	Essential

Appendix C

Topic Elements for core content submission

Element	Definition
KA ABBREVIATION	The abbreviation for the Knowledge Area to which the topic belongs. Example: *VAV* for Software Verification and Validation
COURSE ABBREVIATION	The abbreviation for the course (unit). Example: *tst* for testing
TOPIC NUMBER	A number associated with the topic that may also be used to infer the ordering of the topics. Example: *tst.1* where the *1* denotes this is the first topic in the set.
TOPIC TITLE	The un-abbreviated name of a topic. Example *Unit Testing* where the full abbreviation is *VAV.tst.1*
BLOOM'S ATTRIBUTE	The Bloom's attribute for this topic. Example: One of Knowledge, Comprehension or Application
TOPIC RELEVANCE	The topic relevance assigned to this topic. Example: One of Essential, Desired or Optional.
RELATED TOPICS	A collection of zero or more topics that this topic is related to.
DESCRIPTION	A short description of what this topic is about
SUPPORTING CONTENT	Sample content for this topic or guidance on what this topic should constitute. At the time of submission, the user can provide links to a website or upload attachments for this content.

Topic Metadata model for core content submission

Element	Definition
PARENT	The immediate parent (*Course*) of this topic.
CHILDREN	A collection of zero or more references to linked topics which are this topic's sub-topics.

Sample Topic Elements for rejected core content submission

Element	Definition
KA ABBREVIATION	Software Modeling and Analysis
COURSE ABBREVIATION	*tm* for Types of Models
TOPIC NUMBER	10
TOPIC TITLE	Cloud Modeling
BLOOM'S ATTRIBUTE	Knowledge
TOPIC RELEVANCE	Desired
RELATED TOPICS	-
DESCRIPTION	This topic will provide an introduction to the modeling of cloud-based computer systems and platforms.
SUPPORTING CONTENT	-

Sample Topic Metadata model for rejected core content submission

Element	Definition
PARENT	Types of Models
CHILDREN	-

Sample Topic Elements for accepted core content submission

Element	Definition
KA ABBREVIATION	Software Management
COURSE ABBREVIATION	*cm* for Configuration Management
TOPIC NUMBER	6
TOPIC TITLE	Maintenance Issues
BLOOM'S ATTRIBUTE	Knowledge
TOPIC RELEVANCE	Essential
RELATED TOPICS	EVO.ac (Evolution Activities)
DESCRIPTION	This topic will present issues that arise when dealing with software maintenance.
SUPPORTING CONTENT	-

Sample Topic Metadata model for accepted core content submission

Element	Definition
PARENT	Software Configuration Management
CHILDREN	-

REFERENCES

[1] B. Hokanson, S. Hooper. Integrating Technology in Classrooms: We Have Met the Enemy and He is Us. Association for Educational Communications and Technology, 27th, Chicago, IL, October 19-23, 2004.

[2] Brooks, J., & Brooks, M. (2009). In search of understanding: The case for constructivist classrooms. Revised Edition. Retrieved 22 January, 2009: http://www.philomath.k12.or.us/superintendent/articles/Constructivist%20Classrooms.pdf.

[3] English, F.W. (2000). *Deciding what to teach and test: Developing, aligning and auditing the curriculum*. Thousand Oaks, CA: Corwin Press.

[4] Cuban, L. (1995). The Hidden Variable: How Organizations Influence Teacher Responses to Curriculum Reform. *Theory Into Practice, Vol. 34, No. 1*, 4-11.

[5] Yero, J., L. *Teaching In Mind: How Teacher Thinking Shapes Education.* 2nd Edition ©2010 Judith Lloyd Yero MindFlight Publishing Hamilton, MT.

[6] Miller, J. Computer Science Innovation in Thailand. *ICER 2009, Berkeley, California, August 10-11, 2009*

[7] Kaprelian, V. S. Faculty Development Module Instruction. Duke University Medical Center 2002. Retrieved from http://cfmmodules.mc.duke.edu/curriculum/index.html on March 13, 2012.

[8] Kaprelian, V. S. Faculty Development Module Instruction. Duke University Medical Center 2002. Retrieved from http://cfmmodules.mc.duke.edu/curriculum/intro/process.html on March 20, 2012.

[9] Haworth, K. (1999). *More Community Colleges Push to Hire Ph.D.'s as Professors*. The Chronicle pf Higher Education. Retrieved from http://chronicle.com/article/More-Community-Colleges-Push/8025/ on August 11, 2012

[10] Beetham, H. (2009) *Synthesis Report: Baselining the Institutional Processes of Curriculum Design*. Retrieved from http://www.jisc.ac.uk/curriculumdesign on August 11, 2012.

[11] Knight, S. *Institutional approaches to curriculum* design. Joint Information Systems Committee 2009. Retrieved from http://www.jisc.ac.uk/curriculumdesign on August 25, 2012.

[12]Beetham, H., Findlay, P., Comrie, A. *Managing Curriculum Change: Transforming curriculum design and delivery through technology*. Higher Education Funding Council for England 2009. Retrieved from

http://www.jisc.ac.uk/media/documents/publications/managingcurriculumchange.pdf

[13] Beetham, H. *Synthesis Report: Baselining the Institutional Processes of Curriculum Design*. Joint Information Systems Committee, November 2009. Retrieved from http://www.jisc.ac.uk/media/documents/programmes/curriculumdesign/designbaselinere port09.doc on August 25, 2012.

[14] Everett, J. *PiP: Principles in Patterns*. University of Strathclyde, 2008. Retrieved from http://www.principlesinpatterns.ac.uk/ on August 25, 2012.

[15] *Institutional Approaches to Curriculum Design. Update from Project Interim Reports May 2010*. Joint Information Systems Committee, May 2010. Retrieved from http://www.jisc.ac.uk/media/documents/programmes/elearning/Design%20Overview%20 May%2010%20i3.pdf on August 26, 2012.

[16] Palmer, A. *Enhancing learning and teaching through the use of technology: a revised approach to HEFCE's strategy for e-learning*. HEFCE, March 2009.

[17] Browne, T., Hewitt, R., Jenkins, M., Walker, R. *2008 Survey of Technology Enhanced Learning for higher education in the UK*. Universities and Colleges Information Systems Association. University of Oxford, November 2008.

[18] Nicol, D. *Increasing success in first year courses: Assessment re-design, self-regulation and learning technologies*. Proceedings of the 23rd annual ASCILITE conference, December 2006.

[19] Gluga, R. *Improving University Curricula*. Research Conversazione October, 2011.The University of Sydney. Retrieved from http://sydney.edu.au/engineering/it/research/conversazione_2011/glugar.pdf on September 16, 2012.

[20] Gluga, R., Kay, J., Lever, T., Lister, R. *An architecture for systematic tracking of skill and competence level progression in Computer Science.* In Proceedings of the 2nd Annual

International Conference on Computer Science Education: Innovation and Technology (CSEIT 2011), pages 65-69, 2011.

[21] Gluga, R., Kay, J., Lever, T. *Modeling long term learning of generic skills*. In Proceedings of the 10th International Conference on Intelligent Tutoring Systems, pages 85-94. Springer, 2010.

[22] Gluga, R., Kay, J., Lever, T., Lister, R. *An architecture for systematic tracking of skill and competence level progression in Computer Science.* In Proceedings of the 2nd Annual International Conference on Computer Science Education: Innovation and Technology (CSEIT 2011), pages 65-69, 2011.

[23] Grant, E., France, R. *Towards an Internet-based Education Model for Caribbean Countries*. Journal of Education Media, Vol. 25, No. 1, 2000.

[24] Golding, P., Tennant, V., Virtue, T. *Telecommunications in Jamaica: Monopoly to Liberalized Competition to Monopoly (2000 – 2011)*. Proceedings of the 4th Annual SIG GlobDev Workshop, Shanghai, China, December 2011.

[25] Grant, E. *Online Collaborative Teaching of Software Engineering in an International Multi-Campus Environment*. First HAU International Workshop on Enhancing Teaching & Learning of Software Engineering, Holy Angel University Angeles City, Philippines August 12, 2011

[26] Software Engineering 2004 Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. *A Volume of the Computing Curricula Series,* August 23, 2004

[27] Grant, E*.,* Stokke, T., Helps, W., Esquivel, J., Concepcion, W. *Enhancing Teaching & Learning of Software Engineering in an International Environment*. In Proceedings of the 2nd Annual International Conference on Computer Science Education: Innovation and Technology (CSEIT 2011), pages 36-41, 2011.

[28] Grant, E., Helps, W. *Enhancing Teaching & Learning of Software Engineering in an International Environment*. Journal on Computing (JoC) Vol.2 No.1 April 2012.

[29] Wikipedia: About. Retrieved from http://en.wikipedia.org/wiki/Wikipedia:About on October 14, 2012.

[30] Bloom, Benjamin S. Reflections on the development and use of the taxonomy in Anderson, Lorin W. & Lauren A. Sosniak, eds. (1994), Bloom's Taxonomy: A Forty-Year Retrospective. Chicago National Society for the Study of Education

[31] Anderson, L. & Krathwohl, D. A. (2001) Taxonomy for Learning, Teaching and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives New York: Longman

[32] Lanxon, Nate (June 5, 2008). "The greatest defunct Web sites and dotcom disasters". CNET UK. p. 5. Retrieved from http://crave.cnet.co.uk/gadgets/the-greatest-defunct-web-sites-and-dotcom-disasters-49296926/5/ on January 21, 2013.

[33] Guenther, R., Radebaugh, J. *Understanding Metadata*. National Information Standards Institute. NISO Press, 2004. Retrieved from http://www.niso.org/publications/press/UnderstandingMetadata.pdf on February 10, 2013.