A Concept Map Approach for Introduction to Computer Engineering Course Curriculum

Gul Tokdemir*, Nergiz Ercil Cagiltay***

*Computer Engineering Department, **Software Engineering Department,

Atilim University, Ankara, TURKEY

{gtokdemir, nergiz}@atilim.edu.tr

***Abstract*— As in any discipline, in Computer Engineering, students start learning the basic concepts of the discipline in their first year through an Introduction to Computer Engineering course. The topics toughed in this course can be grouped into two. The first group includes simple concepts like binary numbering system, hard disk, memory, and I/O devices. In the second group, an introduction to the courses that they will take in the next semesters of the program which includes programming, networking, software engineering, artificial intelligence and database systems. The main objective of this course is to give an introduction about the general concepts of the field to the first year Computer Engineering students and prepare them to understand the connections between them for their future studies.**

However, students and instructors face with many problems in this course. First, because of the diversity of the concepts given in the course, it is very difficult for the students to see the big picture of the Computer Engineering domain. Similarly, it is difficult for the instructors to prepare the course content in an integrated manner at the students' level. Additionally, the perception of the theory and practice behind the hardware and software topics and their connections is not an easy task for the beginners. Moreover, the topics are mostly abstract topics, which do not allow application of any laboratory sessions. Students usually find this course difficult to understand, which decreases their motivation about the department and success of the course.

This study is established to propose a concept map approach to better visualize and discover all the connections between the concepts of Computer Engineering field which can be used in the curriculum of the programs and introductory courses of the field addressing the above problems. The proposed concept map helps to visualize the general picture of the field.

***Index Terms*—curriculum development, concept maps, computer engineering**

I. INTRODUCTION

Because of the very nature of their rapidly changing technologies, several problems have been faced with for establishing the computer engineering curriculum. The curriculum of such programs need to provide theory and practice on basic concepts of the field such as hardware, digital logics, computer organization, and architecture, programming languages, operating systems, computer networks, database systems and data structures. As declared by Nisan [1], the overall interactions among hardware, software, compilers and operating systems used to be simple and transparent enough for understanding computer systems. However the modern computer technologies have become increasingly more complex which makes it very difficult to understand the whole system of the computers [1]. Accordingly, this complex structure puts pressure on the designers of introductory courses [2] and curriculum developers of such programs. Main problems addressed in the literature can be summarized as below:

- Until their junior or senior year, potential majors do not find what the major is really about [3].
- Many students often assume that computer science education is not about ideas and creativity, but is about learning technology and syntax of programming languages [4].
- Non-majors believe that computer science is only about programming [3].
- Introductory computer science courses do not introduce students how computer scientists address important problems of the field [3].
- The curriculum of these programs provide the theory and practice on main concepts of the field however it is hard to build connections among different components of the computer systems.

Accordingly, two main problems can be addressed from these views. Firstly, for the introductory courses, it is hard to organize the course for better providing the relationships among concepts of computer engineering programs which would make students lose their interest and motivation at their first year. Secondly, while developing a curriculum of such programs it is hard to show the relationships among courses and additionally build a balance on different dimensions of the programs. Since the field of the computer engineering getting larger by including many hardware and software issues as well as the human factors, the curriculum of such programs may vary on the ratio of distribution of different tracks of the field. In that case, defining different dimensions of the field becomes a necessity. For example, for balancing the programming courses and hardware courses ratio based on the general vision of the whole program sometimes need to be

978-1-4244-6571-2/10/$26.00 ©2010 IEEE

considered. This helps the students as well as the program developers to see how well the program fits the expectations.

On the other hand, concept maps are the tools used to build relationships among concepts. These tools have been used in educational environments to better connect the relationships among theory and practice as well as among other concepts covered in a course. These tools also help the learners build relationships between previous knowledge and newly introduced concepts, encouraging meaningful learning rather than rote learning (memorizing concepts, no relationship to previous learning) [5].

In our university, an introductory course has been offered for the computer engineering students. The problems discussed above are all faced in this course and the computer engineering program as well. Accordingly, in this study to address these problems a concept map is developed which builds relationships among the tracks in the field of computer engineering. Main objective of this concept map was three folded:

- to build connections among the concepts given in the introductory course of the field,
- to provide a general idea about the general structure of the program,
- to build the connections among the courses in the computer engineering curriculum.

This study describes the introductory course and the curriculum of the program. Then, it provides the proposed concept map for describing general tracks and dimensions of the field. For the development of the concept map, a new approach called "Goal-Question-Concept" is applied. Lastly, it shows the relationships among the introductory course as well as the curriculum of the program. We believe that, this concept map can be used as a base to build connections about different subjects given in the introductory course and can be introduced in each course to address the relationship between the content of the course and the whole program. This would help the educators to better show the connections between theory and practice issued introduced in a specific course and their connections with the field and prepare course content according to the concept map interactions. Also it would help the students to better build the connections among different concepts of the whole computer engineering program.

II. Course Description

The course "Introduction to Computer Engineering", is taught as the first course for Computer Engineering, and Software Engineering majors. Students with diverse backgrounds and different expectations are introduced to the basic concepts of the domain. Some of them know some coding and think computer engineering is just coding, and some do not have any idea about the domain and feel very insecure and scared. The class is not homogeneous which makes the things more difficult for the instructor.

The main objective is to teach general computer engineering and engineering concepts together with programming fundamentals. At the completion of this course, students are expected to:

- Discuss computer data representation and basic computer operations
- Appreciate the use of machine language
- Describe basic components of a computer system
- Acquire basics of problem solving and programming
- Have a general knowledge on different aspects of computer engineering

Course is planned as 14 week period, 2 hours of lectures in a class environment and 2 hours of lab sessions each week. The basic concepts of the Computer Engineering are taught in the lectures, while basic computer literacy practices like word processing, spreadsheet usage are taught at the lab sessions. The lectures include broad coverage; simple concepts like binary numbering system, hard disk, memory, and I/O devices and more advanced ones like programming languages, networking, software engineering, artificial intelligence and database systems. As a text book, we follow J.Glenn Brookshear's Computer Science: An overview book. the content of the book [8] is as follows:

- Chapter 0 Introduction
- Chapter 1 Data Storage
- Chapter 2 Data Manipulation
- Chapter 3 Operating Systems
- Chapter 4 Networking and the Internet
- Chapter 5 Algorithms
- Chapter 6 Programming Languages
- Chapter 7 Software Engineering
- Chapter 8 Data Abstractions
- Chapter 9 Database Systems
- Chapter 10 Computer Graphics
- Chapter 11 Artificial Intelligence
- Chapter 12 Theory of Computation

We start introducing algorithms, history of computation, then continue with data storage concepts which include storage of bits, main memory, mass storage, representing different type of information in computer systems, numbering systems, Boolean algebra, gates, flip-flops, machine architecture, machine language and program execution. Then concepts of operating systems, networking & Internet, software engineering, programming languages, database systems, artificial intelligence and Algorithms and C language Programming Basics are taught.

As these topics are covered in 14 weeks, each related chapter from the book is referred. However, the connection between them is not emphasized anywhere in the course which creates an unclear, complex picture in the students minds. They just perceive each topic as independent from each other and cannot elicit the correlation of them. They generally come up with the argument that the course is boring, has no use, difficult to understand. Consequently, since they cannot realize the importance of these concepts, they do not appreciate computer engineering area, get demotivated at their first semester in the department.

III. COMPUTER ENGINEERING PROGRAM

The computer Engineering program is a 4 years program and containing hardware and software issues. The courses offered by the department are described in the Appendix. The first digit of the course number indicates the year it is offered in the curriculum. TE courses are technical electives (students should choose 5 of them according to the their interests). For choosing the TE courses students face problems because they cannot relate the courses and the field applications. Therefore, they confuse to decide which course is more beneficial for their professional life. Remaining courses are the core courses that are mandatory. Additional to those courses students should take general purpose courses like math, physics, chemistry, and English.

Each course listed in the Appendix, provides subjects related to its own domain and no connection is built between the course and the other courses of the curriculum. The only course students integrate the knowledge they obtained during their 4 year study is the senior project courses, Compe 491-492, which are taught at the fourth year. Until that time, they do not have clear picture of which part of the real life problems of the area each course addresses and how they are linked together. They complain about how they will use the knowledge they gain in these courses some of which are pure theoretical courses. Consequently, they learn the subject without awareness and they generally conclude that other than the programming courses, most of courses are unnecessary for them, and they are wasting time with those subjects.

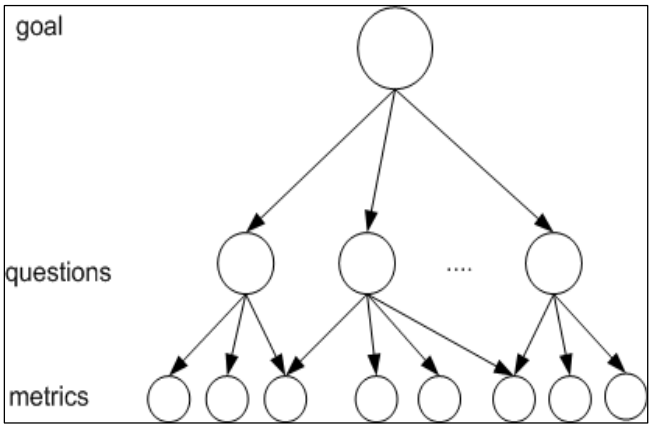
IV. PROPOSED CONCEPT MAP

We propose to use a concept map to built connections between the concepts taught in Introduction to Computer Engineering course. While preparing the concept map, we applied a new paradigm called "Goal- Question-Concept" inspired from a well-known GQM (Goal-Question-Metric) method of software engineering field. GQM was introduced to identify problems in a software process or product and define improvement goals for them for software process improvement. It builds a connection between software goals; questions to be answered for each goal and metrics as answers to the questions [6] as illustrated in Figure 1.

The ultimate goal of a computer system is to serve users by supporting them to improve their performance in their daily lives.

In order to reach this goal we have to answer the following questions:

1. What type of requirements do the users have?
2. What type of system parts needed to be developed based on these requirements?
3. What are the relations between these system parts
4. How these parts should be integrated to create the required system?



5. How this system will serve users to support their tasks?

Figure 1. GQM Technique [7]

The next step is to identify and categorize the concepts related to each question, which should be placed in the concept map. Accordingly, we came up with the top level concepts that should be considered as answers to these questions. These top level concepts can be decomposed to lower levels to reveal the partially detailed concept maps.

For the top level concept map, for answering the first question, we categorized the user requirements as shown in Figure 2.

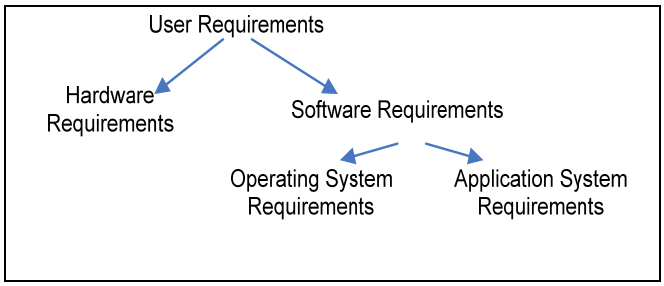


Figure 2. Requirements decomposition

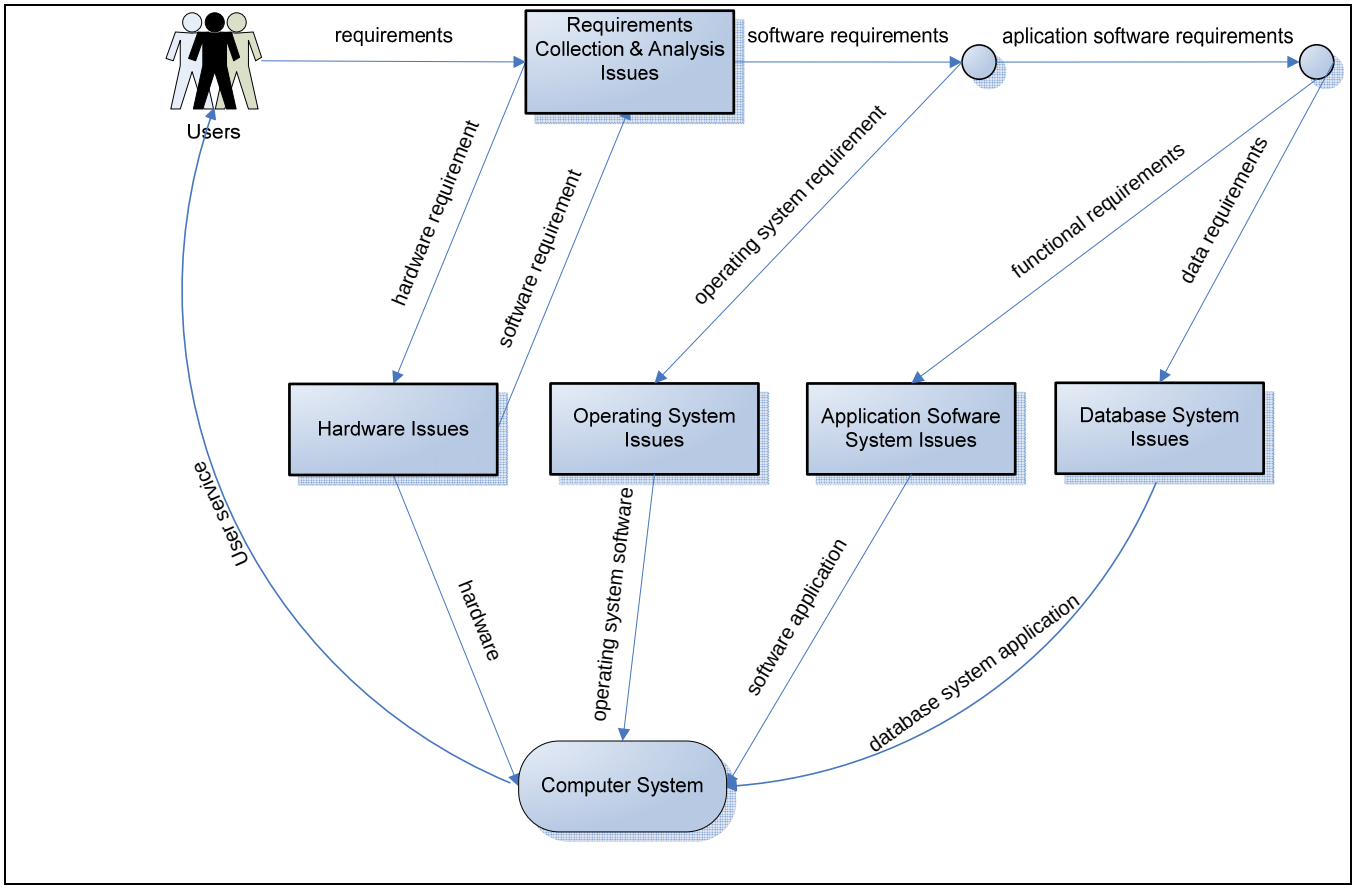


Figure 3. Concept map for Introduction to Computer Engineering course

These requirements triggers the system parts needed to be developed which will be the answer of the second question. We defined these system parts related to the following issues:

- Hardware Systems
- Operating Systems
- Application Software Systems
- Database Systems

We built the connections between those system parts related to system requirements as shown in Figure 3. The integration of those parts will establish the computer system, which is expected to serve the user requests.

Technology, all the theory and developments produced in this area is for the human beings. Main purpose is to provide support for them to work easily, quickly and effectively. Therefore, when the students, they are expected to provide solutions for computer systems to provide support to users. Additionally, they can work in academy to develop new theories, methods, and tools for this field. Accordingly, *user requirements* drive the studies in this field. Hardware requirements trigger new theories, technologies, methods, and products in this field. For this purpose, students should understand the *hardware system issues*. Therefore, in the curriculum of the computer engineering programs, we have several courses addressing main issues of the hardware type studies in this field. Accordingly, in the introduction to computer engineering course the basic hardware concepts are summarized. On the other hand, software requirements may be about the application software or operating software. In order to understand these concepts, one should understand how the application software and operating system software issues are handled. In the concept map, *operating system issues* and *software system issues* provide these concepts. These cover a broad range of design, development, and implementation, management of these systems, and their theories and methods.

The application software requirements are characterized as data and functional requirements. The data requirements triggers *database system issues*, which includes concepts like database management systems, file organization, database design, development, administration, optimization, and monitoring. Functional requirements generate *Software System issues* which cover a wide range of topics like software design, development, testing and management.

All the theories, products, and technologies related to the above-mentioned issues are integrated to construct a *computer system* to better serve for the end users. Based on the requirements, the end user can be able to get services from the computer systems.

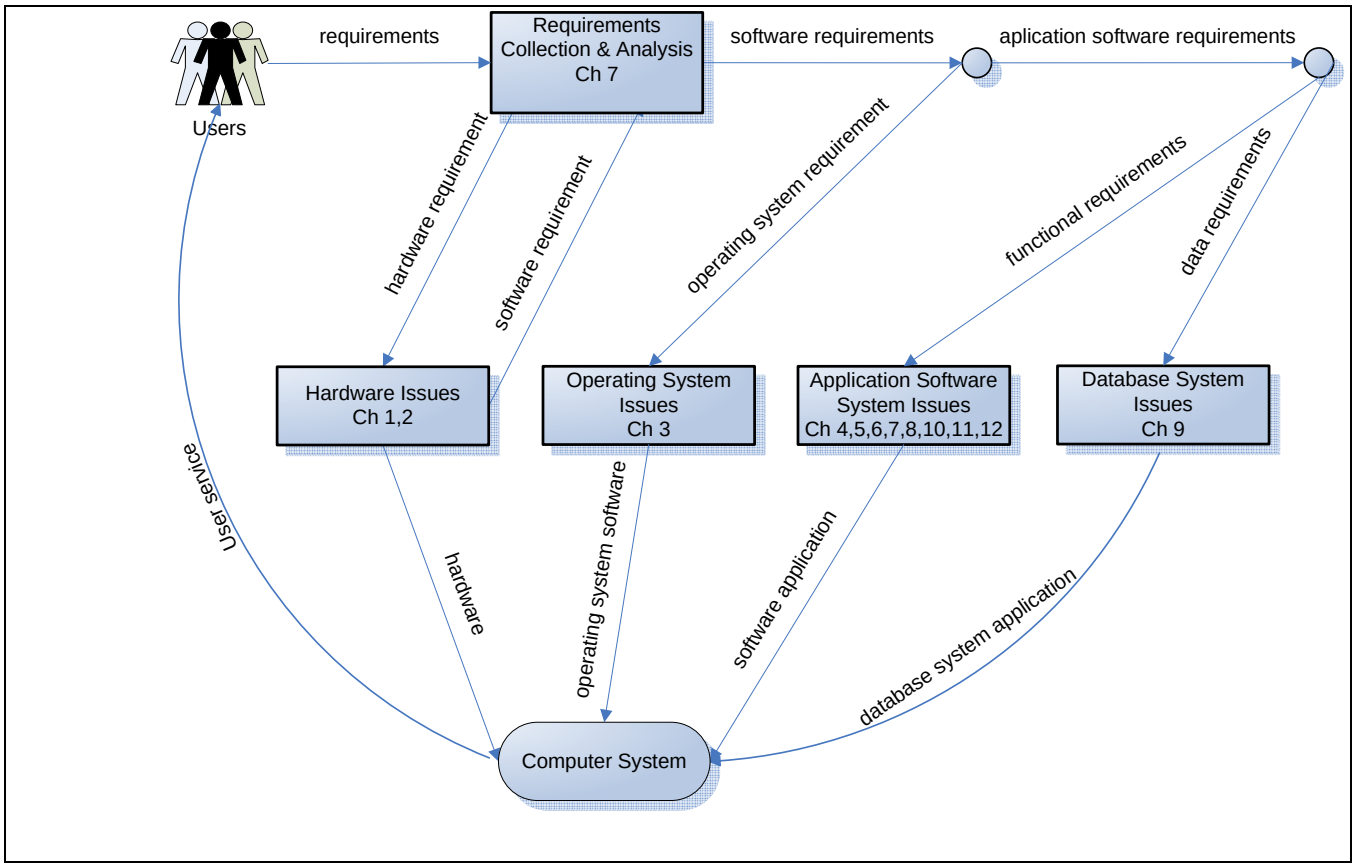


Figure 4. Concept Map- course content match

After we have developed the concept map for the course, we have mapped the chapters covered in the course with the concepts stated in the concept map as described in Figure 4. This shows that every topic covered in the course has a corresponding item in the concept map, which expresses the need for a computer system in the real world.

Describing the concept map to students at each step of the course would make them visualize the big picture and recognize the connection of different concepts they will learn in detail in their curriculum. this way, they can link their previous knowledge, with the new subject which will create a meaningful learning for them.

Additionally, we mapped the concept map with computer engineering curriculum as illustrated in Figure 5. This figure has two important results.

First, it reveals the distribution of the courses in the curriculum based on real world system requirements.

We believe that, the computer engineering issues covered in the curriculum should give insights to the students and make them realize the need for those systems in real world that motivates usage or development of computer systems in the real-life processes. They should be able to map the practical and theoretical aspects of the concepts they learn, to the real world entities where these concepts are employed.

Secondly, this map provides valuable measure to assess how much the program mission is reflected in the curriculum as well as detecting the related problems and improvements of the curriculum.

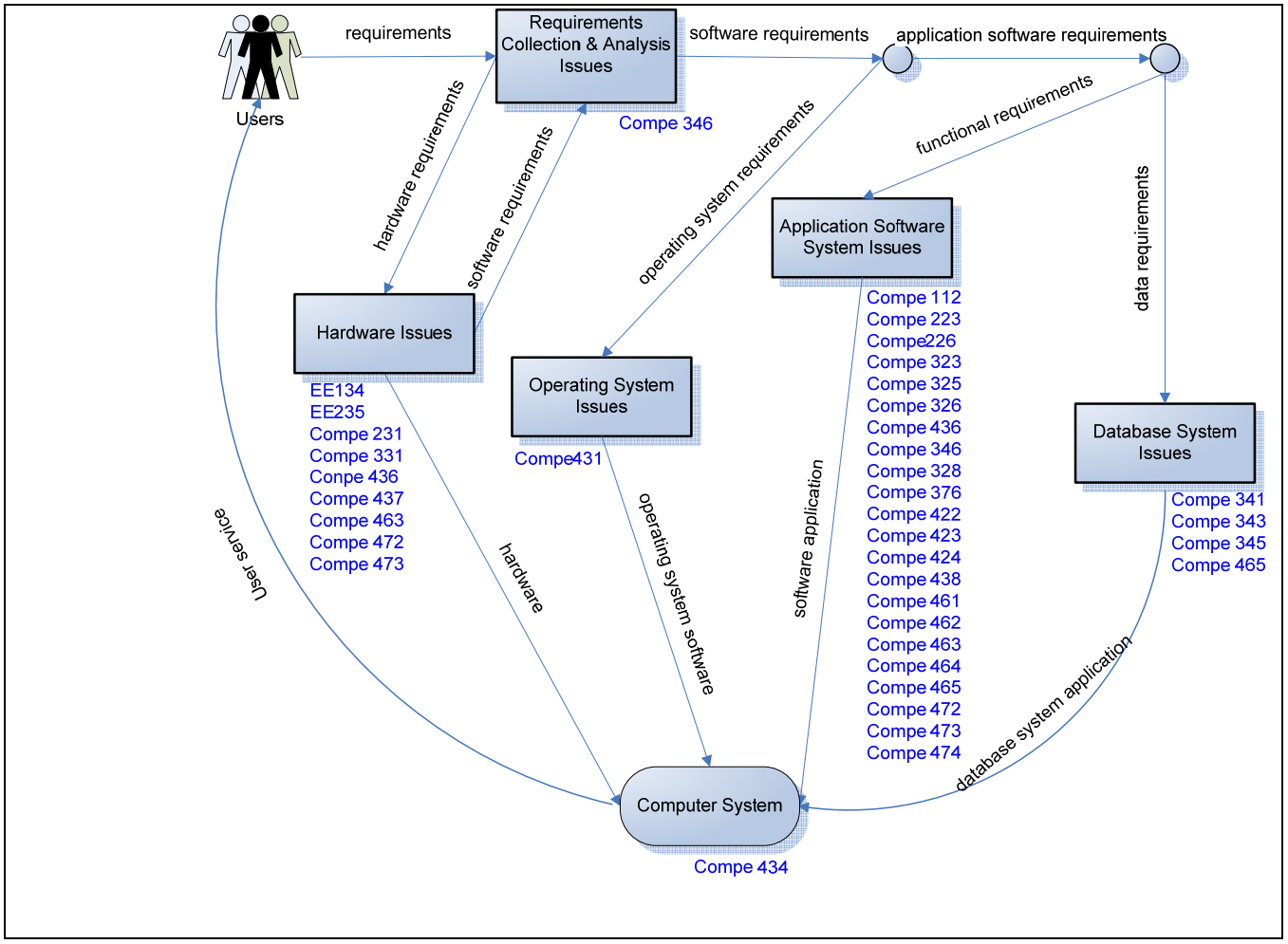


Figure 5. Concept Map –curriculum match

V. DISCUSSIONS AND CONCLUSIONS

In this study, we have addresses two main problems of the field of Computer Engineering education: complexity of the introduction to Computer Engineering courses and the Computer Engineering curriculum. to address these problems a concept map approach is employed. For development of the concept map an ovelap proach "Goal-Qestion-Concept" method has been applied.

The proposed concept maps possibly have several benefits for the Computer Engineering education programs. First, they can address the concepts of Introduction to Computer Engineering course which provides a clear view of the field for freshmans. Secondly, the concept map and curriculum match diagram helps students to address each course of their curriculum, with their conceptual view provided for the introduction course. this helps them to relate their general knowledge of the domain with the courses they will take.

Additionally, lower level concept maps can be prepared for each course and the same concept map and curriculum match diagrams can be applied to the lower level, detailed course contents. This way, students may master the details of the field without getting lost in the complexity of the field.

We believe that, if the students are introduced with these concept maps at the beginning of each course, addressing the position of that course in the concept map, they will build connections between the topics of the area effectively.

As a future study, these concept maps should be decomposed and detailed to lower levels and applied for each course. The benefits of this new approach for the computer engineering programs should be evaluated pedagogically. We believe this approach can be applied to any discipline.

REFERENCES

[1] N. Nisan, S. Schocken, *The elements of Computing Systems, Building a Modern Computer from First Principles*, MIT Press, 2005.

[2] E. Roberts, "The Dream of a Com mon Language: T he Se arch for Simplicity and Stability in Computer Science Education", *SIGCSE'04*, March 2004, Virginia, USA.

[3] E.H. Turner, R.M. Turner, "Teaching Entering Students to Think Like Computer Scientists", *SIGCSE'05*, February, 2005, Mossouri, USA.

[4] K.J. Goldman, "A Concepts-First Introduction to Computer Sceince", *SIGCSE'04*, Vrginia, USA.

[5] K. C. Gupta, R. Ramadoss, and H. Zhang, "Concept mapping and concept modules for Web-based and CD-ROM-based RF and microwave education," IEEE Trans. Microwave Theory Tech., vol. 51, Mar. 2003, pp. 1306–1311.

[6] Mendonça, M. G. & Basili, V. R. "Validation of an Approach for Improving Existing Measurement Frameworks", IEEE Transactions on Software Engineering,2000. 26(6).

[7] Dumke, GQM Method, accesed from http://ivs.cs.uni-magdeburg.de/sw-eng/us/java/GQM/link1.shtml, in 2007.

[8] J.G. Brookshear, *Computer Science: An overview*, Addison-Wesley; 10th edition.

APPENDIX

Curriculum of Computer Engineering Department

COMPE 111-Intro. Computer Engineering
COMPE 112-Computer Programming in C
CHEM 102 - General Chemistry
E 101 - Engineering Fundamentals
ENG 111 - Introduction to Communication Skills
ENG 113 - Academic Listening and Note-taking
MATH 151 - Calculus I
PHYS 101 - General Physics I
EE 134 - Circuit Analysis
ENG 104 - Communication Skills II
MATH 152 - Calculus II
PHYS 102 - General Physics II
EE 235 - Digital Electronics
ENG 211 - Communication Skills III
MATH 275 - Linear Algebra
ENG 212 - Technical Report Writing&Communication
IE 220 - Probability and Statistics
MATH 276 - Differential Equations
TURK 101 - Turkish Language I
COMPE 251- Discrete Computational Structures
COMPE 231- Digital Circuits and Systems
COMPE 223- Object Oriented Programming
COMPE 236- Intro. to Microprocessors&Microcontrollers
COMPE 226- Data Structures
IE 305 - Engineering Economic Analysis
TURK 102 - Turkish Language II
HIST 101 - Principles of Atatürk and the History of Turkish Revolution I
HIST 102 - Principles of Atatürk &History of Turkish Revolution II
ORY 400 - Social and Cultural Activites
COMPE 341- Database Design and Management
COMPE 331- Computer Architecture and Organization
COMPE 325- Study of Programming Languages
COMPE 323- Algorithms
COMPE 350- Numerical Methods
COMPE 346- Software Engineering
COMPE 326- Formal Languages And Automata
COMPE399- Summer Practice I
COMPE 499 - Summer Practice II
COMPE 491- Senior Project I
COMPE 431- Operating Systems
COMPE 492- Senior Project II
COMPE 436- Data Communications & Networks
COMPE 328- Object-Oriented Analysis and Design(TE)
COMPE 343- Database Systems & Programming (TE)
COMPE 345- Data Warehousing&Business Intelligence(TE)
COMPE 376- Computer Games and Simulation (TE)
COMPE 422- Visual Programming (TE)
COMPE 423- Logic Programming (TE)
COMPE 424- Language Processors (TE)
COMPE 434- Embedded System Design (TE)
COMPE 437- VLSI Design (TE)
COMPE 438- Java Programming(TE)
COMPE 461- Applied Neural Computing (TE)
COMPE 462- Artificial Intelligence (TE)
COMPE 463- Digital Signal Processing (TE)
COMPE 464- Pattern Recognition&Image Processing (TE)
COMPE 465- Knowledge Engineering (TE)
COMPE 472- Parallel Computing (TE)
COMPE 473- Computer Graphics (TE)
COMPE 474- Soft Computing (TE)