Fall 2007

Interactive and dialogue based learning in engineering education

Siddartha Thummuri

INTERACTIVE AND DIALOGUE BASED LEARNING IN ENGINEERING EDUCATION

by

SIDDARTHA THUMMURI

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ENGINEERING MANAGEMENT

2007

Approved by

Ray Luechtefeld, Advisor	Donald Myers

Stephen Raper

PUBLICATION THESIS OPTION

This thesis consists of the following three articles that have been submitted or intended to submit for publication as follows:

Pages 3-13 are submitted to the DISTANCE TEACHING AND LEARNING CONFERENCE Proceedings.

Pages 14-39 are intended for submission to JOURNAL OF ENGINEERING EDUCATION.

Pages 40-59 are intended for submission to the IEEE TRANSACTIONS ON PROFESSIONAL COMMUNICATION JOURNAL.

ABSTRACT

ABET has established a set of “professional” skills, that describe the attributes of graduates from engineering programs. These “professional” skills include effective communication, teamwork, understanding ethics and professionalism and knowledge of contemporary issues. But, how these skills are taught or rather learned and assessed is not readily apparent. This issue is elevated in a distance learning environment. Distance learning differs from conventional learning primarily in the isolation felt by and greater self- discipline required of its students. Because of these characteristics, ensuring that distance learning systems provide adequate support to, and interaction with, students is crucial. Researchers have provided suggestions to redesign classes to ensure learning of these skills. One method to apply these techniques would be to have a “practical training” under professional guidance while incorporating student interaction through cooperative learning approaches. This thesis focuses on a class room architecture that provides high levels of interactivity which enhances the students’ learning environment both in distance and on-campus contexts and forms a basis for practical learning. Two analyses are presented in this thesis, the first one being a simple measure of levels of interactivity in this course architecture, based on rubric and the second one, a comparative analysis, to demonstrate the effectiveness of interactive (multi-directional information flow) classrooms as opposed to conventional (unidirectional information flow) classrooms and that this course architecture is ideal for enhancement of distance learning and an excellent route to professional skills specified by ABET.

ACKNOWLEDGMENTS

I am extremely grateful to my advisor, Dr. Ray Luechtefeld for the encouragement and guidance he has given me and the extreme patience he has shown in my research work. He has also given me sufficient freedom to explore avenues of research while correcting my course and guiding me at all times.

I thank Dr. Myers and Dr. Stephen Raper, my committee members, for the help and support they have provided throughout my Masters' degree program.

A special thanks to my fiancée Kirthi Deshpande, for the help and support she has provided to me. She stood by me at all times and helped me in all ways she can. Without her, this success would not have been possible.

I should especially mention Rajesh Chatrathy, Santhosh Muthyapu, Anil Jade for the help they provided me in Programming and Raj Kanwar Singh and Paul, for helping me with the analysis. Without their help, this effort and its successful completion would not have been possible.

On a personal note, I thank my roommates at Rolla and the Indian community here who have been supportive in all my ventures.

Last, but, at the top of my list, I thank my parents Ram Mohan Rao and G. Usha Rani, my sister Spandana and my brother-in-law Rajnikanth Vanam, for the tremendous encouragement and support I have received from them throughout my life which has enabled me to face the challenges and achieve success.

TABLE OF CONTENTS

Page

LIST OF FIGURES

PAPER-1 Page

PAPER-2

PAPER-3

LIST OF TABLES

PAPER-1 Page

PAPER-2

INTRODUCTION

Distance education is becoming increasingly popular due to economic forces and new technologies. This growing popularity is challenging educational systems to provide increased educational opportunities while maintaining (or even reducing) budgets. As an answer to this challenge many universities now offer many varieties of ‘distance education’, ranging from one way technologies like print based correspondence courses to interactive technologies like computer and video conferencing, with the later becoming increasingly popular. The challenge is to make these courses as effective as face-to-face education. Moreover, The 2000 ABET engineering criteria included a set of “professional skills,” apart from the hard engineering skills like ability to apply scientific and mathematical skills, ability to design experiments, system and components, to analyze and interpret data and to formulate and solve engineering problems, which are among the most important initiatives to impact engineering education in the past fifteen years.” While it is relatively easier to teach the technical skills, it can be difficult to teach the second set of professional skills. Doing so calls for course designs that create a learning environment to deliver both technical and professional skills in both on-campus and distance learning environments. In this context, researchers have proposed several course designs that could integrate these skills. Despite these developments in course design for delivering professional skills, their effectiveness is questionable. The first part of this thesis focuses on how this might be accomplished by providing a model for a highly interactive course. This model aims at providing a better environment for attaining the “professional skills and also to enhance distance learning. Arguments are made supporting the effectiveness of interactive learning and these arguments are backed by two kinds of analyses. The results of the analysis support that, the course architecture that is discussed in this thesis provides an excellent environment for interactive learning and also that, it enhances distance learning.

The second part of this thesis focuses on methods to enhance the project management process. Increasing complexity and high stakes in business projects call for new project management systems which can handle those complexities and be able to offer an efficient change management feature. Traditional project management tools do not take into account some sources of change, task interactions and the necessity for distributed planning. Today's projects involve huge amounts of data and information. This makes it necessary to have a robust and intelligent system that can handle and transfer information on time and to people who are responsible or affected by that information. The paper presents a software tool (for project management) that integrates several components related to project management. This tool attempts to provide novel features like efficient change management, time saving, reduced errors, automatic updating of information and a speech interface for ease of use.

PAPER - 1

CLASSROOM AS LEARNING ORGANIZATION:

A COURSE STRUCTURE TO ENHANCE INTERACTIVE DISTANCE LEARNING

Siddartha Thummuri
Research Assistant
University of Missouri-Rolla, Missouri, U.S.A

Dr. Ray Luechtefeld
Assistant Professor
University of Missouri – Rolla, Missouri, U.S.A

ABSTRACT

As distance education extends its reach and uses new delivery tools, concerns about its effectiveness increase. Distance learning differs from conventional learning primarily in the isolation felt by and greater self- discipline required of its students. Because of these characteristics, ensuring that distance learning systems provide adequate support to, and interaction with, students is crucial. But, how distance learning may provide a greater degree of interaction is not entirely clear. To become effective in a distance environment, instructors of distance courses must alter both course design and teaching strategies to leverage technologies and assure maximum interaction. This paper presents a unique and novel classroom architecture for a course called "Management for Engineers-XB (eXperience Based)" offered at the University of Missouri-Rolla. This course is primarily targeted to resolve interaction issues and is designed to ensure interaction among the students that reduce isolation and disconnectedness. This course provides an excellent environment for "mutual learning" as illustrated by examples from sections of the class.

KEYWORDS: Distance learning, Learner interaction, effectiveness, Mutual learning

1. INTRODUCTION

Distance education is becoming increasingly popular due to economic forces and new technologies. This growing popularity is challenging educational systems to provide increased educational opportunities while maintaining (or even reducing) budgets. As an answer to this challenge many universities now offer many varieties of ‘distance education’, ranging from one way technologies like print based correspondence courses to interactive technologies like computer and video conferencing, with the later becoming increasingly popular. The challenge is to make these courses as effective as face-to-face education. This paper focuses on how this might be accomplished by providing a model for a highly interactive distance course.

2. EFFECTIVENESS OF DISTANCE EDUCATION

Some researchers consider distance education courses could potentially be as effective as traditional classrooms. “Well-designed distance education programs are equally effective in terms of learner outcomes with resident instruction, in general, and produce superior learning outcomes in specific applications" (Kelly, 1993, p. 76). This can occur if the methods and technologies used are 1) appropriate to the instructional tasks, 2) there is student-to-student interaction, and 3) when there is timely teacher-to-student feedback (Moore & Thompson, 1990).

The design of distance education delivery methods are of critical importance. While many factors, such as the use of technology, course structure, content delivery method, etc., are important, the extent of interaction designed into a distance course is, perhaps, one of the most important. High levels of interaction tend to reduce problems of student isolation and can create a "mutual learning" environment as well as enhancing levels of personal feedback to the students. While personal feedback is not practical in a lecture setting, highly interactive courses can increase student to student feedback. Moore (1989) categorized classroom interaction into three types: learner-learner, learner-instructor and learner-interface. The prime focus of this paper is the student-student and student-instructor interactions, especially including the student-student (distant student-in class student in this case) interactions.

3. "XB" AS AN INTERACTIVE COURSE EXEMPLAR

The course called "Management for Engineers- XB (eXperience Based)" (known simply as "XB") offered at the University of Missouri-Rolla follows a model originally developed for on-campus students by Dr. Roger Putzel, of St. Michaels College [Putzel, 1992]. The unique design of the course content delivery methods and course structure improve the learning environment for the mixed (distance and on-campus students) class by enhancing student-student interactions. This paper presents an exemplar for a unique course structure that can enhance a classroom by implementing an interactive learning environment, which is quite different than a traditional classroom. The interactive classroom described provides a multidirectional flow of information, compared to traditional classrooms that have a largely one-way flow of information and knowledge.

4. ABOUT XB

While the course entitled “Management for Engineers” is a core Senior/Graduate level course offered by several faculty members in the department of Engineering Management and Systems Engineering at the University of Missouri-Rolla, not all distance education instructors use the highly interactive format called “XB”. The sections offered as “XB” have significant levels of enrollment ranging from around twelve to forty students per semester. Students enrolled in XB include senior undergraduates, Master’s and doctoral students, as well as working professionals. Generally, there are almost equal numbers of distance (generally, consisting of working professionals) and on-campus students. This adds the additional dilemma of delivering course contents in a manner that provides meaningful learning to students at a variety of levels. The course is designed to ensure that all students (especially those at a distance) can participate actively at their skill level. For this reason interactive learning is central to the course structure. Students are strongly required to interact with one another throughout the class sessions.

5. THE CLASSROOM SETTING

Enrollment typically, consists of almost equal numbers of distance and on-campus students. The course is presented via an audio-conference where the instructor, the on-campus and the distance students can interact with each other at all times using phone conferencing and push-to-talk microphones. While the course uses Blackboard® for online discussions and document exchange, Webex ® is the primary conferencing tool for the classroom. The distance students can view a live video of the class where the camera is focused on whoever in the classroom may be speaking at the moment, be it the instructor, the presenter or another student. In addition to the audio and one-way video communications during class sessions, the students are provided with a chat room, where they can send messages to the entire class or even send a private message to a selected student or the instructor. Students also communicate with one another outside the classroom. This increases the interactivity among the students. Most importantly, the instructor is available outside the classroom on the phone or for personal meetings during normal university working hours. Archived videos of class sessions are also available.

6. COURSE STRUCTURE AND CONTENT DELIVERY

The main course objective is to provide students with a broad understanding of principles, processes, and practices of management in contemporary organizations. The course structure is designed to fulfill this objective by setting up the class as a real organization, centered on the completion of more than 200 specific learning objectives, using a peer-education framework. Every student is evaluated through a system of peer evaluation and objective completion designed to enhance practical learning. The basis for the design is simple- the more the students participate, the more they interact with one another and the more they make mistakes and reflect on them, the more effectively mutual learning occurs.

Every class session (framed as a “meeting” rather than a “class”) is facilitated (“led without creating unnecessary dependency”) by the Senior Manager (the instructor). The Senior Manager coaches and advises class members, acting more like a Chairman of the Board than a CEO.

The class is organized into departments and groups, each bearing different administrative and learning responsibilities, as illustrated in Figure-1.

Students choose among the departments and groups based on their interests and expertise, as long as all responsibilities are covered. So, for example, students may negotiate among themselves to ensure that all positions are filled. This resembles a real organization, where an employee is able to choose a position based on his interests, while at the same time limited by certain boundaries.

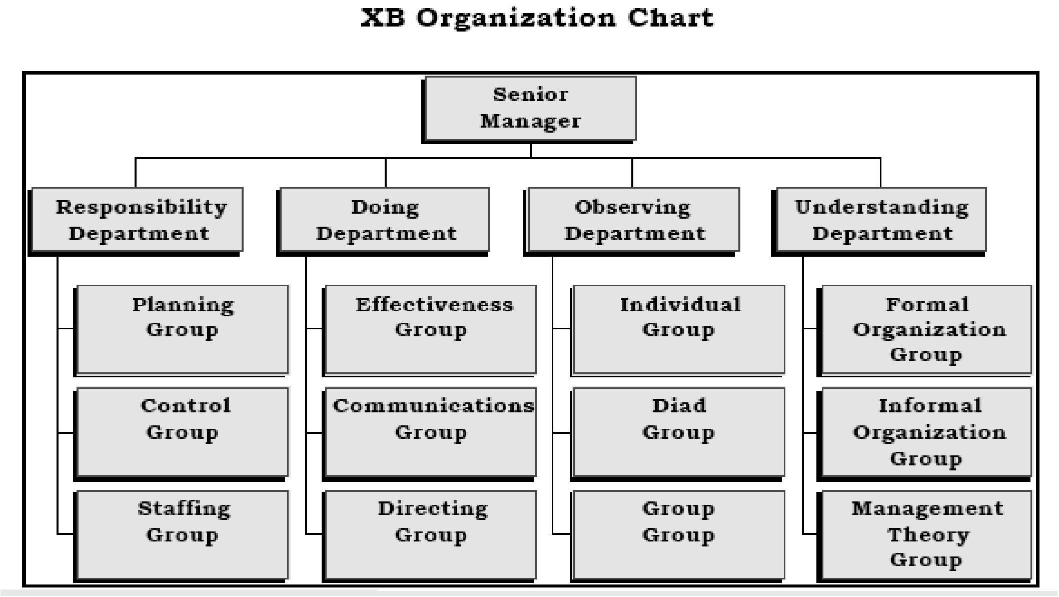


Figure-1: XB Organization Chart

The "XB Manual" (Putzel, 2006) provides a detailed description of the duties and responsibilities of each department and group. Class sessions generally, contain presentations by students to these ends. Decisions about when and what to present are left to the students. Every meeting also has sufficient time for student-led discussions about topics of interest to the students. These discussions include a wide range of topics, from the design of the peer evaluation system to the participation and performance of different groups.

Typically, the first class of the semester begins with a brief overview of the course, led by the instructor. However, on the second day it becomes apparent that this is not a typical course when the instructor sits at the back of the room and does not step in to direct class activities. This initially leads to consternation among the class, but, leadership emerges, often from among the off-campus students, who generally, have more professional experience than the on-campus students. Students are usually quite confused by the course structure at the beginning, but, within the broad outlines of the XB manual they are able to form their own rules regarding the class. The learning process is most effective when the students practice while they learn. For this reason, most of the design elements of the course are similar to functioning organizational systems. For example, at the organization level, the instructor provides policies (the XB manual), measurement systems (ranking systems and learning measures), and tasks (administrative and learning objectives). Students are free to do whatever they want, within the limits of the course.

7. MEASURES OF INTERACTIVITY

Student perceptions of the degree of interaction between students and the instructor and among students play a primary role in determinations of course quality. A significant problem faced by many students in today’s distance courses is a high level of isolation. The difficulty of getting attention from the instructor and other students may lead to passivity and a lack of interest in the course. Horn (1994) and Hirumi and

Bermudez (1996) found that, with proper instructional design, distance courses can actually be more interactive than traditional ones, providing more personal and timely feedback than is possible in large, face-to-face courses. Due to time pressures, it can be impractical for the instructor to give one-on-one feedback to every student. For this reason, distance courses may leave student feeling dissatisfied. Increasing levels of interaction can help students receive greater feedback from other students as well as the instructor.

Levels of interactivity are measured by a rubric proposed by Roblyer and Ekhaml [2000]. It evaluates levels of interactivity in the following four dimensions:

1) Social Goals of Interaction: rapport and collaboration among students

2) Instructional Goals of Interaction: interaction serves important functions in learning

3) Types and Uses of Technologies: technologies that can be used to facilitate interaction. (Hughes and Hewson, 1998)

4) Impact of Interactivity-Changes in Learner Behaviors: students respond to effectively (or ineffectively) designed distance courses with observable behaviors (McHenry and Bozik, 1997)

8. THE PROCEDURE

Class sessions from several semesters of “XB” were analyzed. Two independent coders watched archived class videos of these semesters and qualitatively coded the class sessions. After coding the archived videos, their results were compared and averaged to get a final measure of course interactivity. Results are in Table-I.

Measures of Interactivity in XB

Semester	Fall 2005 (C1)	Summer 2006 (C2)	Fall 2006 (C1)	Spring 2007 (C2)
Total No of students	19	22	18	11
No of On-campus students	13	9	8	7
No of Distance students	6	13	10	4
Points for Elements 1-4	4, 4, 4, 5	2, 5, 4, 3	4, 4, 4, 4	2, 5, 4, 4
Total No of Points	17	14	16	15
Comments	High interactive qualities	Moderate interactive qualities	High Interactive Qualities	High interactive Qualities

* C1- Archived videos evaluated by Coder-1, C2- Archived videos evaluated by Coder-2

Table-I: Measures of Interactivity in XB

9. CONCLUSION

This paper discusses a unique classroom architecture aimed at maximizing interaction. Analysis supports the high interactive qualities of this class. The results found infer that there is good deal of rapport among the students, use of advanced delivery tools overcomes communication barriers, timely feedbacks from peers and instructor and interactivity on voluntary basis in over 70 percent of the students. The scores also suggest improvement in activities improving rapport among students and also in the interactivity levels of the students, but, the later are limited by an initiative from the students end. This course architecture resolves the isolation issue as well. Elements of “mutual learning” make it a better learning environment. The arguments made in the paper are supported by snippets of videos from previous semesters.

10. REFERENCES

Cavanaugh, Catherine, (2001). *The Effectiveness of Interactive Distance Education Technologies in K-12 Learning: A Meta-Analysis*. Jacksonville, FL: International Jl. of Educational Telecommunications (2001) **7**(1), 73-88.

Gilbert, L., & Moore, D. R. (1998). *Building interactivity into web courses: Tools for social and instructional interaction*. Educational Technology, 38(3), 29-35.

Hirumi, A., & Bermudez, A. (1996). *Interactivity, distance education, and instructional systems design converge on the information superhighway*. Journal of Research on Computing in Education, 29(1), 1-16.

Horn, D. (1994). *Distance education: Is interactivity compromised?* Performance and Instruction*, 33*(9), 12-15.

Hughes, C., & Hewson, L. (1998). *Online interactions: Developing a neglected aspect of the virtual classroom.* Educational Technology*, 38*(4), 48-55.

Kelly, W. (1993). *Interview*. *Speaking personally with William J.Kelly*. The American Journal of Distance Education 7(1): 74-82.

Klesius, J., Homan, S., & Thompson, T. (1997). *Distance education compared to traditional instruction: The students' view.* International Journal of Instructional Media, 24(3), 207-220.

Leigh, E.; Luechtefeld, R; & Nelson, E (2007). *Out of Chaos comes Learning - student and staff perspectives on co-creating process-management oriented workplaces in classrooms*. In State of the Art: Enterprise and Business Management: A Handbook for Educators, Consulters, and Practitioners. Altepkin Erkollar, Editor. Education, Training, Consulting, and Process Management: Sydney, Australia.

McHenry, L., & Bozik, M. (1997). *From a distance: Student voices from the interactive video classroom. TechTrends, 42(*6), 20-24.

Moore, M. G. (1989). *Three types of interaction*. The American Journal of Distance Education, *3*(2), 1-6.

Moore M.G. and M. Thompson (1990). *The Effects of Distance Learning: A Summary of Literature*. University Park, Pa: The American Center for the Study of Distance Education.

Putzel, Roger (1992). Experience Base Learning: A Classroom-as-Organization Using Delegated, Rank-Order Grading. Journal of Management Education 5(16), 204-219.

Roblyer M. D. & Ekhaml Leticia, (2000)**.** *How Interactive are YOUR Distance Courses? A Rubric for Assessing Interaction in Distance Learning.* Callaway, Georgia: DLA 2000 proceedings.

Rubash Arlyn R., (2004). *Designing Distance Delivery Courses.* retrieved November, 2004 from http://thejournal.com/articles/17078.

Zirkin, B. & Sumler, D. (1995). *Interactive or non-interactive? That is the question! An annotated bibliography*. Journal of Distance Education, 10(1), 95-112.

Thurmond Veronica & Wambach Karen. *Understanding Interactions in Distance Education: A Review of the Literature.*Internation Jl of Instructional Technology & Distance learning

ABOUT THE AUTHORS

Siddartha Thummuri is pursuing his M.S. in Engineering Management and he received his B.S. in Chemical Engineering in 2004 from the Osmania University, Hyderabad (INDIA). His research interests include Effective Communications in organizations and Enhancements to Project Management techniques. He is currently a Research Assistant to Dr. Ray Luechtefeld at University of Missouri-Rolla (U.S.A).

Ray A. Luechtefeld received his Ph.D. from Boston College in Organization Studies. He holds an MBA degree from the University of Minnesota and a B.S. degree in Electrical Engineering from the University of Missouri - Rolla. He is currently an Assistant Professor at UMR. His research interests include dialogue within teams, group and organizational learning, and the improvement of quality practices through communications improvements.

PAPER-2
INTERACTIVE LEARNING AS A ROUTE TO “PROFESSIONAL SKILLS”: A COMPARATIVE ANALYSIS

Siddartha Thummuri
Department of Engineering Management and Systems Engineering
University of Missouri-Rolla

Dr. Ray Luechtefeld
Department of Engineering Management and Systems Engineering
University of Missouri-Rolla

ABSTRACT

ABET has established a set of “professional” skills, that describe the attributes of graduates from engineering programs. These “professional” skills include effective communication, teamwork, understanding ethics and professionalism and knowledge of contemporary issues. But, how these skills are taught or rather learned and assessed is not readily apparent. Researchers have provided suggestions to redesign classes to ensure learning of these skills. One method to apply these techniques would be to have a “practical training” under professional guidance while incorporating student interaction through cooperative learning approaches. This paper focuses on a class room architecture that provides high levels of interactivity which enhances the students’ learning environment and forms a basis for practical learning. This paper also compares the interactive learning class environments (the “keep it flowing” model of learning) to conventional class environments (the “pour it in” model of learning). A comparative analysis is presented to demonstrate the effectiveness of interactive (multi-directional

information flow) classrooms as opposed to conventional (unidirectional information flow) classrooms.

Keywords: Interactive learning, Practice in learning

1. INTRODUCTION

ABET's accreditation criteria specifically outlined the major elements required by accredited engineering programs, including the program curricula, the faculty type, and the facilities. In the mid-1990s, the engineering community began to question the appropriateness of such rigid accreditation requirements. In 1997, ABET adopted Engineering Criteria 2000 (EC2000). The EC2000 criteria shifted the focus away from the inputs (what material is taught) and to the outputs (what students learned) [1]. EC2000 stresses continuous improvement, and accounts [1] for specific missions and goals of the individual institutions and programs. The intention of this approach was to enable innovation in engineering programs rather than forcing all programs to conform to a standard, as well as to encourage new assessment processes and program improvements

The new and revised "ABET Engineering criteria" are among the most important initiatives to impact engineering education in the past fifteen years."[5]. The 2000 ABET engineering criteria included a set of "professional skills"[5], apart from the hard engineering skills like ability to apply scientific and mathematical skills, ability to design experiments, system and components, to analyze and interpret data and to formulate and solve engineering problems. These hard engineering skills introduced by ABET are just a revision of the engineering criteria it had previously, with a few modifications to meet the contemporary needs. On the contrary, the new and fresh set of “professional” skills specified by ABET have gained a lot of focus. These professional skills include [5]:

- Ability to function on multi-disciplinary teams (3.d)
- An understanding of professional and ethical responsibility (3.f)
- An ability to communicate effectively (3.g)
- Have education necessary to understand the impact of engineering solutions in global, economic, environmental and social context (3.h)
- A recognition of the need for, and an ability to engage in life-long learning (3.i); and
- A knowledge of contemporary issues (3.j)

The skills required by the ABET EEC range from core technical to exclusively managerial and interpersonal skills. While it is relatively easier to teach the technical skills, it can be difficult to teach the second set of professional skills. Doing so calls for course designs that create a learning environment to deliver both technical and professional skills. In this context, researchers have proposed several course designs [2] [4] that could integrate these skills. Despite these developments in course design for delivering professional skills, their effectiveness is questionable [7]. There is a direct relationship between the course design and the mode of content delivery used in its design. Simply put, the efficiency of a course to deliver the required skills depends on how well the class is engaged [2].

Several modes of handling a classroom have been proposed [2] [4] [8]. The underlying concept behind these pedagogies of engagement is interaction, interactions among the students and the teacher and the interaction betweens the students themselves.

The effectiveness of a course bears a direct relationship with the level of interactivity present in that environment. A closer look at the ABET required professional skills reinforces the importance of interactivity in course designs. For example, one path to attain the “ability to function on multi disciplinary teams” criteria mentioned above, is to develop skills in the areas of inquiry and conflict resolution among the team members. This could be achieved via interactions. The more effective the team members interactions, the better they will understand each others concerns and become open to other’s viewpoints, leading to higher team performance. As a result of globalization, the enrollment of students into engineering programs abroad has grown considerably. This creates a much more complex classroom environment, making it difficult for the instructors to engage the class effectively.

Due to an increasing awareness of the above mentioned issues, many universities have been attempting to develop and implement new course architectures with an integration of problem/project based learning and technical content delivery to teach these skills [2]. Most of these programs lacks an important ingredient which is what we term practice-based learning or simply practical learning [12]. Wagner and Sternberg have generated a stream of research on tacit knowledge springing from this position. They share a perspective on tacit knowledge as providing the ability to enact skilled, practical action. Wagner (Wagner, 1987) focused on the role of tacit knowledge in permitting the execution of skilled actions. He cited the definition offered by Wagner and Sternberg (Wagner and Sternberg, 1985) as the "practical know-how that usually is not openly expressed or stated and which must be acquired in the absence of direct instruction. The only way one to learn something is by practicing it. For example, a student can learn circuit design is by designing a circuit, of course under professional guidance, but, not merely by studying about circuited design or by being instructed about it [13].

This paper presents a course design that creates a learning environment not only to teach the theories behind professional skills but also to engage students, and help them in practicing those skills in the classroom itself. An analysis of archived videos of this course compares it with the archived lectures of another section of the same course which takes a conventional approach. The results clearly indicate that the more interactive format creates an environment where there is a multi-directional information flow and a practical learning approach. These arguments are reinforced with surveys of the students from several semesters.

2. THE ESSENTIAL COMPONENTS OF COURSE DESIGNS

It is a widely accepted fact among researchers and institutions considering redesign on their courses that the basic elements of an effective course is to provide good contact among faculty and student and the students themselves, encourages active learning, gives prompt feedback, and emphasizes time on task and respects diverse talents and ways of learning [2]. Many course designs lack these basic elements [10]. However, many courses adopt project based learning as their mode of content delivery [2]. While project based learning is effective compared to the conventional class rooms, it fails to integrate the basic elements. For example, in a project based learning course, the instructor does an excellent job designing a curriculum for the class (technical content), assigns students to teams and assigns a project to them. But, rarely in such cases do the instructors provide feedback to them and monitor or assess team execution performance. The only measure they take to assess the team performance is to evaluate the outcome of the project. But, team performance is not limited to the outcome of the project. Team dynamics based issues like conflicts among team members, compatibility issues, and work load distribution issues need to be monitored and “facilitated” [14] on a regular basis. Students need constant feedback regarding how to communicate intent and function effectively in teams.

The process of course design includes three basic components; the mode of instruction, assessment of performance, and the learning outcomes achieved [10]. While the assessment techniques would not marginally affect the learning outcomes, the mode of instruction does to a larger extent [10]. Conventional class room designs include lectures and labs with perhaps some discussion sessions in their instructional technology [10]. Many faculty members with a pedagogical vision of the basic elements for a successful course design have incorporated project based-learning into their courses [4]. But, the efficacy of project based learning is still questionable [7]. This is because project based learning (PBL) classes often lack the ability to transmit “soft skills” efficiently. PBL is an efficient means of transferring technical or engineering skills. So students exposed to PBL are more aware of what technical skills they have learned, better at articulating their design process and at defining what they learnt from the product embodiment [4]. But, PBL is ineffective at helping students attain process oriented skills like effective communication, teamwork and the ability to resolve ethical dilemmas. It also lacks in helping students gain awareness in areas like understanding the impact of global and social factors, knowledge of contemporary issues and developing lifelong learning.

Project based learning often does not serve as an effective means of teaching the professional skills required by the ABET criteria. Moreover, a majority of the courses using project based learning, lack the instructor’s direct feedbacks to individual students’ performance and involvement into student team dynamics [2]. This, points out the need for alternative pedagogical features to be introduced into the instructional technology apart from the lectures, labs and project-based learning. These features would include incorporation of active and interactive learning into course designs to provide maximum interactions among the faculty and the student and the students themselves. Another important component is to provide a learning environment that would promote practice based learning [10]. This would allow students an opportunity to practice the elements that they have learnt in the class. For example many students may experience conflicts when in teams, but, learning is more likely if they experience conflict in the class and are able to resolve it under the guidance of the instructor. This type of course design would integrate Problem based learning with active or interactive learning along with practical learning [10] in “soft skills” to make it a “team facilitated-problem/project and practice based learning” environment.

Assessment methods used for evaluation are another important area of course design [10]. Conventional courses use methods like tests, assignments, project reports, etc. to test for the expertise gained by the students in engineering and technical domains, but, no attempts are made to evaluate the non-technical, that is, professional skills gained by the students [10]. Hence, the students tend to be unaware of the fact that they need to learn these skills. For example, if there is no assessment of communications skills, they may become careless of the fact that it is as important to gain those skills also, in addition to the engineering skills. Thus they tend to concentrate more towards learning the engineering skills only. This is the reason why new assessment techniques like surveys and assessment of participation in class should be incorporated into course designs [10]. While some instructors may use additional assessment methods like considering class participation as a measure to grade students, the percentage considered towards that is often small. This can send a message to students about the importance of the ABET specified skills [2].

In recent years there have been significant strides in developing assessment tools for the professional skills [10]. Schuman et al. developed a tool to assess students' ability to evaluate and resolve ethical dilemmas. However, even a positive assessment in this area would not ensure that a student can apply it in actual practice. Incorporating practice into learning is another dimension that needs to be considered while designing courses to teach professional skills.

The underlying precept in designing courses is to incorporate the four most important and missing elements; active and interactive learning, problem based learning, practice based learning and innovative assessment methods. There is also a need for a clear statement of goals and requirements by the instructor [10]. The next section reviews the importance of active and interactive learning in course designs.

2.1. WHY INTERACTIVE OR ACTIVE LEARNING?

Though the terms active learning and interactive learning have a different meaning they are used in the same context here because interactive learning (often referred to as “co-operative learning”) is one version of active learning [2]. The general goal of active learning is to “engage students in the learning process” [2]. While the term “engaging students” is a vague and open-ended statement, in a deeper sense it is intended to motivate the students and involve them to participate in the classroom. This can be misconstrued. If a professor is asked whether he or she is engaging the class well, they might respond positively, because their definition of “engaging the class” might be assigning work to students, which is then completed on time. However, engagement of classrooms has come to mean having high levels of student-teacher interaction and cooperation among students [2].

Similarly, it important to know what instructors mean when they say they use “active learning”. One commonly used definition of active learning is simply the introduction or enhancement of student activity in the classroom. However this is one end of a continuum of course design approaches, with student-managed courses anchoring the other extreme.

The effectiveness of using active learning for the acquisition of professional skills can be assessed based on the outcomes that the course has achieved in terms of what the students have gained apart from the engineering skills [7]. Proponents of active learning assert that the effectiveness of this approach depends on the student’s attention [6]. There is considerable evidence [7] to support the effectiveness of the use of active learning in promoting “professional skills”, when compared to traditional instruction methods, but, this requires a supporting course design that does not neglect the basic elements of instructional design as well as active participation and involvement from the instructors side.

Interactive learning is a subset of active learning and it has its own significance. It is synonymous to co-operative learning where the students work together towards a common goal but, are assessed individually [2]. Putting interactive learning to practice is a complex task and requires a detailed attention to the learning outcomes that need to be achieved. The major advantage of incorporating interactive learning is to keep the information in the classroom flowing in all directions as opposed to a unidirectional flow of information as in traditional approaches. At its core, interactive learning is based on the premise that co-operation among students is more effective than competition in terms of producing positive learning outcomes [2]. It provides a learning environment that is specifically designed to promote effective “professional” skills like teamwork. From the student’s side, interactive learning provides a more liberal environment in the classroom, of course bounded by the course limits, and tends to create a higher degree of self motivation among the students. Students are able to express their opinions and beliefs in this type of environment, which can lead to more effective learning. This can help students maximize their skill levels while helping others achieve the same. Hence, incorporating interactive learning into course designs is an important means to attain professional skills [7], as research evidence supports [2].

2.2. ASPECTS OF CLASSROOM BASED PRACTICES

Apart from incorporating the principal approaches like interactive learning into course designs, several other classroom based practices [2] can positively affect the learning process. These practices include preparing instructional objectives, syllabus design, in-class activities etc [8]. One such classroom based practice is to integrate practical learning [12] in the classroom, which provides students with a chance to practice what they learn while they are learning it. Coupled with project based learning in core engineering courses, practice based learning can help students acquire professional skills more effectively.

2.3. INTEGRATING PRACTICE WITH LEARNING

Practice is an important and integral component of modern day course designs, but, is often the least developed aspect. Despite the extensive literature on the value of practice based learning, the perception remains prevalent in many academic circles, that practice based learning is somehow inferior to traditional classroom learning and not an industry standard [15]. Integrating practice with learning is one of the more complex elements in structuring courses [15].

Professional skills cannot be learnt inform a purely theoretical perspective. For example, to learn about communicating effectively in a team or otherwise, a student has to practice how to communicate but, cannot learn by reading about it or by simply hearing about it. While instruction or reading can provide deep theoretical insights, one cannot gain a skill without putting it to practice. Of course, practicing skills in class is not just limited to the professional skills. It is certainly required for technical or engineering skills as well. For example, one cannot claim that one knows how to “do” circuit design without actually designing a circuit [13].

The implementation part of integrating practice with learning process is a complex task. It is easier in terms to practicing engineering skills. For example to practice about setting up an electrical device is easier by introducing a laboratory sessions and demonstrate or rather make the students practice how to do it. But, designing elements of practice for learning professional skills requires clear focus on course design. Several instructors have developed their own techniques like use of informal discussions or even role plays to put the skills to practice. But, assessing what was practiced is again a difficult task [9].

3. CONVENTIONAL VERSUS INTERACTIVE LEARNING

One major difference between conventional and interactive learning environments is that the conventional classroom has information flowing in only one direction (from the instructor to the students) while, in an interactive learning environment information flows in all directions. Smith et al. [2] describe two models of classroom based teaching practices called “pour-it in” and “keep it flowing” where the former model illustrates a traditional instructional approach, while the latter illustrates an interactive environment. These models best describe the contrast between these two environments.

Apart from information flow, other classroom attributes like interdependence, accountability, and instructor involvement differentiate between the two environments [2]. In a conventional environment there is lower interdependence and members take responsibility only for themselves while there is higher interdependence in interactive learning where the students take responsibility for themselves as well as their peers. In conventional learning individual accountability is the norm, as opposed to shared individual and group accountability in interactive learning. Conventional environments usually do not include active involvement of the instructor in group dynamics, such as group formation or assessing teamwork skills. However in interactive learning, the instructor may act as a facilitator and be virtually a part of all the teams, while a strong emphasis is given to teamwork skills.

4. XB

The course called “Management for Engineers XB (eXperience Based)” is offered at the University of Missouri-Rolla. As the title of the course suggests, it incorporates experience into the learning process. The course design includes active learning, which promote the ability to learn professional skills. This course creates a learning environment that ensures maximum interactions among the students and the instructor, hence, making the information and knowledge flow in all directions. Evidence of this can be found in the analysis of archived videos of this class compared to videos of the same course taught in a conventional manner.

XB creates a classroom as a learning organization. The students are members of this organization and the instructor plays the role of a senior manager. This creates high levels of engagement among the students. All students are assigned to one of four departments; Responsibility, Doing, Observing or Understanding. Each department is sub-divided into several groups based on the responsibilities that need to be carried out by that particular department. The assignment of the students to these department is neither random nor mandated by the instructor, rather the students get to choose the departments they want to be in, based on their interests or self evaluations of their strengths and weaknesses. XB provides a much more liberal environment to students with in the legal boundaries of the classroom. Students have the opportunity to modify the grading system for the class and they grade themselves throughout the semester, based on a set of measures, including evaluations of weekly memos, reading memos, presentations, and comments, which are a part of the coursework. Two major responsibilities of each student are to act as a CAT-Coach/Administrator/Teacher and ALE-Apprentice/Learner/Employee. These activities are so designed to ensure that every student gets a chance to participate actively in the class. Unlike traditional courses, only a small percentage of the course is dedicated to the presentation by the instructor. Instead peer instruction is used, where the students themselves read and understand the concepts and discuss them via presentations that are scheduled by students.

The instructor plays the role of a "facilitator" and helps the students with difficulties they face in adopting this proactive, almost "real world" perspective. Every class session (termed as a meeting) dedicates time for general discussion of the issues the students have. Most of the issues or conflicts are resolved by the students themselves and if they cannot, then they approach the instructor for help. The instructor does not provide them with a solution, but tries to reach to the bottom of the issue and "facilitates" them and guides them to help them resolve the issue. Moreover, XB takes co-operative learning as the baseline and the classroom activities are designed such that the students work together in teams towards a common goal. They learn and help other students learn.

Apart from these, another distinguishing feature in XB is that it provides an excellent platform for the students to practice the skills that they learn. Examples include students practicing how to communicate effectively (under the professional guidance of the instructor), engage in conflict resolution (they are encouraged to raise and resolve conflicts, rather than covering them up), working effectively in teams and many more.

All the above mentioned features indicate that XB has been designed with the entire integral component for successful instruction. It provides a strong motivation and a clear statement of goals to students and also helps the students get self motivated. XB can provide an exemplary learning structure for students to learn, practice and hence, successfully develop professional skills.

5. ANALYSIS

One important measure distinguishing conventional and interactive approaches to learning is the level of interaction present in courses. To evaluate this measure, the archived videos of three sections of the same course, were analyzed, one using the interactive learning approach and the other two taking the conventional approach to teaching. (Archived videos are available for these courses because they are offered as part of a distance education program.) The archived videos of three different sections of the course called “Management for Engineers” offered at the University of Missouri-Rolla were analyzed. The section that takes the conventional route is called “Management for Engineers” and the one that takes the interactive route is called “Management for Engineers- XB (eXperience Based)”. Randomly selected portions of each section were selected and evaluated for elements of interactions taking place in the classroom. These interactions include instructor talking to all students, a student talking to all other students (and the instructor too) and individual interactions between the students or the student and the instructor. The results obtained were recorded and a network plot was generated using the results. The results indicated that the amount of individual interactions taking place in the XB was higher compared to the other section.

6. THE PROCEDURE

All the three sections of the class had class sessions of approximately two and a half hours (that is one hundred and fifty minutes) in length. The number of class sessions ranged from 14 to 17 based on the length of the semester. Uniformly distributed random numbers corresponding to the number of videos and the length of the class sessions were generated. To be more precise, if the semester had 15 class sessions, random numbers ranging from 1 to 15 were generated. If the class length was 150 minutes, random numbers were generated ranging from 1 to 150. These random numbers were truncated to the nearest integer for the purpose of ease of analysis. The class number was then matched with the minute (randomly generated) and that particular moment was watched in the video to see who was talking and to whom he/she was talking. The number of random numbers generated was 25 so there were 25 recordings per semester of the random interactions in the class. The interactions among the instructor and all the students was recorded as “Prof to all” and the student’s interaction to all other students was recorded as “Student “X” to all” (where X is whatever letter the student is assigned with) and the individual interactions were recorded as “Student X to Student Y” or

“Student X to Prof” or “Prof to Student X”. The results obtained are presented in Tables 1 and 2 followed by the network diagrams generated in Figures 1 and 2.

Additional data from student surveys made in the XB section for several semesters was collected and analyzed. The questionnaire in the survey measured: “Constructive Controversy” [16], “Goal Interdependence”, and “Perceived Autonomy Support” [17]. This was a measure of their experience after they have taken this course. The responses to this survey from several semesters were collected and then plotted. These plots are presented in Figures 3, 4 and 5.

7. RESULTS

The results obtained from the analysis of the archived videos and the plots from the surveys are presented in this section. The results are as follows:

Table-1 presents the data recorded for interaction in the class sessions in the Sections 1 and 2, which use conventional teaching approaches.

Class No	Minute	Interaction	Class No	minute	Interaction
1	7	Student A to ALL	1	34	Prof to ALL
1	8	Student B to ALL	1	22	Prof to ALL
2	69	Prof to ALL	1	51	Prof to ALL
3	97	Student C to ALL	3	3	Prof to ALL
4	51	Prof to ALL	5	64	Prof to ALL
4	63	Prof to Student C	5	99	Prof to ALL
4	6	Prof to Student D	6	17	Prof to ALL
5	115	Student E to ALL	6	122	Prof to ALL
5	45	Prof to ALL	6	119	Prof to ALL
6	88	Student F to ALL	7	88	Prof to ALL
6	9	Prof to ALL	7	99	Prof to ALL
8	72	Prof to ALL	7	118	Prof to ALL
8	14	Prof to ALL	8	50	Prof to ALL
8	26	Prof to ALL	8	82	Prof to ALL
10	104	Prof to Student G	9	115	Prof to ALL
10	102	Prof to ALL	9	22	Prof to ALL
11	97	Student G* to ALL	10	59	Prof to ALL
12	97	Student H to all	10	88	Prof to ALL
12	41	Prof to ALL	11	26	Prof to ALL
12	68	Prof to ALL	11	116	Prof to ALL
13	93	Prof to ALL	12	17	Prof to ALL
14	27	Student C* to ALL	12	68	Prof to ALL
14	64	Student I* to ALL	13	48	Prof to ALL
14	71	Student G to ALL	14	20	Student A to ALL
14	20	Prof to ALL	14	63	Student B to ALL

Table-1: Interactions in Sections 1 and 2, which take conventional route

Table-2 presents data recorded for interactions in XB.

Class No	Minute	Interaction	Class No	minute	Interaction
1	145	Student A to Prof	1	52	Student H to Student A
2	4	Student B to Student C	2	34	Student A To ALL
2	54	Student D to All	3	92	Student A to Student G
3	52	Student E to all	3	101	Prof to Student F*
3	146	Student F to Student C	3	67	Student B to Student A
4	35	Student B to all	4	83	Student A to Prof
4	124	Student E to all	4	55	Student A to ALL
6	99	Student G to all	4	36	Student E* to Student A
6	48	Student C to Student D	5	23	Prof to ALL
6	111	Student A to Student C	5	78	Prof to Student I
8	23	Student A to all	6	18	Student A to Student H
8	86	Student H to Student I	6	134	Student H to ALL
9	51	Prof to all	7	139	Student A to Student B
10	50	Student C to Student J	7	57	Prof to ALL
10	134	Student K to Student C	7	135	Student A to Student B
11	31	Student L to Student C	8	137	Prof to ALL
11	35	Student A to Student G	8	18	Student B to ALL
12	134	Student F to Student C	8	129	Student A to ALL
12	122	Student G to Student A	9	105	Student A to Student H
12	50	Student F to all	9	16	Prof to ALL
13	82	Student B to All	10	19	Student B to ALL
14	101	Prof to all	10	102	Prof to Student C
14	18	Prof to all	11	56	Student G to ALL
15	2	Student H to all	11	119	Student K to Student B
15	102	Prof to Student C	14	65	Prof to ALL

Table-2: Interactions in XB

The network diagrams created from the recorded data are presented in Figures 1 and 2. The interactions among the instructor and the students to all the students are not included in the main network because those interactions are not the focus of this analysis. Those interactions are plotted separately in the same figure. The number beside these group interactions indicate the total number of times the corresponding student (or the instructor) has interacted in that particular semester.

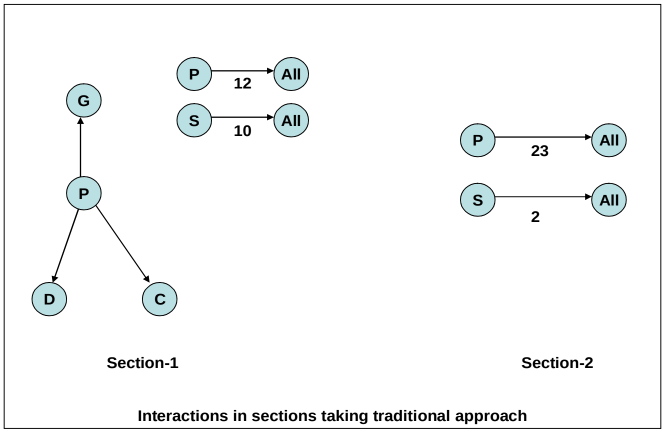


Figure-1: Network diagram for the conventional teaching environment

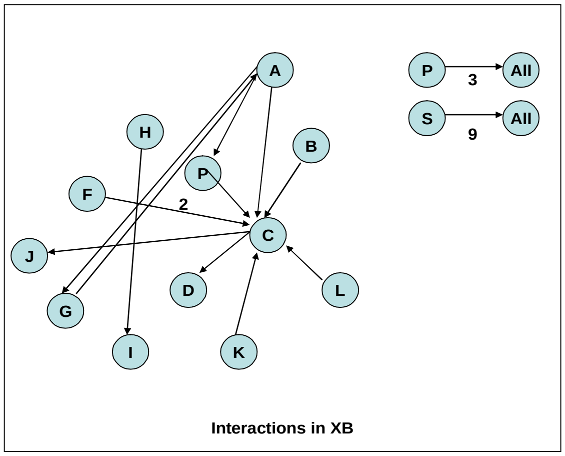


Figure-2: Network diagram for XB, using interactive learning

The student responses for the survey questionnaire are plotted as follows:

No	Average response	St DV
20	5.16	1.05

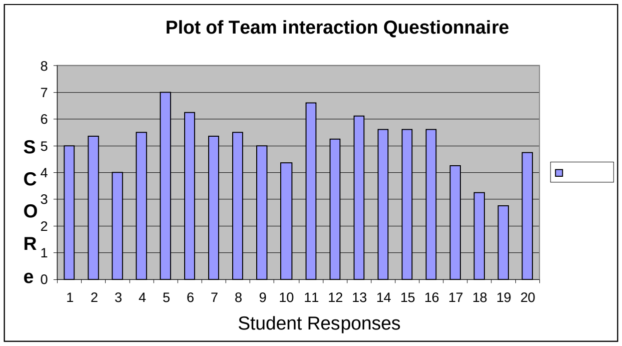


Figure-3: Plot for Survey for Team Interaction: An Assessment of Constructive Controversy

No	Average Response	St Dev
21	5.66	0.76

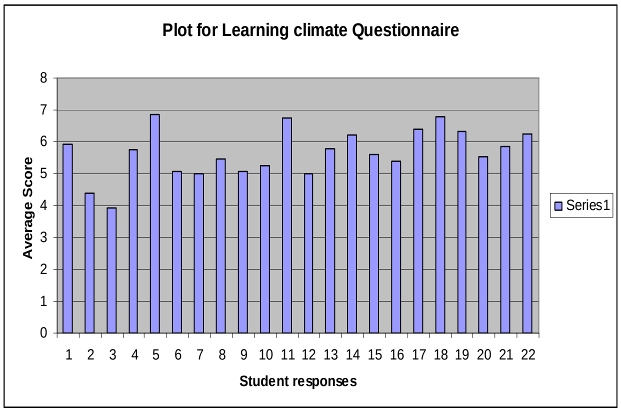


Figure-4: The plot for survey for Learning Climate: An Assessment of Perceived Autonomy Support

No	Average for co-operation	Average for Competition	Average for
21	5.147	3.55	4.35

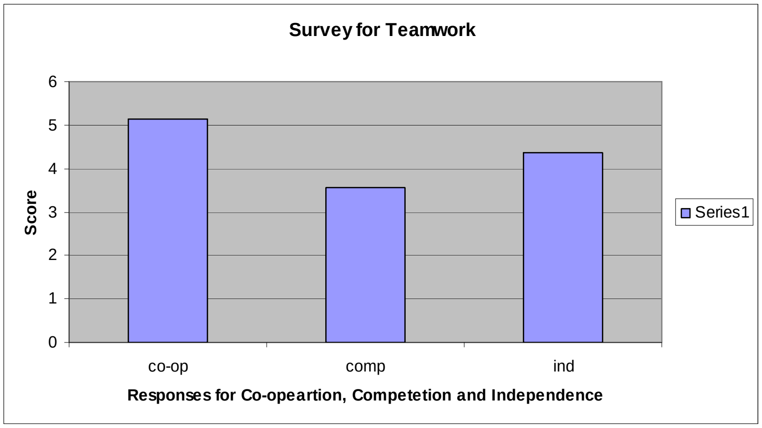


Figure-5: The plot for Survey for Teamwork: An Assessment of Goal Interdependence

8. DISCUSSION

The data obtained and the network charts clearly indicate that the great extent to which the students are interacting on a regular basis in XB is quite large when compared to the section that takes conventional approach. A closer look at the network chart reveals that in the traditional section of the course, almost sixty percent of the time, the instructor is addressing all the students, usually lecturing the students or giving a presentation. In cases where the instructor is not lecturing all the students, the students are usually giving presentations (interacting with all the students). There are only a very few individual interactions in the classroom, limiting the flow of information. This data well illustrates the “pour-it in” model discussed by Smith et al [2]. The process of knowledge transfer is taking place only in one direction and that is from the instructor to the students (or a student to all other students via presentations. This is not the case with XB. The data for XB reveals that there are a large number of individual interactions among the students. Moreover, the other interactions present are the classes are not ones that include presentations. When the data was coded, it was clear that many of the individual interactions were not presentations of a discussion of course material, but rather discussions about issues like, grading or even conflict resolutions. XB plays a better role in “engaging” the students in an efficient manner that the section that takes the traditional approach.

The plots from the student surveys presented above indicate high levels of perceived autonomy support, that is, that their autonomy in the course was supported by the instructor. They also exhibited relatively high levels of cooperation and self-reported abilities to use conflict to integrate diverse perspectives.

The main purpose of this analysis was to demonstrate a method of incorporating interactivity and practice in instructional design and some of the results of doing so. The results illustrate that interactive and practice based learning can offer a valuable approach to student learning.

9. CONCLUSION

This paper concludes that interactive learning yields better results when compared to traditional classrooms when incorporated in to the instructional design. The results of the surveys indicate that, in XB, the instructor supports the student’s autonomy to a greater extent and hence, provides a liberal atmosphere for them which can lead to greater motivation levels in the students. Considering the strategies provided by researcher and the facts provided them and the results of the analysis made in this paper, it can be concluded that integrating interactive and experiential (practical) learning with traditional instruction methods provides an opportunity for students to practice the professional skills specified by ABET and a basis for them to better learn those skills. But, this process needs attention to details of the specific components of instructional design and a careful consideration of the class-room based practices.

10. REFERENCES

[1] http://en.wikipedia.org/wiki/Accreditation_Board_for EngineeringandTechnolog.

[2] Smith, A.K., Sheppard, S.D., Johnson, W.D, Johnson, R.T., Pedagogies of Engagement-Classroom based practices, *Journal of Engineering Education,* Vol., No. 2005, pp. 87-100.

[3] Froyd, J.E., Ohland, M.W., Integrated Engineering Curricula, *Journal of Engineering Education,* Vol., No. , 2005, pp. 147-164.

[4] Dym, C.L., Agogino,.A. M, Eris, O., Frey, D.D., Liefer, L.J., "Engineering Design Thinking, Teaching and Learning", *Journal of Engineering Education,* Vol. , No.-, 2005, pp. 103-115.

[5] Schuman, L.J., Besterfield-Sacre, M., McGourty, J., "The ABET "Professional Skills"-Can They Be Taught? Can They Be Assessed", *Journal of Engineering Education,* Vol. , No.-, 2005, pp. 41-101.

[6] Woods, D.R., "Participation is More than Attendance", *Journal of Engineering Education,* Vol. , No.-, 1996, pp. 177-181.

[7] Prince, M., "Does Active Learning Work? A Review of the Research", *Journal of Engineering Education,* Vol. , No.-, 2004, pp. 223-231.

[8] Finelli, C.J., Klinger, A., Bundy, D.D., "Strategies for Improving the Classroom Environment", *Journal of Engineering Education,* Vol. , No.-, 2001, pp. 491-497.

[9] Catalano, G. D., Catalano, K., “Transformation: From Teacher-Centered to Student-Centered Engineering Education”, *Journal of Engineering Education,* Vol. , No.-, 1999, pp. 59-64.

[10] Felder, R.M., Brent, R., “Designing and Teaching Courses to Satisfy the ABET Engineering Criteria”, *Journal of Engineering Education,* Vol.92, No.1, 2003, pp. 7-25.

[11] Mehlenbacher, B., Miller, C.R., Covington, D., Larsen, J.S., “Active and Interactive Learning Online: A comparison of Web-Based and Conventional Writing Classes”, *IEEE Transactions on Professional Communication, Vol. 43, No. 2, 2000, pp. 166-184*

[12] Luechtefeld R., “Tacit Knowledge Literature review,” chapter 2 of Dissertation, 2007.

[13] Sternberg R. J., Horvath, Epilogue from "Tacit Knowledge in Professional Practice”, 1999, Mahwah, NJ. pp. 231-236.

[14] Baca D., Luechtefeld R., “Rethinking Facilitation”, prescript version, 2006.

[15] Barrie S., “Assessment: Defining the worth of Professional Practice”, Institute for teaching and learning, 2006.

[16] Alper S., Tjosvold, D., Law, K. S., “Interdependence and Controversy in Group Decision Making: Antecedents to effective self-managing teams”, *organizational Behavior and Human decision Processes,* Vol. 74, pp. 33-52.

[17] Black, A. E., Deci, E. L., “The effects of instructors’ autonomy support and students’ autonomous motivation on learning organic chemistry: A self-determination theory perspective”, *Science Education,* Vol. 84, pp. 740-756.

PAPER - 3

DIALOGUE BASED PROJECT MANAGEMENT SYSTEM

Siddartha Thummuri & Ray Luechtefeld
University of Missouri-Rolla, Missouri, USA
Email- st99d@umr.edu, luechtef@umr.edu

ABSTRACT

Increasing complexity and high stakes in business projects call for new project management systems which can handle those complexities and be able to offer an efficient change management feature. Traditional project management tools do not take into account some sources of change, task interactions and the necessity for distributed planning. Today's projects involve huge amounts of data and information. This makes it necessary to have a robust and intelligent system that can handle and transfer information on time and to people who are responsible or affected by that information. Traditional project management systems do not provide proper "change notification." When a person or a team responsible for a particular task is not aware of changes a loss in terms or time and money can occur. This issue is magnified in "virtual teams", where the people involved with a particular project are geographically distant. To overcome this issue of poor information sharing and propagation the authors propose a system called the "Dialogue Based Project Management System". The user interface of this system is developed in Visual basic.NET and Visual Basic and the back end is MS project and a database. The basic architecture of this system takes information from every project meeting and stores it in a database, then evaluates the information. If there are changes in the project, the system automatically communicates the changes to all the people responsible, ensuring proper "change notification." This enables a system of "information centralization”. Moreover the user interface is speech recognition enabled making it possible for the users to be able to dictate information to the system.

Key Words

Project management, change notification, information centralization, speech-interface

I. Introduction

Today, project management and the tools used for it are significantly advanced to have attained a design plateau. Meanwhile, the complexity of projects continues to increase. While the number of people involved in a project has grown, more project teams are geographically dispersed, due to the advent of new technologies, enhanced communication devices and especially the internet.

These technologies have produced advanced communication systems and in turn have given a new dimension to a project. People at different geographic locations, thousands of miles apart may work together on a same project, partly due to outsourcing and globalization. This has become increasingly important in the case of software projects. Today, few software projects are handled in a single location. However, globalization and outsourcing have spread to all the industries including automobile, construction and medical.

The increasing number of people involved with one project makes it difficult to manage the tasks, schedules, deadlines, and resources, and particularly the flow of information. The complexity is elevated when the systems or people involved are physically or geographically distant. If information is lost or overlooked, a pulse of confusion can be created, which propagates throughout the project.

Change management is particularly difficult in these circumstances. Traditional constraints to project management are time, cost, and the scope of the project, which are the requirements specified for the end result. But, modern projects have another constraint associated with these, and that is “change” or “changes” in the time, cost or scope of the project. The challenge of managing these changes unfortunately often overlooked.

There are several reasons so as to why changes occur in a project. For example, the main sources of change in software development are [2]:

External

- Volatile customer requirements
- Error reports
- Unpredictable obstacles (hardware problems,
  Proprietary interfaces, shrinking budget…)

Internal

- Refactoring (if something works… don’t change it)
- Change of development libraries, technology etc.
- Project expansion (due to iterative development)

These changes affect the times, schedules and budgets of a project. Even a small change, if unnoticed, could propagate throughout the project and might gradually lead to project failure. As an example, consider a software project where several people are working on a large, distributed project. “Distributed” means that the people or systems involved in the project are geographically distant (as is the case with outsourced projects). If there is a sudden change in customer requirements, and the details are not propagated to the development location the total schedule and cost can be impacted. Hence, it is vital to identify the changes in project management information and propagate them throughout the system.

This calls for an efficient “change management” [3] system integrated with the project management system. There are several commercially available software packages for project management. But, few of them feature change management systems. Almost all of these are single user-interface systems, so they require a moderator or administrator to manually update the system with information about changes. This increases the dependency on a single person and hence, introduces opportunities for failure.

Hence, there is an essential need for a change management system, integrated with efficient project management software which can track changes automatically, document and record those changes and notify the right agents (people or systems) about the changes in a timely fashion. In this paper one such system is proposed to resolve the above mentioned issues. This system is integrated with MS Project® , which is a robust and widely used project management (PM) software and a database. The basic systems take information from the users via their task or project meetings using a speech recognition engine and store the information in a database. After every meeting, the system compares the new information with the previous one and checks for changes in the project management system. If the system identifies any changes, it automatically sends an e-mail to the people involved with or affected by the change.

II. Scope of the Paper

This paper proposes a system that is integrated with already available PM software. This software does not work independently; it requires applications like MS project, MS Access and MS speech tools. This design of the system is a prototype to demonstrate the features and functionalities that it could handle. Since it is not fully-functional, it is purely for demonstration purpose.

III. Resources for the System

The system has been developed using VisualBasic.NET and Visual Basic programming languages. The back end of the system involves MS Project (Project management tool, a product of Microsoft Corporation) and a database (here MS Access) which can be replaced with any other database that is compatible with the above mentioned programming languages (according to the users preferences). As will be discussed later in the paper this system has a speech-enabled user interface which enables the user to dictate information to the system. The speech application is developed using Microsoft Speech Recognition, which is provided along with the Windows ® Vista ® operating system. The system involves a speech agent that prompts the user in all tasks within the system. This feature uses MS Agent for speech synthesis, a product of Microsoft Corporation too and is a built-in feature of Microsoft Windows XP operating system based machines.

IV. Applicability and Compatibility of the System

The Dialogue Based Project Management System (from here on, termed as DBPMS in this paper is applicable to any project where there is a need for a Project Management tool ranging from small college or university (academic related) projects to large and complex tasks like construction or software projects where there are high risks involved. The future work suggested in this paper would make it possible for users to customize the system according to their requirements making the DBPMS virtually fit any project whatsoever. The DBPMS is compatible with any windows operating system based machines (Windows XP or later versions) but, it does require some system prerequisites which are a full installation of MS office product suite (Including MS Access and Speech tools) Microsoft .NET framework 1.1 or higher and Speech Recognition Engine for Windows ® Vista ®. Most of these features are most commonly available in the windows based PC's today. The only unavailable component would be Speech Recognition Engine, but, it is available on in Vista ® operating system. But, again, since the project uses speech recognition engine for windows® Vista®, the working of this project is limited to that particular operating system.

V. Basic Design and Architecture

The basic architecture of the system is simple, in the sense it does not involve complex dependencies. The basic components of the system as mentioned previously are: a form that initiates when the system is started, a database that works at the back end and MS Project that works on the back end. Apart from these, there are other speech related applications (like MS Agent and speech recognition) that are initiated when the first form is started. There are three forms in the system and so the system is divided in to three stages. The first form is termed as MS project and Database (here MS Access) loading form, the second form is termed as the Meeting Browser and the third form is termed as Project status display form. The terminology is given to the forms based on the functionality that a form carries out. The "information flow" diagram of the system is given in Figure-1 below. The component wise detailed description follows the flow diagram.

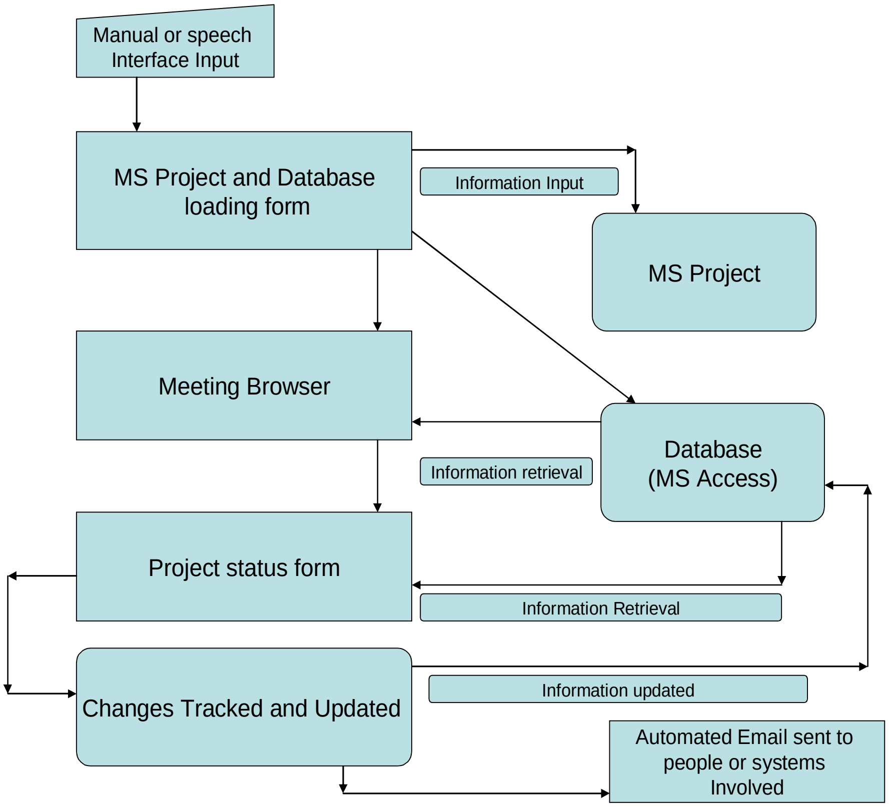

Manual or speech Interface Input
MS Project and Database loading form
Information Input
MS Project
Meeting Browser
Database (MS Access)
Information retrieval
Project status form
Information Retrieval
Changes Tracked and Updated
Information updated
Automated Email sent to people or systems Involved

Figure-1: Information Flow Diagram

A detailed description of the individual components and their functionality are presented below.

VI. Welcome Form

This form has a brief welcome message about the system. Apart from this, it has two buttons named “create a new project” and “open an existing project”. The functionalities of these buttons match their names. A click (or a voice command) on the “create new project” button would initiate a dialog box asking the users to input the “project name” and “number of tasks”. After this is done the system would load the next form. Based on the project name entered a database (MS access file) and project files (MS project file) are created in a predefined directory with the same name. The second button, as the name describes, opens an existing project, details of which are discussed in the later parts of this paper. A screen-shot of the welcome form is presented in Figure-2.

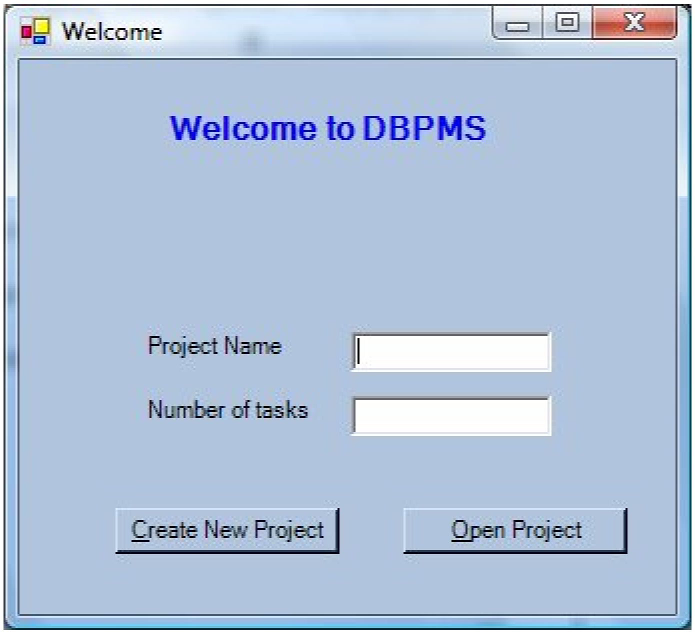


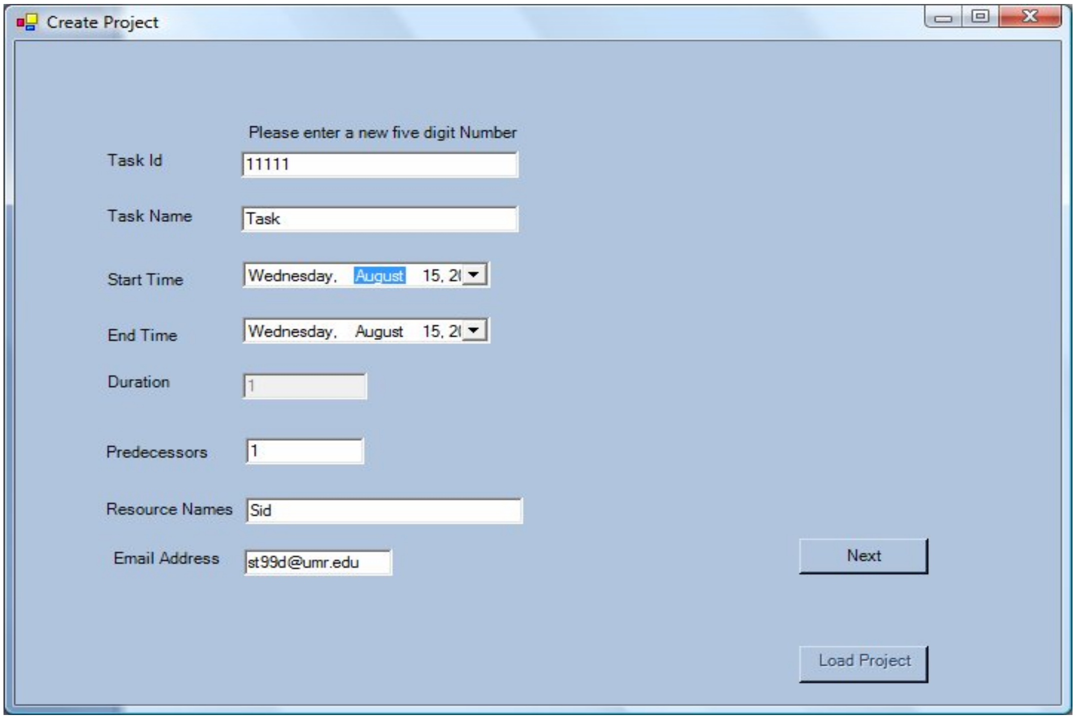
Figure-2: Welcome Form

VII. Ms Project and Database Loading Form

The “create a new project” button of the welcome form after a simple query would lead to this form. As the name of the form suggests, the main functionality of this form is to take information input from the user. The form consists of several input fields or simply a query regarding information pertaining to the project. For a given project this form is a one time input form, so for a particular project the information need not be entered again. The information input fields in this form or the query in the form is more or less similar to a project information query in MS project or any other commercially available project management tools. For example, the query consists of questions related to the project like task names, predecessors, task duration etc. Similar to MS Project, this form consists of information input for each task involved in a project separately instead of taking the whole project information in one single form.

Specifically, the form consists of a field where it asks for the number of tasks involved in the task. After the user inputs the number of tasks, the same form is loaded the number of times corresponding to the number of tasks inputted by the user. This feature is designed to clearly differentiate information regarding each task while it is stored. This makes the data or information entry process less complex and also avoids information loss. Every information field is similar to other PM tools or commonly needed project information. The only unique field in this form is the Task ID (Task Identification) field. This field is triggers the information input and retrieval process that is in the later forms. The Task ID is the primary key in the database too. It is alpha-numeric and has a character limit of 8. This limit is made so as to avoid non- uniformity in task identification and also to avoid confusion with variable Task IDs in the database.

For every project once the Task ID is set to a particular character limit, it has to be maintained the same for the entire project. Needless to say, once a person or a task is assigned with an ID it will remain the same for that particular person or system. Based on the input given by the user regarding the number of tasks in the welcome form, this form is loaded again and again corresponding to the number of tasks entered. A screen-shot of the form is presented in Figure-3.



Please enter a new five digit Number

Task Id: 11111

Task Name: Task

Start Time: Wednesday, August 15, 2(

End Time: Wednesday, August 15, 2(

Duration: 1

Predecessors: 1

Resource Names: Sid

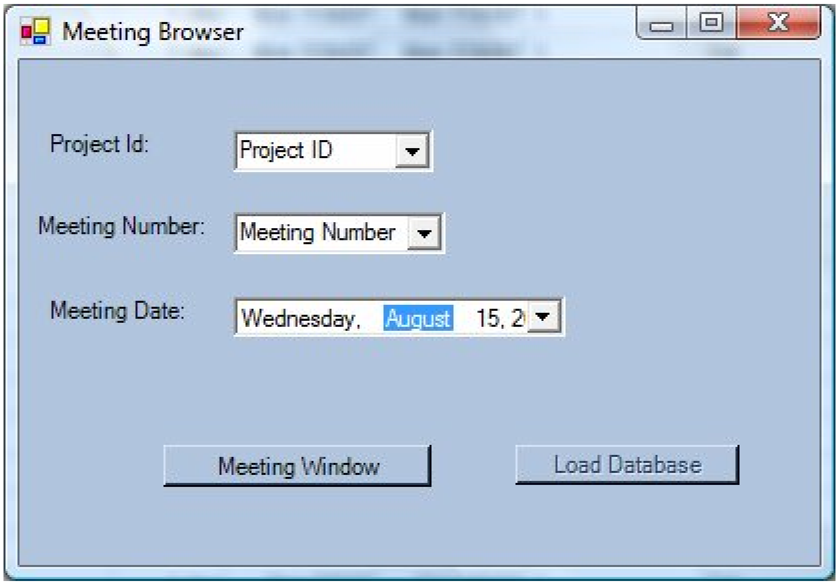
Email Address: st99d@umr.edu

Figure-3: Database and Project Loading Form

MS Agent (which is a speech synthesizer, a feature of this system) guides the users with the query and also prompts them if any corrections are necessary. For example if the user enters an invalid Task ID or if the start and end dates and the task duration does not match, the agent prompts them to correct the information. Like mentioned earlier, the information input process can be done both manually (using the key board) or using the speech recognition engine (built in) tool (using a microphone). After the user is done entering the information in the query, the user can click on the "Load Project" button or command the system using the speech recognition system by saying "Load project". Doing so creates a data base entry with the information provided and also loads MS Project with the same information simultaneously. This process takes a few seconds and soon after the process is done, the system loads or initiates the second form, the meeting browser.

VIII. The Meeting Browser Form

This form is initiated after the first form, the "Load MS project and database form", creates a database entry and loads MS project with the inputted information. This allows the users to browse existing meetings or to create new meetings. This is done so as to create a database entry with the information pertaining to a particular meeting. The basic functionality of this form is simple. It is similar to any other file opening window available in a Windows operating system or other operating systems. This form has only a few fields; a Task ID selection field, a meeting number entry or selection field and a field where a date for a meeting can be assigned or if information is being retrieved, a date can be chosen from the past meeting dates. Other than these, the form has a button named "load database" and its functionality is to load the database with the information regarding the meeting. It has a connection with the database and can input and retrieve information from the database based on the details provided by the user, that is, the Task ID, meeting number and/or the meeting date. Similar to the previous form, this form is also speech recognition engine-enabled so that the browsing process can be done using this engine. This form is also featured with MS Agent to guide through the process or prompt the user to make corrections if any. After the user is finished browsing the necessary information or entering new information, a request is sent to the database to act accordingly. After the request is carried out, this form is closed and the third form, the "Project status form," is initiated. Though this form is closed a button is provided in the project status form to initiate the meeting browser again. This is because the user might need to browse for old meetings sometimes to refer back to some necessary or important information. A screen-shot of this form is given in Figure-4 below.



Project Id: Project ID

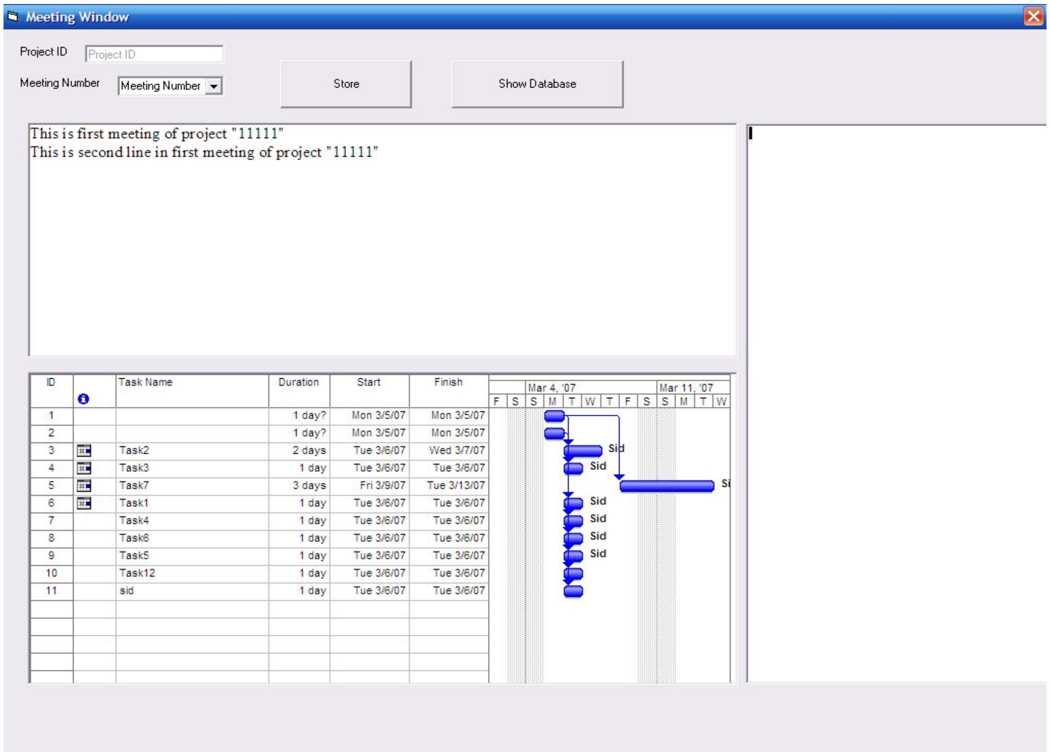
Meeting Number: Meeting Number

Meeting Date: Wednesday, August 15, 2

Figure-4: Meeting Browser

IX. The Meeting Window

This form is called the "Meeting window" and is the heart of the system because it carries out important functionalities. This form monitors the project on a regular basis, corresponding to how frequently the project or task meetings are scheduled. This form handles the change management/change propagation also. The form contains two fields for basic input and output of information for the database or MS project, two buttons to control and command functions like storing information or to display the database itself. It also contains three windows, two of which display the minutes from the previous meeting, the current project status (displayed in MS Project) respectively and the third window which captures the minutes of the meeting directly from a meeting. This is done using the speech recognition engine. A screen-shot of this form is presented in Figure-5 below.



This is first meeting of project "11111"
This is second line in first meeting of project "11111"

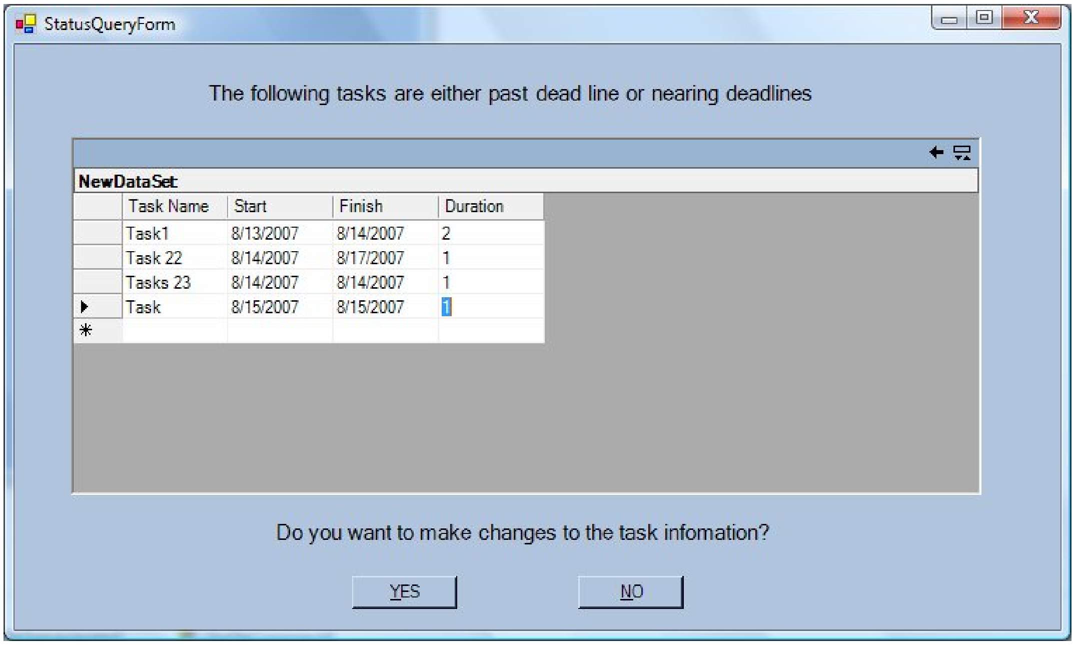
ID	ℹ	Task Name	Duration	Start	Finish
1			1 day?	Mon 3/5/07	Mon 3/5/07
2			1 day?	Mon 3/5/07	Mon 3/5/07
3	▦	Task2	2 days	Tue 3/6/07	Wed 3/7/07
4	▦	Task3	1 day	Tue 3/6/07	Tue 3/6/07
5	▦	Task7	3 days	Fri 3/9/07	Tue 3/13/07
6	▦	Task1	1 day	Tue 3/6/07	Tue 3/6/07
7		Task4	1 day	Tue 3/6/07	Tue 3/6/07
8		Task6	1 day	Tue 3/6/07	Tue 3/6/07
9		Task5	1 day	Tue 3/6/07	Tue 3/6/07
10		Task12	1 day	Tue 3/6/07	Tue 3/6/07
11		sid	1 day	Tue 3/6/07	Tue 3/6/07

Figure-5: Meeting Window

The three windows do not just display information, but, can be edited. For example, at the time of the meeting if the user wants to make changes to the information in MS Project that is displayed, it is possible for the user to do so directly in the meeting window itself. These changes are automatically saved and are updated in the database also. The minutes of the last meeting are displayed in the meeting window purely for information reference purposes. The window that takes minutes of the current meeting, as described above, does so using the speech recognition engine but is not solely dependent on that engine. The minutes can be manually taken by the user by typing directly into the window. The minutes are saved as a MS Word document in the database. When the cursor is placed in any of the windows to activate the manual input process or for editing, the speech engine is automatically deactivated. This is done so as to avoid dual sources of information input. A button is provided on the top of the form to activate or deactivate the speech engine manually. Another button is provided to launch the meeting browser, which the user might need to refer back to previous meetings. After the users are done with the meeting, they click on the Update/close button (at the bottom) which updates the database with the minutes of the meeting. Once the database has updated meeting information and minutes, it prompts the user via the speech agent notifying the user that the database has been updated. In fact whenever the database or MS project are edited or changed or updated the user receives a notification from the system regarding the same. After this process, this window is closed and the next form which is the project status query form pops-up. When the user chooses to open and existing project in the welcome form, the systems starts the meeting window directly for the corresponding project (as chosen) by the user.

X. The Project Status Query Window

This is the final stage of the system and its functionality is to get project or task status information input from the user and update the project information in the database and MS project and then send a notification email to corresponding people or teams. Once the system gets input from the user, it checks for changes in the project (such as any new or changed deadlines, changes in the budgets for tasks or delays in or early completion of any tasks). Once this process is done, an automated email regarding the project status (or any changes) is sent to all the people involved with the tasks or the project as a whole. The email addresses corresponding to people involved in a particular task are stored in the database. This is actually done on the basis of a query embedded in this form. This query consists of question related to task completion, changes in deadlines, new tasks added and/or existing tasks terminated. A change in the deadline or completion date or one task would affect its predecessors or the tasks following it. If a deliverable of a task is the starting point for the next task (as is typical in software development, usually when the product is divided in to chunks of design and implementation) and if there happens to be a change in the delivery date of this task, it would delay the next task unnecessarily. On the contrary, if a task is completed earlier to its completion date, its succeeding task can be pushed back to an earlier date without actually having to wait for the scheduled time. This would propagate through the entire project and the whole project or product can be delivered ahead of time, gaining a lot of customer focus. So coming back to the form, its query is based on the information in the database. It has questions like “a task is due completion today, is it scheduled for completion today?” If the answer is “yes” then, nothing is done, but if the answer is “no” then it asks for new dead line and updates the database with that information. Whenever the database and MS project are updated an email is sent to the user and the corresponding people. The screen-shots of the status query form and the Update Task form are presented in the Figures-6 and 7 below.



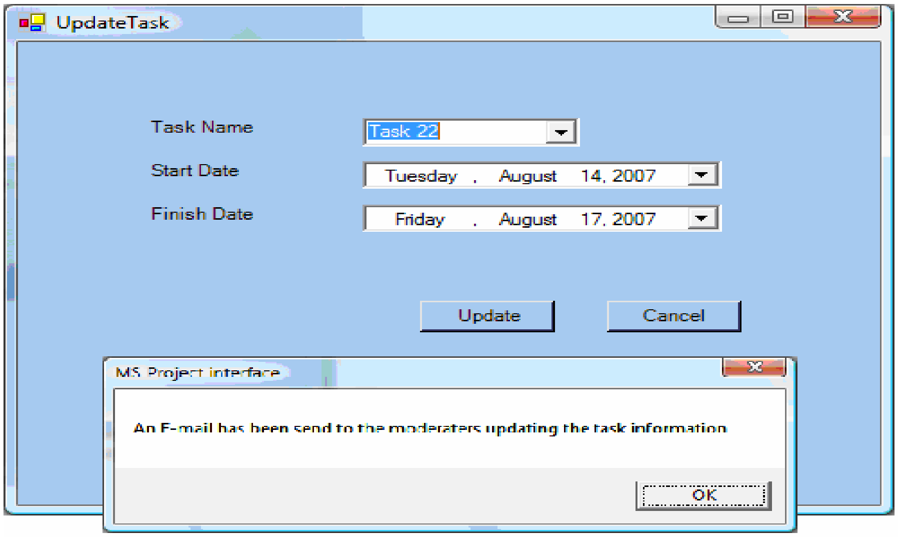
The following tasks are either past dead line or nearing deadlines

NewDataSet

	Task Name	Start	Finish	Duration
	Task1	8/13/2007	8/14/2007	2
	Task 22	8/14/2007	8/17/2007	1
	Tasks 23	8/14/2007	8/14/2007	1
▶	Task	8/15/2007	8/15/2007	1
*				

Do you want to make changes to the task infomation?

Figure-6: Project Status Query Form



Task Name: Task 22

Start Date: Tuesday , August 14, 2007

Finish Date: Friday , August 17, 2007

An E-mail has been send to the moderaters updating the task information.

Figure-7: Project Update Form

XI. The Dynamics of the System

So far, the paper has been describing the individual components of the system and their basic functionalities. This section of the paper describes the actual dynamics of the system and how the information flows in the system.

There are two different ways to initiate this system. One is to create a new project and the other is to open an existing project. We begin by describing how a new project is created and hence, monitored based on the scheduled meetings. When the user opts to create a new project, he is asked to enter the "project name" and the "number of tasks" in the project. After he does that, a database and MS project file are created with the name corresponding to name inputted by the user, usually something like "Project_XYZ". These files are created in the same directory where the project related setup files are stored. If the user wishes to store these files on a server for access to all the project related personnel, it has to be done manually by the user. Once these files are created, the system will prompt the user to enter the task and project related information like the task name, start date, end date, duration, predecessors, resources etc. This is done a number of times equal to the "number of tasks" inputted by the user. For example if the user enters the number of tasks as three, this information (regarding the tasks) is asked three times. That is each query form corresponds to one task. When giving out the information regarding the tasks the user is prompted to enter a Task ID for each task. This Task ID is unique to each task and is used for reference purposes, that is, in the further stages of the project, this Task ID is used to update or retrieve information corresponding to that particular task. Once this is done the user commands the system to "load" the project. Doing so will load the task related information into the MS project file already created.

Once this information is loaded in to the project, it automatically creates a Gantt chart (and simultaneously Network Diagram) for the project. This is an inbuilt functionality of MS Project.

By now, the user has the project information stored in the database as well as the information in MS project too. The Gantt chart of the project is available to all the people related to the project. The next step is to prepare the user for the project or task meetings. This is done using the meeting browser form. This form prompts the user to create or open an existing meeting. Upon the action of the user it loads the appropriate form. For example if the user wishes to create a new project meeting, it takes information related to the meeting like the meeting number, meeting date etc. Based on this information, it loads the meeting window. In the meeting window, the user has options like viewing the project file, take meeting minutes or even edit the project file within the form itself. At the end of the meeting the document containing the minutes of the meeting is stored in the database. After this is done, the system prompts the user again to provide the project status information. This is so done on the basis of simple queries like “is the team meeting the task completion deadline” and so on. If the user responds negatively, he is prompted to enter new deadline information or in fact for any changes related to the project like changes in the schedules etc. If the system gets new information regarding the project, it overrides the previous information with the new one in the database and MS project. Finally it sends the updates the people concerned with the new information via automated emails, like “the project information has been updated; please refer to the project file for updated information”.

If the user opts to open an existing project, the system initiates from meeting window and from there the process is the same as “create new project” option. The only difference between these options is that in “create new project” option the user enters the task related information while in the “open existing project” option he does not. The next sections of the paper talk about the advantages and limitations of this system and then a future work related to few modifications are proposed and discussed.

XII. Pros and Cons of the System

The greatest advantage provided by this system is that it forms an effective change management system, by constantly monitoring the project status and providing the user with updates regarding the changes in the information. Best of all, it does all of this automatically ensuring that there is little or no room for information loss. Apart from that, it saves a lot of time because the speech interface allows the users to input information at a faster rate. While it also helps avoid typing errors, because the current speech engines are not highly reliable, there is room for errors due to mispronounced or misspelled words. Also the speech engine needs training by the user and has problems with different accents. It does not support noisy environments. The next issue with the system is the security levels provided by MS Access. MS Access is not as secure as other database tools available commercially. These problems result because of the lack of resources for the development of the system. Other commercially available speech and database tools are robust and secure, but, are expensive. The scope of this paper is not to build a commercial tool, but rather is to design and develop a prototype that could demonstrate the effectiveness of this system.

XIII. Conclusion

The paper presents a software tool that integrates several components related to project management. This tool attempts to provide novel features like efficient change management, time saving, reduced errors, automated updating of information and a speech interface for ease of use. This system is a prototype that has been designed and built to demonstrate features that could enhance the process of project management, which has become a complex task, because of growing complexity and large amount of information involved in projects. To handle such complexity amid volatile customer requirements, a robust project management tool is required. This paper presents one such tool that could render the above mentioned services, hence, making project management and easier process.

XIV. References

[1] F. Hapgood, “Project Management: Curses for Complexity, CIO Website 2004 [Online].Available: http://www.cio.com/article/32175/Project_Management_Cures_ for _Complexity

[2] D. Weiss, “Traceability: Taming Uncontrolled Change in Software Development,” Institution of Computing Science 2002 [Online]. Available: http://www.cs.put. poznan.pl/dweiss/site/publications/slides/kkio-2002-slides.pdf

[3] C. Petrie, S. Goldmann, A. Raquet, “Agent-Based Project Management”, Lecture notes in AI 1999 [Online]. Available: http://www-cdrstanford. edu/ProcessLink /papers/DPM/dpm.html

APPENDIX-A

PAPER-2 SURVEY QUESTIONNAIRE

Learning Climate Questionnaire: Assessment of perceived Autonomy Support

This questionnaire contains items that are related to your experience with your instructors at UMR. Instructors have different styles in dealing with students, and we would like to know more about how you have felt about your encounters with your instructors. Your responses are confidential. Please be honest and candid. Please use the following scale to indicate how true each reason is to you:

1	2	3	4	5	6	7
Strongly Disagree			Neutral			Strongly Agree

1. I feel that my instructors provide me choices and options.

2. I feel understood by my instructors.

3. I am able to be open with my instructors during class.

4. My instructor conveys confidence in my ability to do well in the course.

5. I feel that my instructor accepts me.

6. My instructor makes sure I really understood the goals of the course and what I need to do.

7. My instructor encourages me to ask questions.
8. I feel a lot of trust in my instructor.

9. My instructor answers my questions fully and carefully.

10. My instructor listens to how I would like to do things.

11. My instructor handles people's emotions very well.

12. I feel that my instructor cares about me as a person.

13. I don’t feel very good about the way my instructor talks to me.

14. My instructor tries to understand how I see things before suggesting a new way to do things.

15. I feel able to share my feelings with my instructor.

Teamwork Questionnaire: An Assessment of Goal Interdependence

This questionnaire contains items that are related to your experience with the way course goals are set up in classes at UMR. Classes often create different kinds of goal structures, and we would like to know more about these affect your relationships with students in classes. Your responses are confidential. Please be honest and candid. Please use the following scale to indicate how true each reason is to you:

1	2	3	4	5	6	7
Strongly			Neutral			Strongly
Disagree						Agree

1. Students in this class ‘sink or swim’ together.

2. Students in this class have a ‘win-lose’ relationship.

3. Typically, in this class, each student ‘does his/her own thing’.

4. The goals of students in this class go together.

5. The goals of students in this class are incompatible with each other.

6. Students in this class work for their own independent goals.

7. Typically, in this class, students want each other to succeed

8. Students in this class give high priority to the things they want to accomplish and low priority to the things other students want to accomplish.

9. Typically, in this class, one student’s success is unrelated to others’ success.

10. Typically, in this class students like to be successful by their own individual work.

11. Students in this class like to show that they are superior to each other.

12. Students in this class have compatible goals.

Team Interactions: An Assessment of Constructive Controversy

This questionnaire addresses the type of interactions that occur within your team in this class. Please answer the following questions according to the following scale:

1	2	3	4	5	6	7
Strongly			Neutral			Strongly
Disagree						Agree

1. Team members express their own views directly to each other.

2. Team members listen carefully to each other's opinions.

3. Team members try to understand each other's concerns.

4. Team members try to use each other's ideas.

5. Even when we disagree, team members communicate respect for each other.

6. Team members work for decisions we all accept.

7. All views are listened to, even if they are in the minority.

8. Team members use our opposing views to understand the problem.

APPENDIX- B

CODE FOR THE “DIALOGUE BASED PROJECT MANAGEMENT SYSTEM”

```
Public Class Welcome
    Inherits System.Windows.Forms.Form
#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()
'This call is required by the Windows Form Designer.
        InitializeComponent()
'Add any initialization after the InitializeComponent() call
    End Sub

'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    'Required by the Windows Form Designer
    Private components As System.ComponentModel.IContainer
  #End Region

Private Sub Submit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Submit.Click

        Dim createProject As New MS_Project_interface.Form1
        createProject.lblProjectName.Text = Me.txtProjectName.Text
        createProject.lblNoOfTasks.Text = Me.txtNoOfTasks.Text
        createProject.Show()
        Me.Hide()
    End Sub

    Private Sub Welcome_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        MsgBox("Welcome to the Dialogue Based Project management
system. This tool is equipped with speech recognition, allowing the
user to dictate into the query instead of typing. To disable speech at
any time, just say 'stop listening', to enable it again say 'start
listening'. This is just a demo version and all the features are not
available in this version")
        'Shell("C:\Windows\Speech\Common\sapisvr.exe",
AppWinStyle.NormalFocus, True, -1)
        Dim oStartInfo As ProcessStartInfo = New ProcessStartInfo
        oStartInfo.FileName = "C:\Windows\Speech\Common\sapisvr.exe"
        Process.Start(oStartInfo)
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        MsgBox("Feature available in later version")
    End Sub
End Class
```

```
Imports SpeechLib
Imports MSProject = Microsoft.Office.Interop.MSProject
Public Class Form1
    Inherits System.Windows.Forms.Form
    'Private mSpK As AgentObjects.IAgentCtlCharacter
    Dim appProj As New MSProject.Application
    Dim counterForLoad As Int16

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub
#End Region

        Me.OleDbDataAdapter1.Fill(Me.DataSet11)
        'mSpK.Hide()
        'Dim AxAgent1 As New AxAgentObjects.AxAgent
        counterForLoad = Me.lblNoOfTasks.Text
        TextBox1.Focus()
        AxAgent2.Characters.Load("merlin", "C:\i386\merlin.acs")

        mSpK = AxAgent2.Characters("merlin")
        mSpK.Show()
        mSpK.Speak("Please enter the Following details to create a
database entry")
    End Sub
    Private Sub RC_Recognition(ByVal StreamNumber As Long, ByVal
StreamPosition As Object, ByVal RecognitionType As
SpeechLib.SpeechRecognitionType, ByVal Result As
SpeechLib.ISpeechRecoResult)
        TextBox1.Text = Result.PhraseInfo.GetText
    End Sub

    Private Sub RC_StartStream(ByVal StreamNumber As Long, ByVal
StreamPosition As Object)
        TextBox1.Text = Val(StreamNumber)
    End Sub
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        counterForLoad -= 1
        Dim tblProject As DataTable = New DataTable
        tblProject = Me.DataSet11.Tables("Table1")
        Dim rowNewProject As DataRow
        rowNewProject = tblProject.NewRow()
        '(Duration, EmailAddress, Finish, Predecessors, ProjectID,
[Resource Name], Start, [Task Name])
        rowNewProject("ProjectID") = Me.TextBox1.Text
        rowNewProject("Task Name") = Me.TextBox2.Text
        rowNewProject("Start") = Me.DateTimePicker1.Text
        rowNewProject("Finish") = Me.DateTimePicker2.Text
        rowNewProject("Duration") = Me.TextBox3.Text
        rowNewProject("Predecessors") = Me.TextBox4.Text
        rowNewProject("Resource Name") = Me.TextBox5.Text
        rowNewProject("EmailAddress") = Me.txtEmailAddress.Text

        tblProject.Rows.Add(rowNewProject)

        Try
            Me.OleDbDataAdapter1.Update(Me.DataSet11)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try


        Me.TextBox1.Text = ""
        Me.TextBox2.Text = ""
        Me.TextBox3.Text = ""
        Me.TextBox4.Text = "1"

        'Me.TextBox5.Text = ""
        'Me.txtEmailAddress.Text = ""

        Me.DateTimePicker1.ResetText()
        Me.DateTimePicker2.ResetText()
        Me.Button2.Enabled = False
        Me.TextBox1.Focus()

        If (counterForLoad = 0) Then
            Me.Button2.Enabled = True
            Me.Button1.Enabled = False
        End If

    End Sub

    'Private Sub TextBox1_GotFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TextBox1.GotFocus
    '    mSpK.Speak("Please Enter the Project eye d")
    'End Sub

    'Private Sub TextBox2_GotFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TextBox2.GotFocus
    '    mSpK.Speak("Please enter Project Name")
    'End Sub
    'Private Sub TextBox3_GotFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TextBox3.GotFocus
    '    mSpK.Speak("Please Enter the task duration")
    'End Sub
    'Private Sub TextBox4_GotFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TextBox4.GotFocus
    '    mSpK.Speak("Please enter Predecessor")
    'End Sub

    'Private Sub TextBox5_GotFocus(ByVal sender As Object, ByVal e As
System.EventArgs) Handles TextBox5.GotFocus
    '    mSpK.Speak("Please Enter the Resource Name")
    'End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Dim intCurrTask As Integer = 1
        'If (OpenFileDialog1.ShowDialog = DialogResult.OK) Then

        Try
            Dim oStartInfo As ProcessStartInfo = New ProcessStartInfo
            oStartInfo.FileName = "C:\Users\Santhosh
Muthyapu\Desktop\siddartha thummuri files\Sid Project\Siddu\MS Project
Files\Project1.mpp"
            Process.Start(oStartInfo)
            'Dim projectFile As New MSProject.Project
            ''Dim disp As Object = True
            ''Dim templ As Object = False
            '''Dim file
            'projectFile = appProj.Projects.Add()
            'projectFile.Activate()
            'appProj.Visible = True
            'With appProj
            '    .Visible = True
            '    .DisplayAlerts = False
            '    .DisplayPlanningWizard = True
            'End With
            'Dim projectClass As New MSProject.ApplicationClass
            'Dim projFile As New MSProject.ProjectClass
            'projFile.Name = "C:\Users\Santhosh
Muthyapu\Desktop\siddartha thummuri files\Sid Project\Siddu\MS Project
Files\Project1.mpp"
            'projectClass.Projects.Add(True)
            'projectClass.Projects(0).Open(projFile)
            'projFile.Activate()

            Dim ds As New DataSet
            OleDbDataAdapter1.Fill(DataSet11)

            For Index As Integer = 0 To DataSet11.Tables(0).Rows.Count
- 1
                AddTask(DataSet11.Tables(0).Rows(Index).Item(0),
DataSet11.Tables(0).Rows(Index).Item(1),
DataSet11.Tables(0).Rows(Index).Item(2),
DataSet11.Tables(0).Rows(Index).Item(3),
DataSet11.Tables(0).Rows(Index).Item(4),
DataSet11.Tables(0).Rows(Index).Item(5),
DataSet11.Tables(0).Rows(Index).Item(6)) ',
DataSet11.Tables(0).Rows(Index).Item(7))

            Next
            Dim oSpeechtotext As New MS_Project_interface.SpeechtoText
            'oSpeechtotext.Invoke()
            'oSpeechtotext.Parent = Me

            oSpeechtotext.Show()

            Me.Hide()
        Catch ex As Exception
            MsgBox(ex.Message)
            'Me.Dispose()
            Me.Close()
        End Try
        'End If
    End Sub
    Public Sub AddTask(ByVal Duration As Integer, ByVal FinishDate As
String, ByVal Predecessors As Integer, ByVal ProjectID As String, ByVal
Resource As String, ByVal StartDate As String, ByVal TaskName As
String) ', ByVal EmailAddress As String)
        appProj.SelectTaskField(1, "name")
        'appProj.SetTaskField("ID", ProjectID)
        appProj.SetTaskField("Name", TaskName)
        appProj.SetTaskField("Duration", Duration)
        appProj.SetTaskField("Start", StartDate)
        appProj.SetTaskField("Finish", FinishDate)
        appProj.SetTaskField("Predecessors", Predecessors)
        appProj.SetTaskField("Resource Names", Resource)
        '        appProj.SetTaskField("EmailAddress", EmailAddress)
    End Sub

    Private Sub OleDbDataAdapter1_RowUpdated(ByVal sender As
System.Object, ByVal e As System.Data.OleDb.OleDbRowUpdatedEventArgs)
Handles OleDbDataAdapter1.RowUpdated

    End Sub


    Private Sub TextBox1_Validating(ByVal sender As Object, ByVal e As
System.ComponentModel.CancelEventArgs) Handles TextBox1.Validating
        If TextBox1.Text.Length <> 5 Then
            'mSpK.Speak("Please enter a 5 digit number for project id")
            Exit Sub
        End If
        Dim i As Integer
        For i = 0 To DataSet11.Table1.Rows.Count - 1
            If TextBox1.Text =
DataSet11.Table1.Rows(i).Item("ProjectID") Then
                TextBox1.Text = ""
                TextBox1.Focus()
                'mSpK.Speak("Project ID entered already exists. Please
enter a new Project ID")
                Exit For
            End If
        Next
    End Sub

    Private Sub DateTimePicker2_ValueChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DateTimePicker2.ValueChanged
        Dim i1 As Integer
        Dim oTS As TimeSpan
        Dim oDateStart, oDateEnd As Date
        oDateStart = DateTimePicker1.Text
        oDateEnd = DateTimePicker2.Text
        oTS = oDateEnd.Subtract(oDateStart)
        TextBox3.Text = oTS.Days + 1
    End Sub

    Private Sub ShowErrMsg()

        ' Declare identifiers:
        Const NL = vbNewLine
        Dim T As String

        T = "Desc: " & Err.Description & NL
        T = T & "Err #: " & Err.Number
        MsgBox(T, vbExclamation, "Run-Time Error")
        End
    End Sub

    'Private Sub txtEmailAddress_GotFocus(ByVal sender As Object, ByVal
e As System.EventArgs) Handles txtEmailAddress.GotFocus
    '    mSpK.Speak("Please Enter the Email address")
    'End Sub
End Class

Public Class SpeechtoText
    Inherits System.Windows.Forms.Form
    Private mSpK As AgentObjects.IAgentCtlCharacter
    Dim sMinutesPath As String

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub


    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim tblProject As DataTable = New DataTable
        tblProject = Me.DataSet21.Tables("Table2")
        Dim iCount As Integer
        For iCount = 0 To DataSet21.Table2.Rows.Count - 1

Combobox1.Items.Add(DataSet21.Table2.Rows(iCount).Item("Project ID"))
        Next

        Dim rowNewProject As DataRow
        rowNewProject = tblProject.NewRow()

        'rowNewProject("Project ID") = Me.TextBox1.Text
        'rowNewProject("Task Name") = Me.TextBox2.Text
        'rowNewProject("Meeting Number") = Me.TextBox3.Text
        rowNewProject("Date") = Me.DateTimePicker1.Text
        rowNewProject("Minutes") = sMinutesPath

        tblProject.Rows.Add(rowNewProject)

        Me.OleDbDataAdapter1.Update(Me.DataSet21)


        'Me.TextBox1.Text = ""
        '  Me.TextBox3.Text = ""
        'Me.TextBox4.Text = ""
        Me.DateTimePicker1.ResetText()
        'Me.TextBox1.Focus()
    End Sub


    Private Sub SpeechtoText_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load
        'AxAgent1.Characters.Load("Merlin", "C:\i386\merlin.acs")
        'mSpK = AxAgent1.Characters("Merlin")

        'mSpK.Show()
        'mSpK.Speak("Please enter the Following details to create a
database entry")
        'mSpK.Hide()
        OleDbDataAdapter1.Fill(DataSet21)
        Dim iCount As Integer
        For iCount = 0 To DataSet21.Table2.Rows.Count - 1

Combobox1.Items.Add(DataSet21.Table2.Rows(iCount).Item("Project ID"))
        Next


        'Me.TextBox1.Focus()
    End Sub
    'Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
    '    Dim ds As New DataSet
    '    OleDbDataAdapter1.Fill(ds)
    '    Dim values As Byte()
    '    values = ds.Tables(0).Rows(0).Item("minutes")
    '    RichTextBox1.Text = ds.Tables(0).Rows(0).Item("minutes")

    'End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Shell("C:\Users\Santhosh Muthyapu\Desktop\siddartha thummuri
files\Sid Project\Siddu\Projectcommand.exe " + Combobox1.Text,
AppWinStyle.NormalFocus, False, -1)


        Dim sqf As New MS_Project_interface.StatusQueryForm
        sqf.Show()
        Me.Hide()
    End Sub


    Private Sub Combobox1_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Combobox1.SelectedIndexChanged
        Dim iCount, iCountMeeting As Integer
        iCountMeeting = 0
        For iCount = 0 To DataSet21.Table2.Rows.Count - 1
            If DataSet21.Table2.Rows(iCount).Item("Project ID") =
Combobox1.Text Then

ComboBox2.Items.Add(DataSet21.Table2.Rows(iCountMeeting).Item("Meeting
Number"))
                iCountMeeting = iCountMeeting + 1
            End If
        Next
        ComboBox2.Items.Add("New Meeting")
    End Sub


End Class

Imports System.Object

Public Class Speech
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()
        'Add any initialization after the InitializeComponent() call

    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        OleDbDataAdapter1.Fill(DataSet21)
        TextBox1.Text = DataSet21.Table2.Rows(0).Item("Project ID")
        TextBox2.Text = DataSet21.Table2.Rows(0).Item("Meeting Number")
        'Dim valuesss() As Byte
        Dim oAccess As Microsoft.Office.Interop.MSProject.Application()

        'Dim abcd As System.Object

        'abcd = DataSet21.Table2.Rows(0).Item("Minutes")


        Dim abc As System.Object

        abc = DataSet21.Table2.Rows(0).Item("Minutes")
        'RichTextBox1.Text = T(abc)
        MsgBox(abc.ToString())
        ToString()
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click

        RichTextBox1.SaveFile("current.rtf",
RichTextBoxStreamType.PlainText)

    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        'RichTextBox2.LoadFile("current.rtf",
RichTextBoxStreamType.PlainText)

    End Sub

    Private Sub AxOleInstall1_OnProgress(ByVal sender As System.Object,
ByVal e As AxOLEPRNLib._InstallEvent_OnProgressEvent) Handles
AxOleInstall1.OnProgress

    End Sub
End Class

Public Class StatusQueryForm
    Inherits System.Windows.Forms.Form


#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()
    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As
Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

Private Sub Submit_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Yes.Click
        Dim updateTask As New MS_Project_interface.UpdateTask
        updateTask.Show()
        Me.Hide()
    End Sub

    Private Sub StatusQueryForm_Load(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles MyBase.Load
        Dim ds As New DataSet
        OleDbDataAdapter1.Fill(ds)
        DataGrid1.DataSource = ds
    End Sub

    Private Sub NO_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles NO.Click
        Me.Dispose()
    End Sub
End Class

Public Class UpdateTask
    Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "

    Public Sub New()
        MyBase.New()

        'This call is required by the Windows Form Designer.
        InitializeComponent()

        'Add any initialization after the InitializeComponent() call
    End Sub

    'Form overrides dispose to clean up the component list.
    Protected Overloads Overrides Sub Dispose(ByVal disposing As 
Boolean)
        If disposing Then
            If Not (components Is Nothing) Then
                components.Dispose()
            End If
        End If
        MyBase.Dispose(disposing)
    End Sub

    Dim ds As New DataSet
    Private Sub UpdateTask_Load(ByVal sender As System.Object, ByVal e 
As System.EventArgs) Handles MyBase.Load

        Try
            OleDbDataAdapter1.Fill(ds)
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try

        For i As Integer = 0 To ds.Tables(0).Rows.Count - 1
            ComboBox1.Items.Add(ds.Tables(0).Rows(i)(0))
        Next
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As 
System.Object, ByVal e As System.EventArgs) Handles 
ComboBox1.SelectedIndexChanged
        DateTimePicker1.Text = 
ds.Tables(0).Rows(ComboBox1.SelectedIndex)(1)
        DateTimePicker2.Text = 
ds.Tables(0).Rows(ComboBox1.SelectedIndex)(2)
    End Sub

    Private Sub Update_Click(ByVal sender As System.Object, ByVal e As 
System.EventArgs) Handles Update.Click
        DateTimePicker1.Format = DateTimePickerFormat.Short
        DateTimePicker2.Format = DateTimePickerFormat.Short

        Dim strQuery As String = "Update Table1 SET START = '" + 
DateTimePicker1.Text + "' , FINISH = '" + DateTimePicker2.Text + "' 
where [Task Name] = '" + ComboBox1.SelectedItem + "' and ProjectID = " 
+ (ds.Tables(0).Rows(ComboBox1.SelectedIndex)(3)).ToString
        OleDbCommand1.CommandText = strQuery
        Dim oApp As Outlook._Application
        Dim oMsg As Outlook._MailItem

        Dim oAttachs As Outlook.Attachments
        Dim oAttach As Outlook.Attachment
        Try
            OleDbConnection1.Open()
            OleDbCommand1.ExecuteNonQuery()
            OleDbCommand2.CommandText = "SELECT EmailAddress FROM
Table1 WHERE ([Task Name] = '" + ComboBox1.SelectedItem + "' )"
            ' Create an Outlook application.
            oApp = New Outlook.Application

            ' Create a new MailItem.
            oMsg = oApp.CreateItem(Outlook.OlItemType.olMailItem)
            oMsg.Subject = "Demonstration of DBPM"
            oAttachs = oMsg.Attachments
            oMsg.Body = "This is an automated email" & vbCr & vbCr

            ' TODO: Replace with a valid e-mail address.
            oMsg.To = OleDbCommand2.ExecuteScalar()   '"st99d@umr.edu"

            ' Add an attachment
            ' TODO: Replace with a valid attachment path.
            Dim sSource As String = "C:\Users\Santhosh
Muthyapu\Desktop\siddartha thummuri files\Sid Project\Siddu\MS Project
Files\Project1.mpp"
            ' TODO: Replace with attachment name
            Dim sDisplayName As String = "Hello.txt"

            Dim sBodyLen As String = oMsg.Body.Length
            oAttach = oAttachs.Add(sSource, , sBodyLen + 1,
sDisplayName)
            If oMsg.Sent = True Then
                MsgBox("An email was successfully sent")
            End If
            ' Send
            oMsg.Send()


        Catch ex As Exception
            MsgBox(ex.Message)
        Finally
            If OleDbConnection1.State = ConnectionState.Open Then
                OleDbConnection1.Close()
            End If
            ' Clean up
            oApp = Nothing
            oMsg = Nothing
            oAttach = Nothing
            oAttachs = Nothing
        End Try
    End Sub
Private Sub Cancel_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Cancel.Click
        Me.Dispose()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Me.Dispose()
    End Sub
End Class
```

VITA

Siddartha Thummuri (Sid) was born in Karimnagar, India on September, 3rd, 1981. He completed his Bachelor of Technology in Chemical Engineering at Chaitanya Bharati Institute of Technology in Hyderabad, India in May, 2004. Sid started his Master of Science program with the Engineering Management Department at the University of Missouri-Rolla (UMR) in January, 2006. His areas of interests were Organizational Communications and Implementation of new instructional Techniques in Engineering Education.

He worked with Dr. Ray Luechtefeld in his area of interest. During his research, he designed and developed a Novel Project Management Tool called “Dialogue Based Project Management System”. He completed his research under Dr. Ray Luechtefeld, being the author of two journal and one conference paper. He received his Master of Science Degree in December of 2007.