Multi-Year Design Education for Non-MSE Majors, the West Point Experience

Authors:

Joseph F. Puett III, U.S. Military Academy, West Point, NY 10996, dj3482@usma.edu
Douglas J. Wolfe, U.S. Military Academy, West Point, NY 10996, dd3940@usma.edu
Curtis A. Carver Jr., U.S. Military Academy, West Point, NY 10996, zc8177@usma.edu

Abstract — *The pedagogical benefits of teaching design methodologies in ABET accredited curricula are well documented; however, receiving significantly less attention are the benefits of such methods when applied to non-Math/Science/Engineering (MSE) curricula or to entire undergraduate populations over multiple courses and years. This paper reports on the results of such a curriculum recently implemented at the United States Military Academy (USMA). The Military Academy requires all non-ABET majors (approximately two-thirds of the student population) to undertake two courses in Information Technology (IT), one during the freshman year and one during the junior year. The courses are presented at increasing levels of complexity, both focusing on problem-solving and design using IT. Teaching strategies include exposing the students to the underlying physical and mathematical concepts relevant to IT, IT systems' functionality, processes for successfully employing IT, and the importance and implications of IT. Originally, design methodology was taught only during the first course with the expectation students would successfully recall and apply the methodology during the second course. The design methodology taught at the freshman level is a four-step design methodology (analyze the problem, design a solution, implement the solution, test the solution) which can be effectively applied to most everyday problems, not just technical ones. During the junior IT course, students expand upon what they learned in their freshman course and attempt to apply the four-step design methodology to design, build, and test different components of an information system. For a number of reasons, the expectation that students in the second course would successfully recall and apply the design methodology presented in the first course was not met. In response, several changes were incorporated into the second course to reinforce the use of design methodology – changes which have produced substantial improvement in the quality of project products. This paper reports on this case study, identifying and characterizing the reasons why design methodology was not initially employed by the students in the second course and identifying the revisions to the second course that corrected this and reinforced earlier learning from the first course. It also presents an updated paradigm of how design methodology could be successfully presented so that it is embraced by non-MSE majors and presents an analysis of results.*

Index Terms — *Information System Design, Design Education Paradigms, Non-MSE Majors, Core IT Courses*

INTRODUCTION

The United States Military Academy (USMA) has a very specific mission to prepare young men and women for successful careers in an increasingly technological Army. As an undergraduate institution, the curriculum is notably oriented towards developing leaders that can creatively solve complex problems in dynamic situations. One considerable component of this curriculum is a two course core sequence in Information Technology (IT).

While both courses are discussed to some degree in this paper, the paper principally focuses on the second course in the sequence. The paper is organized into an introduction which covers the historical motivation for the two-course sequence, a description of the two courses, and the pedagogy employed in those courses. The introduction is followed by a description of the design methodology taught in the sequence. Next, the paper highlights some of the initial observations from the second course and then discusses the corrections made to the course to correct negative aspects of these observations. Finally, the paper concludes with some lessons learned in implementing the two-course IT sequence.

Historical Motivation

A number of relevant considerations led the Academy to the decision to allocate two semester-long courses to IT, with emphasis on "design" and "creativity" in both. The first consideration was recognition that IT plays an increasing role in the lives and careers of military officers. As [1] points out, human knowledge is currently doubling every five years and that this information explosion is a major driver of global "sociocultural" change. As information technologies advance to keep pace, "design" plays an ever increasing role in both managing/directing this change as well as in leveraging new technologies to solve increasingly complex problems. No where is this more true than in the U.S. Military which is committed to maintaining a technological edge over all potential adversaries. Reference [1] goes even further and posits:

> To the extent that complex adaptive systems are increasingly understood to follow discernible evolutionary pathways, design will become the means by which the conscious evolution of complex adaptive systems, ultimately of socioculture itself, takes place.

The implication of this is that IT and design will be progressively more important factors in problem solving, not just in the military, but in future everyday life. Those adept at design will be able to dynamically "customize" solutions to complex problems – those that cannot are disadvantaged.

A second important consideration that guided the way in which the two-course IT sequence was implemented was recognition of the ever-changing nature of IT. Such continuous change dictates the need for a focus on creativity and life-long learning. Creativity is a highly coveted trait in military academy graduates; history has shown that military leaders with agile intellects and the ability to quickly synthesize creative solutions in hazy, time and resource-constrained environments win wars. Thus, the challenge for the Academy continues to be how best to develop "creativity" in its graduates. Reference [4] notes that Information Technology shares many characteristics with those professions in which "creativity" is an integral part of the profession; these characteristics include: a shifting body of knowledge and a changing work environment. This corresponds closely to the work environment of the military professional where the management of change is an essential and necessary component. Thus, it made sense to use IT courses to reinforce the development of creativity in the West Point student body. However, the changing nature of IT also presents a significant dilemma to the educator. As [3] comments:

> The education of IT professionals becomes more difficult as our curricula seem to become more obsolete each day. As educators, we struggle with the problem of trying to decide which topics to include and their appropriate mastery levels.

This is supported by [1] who observes that because of the rapid transformation of everyday IT and design practice, educational response is required in which "life-long learning" becomes the emergent need. Consequently, in developing our IT courses, we recognized that we needed to focus on skills required by our students that encouraged life-long learning of IT – at the same time recognizing that the majority of these students were not Math-Science-Engineering (MSE) majors and that the principal focus of their chosen education was not in IT. Teaching students to, in the long-term, "teach themselves" about various IT topics became an important component of the sequence.

Description of the Two Course IT Sequence

The first course, IT105 "Introduction to Computing and Information Technology," is a one semester introductory course in Information Technology. All freshmen (approximately 1200 students) take this course which focuses on problem solving and introduces some of the underlying physical and mathematical concepts relevant to IT. A substantial part of the course centers on application of the design process associated with programming in Java. Other topics include IT fundamentals associated with website design and construction, networks, sensor systems, and information assurance.

The second course, IT305 "The Theory and Practice of Military IT Systems," was first taught in the fall of 2003 and is a one semester course required by all juniors that are not majoring in an ABET accredited program (approximately 650 students). The underlying assumption is that those students enrolled in ABET programs are already adequately exposed to IT topics and design. Topics in IT305 include the fundamentals of digitization and information assurance; but more importantly, the design and implementation of databases, networks, and dynamic websites. The course is structured around the acquisition of data, the transmission of data, the processing of data into information, and the display of information. Threaded throughout the course are the topics of problem solving and design, military IT systems, the ethical implications in the use of IT, and the implications of the changing nature of IT upon society and the military.

Breadth versus Depth in IT Topics

One of the biggest challenges associated with the design of our two-course IT sequence was settling on the breadth and depth of IT topics to be presented in the sequence. This challenge was not unique to our institution, as [3] points out:

> The current trend in technical education is to begin with a breadth first approach and then specialize. This approach works to some extent for IT education but as the number of areas of specialization grow, curriculum design becomes more difficult.

However, in our particular case, this challenge was somewhat eased by the fact that our focus was not on developing IT professionals, but instead on providing a foundation for the life-long learning associated with IT needed by the military professional. This distinction allowed us to limit the number of topics addressed in the courses and to provide sufficient depth in these limited topics that allowed students to experience non-trivial IT design problems in each. In the case of IT305, we settled on the topics of network design, database design, and web-based information system design.

Problem Based Learning (PBL)

A fundamental aspect of our course design was our decision to employ Problem Based Learning (PBL) as the pedagogical method to attain our educational goals. We decided that the best reinforcement of IT topics taught in the classroom and the best way to teach design and reward creativity was to provide our students substantive design problems oriented around IT. Our rationale in using PBL to teach IT parallels that of [2] to teach science:

> Once students see that the best scientists (and the best science students) are not the ones who have memorized the most facts but are the ones who apply those facts in the most creative manner, we will have gone a long way toward conveying what science does. Science is, in fact, a problem-solving discipline, and we must shift the paradigm from the accumulation of facts to problem-solving, and PBL can help accomplish this.

Similar to design problems in other engineering disciplines, non-trivial design problems in IT are extremely difficult to scope so that they can be individually completed during a class period. Thus, the key component of IT305 became the out-of-class group IT design problem. To provide real-world relevance, these problems are wrapped within a context of military IT problems requiring considerations of limited time, resources, and active adversaries attempting to disrupt implemented IT services.

THE FOUR-STEP PROBLEM SOLVING METHODOLOGY

Two goals of the two-course IT sequence are to learn about information technology and obtain hands-on experience solving problems using IT. The four-step problem solving methodology taught in the courses is not much different from the methodology that is used in other classes such as calculus, biology, or any of the engineering sciences. The methodology is based on the scientific process of analyze, design, implement, and test. This methodology is first introduced in IT105; but students are expected to recall it in later courses. They are shown how the process also applies to problems that they will need to solve throughout their collegiate and professional lives. The methodology taught to the students is diagramed in Figure 1 and described in the subsections below. Readers experienced in design methodologies may want to skip or skim these subsections and move forward to the next major section of the paper: "Initial Observations of IT305."

Analyze the Problem

In this step we teach the students to decompose the problem and think critically about its nature. The students gain an understanding of the problem by determining the problem's objectives, assumptions, constraints, specified tasks, and implied tasks – together referred to as the "problem specification." This step involves transforming a general problem statement into a problem specification that is more specific and more complete than the original problem statement. The problem specification documents the analysis of the problem:

- Objective(s): students identify the goal(s) or objective(s) of the solution. If they achieve the goal, they have successfully solved the problem.
- Assumptions: students identify all relevant assumptions associated with the problem. Students are taught that assumptions are suppositions about the current or future situation that are assumed to be true in the absence of facts. They take the place of necessary, but unavailable, facts and fill in gaps in the problem. An assumption is appropriate if it meets the conditions of "validity" and "necessity." Validity means the assumption is likely to be true. Necessity is whether or not the assumption is essential to solve the problem. If planning can continue without the assumption, it is not necessary and should be discarded. Assumptions are replaced with facts if possible as more information becomes available.
- Constraints: students identify the problem's constraints. They are taught that a constraint is a design requirement which bounds the potential design solution space and generally takes the form of a requirement to do something or a prohibition on action. It is an element that limits the options, depth, and breadth of the solution in terms of resources (i.e., time, money, space, etc.).

- Specified Tasks: students identify all specified tasks. They are taught that specified tasks are those specifically appearing within the problem statement.
- Implied Tasks: students identify the implied tasks. Implied tasks are those that must be performed to accomplish a specified task, but which are not explicitly stated in the problem. Implied tasks are derived from a detailed analysis of the problem.

Design a Solution

Using the problem specification as a starting point, the next step is to design a solution to the problem. The design will not solve the problem any more than a blueprint of a bridge allows someone to cross a river, but it serves as a guide when implementing the solution. The design step involves creating different design artifacts depending on the type of problem being solved. For instance, in creating a website, the design artifacts might include web "site maps" (see Figure 2 for a simple example) and webpage sketches and potential themes. In the case of designing a network, the artifacts might include physical or logical network diagrams (see Figure 3 for an example logical network diagram). Finally, towards the end of the design step, students are taught to develop a test plan against which they will test both the functional and technical requirements of their implementation.

An important distinction is made in this step identifying the difference between "design" and "implementation." The design should not specify an implementation. Rather, the design should be the documentation of the thought process that the students undertake to arrive at a solution to the problem. Using the problem specification as a guide, the student needs to think about alternative solutions, draw diagrams that graphically depict their ideas, write algorithms if the solution involves procedures, and documents discussions with other project members, if any. Just as the problem specification sets the stage for the design, the design sets the stage for the implementation. Therefore, the design must be carefully considered before implementation begins. Much effort, time, and resources will be wasted if design flaws are discovered partway through the solution's implementation.

Given the IT topics covered in IT305 and the design projects the students undertake, the primary design artifacts taught are web site maps, network design diagrams, database design diagrams, and algorithms. However, students are not limited to these artifacts. Students may also draw sketches, construct models, and build prototypes among many other techniques that help them depict their design.

During the last portion of the design step, the students develop a test plan against which they will test their implementation to ensure that it does what it is supposed to do. The test plan varies depending on the type of problem they are trying to solve. For instance, if they are designing and building a web portal, they would test to ensure the site looks and functions correctly. In this case "correctly" means that site looks the way their supervisor (or customer) wanted it, has working navigation links that are easy to use, and can technically be accessed by common web browsers. Students are taught that an excellent starting place for determining which tests are needed in the test plan is to look at their specified and implied tasks. These tasks often lead directly to particular tests in the test plan.

Implement the Solution

The next step in the methodology is to implement the design. If the design requires an information system, then the student builds the database and user interface. If the design requires a local area network, then the student constructs the network.

Students are taught that implementation should occur only after the design is complete. If the design is discovered to be inadequate after implementation begins, then the student should return to the design step to make the necessary changes to the design before continuing with the implementation. While some testing should occur during the implementation, most formal testing outline in the test plan is conducted during the next step in the process. In IT305, students fully implement a web portal, a network, a database, and a web-based information system employing PHP (PHP Hypertext Preprocessor) server-side executables. Instructors emphasize that trying to perform the implementation prior to designing a solution leads to frustration and possible system failure.

Test the Solution

The last step in the process is to test the implementation. The solution must be tested to ensure that it meets the requirements outlined in the problem specification. This step could involve testing the links of a web portal from a different computer, testing the connectivity of a local area network, or testing to ensure that PHP code correctly inputs and retrieves data from a database. When errors are identified, students are instructed to return to a previous step in the process and begin again from there.

In considering what to test, the solution should meet the requirements outlined in the problem specification. Again, what students will need to test depends upon what is being implemented. For a web portal, students are taught to test all links from a computer other than their own to ensure they work as well as to ensure that all images appear as intended. For a network, tests should confirm that network traffic can be passed between the different network components (routers, switches, hosts). For a database, students should populate the tables with enough different types of data to ensure the database works for all types of data and then to test that all forms, queries, and reports function as intended. For PHP code, students test that data is correctly entered into and retrieved from a database and that the display of information matches the intended design.

Sequence and Feedback

Each step of the four-step problem solving methodology is theoretically performed in sequence; however, the complexity of real design problems complicates this. Typically, each step provides feedback to the previous step – when the designer sees a hole in the design it is usually because the specification was incomplete; when the tester finds a failed test it might be caused by a mistake in the design or implementation. New requirements may be discovered during the design, which would then be included in the problem specification. If feedback does occur, then the student is taught to return to previous steps to ensure they are refined before continuing with later steps. The feedback in the four-step problem solving methodology is graphically depicted in Figure 1.

INITIAL OBSERVATIONS OF IT305

As a core IT course, IT305 was first launched in the fall of 2003. While the initial semester proved very successful in meeting the Academy's course and curriculum expectations, there were several noteworthy observations – both positive and negative. On the positive side, the course proved to be an effective addition to the core non-MSE curriculum of the Academy. Student perceptions that the course lacked relevance were largely overcome during the course of the semester. On the negative side, however, the expectation that students would recall and correctly apply the design methodology they had learned during their freshman year in IT105 was not met.

Effectiveness of IT305 as a Core Course

One model of evaluating the effectiveness of a core course is that proposed by [5]. In this model, [5] identifies five features of effective core courses. In each case, IT305 makes a positive contribution towards that feature:

- Creating community through collaborative learning: IT305 uses team problem solving experiences so that students learn from each other and are "guided" not "directed" by the instructor.
- Fostering student ownership of learning which in turn fosters life-long learning: This feature is especially appropriate in dynamic disciplines such as IT.
- Connecting academic ideas with other disciplines and with the real world: IT305 is particularly relevant as the students can leverage what they learn in the class and apply it to other classes – such connections are frequently identified during the semester. Also, since a major thread in the course is to expose the students to the types of Information Technology they will encounter in the Army, the course has significant real-world application to their future careers as Army Officers.
- Evaluating student learning through active experiences: A significant portion (approximately 50%) of student learning in IT305 is evaluated through active, team-based IT design experiences.
- Sharing the experience of the discipline: The problem solving approach used in IT305 encourages non-MSE students to understand and embrace a new perspective when dealing with problems in their other, non-IT disciplines.

Student Perceptions as to Relevance of the Course

By their junior year, the majority of students in IT305 have already chosen non-MSE majors. Thus, at the beginning of the course, there is significant student resistance in appreciating the relevance of the course to their education. To counter this resistance to IT, the course takes a similar approach to that espoused by [2]:

> By emphasizing the process of science, rather than the inundation of specific facts, we can eliminate the fear of science that many students bring with them. We can also alleviate the general disdain for science expressed by some students, who find science dull, repetitive, dehumanizing, and irrelevant to their lives.

While such an approach goes a long way in breaking down student resistance to the course, it has not been totally successful. Student feedback indicates that some students, even at the end of the course, are significantly dubious of the usefulness of the course to their education and their careers. However, we have been greatly encouraged by the amount of feedback (even after only two semesters of teaching the course) where students write back to their instructors and tell them about experiences in which they applied what they learned in IT305 to other courses in their major or during their summer Army internships.

Creativity and Application of Design Methodology

Perhaps the most disappointing observation from the first semester (Fall 2003) in which IT305 was taught was that a substantial number of design teams failed to fully employ the design methodology which they had been taught just two years prior during IT105. Many designs lacked creativity and some design solutions contained errors that should have been caught and corrected had the design methodology been properly applied.

Three main reasons why students fail to incorporate creative processes into their problem solving techniques are identified by [4]: 1) because they are inadequately trained on how to do so, 2) they lack confidence in using the creativity techniques that have been shown, and/or 3) because they are discouraged from being creative. In the case of the initial semester of IT305, our faculty decided that the problem lay mainly with causes 1 and 3. The students forgot the methodology taught to them previously and the structure and feedback of the problems themselves failed to encourage application of the proper process. It was this observation that led to the main course revisions when IT305 was taught the following semester.

IT305 COURSE REVISIONS

When IT305 was taught for the second semester in Spring 2004, three significant improvements were implemented with the aim of improving student application of the design process. These three improvements included adding an early lesson in the course and practical exercise which reviewed the design process previously taught in IT105, restructuring many of the PBL exercises throughout the course so that artifacts of the complete design process were present, and finally, introducing online “In-Progress-Reviews” for the major team-oriented IT design problems in the course.

Review Design / Problem Solving Process

In the second semester in which IT305 was taught, the entire second lesson was devoted to the review of design and the problem solving process. In preparation for the lesson, the students reviewed the methodology they had been taught during IT105. During the lesson, this review was reinforced with a practical exercise which required the students to complete a problem specification and design of a simple organization’s informational website. After class, the students completed the implementation of the website using HTML and then tested the site using the test plan they developed in class during the design step of the process. Our experience during the semester was that this single lesson devoted to the review of methodology was sufficient to allow the students to initially recall the methodology, but that additional reinforcement was required in order for them to be able to consistently apply it.

Exercises Rewritten to Emphasize “Process”

In addition to adding a specific lesson devoted to methodology, most of the practical exercises previously used in the course were rewritten to explicitly reflect where that particular activity fit within the overall problem solving process. For instance, when a particular exercise required implementation (for example, the physical implementation of a network), the students were provided artifacts from the early portions of the process (the problem’s specification for the network, the logical and physical network diagrams, and the test plan for the network). Providing these additional artifacts reinforced in the students’ minds the importance and relevance of the overall methodology.

Online In-Progress-Reviews (IPRs)

The most successful course correction made during the second semester was the additional requirement for students to complete online In-Progress Reviews (IPRs) for their teams’ IT design projects. The IPR is a type of meeting used widely in the Army in a number of different formats. In most respects it resembles an engineering design review; however, because IT305 focuses on military IT, the use of an Army meeting construct added relevance to the design problems. In the Army, the main purpose of an IPR is to update the supervisor as to the status of a particular project. The format of the IPR is based on the complexity, size, and importance of the project, the degree to which the supervisor requires updates, and the amount of time the supervisor has to consider those updates. Examples of IPR formats include:

- Informal Notes and Emails: periodic notes from the project team to the supervisor providing updates on progress of the project. These notes might even be handwritten or sent as email. Students are instructed that this format is good for less important projects when the supervisor does not have time to meet directly with the project team.
- Formal Memorandums: a periodic formal memorandum that specifically states the progress of the project as of a given date. Students are taught that this format is good for important projects when a formal chain of progress is necessary.
- On-Line IPRs: a website that the project team maintains and updates as progress on the project changes. This form of IPR is excellent when the supervisor does not have time to hold meetings to check progress but might need to be able to frequently check on the progress of a project or check on it from multiple/diverse locations. Another advantage of this format is that multiple people/agencies can check on the progress of the project concurrently. A disadvantage of this form of IPR is that the IPR website has to be frequently updated to be kept up-to-date.
- Desk-side Briefings: a common form of IPR in which the project team provides a desk-side brief (e.g., using PowerPoint slides, a written outline, etc.) to the supervisor.
- Formal Briefings: probably the most common form of Army IPR in which a formal meeting of all project participants and stakeholders is held on a regular basis (weekly, monthly, etc.). Typically, the project team develops a formal agenda, provides a briefing or leads a discussion based on the agenda, takes notes, and after the IPR provides minutes to all participants.

Students were taught that any IPR should include the following key points:

- Task Analysis: Has the design team identified all of the stated and (just as importantly) implied tasks associated with the project?
- Plan: Does the team have a plan for completing these tasks? Which tasks can be done concurrently; do any tasks have to be done in a particular order? On what dates does the team intend to have particular tasks completed?
- Task Progress: Has the team made the progress intended? What is the status of completion for large tasks? This status might be in the form of color coding (red, yellow, green) or percentage (10%, 40%, 55%, 100%).
- Problems: What problems are holding the team back from completing the project? What can the supervisor do to help?
- Briefing Style: Your supervisor expects a professional briefing – take charge. Do not expect the supervisor to lead the team through the project; however, the team should leave the supervisor an opportunity to ask questions. The design team is taught that everything (the briefing, the slides, any written outline, or an online IPR website) directly reflects on the team and the supervisor's impression of how well the team is handling the project.

As a second semester change to IT305, project design teams were required to post an online IPR approximately one week before the project was due. An example of a project team online IPR for a database design project is shown in Figure 4. This online IPR required the team to complete and deliver evidence of completion of steps 1 and 2 of the problem solving process (the problem specification and the design). The usefulness of the online IPR during the semester was significant. Not only did teams start their projects earlier, but they delivered excellent artifacts of the design process. Their solutions were more creative and of higher quality. It was clear from their work and their comments that they were starting their projects earlier and were appreciating and internalizing the important role that "process" places in producing quality designs.

CONCLUDING THOUGHTS

The case reported in this paper supports the premise that there is pedagogical benefit for teaching design methodologies to students in non-Math/Science/Engineering (MSE) curricula. The two-course IT sequence recently implemented at the U.S. Military Academy requires all non-ABET majors to undertake two courses in IT, one during the freshman year and one during the junior year. While the courses are presented at increasing levels of complexity, both focus on problem-solving and design using IT. Teaching strategies expose the students to the underlying physical and mathematical concepts relevant to IT, IT systems' functionality, processes for successfully employing IT, and the importance and implications of IT. Originally, design methodology was taught only during the first course with the expectation students would successfully recall and apply the methodology during the second course; however, this expectation was not met. In response, several improvements were incorporated into the second course to reinforce the use of design methodology. These improvements included adding an early lesson in the course and practical exercise which reviewed the design process previously taught in the first course, restructuring many of the problem based learning exercises throughout the course so that artifacts of the complete design process were always present, and finally, introducing online "In-Progress-Reviews" for the major team-oriented IT design problems in the course. These changes have produced substantial improvement in the quality and creativity of project products.

REFERENCES

[1] Broadbent, John A. and Cross, Nigel; "Design Education in the Information Age," *Journal of Engineering Design*, Vol. 14 Issue 4, December 2003, pp 439-447.

[2] Keller, George E. III, "Using Problem-Based and Active Learning in an Interdisciplinary Science Course for Non-Science Majors" *The Journal of General Education*, Volume 51, Number 4, 2002, pp 272-281.

[3] Owen, William, "Specializing in IT Education," *Proceedings of the 4th Conference on Information Technology Curriculum*, October 2003, pp 135-138.

[4] Sweeney, Robert B., "Creativity in the Information Technology Curriculum Proposal," Proceedings *of the 4th Conference on Information Technology Curriculum*, October 2003, pp 139-141.

[5] Weissman, Julie and Boning, Kenneth J., "Five Features of Effective Core Courses," *The Journal of General Education*, Volume 52 Number 3, 2003, pp150-174.

FIGURES AND TABLES

FIGURE 1
THE FOUR-STEP PROBLEM SOLVING METHODOLOGY

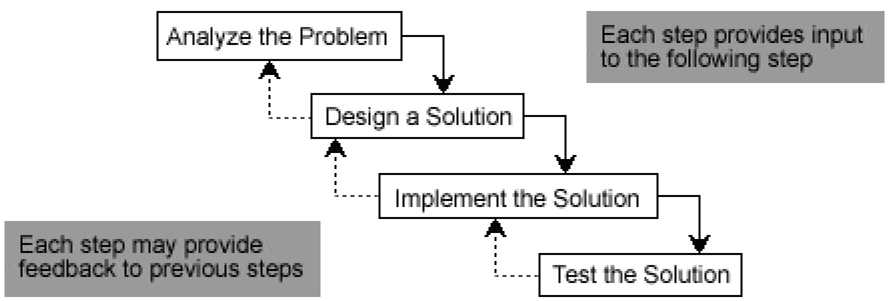

FIGURE 2
WEB SITE MAP EXAMPLE

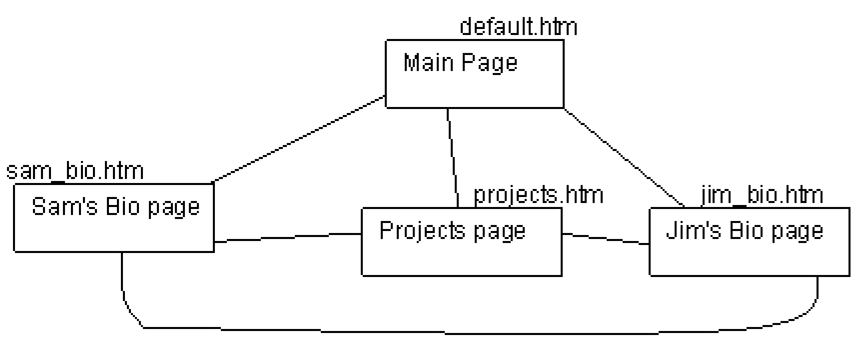

FIGURE 3
LOGICAL NETWORK DIAGRAM EXAMPLE

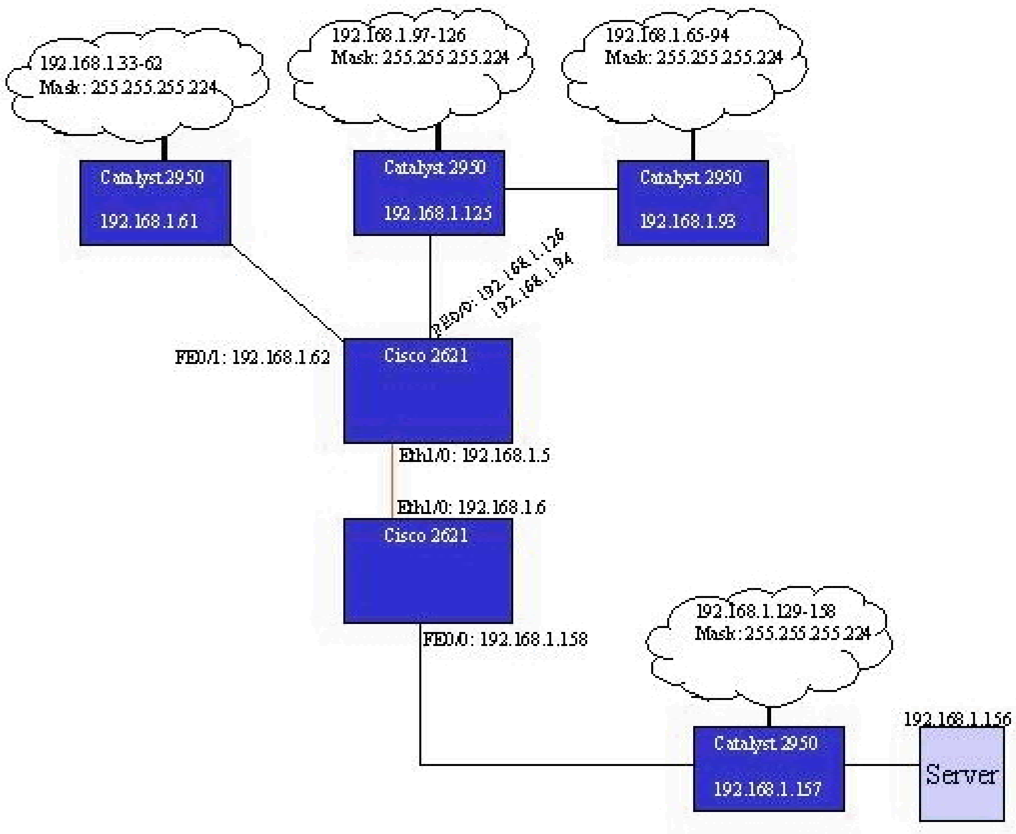

FIGURE 4
ONLINE IPR EXAMPLE

Database IPR

Last Updated: 17 Mar 2004

Objective: Design a database for use by the HQ elements of the 3rd Brigade 82nd Airborne Division (505 PIR).

Constraints: i. The database must be capable of tracking the personnel status of all soldiers in the Brigade
ii. The database must be capable of tracking the status of all major combat systems in the Brigade down to the Company level
iii. The database must be capable of tracking all requested and actual air missions in support of all operations within the brigade, down to the company level
iv. The database must be capable of meeting future staff requirement needs
v. The Brigade commander has requested that all information in the database be tied together by unit.
vi. Microsoft Access must be used to create the database.

Assumptions: i. We assume that the Brigade has the computer systems and network available to support the database.
ii. Instruction on use of the database will be provided to all parties that will be required to input data into the database.
iii. We assume that additional queries and reports will be able to be constructed to acquire new information as it is entered into the database.
iv. We assume that proper security precautions will be taken so that all confidential or classified information entered into the database will be properly protected.

Project Plan and Status:

Implied or Specified Task	Description	Status	Expected Completion Date
S1	Create Database segment for S1 to record soldier personnel information	Completed	3/17/04
S2	Create Database segment for S4 to track all major weapons systems in brigade	0%	3/20/04
S3	Create Database segment for S3 to track all air requests made by the brigade	25%	3/19/04
S4	Create weapons query	0%	3/21/04
S5	Create weapons report	0%	3/21/04
S6	Create Personnel input form	Completed	3/17/04
I1	Create Title Page and Works Cited Page for project	50%	3/28/04
I2	Create a problem specification and design for the database	Completed	3/17/04