

Teaching Systems Analysis and Design in a Practical Way: A Collaborative Effort Between Computer Science and Business School

Ken Surendran
Department of Computer Science
Southeast Missouri State University, Cape Girardeau, MO 63703

Ike C. Ehie
Department of Management
Kansas State University
Manhattan, KS 66506.

and

Chellappa Somarajan
Department of Management and Management Information Systems
Southeast Missouri State University, Cape Girardeau, MO 63703.

Abstract

Nowadays, most Management Information Systems (MIS) Curricula include a course in Systems Analysis and Design (SA&D) followed by a project course involving system implementation. The SA&D course focuses on the earlier phases of the System Development Life Cycle (SDLC), and these days, it is delivered using either or both of the Procedure Centric and Object Oriented paradigms. In the Computer Science (CS) Curricula, usually, a two-part Software Engineering (SE) course is offered which consists of analysis and design in the first course and a system development project in the second course. One of the objectives of all these courses is to prepare the students for the real world challenges of system development. For this reason, system development projects are used in these courses for assessment purposes. In this paper, the authors discuss the merits of teaching a cross-listed SA&D course to both the MIS and the Applied CS majors that enhances the simulated real-world experience. First, they discuss how they designed the course for meeting the essential requirements of the two majors and then explain how they made use of the class diversity to simulate real-world experience through team-based project assignments. Based on the students' feedback, the authors share their findings concerning the effectiveness of this strategy and provide suggestions for further improvements.

Keywords: Cross-listed course, Real-world experience, Systems analysis and design

1. INTRODUCTION

Systems Analysis and Design (SA&D) is a core course in the model IS curriculum (Davis 1997). In the IS 2002 Curriculum draft, the SA&D course is spread across three model courses: *Analysis and Logical Design*, *Physical Design and Implementation with DBMS* and *Physical Design and Implementation in Emerging Environments*. At present, the MIS program (Ehie 2002) offered at the authors' university has been designed to have SA&D (that includes both the logical

and physical models) as a core course. A course called *Business Database Systems* and a programming course are prerequisites for the SA&D course. Further, the SA&D course is followed by a project course called *Systems Implementation and Practice*, which essentially caters for the curriculum requirement that calls for the knowledge of systems implementation involving database and distributed applications. The Applied Computer Science majors specializing in IS in this university have a core course titled *Information Systems Analysis and Design* (ISA&D), in addition to *Software*

Engineering, a project-based capstone course required by all CS majors. A second-level Information System course emphasizing the use of databases is the prerequisite for the ISA&D course.

Last year, due to resource constraints, the CS and MIS chairpersons at the University agreed to cross-list the SA&D and ISA&D courses. Such an unusual collaboration between two competing departments offers challenges and opportunities to both students and course facilitators. The main challenge is in ensuring the course meets both the CS and MIS curricula requirements and prepares both majors adequately for their subsequent system implementation courses. The other challenge is to diffuse the possible cultural tension between the two differently - Business and IT – focused majors during the class sessions. The main opportunity lies in constructively using the same diversity to simulate the real-world situation, where the students with different interests work together to achieve the common course objectives.

The next section is concerned with the details of the cross-listed course, elaborating on how its aims and contents meet the two curricula requirements. One of the strategies used was to compose class teams consisting of students from both the majors to work on a semester-long project, which accounts for 50% of the course grade. In the third section, the authors discuss how such composite teams functioned synergistically, pooling their skills in realizing the project objectives and in analyzing mini cases given in the textbook (Dennis 2000). Specific feedback from the participating students was collected to determine the effectiveness of the collaborative project approach. Further, the graduating seniors were asked, during their exit interviews, to share their concerns regarding this cross-listed course. In the fourth section, the authors present a summary of these feedbacks. Finally, refinements to this course and the possibility of extending such a cross-listed SA&D course for all the CS majors are considered.

2. COURSE OVERVIEW

The knowledge elements of the cross-listed SA&D course pertain to the principles and techniques used in the analysis and design aspects of system development. The course lets the participants apply the techniques and tools of the procedure-centric structured methodology for producing the intermediary system artifacts (from inception to design) of software development. In addition to the structured paradigm (which is considered in detail), some elements of object oriented (OO) paradigm are also introduced so as to develop skills in the use of an UML-compliant tool (Rational Rose) for documenting system artifacts. For the sake of completeness, the participants produce prototypes based on the designs.

The course outline, by week, is given in Table-1. The last column in the Table shows the corresponding chapter references in the textbook (Dennis 2000). The course, spread over 17 weeks, covers all the core analysis and design activities in detail. In addition, topics on planning and implementation phases are added. Thus the course addresses four areas:

1. System development project planning (with focus on preparing a proposal and feasibility analysis)
2. Requirements specification (with focus on information gathering and building process & entity models)
3. Design Specification (with focus on system architecture, Input/Output design, database schema, and structure chart)
4. Implementation (with focus on creating user interfaces and prototypes, planning installation and training)

The business value of an information system is emphasized in the planning and requirements specification phases. Techniques for information gathering and models like Data Flow Diagrams and Entity Relationship diagrams are introduced in the analysis phase. The distinction between architectural design and detailed design is then emphasized. During the architectural design, network models are discussed and alternative development plans are examined. In the detailed design, schema for the optimized database, structure chart (guided by the concepts of cohesion and coupling) for the modules, test cases, and user interface designs are covered. The implementation phase is concerned with test plans, creation of user interfaces and installation plans. The implementation here, however, is limited to building a prototype.

The authors consider it important to teach the course using just one paradigm in depth (in this case, the procedure-centric / structured paradigm) and provide an overview of the object-oriented paradigm. Only two weeks (7th and 13th) are allocated for OO paradigm. The OO concepts, UML, use-case model are introduced in one week and, in the other, the heuristics of finding analysis classes using interaction diagrams pertaining to scenarios of use cases are discussed. The students are encouraged to learn and use the various UML diagrams on their own outside the class time, using the on-line Rational Rose workbook.

Usually, the SA&D course is primarily concerned with requirements specification and logical and physical models. However, by stretching the course on either side with suitable SDLC phases, both the *business* focus and the *IT solution* focus required for the two majors are provided (see Figure 1). The business focus includes feasibility analysis, evaluation of business process (simple mechanization, improvement, or re-engineering), and preparing the organization for

Week	Topics and Activities	Text
1	Course overview, SDLC, System development paradigms and methodologies [group assignment – team formation]	Chap 1
2	Roles and skills, Proposal for business solution based on IT	Chap 2
3	Proposal evaluation Analysis process [Test-1]	Chap 2, 4
4	Current system & improvements, BPR [Project: Proposal and feasibility report due]	Chap 4, 5
5	Information gathering, Process model Context and DFD	Chap 5, 6
6	Data Model, ERD. [Test 2]	Chap 7
7	<i>UML, Use cases, Introduction to Rational Rose</i>	Chap 6, 16
8	[Requirements Spec. due] Mid-term Exam - Chap 1, 2, 4-7.	
9	General and detailed system designs; Architecture design	Chap 7, 9
10	Design strategies, Revising DFDs and ERDs	Chap 8
11	Program design and Data Storage design (schema)	Chap 13,12
12	User Interface (I/O) designs [Test 3]	Chap 10 -11
13	<i>Finding classes, Class design, Relationships and operations</i> [Design Documents due]	Chapter 16
14	Issues in Implementation	Chap 14,15
15	Course Review and Project Review	
16	[Present prototypes – Peer Reviewed]	
17	Final Exam - Chapters 8-13, 16	

Table 1: SA&D Course Outline

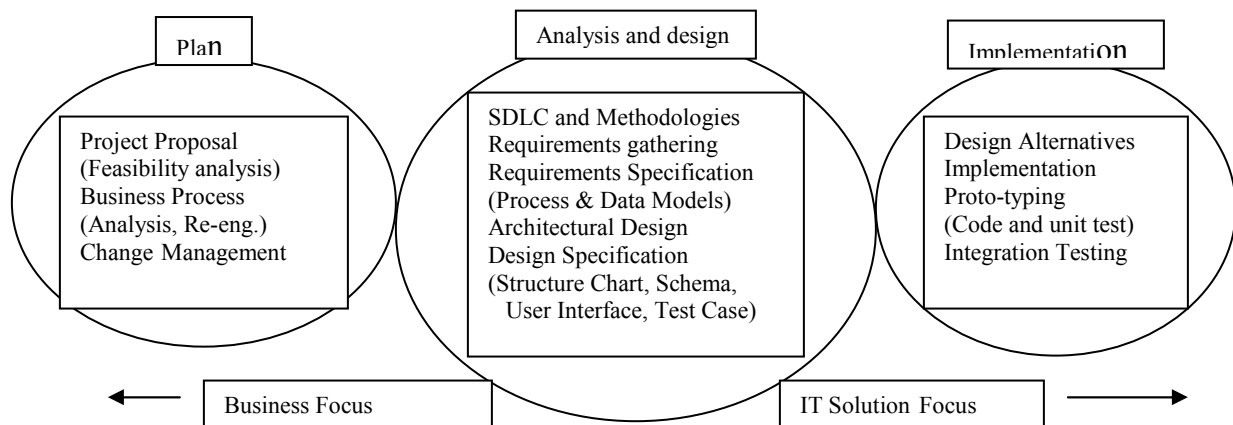


Figure 1 Business-cum-IT Solution focused SA&D Course Model

realizing the value of the new system. These pre-analysis and evaluation activities are of greater value to the MIS majors. The IT solution focus, on the other hand, is concerned with evaluating design alternatives, implementing (making a prototype or a proof of concept) a system design, and planning for a quality product. These post-analysis and design activities are of greater value to the CS majors. By including the Plan and Implementation phases to the core Analysis and Design (as indicated in Figure 1), the two majors are able to appreciate the value their counterparts provide to the overall development of an IT-based business

solution. A textbook like the one used for the course (Dennis 2000) does a good job on the two extended phases as well.

3. COMPOSITE TEAMS

The learning strategies used in this course aim at developing the students' communication skills and working in teams. A group project consisting of four phases was chosen to serve both the objectives. The deliverables involved considerable writing and a short oral presentation. This project assessment accounted for

50% of the final grade, while several individual assessments (tests and exams) accounted for the balance. The team size for the project was limited to 4 or 5 students to match the workload of the assignment (each student was expected to put in about 50 hours of effort for all the four phases). The team selection was left to the students. (However, each team should have ideally two CS and two MIS students.) It was mandatory for each team to have at least one CS and one MIS student.

Project Deliverables

Using team projects is common in software engineering courses (Surendran 2001) as it offers considerable opportunities in simulating real-world experience, especially if the teams get to work on client-sponsored projects. In the SA&D course, all the teams worked on a single project with the instructor as the client. During the class, time was allocated for the teams to interview the client. Also, e-mail was used extensively for gathering information. The four deliverables of the team project were:

1. Project Proposal (10%)
2. Requirements Specification (15%)
3. Design Specification (15%)
4. Prototype for selected subsystems (10%)

Knowledge Sharing

Detailed list of contents for the deliverables in each phase was suggested. For instance, the following were listed for phase-1 deliverables:

1. Current Situation (background, business needs; constraints)
2. System Objectives (functionalities)
3. Solution Description (approaches and alternatives)
4. Resources needed (people, training, equipment)
5. Cost estimates, anticipated benefits leading to economical feasibility
6. Technical and Operational (organizational) feasibilities
7. Schedule (major activities and timelines; use MS Project to draw a Gantt chart)
8. Conclusion (risks, if any; additional notes)

Similar lists with pertinent tasks were provided for the remaining three phases as well. These lists provided the necessary guidelines.

Each team had a leader for each phase. The teams met outside the class to assess and allocate work. In the above list for phase-1, tasks such as *system objects* and *economic feasibility* require business knowledge and tasks such as *solution description* and *technical feasibility* require technical knowledge. Since the teams collectively had members with both business and technical background, they were able to learn from each other and work collaboratively to complete the tasks.

Apart from the project, the teams also worked on mini cases that were discussed in class. Here too, having a

composite team with multiple background helped in enhancing their overall learning. They were able to analyze the mini cases from both the business and solution perspectives.

4. STUDENT FEEDBACKS AND ANALYSIS

A questionnaire at the end of the semester was used as the main instrument for gathering feedback from students concerning the cross-listed SA&D course. In addition, there were two other secondary feedbacks: concerns regarding this cross-listed course during the graduating seniors' exit interviews and individual reflections concerning the group project provided as part of the deliverables for the last phase. About 90% of the seniors made very positive remarks, in particular by MIS students working with the CS majors in their teams. Some of them even had used the project documentation as part of their job interview portfolio. At least, three student teams from the SA&D class entered for the student competition paper at the AITP 2002 annual convention. These students felt very comfortable with the knowledge gained in the course to enter for a national competition. All feedback obtained during the exit interview spoke very laudably about the cross-listed course. The MIS students felt that the make-up of the teams with CS majors allowed them to experience the actual implementation of an IT Project.

The individual project reflections were positive, of which some addressed the soft-skills. The following three sample individual reflections speak for themselves:

(1) This was a good project and helped feel more secure about my degree choice, I have been saying for the past two semesters that I am an MIS major and I really didn't know what it meant to be. I know now that I want to be a business analyst. (2) The project was a way to learn how to work as a team and break down a project between members so that each member was able to do tasks that are best suited to him / her. (3) We learned teamwork, time management, (how to) compromise, and hopefully taught each other things along the way. When doing a project of this size, I found that if you stay actively involved you would know what is going on, and gain a lot more out of this experience.

Questionnaire Feedback

There were some 12 questions covering the following aspects of the team project: phasing, team composition, roles played, workload balance, skill balance, functioning of the team, learning aspects, real-world experience simulation, and overall experience. At this stage, no quantitative methods were used since the main objective was to explore the value of this approach and to identify problems, if any. Since the questionnaire was optional, only ten CS and ten MIS majors in a class of 40 had responded. The analysis indicated that more than 80% liked the idea of working in teams with mixed skills. All liked the idea of phasing, except that a few

(belonging to teams which did not have a perfect balance) had complained about lack time for the final phase. About 75% considered that the project offered them real-world experience (based on the fact they worked on a single project with phased delivery, had to work with people with different personalities and levels of commitment, experienced the stresses and strains of working in a group, delivered something tangible at the end of the day, and dealt with real requirements). Of the eight teams, two seemed to have had problems in balancing the workload. During the project, every one had taken on at least three different roles from among the following: business analyst, systems analyst, process designer, database designer, interface designer, programmer, team coordinator, and researcher. Learning from others was limited to one of filling the gaps (lacking prerequisites such as database tools, network basics) and about one third stated they learned something new from other team members. On the whole, 90% considered the mixed team project a positive experience. Some of the problems identified in the questionnaire include: lack of time for the last phase, lack of good comprehensive examples for the deliverables and schedule conflicts for outside-class team meetings.

5. CONCLUSION

There are several issues, which play a major role in making the project assessment in a cross-listed course a valuable experience to the students. As far as possible, the teams should be balanced in terms of skills. Any weak team (usually identified during phase-1 itself) should be disbanded and members reshuffled to other teams. It is critical to set aside time during class as the project moves into later phases since teams may have problems in scheduling their team meetings. Using on-line resources (like discussion boards for each group) is essential. Phased projects have their own challenges, since output of the previous stage is used as input in the following phase. It is important to scope the activities for each phase, irrespective of what was done in the previous phase, and to post the best solutions of the previous phase so that all the teams can work from a common base. There is a need to be in constant touch with the team leaders to ensure all members are actively participating. For every deliverable, each team should be asked to provide a 'who did what' list. The objective is to have an overall workload balance (not necessarily for each phase). Equity is always an issue in dealing with group work. By having several staged deliverables, the impact of equity issue can be minimized. The previous year's project assignments and consolidated sample deliverables could be posted on the web to serve as comprehensive examples. In order to have more time for the last phase, the whole project assignment could be started during the first week itself. As new topics are introduced in the class, their relevance must be explained in light of the project, in addition to other

examples and mini cases. This reinforces the application of new concepts.

Carrying out formal reviews at the end of each phase is essential. In real world, the results of analysis are reviewed with the clients and walkthrough are conducted for reviewing design specifications and codes. One way to achieve this is to let groups swap artifacts after each phase to provide formative reviews.

The extensions used in the cross-listed SA&D course offer additional learning opportunities to both the CS and MIS majors. It allows the CS majors to learn about business value of the IT solution and the MIS majors to appreciate the challenges of implementing an IT solution. One of the criticisms the CS majors face is that they are not well prepared to appreciate the business environment. The CS curriculum at this university is being revised to convert the current SE course into a client-sponsored system development course and to include the cross-listed SA&D course as its prerequisite. System development being a communication intensive process, the SA&D course should prepare the CS students well for the intended client-sponsored capstone project. In view of the OO emphasis placed in the CS curriculum, the SA&D course may then be taught using primarily the OO paradigm with an overview on procedure centric.

The usual management rhetoric *breaking the departmental silos* need not be limited to industry alone. A course like SA&D, which interfaces CS and MIS majors, lends itself to exploring those possibilities in the academia as well.

REFERENCES

- Davis, G.B., John.J. Gorgone, D. Couger, D. Feinstein, and H. Longenecker Jr. ,1997. *IS 1997 model curriculum and guidelines for undergraduate degree in information systems, ACM, AIS and AITP*. Association for Computing Machinery.
- Dennis, A. and B. H. Wixom, 2000, *Systems Analysis and Design: An Applied Approach*. John Wiley & Sons.
- Ehie, I.C., 2002, "Developing a Management Information Systems (MIS) Curriculum: Perspectives From MIS Practitioners," *Journal of Education for Business*, 77:3, pp. 151-158.
- Surendran, K. and Frank H. Young, 2001, "Teaching Software Engineering in a Practical Way," *The New Zealand Journal of Applied Computing and Information Technology*, 5:2, pp. 75-79.