

An Analysis and Design Case Suitable for Procedural and Object Oriented Approaches

Ken Surendran and David Naugler
{ksurendran, dnaugler}@semo.edu
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, MO 63701

Abstract

The authors present a case suitable for a semester long systems analysis and design project in an upper level course in Information Systems or Computer Science. They further illustrate how this case was used in an IS course using the procedural approach and in a CS course using the object oriented approach. The Multimedia Education Center (MEC) case grew out of one of the authors' experience in developing customized courseware for an industrial client. For IS students, MEC has a front-end proposal and feasibility analysis segment and, for CS students, an in depth requirements analysis component. Appropriate deliverables are specified for the two approaches. Although MEC has three delivery stages under either approach, it can be extended to include a prototype system implementation. Since the case simulates real life experiences such as working in teams, taking on different roles in the system development process, and using information gathering techniques with the instructor playing the role of the client, it prepares students well for a subsequent capstone course, where the students analyze, design, and implement a client-sponsored system development project. The student feedback from both the programs confirms this observation.

Keywords: Case study, Object Oriented and Procedural Systems Analysis and Design

1. INTRODUCTION

Most textbooks in Systems Analysis and Design (SA&D) (e.g., Dennis *et al.*, 2006), provide case studies to illustrate and reinforce the application of concepts. The same trend is observed in Software Engineering (SE) textbooks as well (e.g., Bruegge and Dutoit, 2004). The concepts and heuristics the students learned and the skills developed in these courses are better retained when the students carryout a semester-long project concurrently. This learning experience is further enhanced when the students carry out a client-sponsored system development project in a next course. For instance, the IS2002 Model Curriculum (Gorgone *et al.*, 2003) recommends a capstone course titled Project

Management and Practice. At the authors' institution, both Management Information Systems (MIS) and the Computer Science (CS) students have a concept course that teaches systems analysis and design followed by a capstone course (called Capstone Experience in the CS program and System Implementation and Practice in the MIS program) involving client sponsored system development projects. Currently, the main difference is that the MIS students use the procedural approach and the CS students the object oriented approach.

The use of case studies is a very common approach in most upper level IS courses (Hackney, *et al.*, 2003). Case books (for example Miller, 2007) and teaching cases published in IS education journals like

Journal of Information System Education are good resources.

Characteristics of a good teaching case (Cappel *et. al.*, 2003) are: (1) the case addresses an IS subject matter, (2) it has a clear sense of purpose, (3) it provides realism, (4) it is of appropriate length, (5) it is objective in presentation and tone, (6) it has a hook, (7) it addresses a timely topic, and (8) it has been pre-tested. Here the authors present a case study that grew out of an industrial assignment one of the authors carried out. They endeavor to meet several of the above criteria.

The case presented here was used for the group project (worth about 50% of the course assessment) in both the MIS and CS concept courses. Each group had four or five students. In both courses, the instructor acted as the client and provided supplementary information as required. In the next section, the case is described in detail. The student assignment along with the deliverables for each of the courses is discussed in section 3. Suggestions to case users are provided in section 4. The paper concludes with a discussion on student feedback and the utility of this particular case study.

2. THE MEC CASE

This project is about developing a system for managing a multimedia education center. The case is presented in a format suitable for assigning to students. An outline of the project and the broad requirements are stated in the following. As the project sponsor, the instructor provides additional information or clarifications concerning the requirements by way of answering students' questions. (Time is allocated during class for the students to interview the instructor acting as client.)

2.1 The Case Context

The Multimedia Education Center (MEC) is an entrepreneurial unit in a university that develops computer based training modules on topics in many disciplines such as agriculture, business administration, and computing. These training modules are for non-university clients such as businesses,

government entities, and non-profit organizations. MEC holds the copyright for the modules it develops. The sale of such off-the-shelf modules is MEC's main business. MEC also develops custom-made products to meet special requirements.

To develop a module, MEC allocates a team consisting of a subject matter expert (SME) and module development engineers (MDE). A SME designs modules or components in their subject areas and the MDEs build the modules using a number of authoring and simulation tools. MDEs may use existing modules (or components of the modules) in preparing customized products. MEC needs to update existing modules for maintaining currency and meeting client needs.

2.2 The Organization

Clara Banks, a psychologist by training, is the director of MEC. She is directly supported by five people in MEC: an office manager (Betsy King), a SME coordinator (Austin McDonald), a technology manager (John Joyce), a training manager (Carol Power) and a marketing and sales manager (Adam Hughes). Austin has two full-time SMEs in his team (one in business and the other in technology). Interested faculty members are hired as SMEs (on a part-time basis) for specific contracts. John has five full-time MDEs; he also employs college seniors as needed. Carol, a trainer, has an assistant. Carol hires some faculty members for specific training programs as needed. Adam spends most of his time visiting clients and potential clients. Betsy takes care of routine matters. At times Adam takes SMEs with him for contract negotiations.

2.3 Current System Facility

Every member in MEC has a workstation connected to a LAN with links to the university's mainframe. Managers in MEC keep essential information in their own spreadsheets. The lack of integration makes it difficult to produce consolidated reports. The university finance and accounting (FA) department handles MECs accounting requirements. MEC prepares and sends invoices for products purchased by clients – with copies to FA – and sends payment authorizations to FA for services provided by the faculty and students (part-time). MEC

has access to the related accounting information through its workstations. MEC is, however, not aware of the possibility of a direct interface with the mainframe.

2.4 Problems

MEC is having growing pains. It does not have a computer system for business operations even though it has state-of-the-art systems for making multimedia educational products. Some prospective clients have unconfirmed orders (which must be ready by a deadline). It often takes longer than estimated to determine the nature of a job and to specify the product. The development teams are often unaware of existing modules and components that are very similar to the ones being requested since appropriate records not kept; only an alphabetical component list is available. MEC also buys ready-made components from other companies for use in their own modules, reducing development time considerably. Appropriate product pricing is difficult since component cost estimation is unreliable. An on-line product catalog for both modules and components would alleviate these problems.

2.5 System Request

Clara wants a computer-based solution for effectively supporting MEC operations and, for tactical reasons, she wishes to outsource this project to a computer system consulting company (in this case, the student team). The first stage of this system development initiative, Clara wants a system to support the following four major functions:

- Product Catalog Management (course and module)
- Contract Management (course development / training)
- Resource allocation management (faculty & student)
- Product sales and customer analysis

2.5.1 Product catalog management: Each module addresses one or two topics in a subject area. A module may include video, audio, text and other self-study material – items found in good computer based training programs. A module may be included in several courses (for example, the module on the System Development Life Cycle could be in a course on Systems Analysis and Design

and also a course in Software Engineering). MEC wants to keep the following information for each module: module code, topic, year made (or last updated), author(s), a brief description using (up to ten) keywords, and the current price. Each course needs a unique code and similar information; in addition, a course will have a list of modules (limited to ten) used in it. SMEs should be able to identify modules that are five or more years old for possible update, and be able to search for modules by key word(s).

2.5.2 Contract management: After MEC and the client have signed a contract, the development process goes through the following steps: project plan, architectural design, and implementation (assembling the course material). Austin, Carol, and a suitable faculty member make the initial project plan. The SME (faculty) analyzes requirements and produces an architectural design for the course. John and the SME then identify a team of suitable MDEs to assist in assembling the course material. During the architectural design, the SME may contact the client for additional information. A training component is an optional phase. MEC management wants to keep track of the contract status from initiation to training (or product delivery), and who worked on what and for how long. This means keeping an activity log for each product development project. A product is revised after its first use. After this revision, the course and its modules are added to the catalog. The contract management facility will have the following information: contract code (to identify the contract), client code, start date, required completion date, SME and the MDEs assigned to it, and the development status code. The activity log will contain employee codes (for faculty and students), activity codes, time spent and other details.

2.5.3 Resource allocation management: A system facility is required to keep track of faculty and students who work on various MEC projects. Information on their subject expertise (for faculty) or skills (for students) and availability will be kept. Combined with other systems, MEC could find the courses a faculty has developed or is currently working

on. Also, MEC could check the availability of suitable faculty members.

2.5.4 Product sales and customer analysis: MEC must promote their products. Clara wants a customer database with customer codes, contact details and type of business. Adam wants to track sales. It would be very helpful to know who bought which products when preparing promotional letters for new products. MEC needs management reports containing customer and sales information. MEC should be able to identify products that are not selling and those that sell well. Based on such information, MEC could remove unpopular courses and develop innovative courses in popular areas.

2.5.5 Additional requirements for CS: Clara wants to examine the possibility of generating invoices for sales through the system and automatically sending sales details to the university accounting system. This will require an interface for interacting with the university accounting system.

2.5.6 Quality of service (QoS) requirements: All users of the system must log on to access any of its facilities. System access authorization should be tailored to the user's tasks. Each group of users sees only the facilities they need. It should be possible for all users to access the system from off-campus.

3. COURSES & STUDENT ASSIGNMENT

The SA&D course, taught using procedural approach, includes the following major topics: project proposal (feasibility analysis), system development methodologies, requirements gathering, requirements specification (process and data models), architectural design, and detailed design specifications (structure chart, schema, user interface, and test case). The SE course, taught using object oriented approach, includes the following major topics: system development methodologies, unified modeling language and unified process, use case model (use case diagram), use case analysis (interaction diagrams, identifying analysis classes, view of participating classes), and system and object design (system architecture, patterns, interface,

object, class design, and database design). In both the courses, the students are taught the necessary analysis and design heuristics and concepts appropriate to the approach used in the course.

Case studies such as the one described in section 2 are used for a SA&D or SE project assignment. A group project assignment consisting of three phases accounts for about 50% of the final grade in both the courses. Several individual assessments account for the balance. The deliverables by stage for the two approaches are presented below. These provide guidelines on formatting system documents. Different tools are used in the two approaches for expressing the results of analysis and design. Currently, Rational Rose is used in the SE courses and Visio is used SA&D course. Entity relationship diagrams (ERDs) are used in both courses for data modeling. The deliverables for the three stages under the two approaches are described in detail in the next two subsections.

3.1 Project Deliverables for the Object Oriented Approach

The analysis and design assignment is carried out in three stages: use case model, use case analysis and design specification. Teams should explain what is done in each section of the deliverables referencing model diagrams and should provide a softcopy of all the model diagrams for each stage. All team members should contribute to the production of the deliverables. Summary sheets should indicate each member's contribution. The suggested deliverables for the three stages are discussed below.

3.1.1 Stage I - Deliverables for the use-case model (UCM): In this stage, the system requirements are captured. The system functionalities are described using use cases. It is important that all the members in the development team have a common understanding of the various terms used in the application domain. Hence it is useful to have a glossary of terms at this stage itself. Quality of service aspects (non-functional requirements) are captured in the supplementary specifications. The suggested contents for the group report on use case model include:

- Introduction (Context and Scope of your project)
- Use Case Model (use case diagrams, use case descriptions in the standard format)
- Glossary of terms
- Supplementary Specification

Note: Teams will interview the client (during class sessions) for additional information. Teams are advised to write down the questions they have for the interview when they discuss the case. The teams should seek clarifications on both the functional and QoS requirements.

3.1.2 Stage II - Deliverables for use case analysis (UCA): Each use case is analyzed using the flow descriptions developed under UCM. Appropriate sequence or collaboration diagrams are drawn to identify the participating classes which can take on the responsibilities to fulfill the service specified in the use case. These classes are rationalized and put into packages. The main classes in packages are then shown. A table that maps the quality of service aspects (called analysis mechanisms) with the identified classes is required. The deliverables for the use case analysis are listed below:

- Cover page (project title; authors) and Contents page
- Introduction (modified use case diagram, summary of the contents)
- Interaction diagrams (Collaboration or Sequence) for the main flow and one or two alternate flows for each use case
- View of participating classes for each use case (class diagrams for the use cases).
- Package diagram
- Main class diagram for packages
- List of identified analysis classes with their attributes and responsibilities
- A table showing the mapping of class versus analysis mechanisms.

3.1.3 Stage III - Deliverables for the design specification: In this final stage, all the design activities pertaining to both the architectural and detailed designs are performed. The system architecture, subsystem designs, and detailed class designs, user interface and database designs are presented. Even though the analysis

and design is carried out in the OO paradigm, persistence may have to be realized using a relational database management system rather than an OO database management system. The following is the list of items suggested for design specification:

- Cover page and Contents page Introduction (summary of contents and major design decisions)
- The software architecture
- Sample transformations of packages into subsystems (apply coupling and cohesion principles)
- Sample subsystem design (use sequence diagrams and class diagrams for realizing some operations stated in subsystem interface)
- Patterns or frameworks used
- Object design (provide state diagrams for dynamic objects)
- Final class diagrams (apply inheritance and refine classes)
- Database design (provide normalized ERDs)
- User interface design (include sample forms and reports)
- Conclusion (list individual contribution and time log)
- Softcopy for all model diagrams

3.2. Project Deliverables for the Procedural Approach

The analysis and design phases of the project are carried out in three stages: proposal, requirements analysis, and design specification. Teams should explain what is done in each section of the deliverables referencing model diagrams and should provide a softcopy of all the model diagrams for each stage. All team members should contribute to the production of the deliverables. Summary sheets should indicate each member's contribution. The suggested deliverables for the three stages are discussed below.

3.2.1 Stage I - Deliverables for project proposal with feasibility analysis: Teams prepare a viable IT solution proposal for MEC to support and even improve its operations. Each team costs the solution (covering all aspects) and present a cost-benefit analysis, stating any assumptions (e.g., cost of server or software; systems professionals' time).

Each team should examine the four feasibilities (schedule, economic, technical and operational) of the suggested solution.

The project proposal includes:

- Executive summary (about 200 words describing the context, content, highlights)
- Current Situation (background, business needs; constraints)
- System Objectives (functionalities)
- Solution Description (approaches and alternatives)
- Resources needed (people, training, equipment)
- Cost estimates, anticipated benefits leading to economical feasibility
- Technical and operational (organizational) feasibilities
- Schedule (a Gantt chart)
- Conclusion (risks, if any; additional notes)

(The note under 3.1.1 applies here as well.)

3.2.2 Stage II - Deliverable for the requirements analysis: Teams are expected to prepare a Requirements Specification for the MEC system. The process description and the data descriptions are the main items in this specification. The suggested contents for requirements specification include the following:

- Executive summary (summary of report content and highlights)
- Revised Schedule (for design, implementation – in some detail)
- Process description of the system (using context and data flow diagrams)
- Data description of the system (using ERDs)
- Conclusion (plans on further work)

3.2.3 Stage III - Deliverables for detailed design: The first task at this stage is transforming the logical models to physical models. Teams carry out an optimized database design and a process design. In addition, forms, reports, and displays are designed. Also test plans and important test cases are developed. Suggested contents for design specification include:

- Executive summary (design report summary)
- System Architecture and transformation to physical models
- User interface design (forms, reports and displays)

- Process design (structure chart and pseudo code where needed)
- Database design (third normal form and optimization –database schema)
- Test plans (integration and system testing along with test cases)

3.3.4 Optional Stage IV: Implementation (prototype): A prototype for the Product Catalog and Product Sales & Customer Analysis sub-systems needed by MEC is developed based on the design specifications. The prototype can run on a PC with no separate database or application server. Teams can any tool they are familiar with for developing the prototype.

It is important that the teams develop error-free working prototypes which meet the expectations and requirements of MEC and follow the test plan properly. Teams prepare user notes, which must truly reflect what the system does, and compile all the system documentation. Each team will demonstrate the working system during a class session.

4. SUGGESTIONS TO CASE USERS

The purpose of the case project is to help students apply the heuristics for analysis and design in a systematic manner. In the course they learn these heuristics and the basic concepts used in analysis and design. The case project is a means to let the students work in teams taking on different roles for producing the required system artifacts. With this mind, a few suggestions are provided below about how the case can be used.

4.1 Additional information

At all stages the instructor has to provide additional information and should have several interview sessions for the class to gather additional information concerning the project. Alternatively, the instructor can assign roles (described in section 2.2) to a few students and coach them before class for a comprehensive information gathering session. For the economic feasibility analysis, the instructor might enumerate the benefits of the system (such as increase in sales or cost savings due to new and integrated system features). Teams may be asked to research computer hardware and

software costs and to do financial calculations (such as net present value) using the prevailing bank lending rates. Teams must be asked to state all assumptions they make in their report.

It is important to ensure the students understand that the project is about a management tool, not about operational tools. It should be made clear that system tools already exist for supporting the creation of course modules.

4.2 Flexibility

The scope of the system should be customized for team size, student background, and time availability. For example, if there are accounting majors involved a separate accounting module could replace the interface requirement to the mainframe. Implementation is an option.

4.3 Modeling Tools

In either approach teams should use standard tools for modeling diagrams. One or two lab-sessions for introducing these tools may be needed. A simpler tool such as Visio should be an option for students with a less technical background.

4.4 Example Case Solutions

The instructor could provide, if available, an earlier project in its entirety (problem and essential parts of the solution) as an example. The groups should be assigned to work on exercises relating to the preparation of model diagrams. Since there are several heuristics to learn and apply, such in-class exercises are essential.

4.5 Grading

About 30%-50% of the course grade was assigned to the case project. The team size was limited to four or five students. Since the case has a set of functional requirements, it lends itself nicely for distributing the tasks amongst team members. For grading purposes, each team was asked to list individual contributions. Higher weights were given to the second and third stage deliverables (the ratio for the three stages being 1:2:2). Oral

presentations could be used to supplement the reports. Topics might include context diagrams, 0-level data flow diagrams, entity relationship diagrams, structure charts, and user interfaces for the procedural approach and use case diagrams, selected interaction diagrams, class diagrams, and user interfaces for the OO approach.

5. STUDENT FEEDBACK

Both the CS and MIS majors have their own subsequent capstone project course where students work on client sponsored system development projects. One of the objectives of using such a case study assignment is to prepare the students for successfully completing the client sponsored project.

The MEC case assignment was used with procedural approach in a SA&D course for MIS students and with the OO approach in a SE course for CS students. Surveys were conducted to find out to what extent the two conceptual courses (SE and SA&D) - in which the case project constituted about 50% of the assessment - helped in successfully completing project tasks and in developing the required technical and professional skills (a.k.a. soft skills). Students were asked to indicate their perceptions on a five-point scale, ranging from definitely disagree (value 1) to definitely agree (value 5). Four (out of seven) students from MIS and 13 (out of 19) from CS responded.

Table 1 and 2 show the tasks normally required in client-sponsored project courses and the students' perception on how the respective concept course in which the case project was used helped in learning those tasks. In general, the MIS students seem to think the concept course in which the case was used helped considerably in carrying out several of the activities in their capstone project. The significant ones are planning, requirements specifications and user interface designs. The CS students seem to think the concept course helped them considerably in carrying out the requirements specifications and moderately in other activities. Since there were no real users (or clients) the approach has not helped them much in terms of gathering user information.

Since the CS students learn programming and database in other courses, the SE course did not provide them with much help in developing prototypes and in database design. The course did help them in the analysis, system design tasks and documentation. The MIS students found the SA&D course helpful in learning planning and analysis and design specifications tasks. Although the sample sizes are quite small, these results are in line with the courses' overall objectives.

Table 3 shows various pertinent skills (some technical and some professional) and the students' perception of how well the SE and SA&D courses helped in the client-sponsored projects in their respective capstone course. The courses seem to have helped both CS and MIS students equally well with soft skills such as team building, leadership and communications. Since feasibility (cost-benefit analysis) is included only in SA&D, the MIS students found the course more helpful in preparing them for the analytical aspects of the project. Since abstraction in the OO approach is harder than in the procedural approach, the CS students found that the SE course did not help them adequately with system wide concepts. This may not be a limitation of the case assignment. Perhaps more exercises should be included in the SE course to address this issue.

6. CONCLUSION

The Systems Analysis and Design case project described above was successfully used in both the OO (for CS) and the procedural (for MIS) approaches and may be of particular interest to instructors who teach both the OO and procedural approaches in their SA&D course. However since abstraction is more involved and difficult to learn in the OO approach than in the procedural, the prerequisites for the SA&D course should include exposure to OO concepts.

The MEC case takes place in a familiar environment. The context is easy for the students to appreciate. It also has an entrepreneurial element that makes it more interesting. The application domain will not require additional learning. Both the instructor and the students can easily

identify themselves as stakeholders in the project. This case was carefully designed to include sufficient complexity. The idea of interfacing the new system with a legacy system provides an opportunity to think about system interfaces. The SE students do see an actor that is an external system. The case offers considerable scope for group work. It offers a sufficient number of uses cases or level-0 DFD processes so that they can be assigned to individual team members.

It is important to provide clear guidelines for preparing the various system artifacts. Fairly comprehensive lists of deliverables for three stages are provided under sub-sections 3.1 and 3.2. These deliverables can be customized to suit a variety of environments. For instance, an SA&D course in an MIS program that uses OO approach may not wish to include sub-system designs and patterns. A few other suggestions on customizing the case project to a particular situation are provided in section 4.

The MEC case meets the realism and purpose criteria of characteristics of a good case (Cappel *et. al.*, 2003). The case provides sufficient opportunity for the students to develop "higher order reasoning skills" as described in (Hackney *et. al.*, 2003) and other technical skills. Using a team-based case project in a conceptual course seems to help develop professional skills with either approach. Further, such an instructor-managed project assignment provides the students a safe environment in which to learn the concepts.

7. REFERENCES

- Bruegge, B. and Dutoit, A. H. (2004). *Object-Oriented Software Engineering*, Prentice Hall
- Cappel, J. J. and Schwager, P.H. (2003) "Writing IS Teaching Cases: Guidelines for JISE Submission," *Journal of Information Systems Education*, Vol 13, No.4, PP. 287-93.
- Dennis, A., Wixom, B. H, and Roth, R. M. (2006). *Systems Analysis and Design*, John Wiley & Sons

Gorgone, J. T., Davis, G. B., Valacich, J. S., Topi, H., Feinstein, D. L., Longenecker, Jr. H. E., (2003). "IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems," *The Database for Advances in Information Systems*, Vol. 34, No. 1.

Hackney, R., McMaster, T., Harris, A. L., (2003) "Using Cases as a Teaching Tool in IS

Education," *Journal of Information Systems Education*, Vol. 14, No.3, PP. 229-234.

Miller, L. (2007) *MIS Cases: Decision Making with Application Software*, Pearson Prentice Hall.

Table-1: CS Students on value of SE course concerning project tasks

Tasks	Average Score out of 5
Prepare a project plan (objectives, work breakdown structure, resources and schedule) for developing a system.	4.2
Gather information (interview) from users for the intended system	3.6
Prepare requirements specification and develop logical system models (use cases, sequence, class, etc.)	4.7
Prepare system architectural design and choose appropriate Database design (example: client-server model)	3.7
Prepare input/output designs (proto-type user interfaces)	4
Prepare detailed design specifications for programs and database	3.6
Develop a prototype for selected system facilities	3.3
Prepare system and user manuals	3.7

Table-2: MIS Students on value of SA&D course concerning project tasks

Tasks	Average score out of 5
Prepare a proposal for an IT- based business solution	4.7
Carry out a feasibility analysis for system solution (economic, technical and operational feasibilities)	4.5
Prepare a project plan (objectives, work breakdown structure, resources and schedule) for developing a system.	4.5
Gather information (interview) from users for the intended system	4
Prepare requirements specification and develop logical system models (ERD, context, Data Flow diagrams)	4.7
Prepare system architectural design and choose appropriate design strategies (Example: client-server model, physical DFD and ERD)	4.5
Prepare input/output designs (proto-type user interfaces)	4.7
Prepare detailed design specifications for programs and database (database schema, structure chart)	4.2
Develop a prototype for selected system facilities	4.2
Prepare system and user manuals	4.2

Table-3: Value of SE and SA&D courses in developing skills for project

Skill Categories	CS average out of 5 (SE)	MIS average out of 5 (SA&D)
Interpersonal skills (get along well; work in team)	4	4.7
Technical skills (know the concepts, apply appropriate tools)	4.1	4.7
Analytical skills (abstraction, unbiased situation analysis and scoping, cost-benefit analysis)	3.7	4.7
Communication skills (report writing, discussion and presentation at meetings)	4	4
Team-building skills (negotiation, organizing / managing meetings, brainstorming)	4.2	4.2
Knowledge of systems wide concept (abstraction, partitioning, scaling, system interface, SDLC)	3.2	4.2
Planning skills (plan activities, assign tasks, estimate resource)	4.1	4.3
Leadership skills (lead by example, coach, resolve resource early on, delegate responsibilities)	4.3	4.3