A Case-Based Approach to Integrating an Information Technology Curriculum

Kathleen S. Hartzel[1]
William E. Spangler
Mordechai Gal-Or
Trevor H. Jones

A. J. Palumbo School of Business Administration
Duquesne University
Pittsburgh, PA 15282 USA

Abstract

This paper describes our plan to use a single business case scenario as a mechanism for integrating our undergraduate IT curriculum. Our proposal is to use an on-line course management tool, as well as an integrated CASE tool, as repositories to facilitate the conceptual and physical integration of courses in our undergraduate Information Technology major. The business case will serve as the basis for student deliverables throughout the courses. As a student progresses through the curriculum, deliverables from prerequisite courses will be available as inputs into subsequent courses. We will use the requirements of the capstone course to determine the scope and requirements of the business case. Then we will incorporate the case requirements into the objectives and deliverables of the preceding foundation courses.

Keywords: IT systems curriculum, business case study, course integration

1. INTRODUCTION

Within the business school we are developing a new undergraduate curriculum in Information Technology with a strong emphasis and focus on distributed application development. An important component of our new curriculum will be a comprehensive business case that focuses on each of the critical areas of information systems development. The specific purpose of the case is to facilitate the conceptual and physical integration of courses addressing the phases of the systems development life cycle (SDLC) in our Information Technology major. As a student progresses through the curriculum, deliverables from previous courses will be available as inputs into subsequent courses. When the student reaches the capstone course, s/he will have completed many of the deliverables for the integrated case, and in so doing will have accumulated the requisite knowledge for completing the case requirements within the capstone. The student also will have a strong understanding of the *role* and *relevancy* of each course, because the objectives of each course will be clearly defined within the context of the overall case.

The requirements of the capstone course will be our starting point. In our capstone course, students must build a multi-tier Internet application, and therefore must understand object-oriented analysis concepts and be able to use them in database development and application programming.

[1] hartzel@duq.edu

In the prerequisite courses, we establish the foundation for the capstone course by introducing the multi-tier environment in the Advanced Systems course, object-oriented analysis in Systems Analysis and Design, database concepts in Database Management, and programming in the Object-Oriented (Java) Programming course.

As suggested, relevancy is a critically important issue in curriculum development. Although our faculty members understand the synergistic nature of the curriculum, it may not be apparent to students. It is important that as new techniques and methods are presented, students will be better able to appreciate how those practices fit into the development process. Therefore, we believe that using a consistent business case across courses would allow students to have a clearer picture, when they reach the capstone course, of what is required to develop a multi-tier Internet application.

2. CURRENT IT CURRICULUM

In creating our IT curriculum we essentially focused on the development of meta-skills in our students. A meta-skill is the ability to adapt and perform a generic activity, such as technology evaluation, assimilation, and debugging, within a variety of technological environments (Westfall 2000). Therefore we have taken a three-pronged approach: 1) determine the *fundamental* (i.e., relatively persistent) skills and knowledge that an IT professional should possess: 2) teach those technologies and concepts in a way that creates and reinforces IT meta-skills: 3) deliver the complete program within a pedagogical environment that stresses the use of meta-skills as a response to business needs. Our initial exploration of the marketplace, as well as the types of positions acquired and skills required in our graduates, indicated that our most successful graduates possessed a strong grounding in business systems – particularly distributed, integrated systems. This initial conclusion was confirmed and fine-tuned through numerous meetings with our industry partners, including companies such as Alcoa, Bayer, Deloitte Consulting, and FedEx, among others. Our partners helped us to better understand the market forces, the particular trends and pressures these companies face, and the characteristics required of their new hires.

From this analysis, we concluded that the transition toward distributed applications – accessible within and across organizations via various types of systems architectures – is the primary technological transformation affecting the analysis, design and implementation of business systems. Toward this end, we sought to encourage students to explore the fundamental constituents of these systems – including object-orientation, internet interface, language protocols, and database connectivity – via diverse implementations that provide students with variations on specific conceptual themes. Over time, students should begin to assimilate and understand those themes. They should begin to develop meta-skills by solving similar problems in different development environments, using different technologies, and they should begin to think fundamentally in terms of object-orientation and inter-related components (Nelson, Armstrong and Ghods 2002).

On a more general level, our approach also has been motivated by the need to develop better critical thinking skills in our students. The sequencing of courses is intended to engage student critical thinking at progressively higher levels, as characterized by Bloom's taxonomy (Bloom, Krathwholl, and Masia 1984). Bloom's taxonomy identifies six major levels or categories of learning. The most basic category is 1) *knowledge*, which essentially entails memorization or identification of facts. After knowledge comes 2) *comprehension*, which focuses on meaning and intent, 3) *application*, which applies existing knowledge to new situations, 4) *analysis*, which enables deeper understanding through decomposition of concepts into components, 5) *synthesis*, the final level of learning which combines components to form original conclusions, and 6) *evaluation* which is used in conjunction with the other five categories. It is well established that individuals cannot attain a high level of critical thinking without first experiencing the lowest levels of knowledge and comprehension. Hence, the curriculum is designed to follow the hierarchy, with the emphasis shifting from knowledge, comprehension and (some) application in earlier

courses, to the more analytic, synthetic and evaluative aspects of the advanced systems and integration courses.

With these objectives in mind, our current curriculum takes an iterative, evolutionary approach that builds on prior learning and requires students to apply that learning in new and more complex situations. We begin by exposing students to fundamental business requirements and potential technology solutions. In terms of creating solutions to business requirements, we go on to cover systems architecture, data and process management, and programming issues. After students have been schooled in those concepts, they use their skills to construct a small, multi-tier application – focusing primarily on the application's interface to a database. This activity serves to reinforce previous learning, demonstrate the relevance of that learning, and introduce the concept of component integration and communication. Finally, the students incorporate all of these conceptual and technical skills in a capstone experience where they "put it all together" by developing a distributed (web-based) application – using technologies and approaches most appropriate to the problem at hand.

While our current curriculum is composed of a set of nine required courses, the integrative business case will initially apply to five essential courses. The sequence of courses and their essential 'theme' is illustrated in Figure 1 and is discussed as follows:

Systems Analysis (Process and Object Modeling): introduces students to a process orientation in systems design, focusing on process modeling tools and techniques. It also introduces students to object-oriented, Unified Modeling Language (UML) constructs. As part of the systems development life cycle, the course emphasizes the movement from high-level (context) models through the decomposition of process definitions to the level of functional program definition. This approach provides a functional, as well as logical, understanding of subsequent application development, which then provides a contextual platform for advanced development classes.

Object-Oriented Programming: builds on the introduction to programming, by introducing the principles of object-oriented design and implementation, as well as more complex syntactic structures necessary for more advanced application and solution development. It currently uses the Java programming language as the development environment.

Database Management (Data Modeling): focuses primarily on conceptual business database modeling, leading to logically correct, effective and efficient relational models. Significant emphasis is placed on dealing with complex business data requirements and how the solutions can be modeled and subsequently reflected in the database design. Since the de-facto standard for database retrieval is SQL, the course provides an introduction which lends itself to the continuation of SQL development and integration in subsequent courses.

Advanced Application Development: introduces students to the essential practices and tools of distributed application and database development, from an object-oriented perspective. This course focuses on understanding three primary technologies of systems integration: 1) object-oriented development using Java, focusing on data structures, object communication and multi-tier applications, 2) a database management system as a central repository linking distributed modules, and 3) Structured Query Language (SQL) as the primary protocol for communication between applications and databases. While each of these technologies is introduced in the foundation courses, this course extends those technologies in the context of their use in building integrated applications.

Systems Integration: utilizes the concepts and techniques of systems integration as a medium for curriculum integration, with the Internet serving as the context for developing front-to-back integrated applications. The course uses the Internet as an example of a complex, protocol-based distributed environment typically encountered by systems architects, and as such is intended to provide students with a capstone experience that brings all of the issues of

application development together in a project-oriented environment.

The integration course supplements the technical knowledge provided in previous courses with concepts and techniques specific to the task of building Internet applications. Because the Internet application infrastructure is inherently object-oriented, student understanding of message passing and object communication is reinforced.

The integration course also reinforces the concept of application-database connectivity by exploring how programs store and retrieve transactional information in a database – through implementation in a different programming languages (i.e., Visual Basic and Java), and using different databases (i.e., Microsoft Access, SQL Server, and/or Oracle). The issues of application-database connectivity and object-oriented communication then are further generalized through the introduction of additional technologies such as Active Server Pages (ASP) and the ASP object model, as well as the role of Extensible Markup Language (XML) in the object-oriented world of web services.

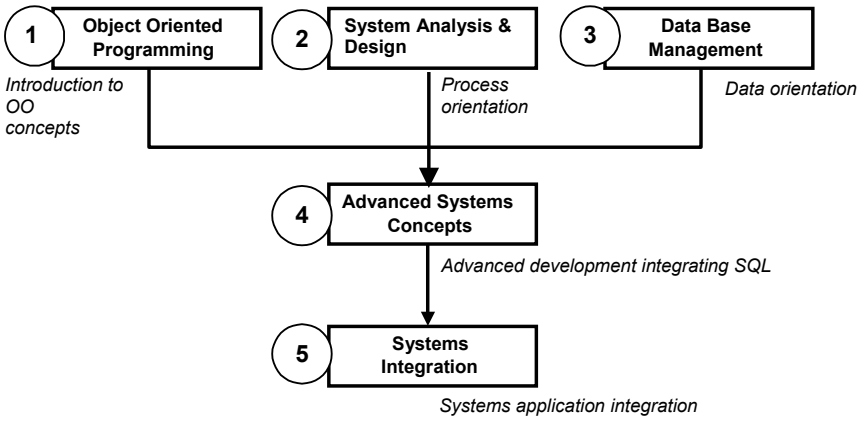

Figure 1: Sequence of courses and themes covered by the business case

3. INTRODUCING THE BUSINESS CASE

The business case scenario will serve as a foundation for each course's homework assignments and major course deliverables. Each course will be modified to (1) use the business case and (2) to discuss how the current case deliverable relate to previous course deliverables as well as what will be addressed in subsequent courses. A Course Management System (CMS) – most likely Blackboard – will be the central, coordinating repository for the case, deliverables, and supporting documentation generated during the sequence of courses. The CMS will allow the maintenance of documents generated throughout the courses. Electronic submission of deliverables will provide a means for the faculty to easily reference and use student work as a teaching technique within the classroom, and to more effectively distribute solutions, reference materials and other handouts. In addition, we intend to utilize the capabilities of a CASE (Computer Assisted Software Engineering) tool to supplement the role of the CMS. A CASE tool will coordinate activities within and between courses, while also exposing students to this important constituent of business application development.

One potential case scenario would be to design a system to support the distribution of auto parts. Initially, in the Systems Analysis course, the students study product procure-

ment and sales management functions of the business in order to build a requirements model. The requirements model would include diagramming the flow of data through the organization and modeling the behavior and attributes of business entities. The requirements model would be further defined in the Database course, where students would produce a data model using Entity-Relationship Diagrams and then generate the database. The physical design and implementation of the behaviors, which were defined in the Systems Analysis course, would occur in the Object-Oriented Programming course. This program code would be linked to the database in the Advances System course. Finally, in the capstone Integration course, students would create user interfaces and distribute the system's pieces over a multi-tier platform. Then they will be able to simulate the act of a customer ordering products via an Internet site and the generation of purchase orders and their electronic transfer to another site.

4. GOALS, EXPECTED OUTCOMES AND ASSESSMENT

Our goal is to instill in students an ability to fully appreciate the complexity and requirement of a realistic business case scenario. The problem-directed approach inherent in the case should provide a number of positive outcomes related directly to the study and assimilation of knowledge *in context*. For example, the use of problem-directed learning should enable students to develop generic problem-solving (or *meta*) skills that are transferable across different situations. Meta-skills transcend individual – and generally transient – technologies, and include skills such as technology evaluation, assimilation and debugging [2]. These types of skills are critical to a successful long-term career in a technology marketplace that is undergoing continuous and rapid change.

With respect to specific courses and concepts, we expect students to demonstrate an improved understanding of systems architectures, processes, and interoperability – knowledge that is fundamentally important for business IT majors. Students also should better understand the overall development process, the methodologies used in the development life cycle, and the relationships between and among supporting technologies in a distributed application. Finally, students should become more aware of the relevancy, utility and objectives of each individual course. They will realize their course requirements are part of a complete integrated program of study, not merely a group of individual, ad hoc requirements.

Assessment

A curriculum bound by a common, unifying case will allow us to determine the overall knowledge set required for the case solution, as well as which individual knowledge constituents are required and contributed by each course. This in turn will allow us to establish a goal-driven assessment plan that includes evaluation of deliverables at specific milestones in the program, and assessment of pre-requisite knowledge prior to a student moving on to the next course (and the next phase of the development process). As such, this approach is similar to standard project management review techniques used in real-world systems development projects.

While specific changes in assessment for the lower level courses are not readily apparent at this time, it seems likely that we will assess the quality of deliverables coming from each course. We also can develop 'pre-assessment' tests of student prerequisite knowledge at the beginning of each new course. At the overall program level, students will have a better understanding of the entire development process. Consequently, we will be able to assess the outcome to the extent that we can successfully deliver a more rigorous capstone course.

Specific short-term evidence of success would include:

- Successful completion of more complex and complete assignments
- Development and implementation of more rigorous courses, with a greater focus on problem-directed learning
- Better preparation of students for subsequent courses
- Positive feedback from students regarding perceived relevancy, rigor and challenge.

- Coverage of more and advanced textbook chapters.

Specific long-term evidence of success would include:

- Positive feedback from our Corporate Advisory Board and employers
- Better types of positions and higher success rate of IT graduates

5. CONCLUSION

We initiated our new curriculum at the beginning of the academic year, and currently are into the second semester of the implementation. Thus, the introduction of the business case is part of the process of ongoing improvement in the content of individual courses and the relationship between courses and their technical prerequisites. Feedback from current students has been useful in helping us to understand the general parameters of the business case and how it might be implemented. After implementing the case, we also intend to follow up in the near- and medium term with studies of recent graduates and their employers, to determine how the new curriculum will affect opportunities for graduates and their performances within employer organizations. These follow-on studies should serve as input to a continuing process of program definition, course curriculum refinement and improvement.

6. REFERENCES

Bloom, Benjamin S., D. R. Krathwhol, and Bertram Masia, 1984, Taxonomy of Educational Objectives : The Classification of Educational Goals, Longman, New York

Nelson, H.J., D.J. Armstrong, and M. Ghods, 2002, "Old Dogs and New Tricks." Communications of the ACM, 45(10): pp. 132-137.

Westfall, R.D., 2000, "Meta-Skills in Information Systems Education." Journal of Computer Information Systems, 40(2): pp. 69-74.