A Plan for a Comprehensive and Integrated Information Systems Curriculum

ABSTRACT: Curriculum guidelines have placed a strong emphasis on integration of theory and practice in information systems (IS). A parallel concern, as yet unaddressed, is the integration throughout the IS curriculum of key theoretical concepts from individual courses. This paper suggests an architectural plan that incorporates formal methods, technological team issues, and organizational theory considerations in the development of a fully integrated information systems curriculum.

Shirley Becker
Rick Gibson
Eugene McGuire
Computer Science and Information Systems
The American University
4400 Massachusetts Avenue
Washington, D.C. 20016

KEYWORDS: *Information Systems Curriculum, Capstone Course, CASE*

INTRODUCTION

"Does the final exam have to be comprehensive?" It is tempting to attribute such questions to student indolence or concern about grades. From a different perspective, however, this question may be seen as symptomatic of some educational shortcomings in conveying the fundamental interrelatedness of information systems concepts.

Both AACSB (1991) and DPMA (1991) guidelines have stressed the need for integration of important concepts across the curriculum in terms of learning the theory and practice of information systems (1, 3). In response, Becker et al. developed an instructional model based on DPMA guidelines that initiated a study of a three-phase approach for integrating course content and allowing students to apply this knowledge in information systems development (2).

This paper describes a comprehensive IS curriculum model that is based on the DPMA principles listed in Table 1. These principles provide a foundation from which is derived a set of courses and their relationships within the IS curriculum model. This model addresses a parallel concern, as yet unaddressed, which is the systematic and sustained integration of key theoretical concepts from individual courses into a

Table 1. A MAPPING OF DPMA PRINCIPLES TO I.S. COURSES

The DPMA curriculum is based on a set of eleven principles that represent the underlying philosophy.The following table provides a mapping of these principles to curriculum ideas presented in this paper. It is noteworthy that the DPMA curriculum model is suggested for a baccalaureate degree only, whereas our synergistic plan is applicable to graduate programs as well.

DPMA Principles	Derived Courses
2. IS professionals must be grounded in the principles of systems theory. 4. IS professionals must understand modeling, measurement and simulation	Overview Course
6. As IS becomes more quantitative, IS professionals must develop an understanding of software and hardware engineering principles. 3. A fundamental activity of IS professionals is problem solving.	System Engineering Courses
1. IS is inextricably linked with the organization's strategy and objectives. 7. IS professionals must understand enterprise planning.	Technology and the Enterprise IS Courses
8. Cultivate creativity and develop tolerance and respect for this characteristic in others. 10. Possess a tolerance for change and skillful management of the change process. 5. IS professionals must interact with and understand diverse user groups.	People and Technology Courses
9. Direct increasing attention to problem avoidance—understand and apply error control and risk management. 11. Education must be continuous.	Capstone Course

unified body of knowledge.

Information systems remains an inherently interdisciplinary field of study, but the central focus is on the use of information and information technology to solve problems. Isaksen and Parnes argue that, given the ill-structured nature of the types of problems that exist today, knowledge cannot be acquired as unassociated fragments that are simply handed down (6). Instead, they contend that "capstone" experiences are needed so that knowledge can be acquired via active, experiential learning opportunities. The information systems programs at several universities offer a single course that promises, but usually fails, to provide this experience. For example, consider the following illustration.

A recent section of the capstone course in a graduate IS program required students to form teams to develop a new system component that could be integrated with an existing DB2 administrative database system. The teams successfully developed a formal specification using Joint Application Development, implemented the system to execute on a local workstation, evaluated their effort in terms of its correctness properties, and successfully tested the system. However, problems emerged in subsequent efforts to link the local system to the central DB2 database due to the lack of an overall, organizational perspective during the systems development process. The development team failed to identify the future directions of the university in terms of local database software, which directly impacted the new system's connectivity with the DB2 system. As a result, the local system must now be redesigned to meet the requirements of a different database software system.

This example highlights the need for a more synergistic teaching plan to ensure that students not only develop high-quality systems, but that the resulting systems meet the long-term needs of the target organization. The problems in this project were largely the result of a non-integrated, loosely coupled approach to IS education throughout the curriculum. Students focused only on the technical requirements of this single course and neglected consideration of the more comprehensive view of the project in relation to the people involved and the enterprise at large. It also became apparent that the ***capstone*** seminar in this particular program was itself not strongly connected to the rest of the curriculum but was actually a weak link in the learning process. One conclusion that can be drawn from this oversight is that the capstone course was not reinforcing key theoretical concepts from other courses in the curriculum. Another possibility is that these concepts are not emphasized and integrated thoroughly throughout the curriculum.

INTEGRATING INFORMATION SYSTEMS CONCEPTS

Laudon and Laudon contend that information systems topics are generally addressed within three environments: technical, behavioral, and organizational (9). Consequently, three distinct areas of information systems as a discipline can be identified, each with its own reference discipline and research paradigm.

- *Information Technology* - the reference discipline is computer science and this area focuses on the application of engineering formalisms to problems. Example topics include: database systems, software engineering, and data communications.

- *Management of Information Systems* - the reference discipline is behavioral science (e.g., psychology and sociology) and this area considers the organizational and human behavioral impacts of IS use. Examples of topics include: organizational role and impact of information systems, end-user computing, cooperative development teams, group decision support systems, global information systems, computer supported cooperative work, and computer-mediated communication.

- *Information Resource Management* - the reference disciplines are economics and organizational theory. Information and the enterprise-wide systems developed for its creation and communication are considered to be scarce and valuable resources well suited to the application of the principles from business disciplines.

It is not surprising, given the above spectrum of reference disciplines and the associated diverse preparation required for each, that information systems courses are usually taught independently of one another. Consequently, students fail to identify the practical application of concepts that span more than one course. An integrated plan of instruction for information systems must support all three of these areas and effectively teach students how to apply formal theory, methods, and techniques within and across these domains.

Granger et al. suggest that a curriculum should be an architectural plan rather than a collection of courses and that the proper starting place for any curriculum change is with three learning objectives: awareness, understanding and competency in requiring skilled use of tools and techniques (5). Such a plan requires an integrative approach to curriculum development to tie together formal development approaches, effective teamwork, and organizational factors in order to produce high-quality, fully-integrated information systems.

The successful application of formal design practices to the task-related activities is only one aspect of the management process associated with information systems development. An additional component of the process of systems development is the effective use of methods for project team development. Increasingly, development teams have become responsible for minimizing the number of software defects by enforcing rigorous development methods, testing techniques, and architectural standards. Finally, at the organizational level, there are internal and external considerations that may impact the systems development process. A major organizational factor is the use of existing standards and awareness of potential ones. When these standards are ignored during the development process, systems integration becomes difficult if not impossible without major design modifications.

Flood and Moll observe that teaching involves ***managing*** the learning process by planning (specifying objectives and activities), organizing, leading, and control (evaluating) (4). Furthermore, they suggest that no course be taught in isolation, instead stressing the criticality of the university contextual environment. Many traditional IS curriculums, however, appear to teach system development courses independently of one another. This model introduces students to a granular level of systems development that is expanded to include development team environments and organizational considerations. This becomes a major advantage as students study how technology affects the developer, the team, the project manager, and the organization. With a well-formulated instructional plan, practice and theory come together in the development of systems. To provide students with the knowledge and skills needed to effectively

Figure 1: THE INTEGRATED I.S. CURRICULUM MODEL

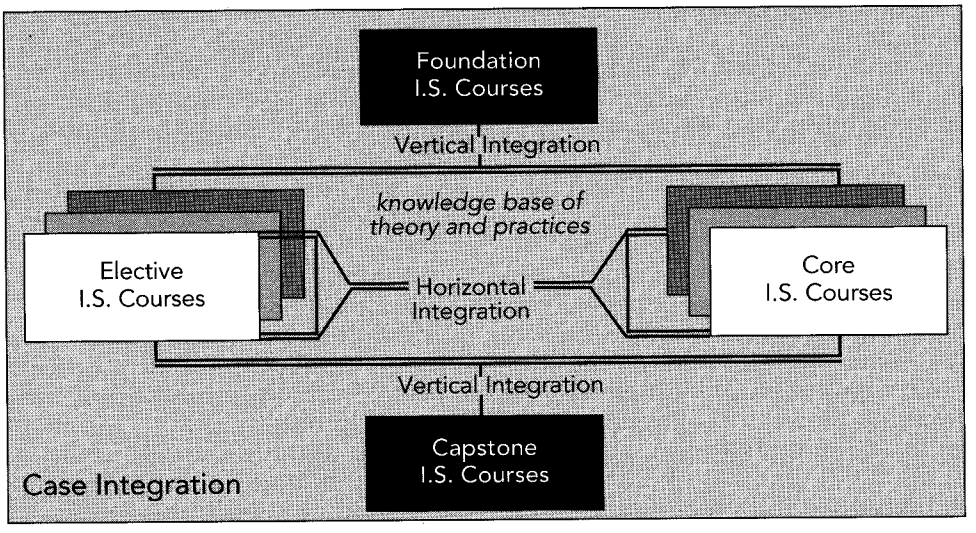
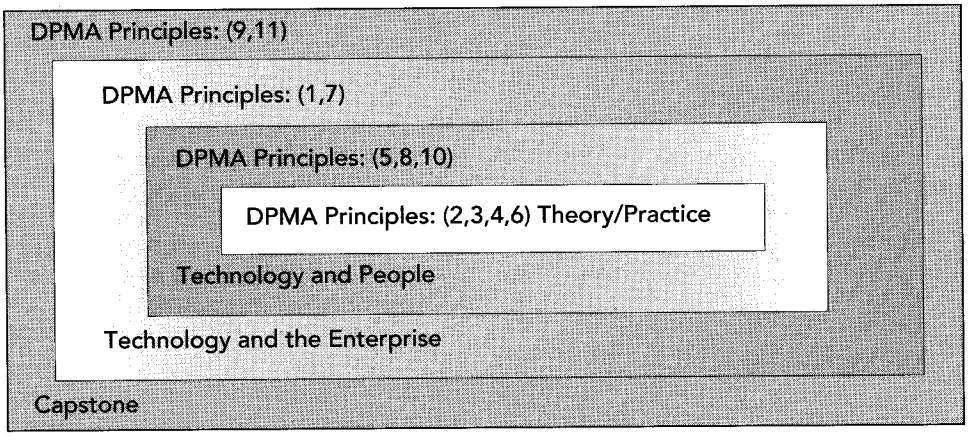


Figure 2: THE INTEGRATED LAYERS OF THE I.S. CURRICULUM



develop integrated systems, the information systems curriculum should include the integrated structure shown in Figure 1 and described below.

The Core and Elective IS Courses

The curriculum model in Figure 1 presents a structure where IS courses build upon one another in an integrated fashion. The foundation courses provide an overview of information systems as well as a skill base for developing systems. The core courses expand upon the theory and techniques introduced in the foundation courses. The elective courses supplement the core courses by allowing students to gain expertise in selected areas of interest.

A classification schema for these courses has been developed in order to show the inherent relationship of our model with the DPMA guidelines. This schema is shown in Figure 2 and is composed of: system engineering technology, people and technology, and technology and the enterprise curriculum layers. All of these layers are encapsulated by the capstone layer. Each layer is defined as follows:

- *System engineering technology* - Courses include systems analysis and design, software (system) engineering, system testing and measurements, database management, data communication, decision support systems, and other related courses. These courses provide the theories that serve as precise descriptions of system behavior, provide insight and control over the systems development process, and form the basis for rigorous evaluation of the correctness and performance of system designs.
- *People and technology* - Courses include human factors and project management, and other related courses. These courses build upon software engineering technology courses as they address the human aspect of developing systems. Students learn that successful systems development is not solely a technical issue. Instead, it concerns individual and collaborative efforts in creating and using a system. These courses focus on the human aspects of the technologically coordinated workplace from both a system developer's and a user's perspective to include creative problem-solving and systems integration strategies.
- *Technology and the enterprise* - This categorization schema is depicted in Figure 2 where it is shown that students become knowledgeable about theoretical and pragmatic applications of systems development. This knowledge base is expanded by exploring the impact of IS technology on the developer, manager, teams, and others. The students then learn about technology's impact on the organization and its external environment.
- *Capstone* - The theory, practices, and tools learned in these courses become the building blocks applied in the systems development capstone course. This supports the DPMA objective of continuous education in information systems.

The integrating factors in this schema are three-fold. Students are required to complete project components that will be used as inputs or developed in parallel with other courses. Students are required to use appropriate CASE tools to complete project components. The capstone course becomes the third integrating factor in our model. These integrating concepts are described in Table 2.

The ideal capstone course is a systems development workshop that requires students to fully integrate and implement all the conceptual ideas and technological aspects of systems development. This provides students with in-depth experience and insight on the actual progression of the development of a real system. The student is also given the opportunity to apply the skills obtained in the prior courses

CONCLUSION

Having begun with a question, this paper now ends with one possible prescription. Just as systems development has evolved beyond the classic waterfall model to consider life cycle and object-oriented approaches, information systems educational efforts must embrace pedagogical improvements centered on content integration. This paper has proposed an architected plan for a ***student*** development life cycle that requires students to integrate concepts and skills in a collaborative team-

centered environment that provides greater emphasis on active student learning and faculty productivity (7). An additional benefit of such an integrated approach is the reduced need for inflexible course sequencing (beyond the need for starting with the overview and ending with the capstone), which places a burden on students, faculty and administrators.

Table 2: THE INTEGRATING FACTORS IN THE I.S. CURRICULUM

Integration in the I.S. Curriculum

Horizontal Integration:
Project components are coordinated among core courses. (Example project: Kowal's Convenient Auto Rental System (8, 2)). Project components include:

- *Systems Analysis and Design* - Develop the formal specifications and design documents.
- *Database Management* - Develop a conceptual and physical schema of a database to support the project.
- *System Testing and Measurements* - Develop test plan and define metrics for assessing system quality.
- *Project Management* - Develop and assess project schedule, critical path, resource utilization, etc.

CASE Tool Integration:
Students use CASE tools to successfully complete project components. Typical CASE tools used include:

- *Systems Analysis and Design* - IEW®, Excelerator®
- *Database Management* - XDB®, Paradox®, Access®
- *System Testing and Measurements* - ExpertChoice®
- *Project Management* - IEW®

Vertical Integration:
The theory, skills, and tools used in the foundation, core, and elective I.S. courses are used to develop an actual system in the capstone course. Course activities include:

- *Form Development Team* - students form development teams and review the behavioral aspects of teams.
- *Gather User Requirements* - students hold JAD sessions with users to gather all system requirements.
- *Develop Formal Specifications* - students produce functional and usage specifications (11, 12).
- *Perform Analysis and Design Activities* - students use a spiral approach to achieve a high-quality system design.
- *Develop test plan and conduct testing* - students develop a test plan, perform alpha testing on their systems, and perform beta testing on other system designs.
- *Develop User and System Documentation* - students develop documentation manuals used during testing for consistency and correctness validation.

REFERENCES

1. AACSB. (The American Assembly of Collegiate Schools of Business), Final Report: The AACSB Accreditation Project, March, 1991.

2. S. A. Becker, E. G. McGuire, and L. R. Medsker, "An Information Systems Instructional Model for Supporting the DPMA 1990 Guidelines," *Journal of Information Systems Education*, Volume 4, 1992, pp. 21-25.

3. DPMA (Data Processing Management Association), *Information Systems: The DPMA Model Curriculum for a Four Year Undergraduate Degree*. Park Ridge, IL: DPMA, 1991.

4. B. Flood, and J. K. Moll. *The Professor Business: A Teaching Primer for Faculty.* Medford, NJ: Learned Information Inc, 1990.

5. M. J. Granger, D. L. Schroeder, B. Rollier, L. Esnault, "A Framework for Internationalization of MIS Curriculum," In *Proceedings of the 1992 Information Resource Management Association Conference*, M. Khosrowpour (Ed.), pp. 257-261.

6. S. G. Isaksen and S. J. Parnes. "Curriculum Planning for Creative Thinking and Problem Solving." *Journal of Creative Behavior* 19 (1), 1985, pp. 1-29.

7. D. W. Johnson, R. T. Johnson, and K. A. Smith, *Cooperative Learning: Increasing College Faculty Instructional Productivity*, ASHE-ERIC Higher Education Report, No. 4, 1991.

8. J. Kowal, *Analyzing Systems*, Prentice-Hall, Inc., 1988.

9. K. C. Laudon and J. P. Laudon, *Management Information Systems: A Contingency Perspective.* New York: Macmillan Publishing Company, 2nd ed, 1991.

10. N.G. Leveson, "Guest Editor's Introduction: Formal Methods in Software Engineering," *IEEE Transactions on Software Engineering*, Vol.16, No. 9, September 1990, pp. 929- 930.

11. H.D. Mills, R. Linger, and A. Hevner, *Principles of Information Systems Analysis and Design*, Academic Press, Inc. 1986.

12. H.D. Mills, "Stepwise Refinement and Verification in Box-structured Systems", *IEEE Computer*, Vol 21, No. 6, June 1988.

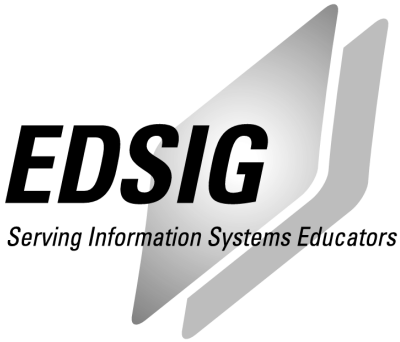
AUTHORS' BIOGRAPHIES

Shirley A. Becker is an assistant professor of computer science and information systems at The American University. Her research interests include system modeling using Petri-nets, Cleanroom system engineering, and object-oriented design using box structures. She received a Ph.D. in information systems from the University of Maryland at College Park. She is a member of IEEE Computer Society and ACM.

Eugene G. McGuire is an instructor of computer science and information systems at The American University. His research interests include human factors, computer-sup ported cooperative work, hypermedia, and the group dynamics of system deveopment. He received a M.S. in information systems from The American University and is completing a Ph.D. He is a member of IEEE Computer Society and ACM.

Dr. Gibson received his Ph.D. in Information Systems from the University of Maryland at College Park in 1992. His four years of teaching in Italy, Germany, Belgium, Korea, Japan, and the Philippines provided the impetus for adopting a global perspective regarding information systems. Related research interests include the evolutionary internationalization of the information systems curriculum and the revolutionary application of chaos theory to information systems development.

Information Systems & Computing
Academic Professionals



STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096