AN INFORMATION SYSTEMS INSTRUCTIONAL MODEL FOR SUPPORTING THE DPMA 1990 GUIDELINES

Shirley A. Becker,
Eugene G. McGuire, and
Larry R. Medsker
Computer Science and Information Systems
The American University
Washington, D.C. 20016

ABSTRACT: The recent DPMA 1990 curriculum guidelines emphasize the integration of theory and practice in information systems. The guidelines are a systematic approach to covering the knowledge clusters and to preparing students to be practitioners who can develop real systems. Traditional I.S. curricula, however, tend to offer theoretical, self-contained courses. The Information Systems Instructional Model proposed here integrates courses and bridges the gap between theory and practice. Our model supports the DPMA 1990 and is based on an underlying methodology called the Cleanroom Systems Development Process (CSDP). This innovative approach to systems development is discussed, and the progress in implementing the model is described.


INTRODUCTION AND BACKGROUND

The existing software crisis has been attributed to the chasm between industry practices and formal systems development theories [5]. Practitioners are faulted for applying ad-hoc development methods that have no formal basis for unambiguous descriptions, correctness evaluation, or statistical testing. One of the reasons that many formalisms have not been widely used by practitioners, however, is because of their complexity when scaled up to real-world problems. What is needed to bridge this chasm is systems development practices with underlying formalisms.

A significant factor in perpetuating this gap is the traditional Information Systems (I.S.) curriculum. Most I.S. programs offer systems development courses that are theoretical in nature, are self-contained, and are not integrated with other I.S. courses. This approach is quite effective in teaching formal methods to students. But what is lacking in many I.S. curricula is the practical implementation of these formalisms not only in case studies but scalable real-world system applications.

The objective of our study is the development of a rigorous yet pragmatic systems development model that emphasizes the application of formal development methods at all levels of coursework in the I.S. curriculum.

EFFECTIVE APPLICATION OF THE DPMA GUIDELINES

We are currently studying how to effectively implement the DPMA IS’90 guidelines so that students gain practical experience applying formal systems development theory. The objective of our study is the development of a rigorous yet pragmatic systems development model that emphasizes the application of formal development methods at all levels of coursework in the I.S. curriculum.

In this ongoing research, we use the Cleanroom Systems Development Process (CSDP) model as the basis for bridging formal theory and development practices. CSDP has been chosen as the integrating factor in our study because it has a well-defined underlying formalism and supporting practices that have been used successfully in industry to develop near-zero defect system applications [3] and [6]. (An overview of CSDP is presented later in this paper.)

The application of CSDP throughout the I.S. curriculum supports the 1990 DPMA curriculum guidelines which
emphasize the integration of theory and practice in Information Systems. These guidelines are significant because of the focus on advances in information technology, the emphasis on functionality rather than the topicality of course content, and the enhancement of the traditional lecture approach to teaching I.S. with lab and/or field experiences [4].

Many of the philosophical elements found in the DPMA IS’90 guidelines are captured in the model IS curriculum we are implementing and researching. Some of the similarities in these two models follow.

IS’90 strongly urges an organizational/practitioner emphasis in I.S. education instead of a purely conceptual emphasis. Our Information Systems Instructional Model shown in Figure 1 supports this emphasis. In our model, student projects are centered in the real world, span several courses and bodies of subject matter, and are under the control of multiple instructors throughout their life cycles. Students are exposed not only to the technical aspects of a particular project but also to an environment that resembles many matrix-managed organizational settings in which technical projects are commonly developed.

IS’90 is focused on the offering of a cohesive body of knowledge instead of isolated islands of knowledge. Our model is complementary to this focus with its emphasis on the instructor collaboration necessary to carry projects across several previously distinct and separate course offerings. This collaboration results in a greater integration of subject matter and a better focus on the core concepts that permeate all phases of a systems development project.

Another of the philosophical guidelines expressed in IS’90, is that the coverage of a knowledge cluster is progressive in that related subject matter is covered in an iterative and integrated fashion until the required depth is achieved. Our model, by integrating subject matter and instructional guidance across courses, achieves this objective as well. Students can attain incremental success as they build towards the completion of a project that is judged by rigorous, real-world standards.

The DPMA’s I.S. curriculum model emphasizes “systems theory, user requirement based problem solving, and software engineering” concepts. The focus of this model is to provide an educational experience that is sufficiently rigorous to meet the challenges faced by today’s practitioners. The goal of this program is to produce graduates that have effectively applied formal knowledge in the development of actual systems. In addition, these graduates will have developed effective communication skills, will be aware that they must continuously update their skills, and will have developed a high level of confidence in their abilities as professionals [4].

THE INFORMATION SYSTEMS INSTRUCTIONAL MODEL

We propose an Information Systems Instructional Model that is the basis for integration among the I.S. courses. The model presented in Figure 1 provides I.S. students an opportunity to learn fundamental systems development skills and apply their form knowledge across all of the systems development activities.

The Information Systems Instructional Model supports the information systems perspectives outlined in the DPMA standards. These I.S. perspectives include: life cycle, organizational hierarchy, conceptual level, content, and depth of knowledge [4]. CSDP which is the basis for the I.S. Instructional Model introduces students to an alternative to the traditional waterfall approach to systems development because it is an iterative and dynamic process [1]. Any of the systems development activities including requirements specification, analysis, design, and implementation can be performed in any sequence and as many times as necessary to develop a high-quality system.

The organizational hierarchy component is introduced during the skills courses so that students gain insight into group dynamics and organizational concepts of systems development work. Students gain practical experience interacting with one another and assessing
this interaction during the advanced systems development coursework when they form CSDP development teams and assess its organizational structure [2].

The conceptual and content levels are critical in that they provide students with the methods and tools necessary for developing today's complex systems. Students require basic underlying theory in systems development, structured programming concepts, information and computer technology, and organizational concepts. These general principles are developed in the foundation courses and are utilized in all advanced I.S. coursework. These I.S. perspectives are discussed in depth in the following sections on CSDP and CASE support.

Students gain a depth of knowledge by integrating concepts across I.S. courses and at each level of course advancement. Students are given an opportunity to expand system designs for which they have gathered requirements, produced documentation, evaluated correctness, applied statistical testing techniques and project management concepts, and evaluated the development team's and team leader's effectiveness.

The key to the success of the I.S. Instructional Model is its underlying methodology called the Cleanroom Systems Development Process (CSDP). This innovative approach to systems development is discussed next.

THE CLEANROOM SYSTEMS DEVELOPMENT PROCESS

CSDP has been successfully applied in the development of systems applications because of its effective integration of theory and practice in developing near-zero-defect systems [3]. CSDP's success can be attributed to its unique approach of combining formal correctness verification and statistical testing during the systems development process. The CSDP activities are shown in Figure 2.

CSDP effectively uses the concept of development teams that gather system and usage requirements and can concurrently perform analysis and design, documentation, and testing for each design increment. A system is divided into design increments for intellectual and management control over the systems development process. Each design increment may be considered a closed system that adds functionality to previous increments [6].

Throughout the systems development process, CSDP enforces a number of essential, formal system development concepts. These concepts include incremental development, box-structured analysis and design, correctness verification, and reliability certification [6].

Box structure theory and principles provide a formal basis for top-down Cleanroom incremental development. System components are described in three levels of increasing detail: black box, state box, and clear box [8]. All box structure designs are verified for consistency and closure. Each design increment is certified for reliability via mean time to failure (MTTF) analysis. (For a comprehensive discussion refer to [3] and [8]).

THE INTEGRATION OF CASE IN THE I.S. INSTRUCTIONAL MODEL

CASE has become a critical component for effective systems development by providing automated support needed to develop and manage complex systems. CASE environments provide a basis for integrating individual or course projects into a larger system design. CASE tools allow students to build upon previous system designs by providing automated support for all of the systems development activities including the traditional activities as well as project management, documentation, and testing activities.

Figure 2: The Cleanroom Systems Development Process (CSDP) [6]
Figure 3: The I.S. Instructional Case Architecture

We are currently developing a CASE workstation to support the integration of systems development theory and practice in our Information Systems curriculum. The CASE architecture of the proposed workstation is presented in Figure 3 and is based on the CSDP CASE architecture proposed by Hevner and Becker [5].

The CSDP workstation thus far is comprised of the following CASE tools (note: other commercial CASE tools can be easily substituted in this architecture as long as students are provided CASE support for each of the system development activities):

- XDB® PC-based relational database system
- XDB link software to DB2® mainframe database system
- Microfocus COBOL® application language
- Microsoft® C application language
- IEW Knowledgeware® systems development tool
- WordPerfect® and DrawPerfect® word processing and graphics tool.

The I.S. Instructional Model is fully supported by the proposed CASE architecture. Individual CASE tools are introduced in the foundation courses and are used in an integrated fashion during the advanced systems development courses.

The CASE tools essential to supporting student projects include graphics and editing capabilities for requirements specification and analysis and design activities, simulation and statistical packages for system testing, and intelligent editors for documentation. A common repository is the underlying basis for course integration by storing information obtained during each development activity [5].

The Cleanroom process enforces rigorous correctness criteria for developing high-quality systems from the conceptual design (supervised by Knowledgeware and WordPerfect), database design (supervised by XDB and DB2) to application code generation (supported by MicroFocus COBOL and Microsoft C).

COMPARATIVE RESULTS OF I.S. CASE STUDIES

A view of the instructional differences between traditional I.S. curriculum and the I.S. Instructional Model is presented in Table 1. These observations can be regarded as preliminary, but appear to be valid. This table shows that the I.S. Instructional Model requires a more comprehensive and integrated course preparation on the part of the instructors. In addition, instructors must be willing to enter into a collaborative and facilitative mode for the duration of a student's program.

Table 2 indicates that projects in the I.S. Instructional Model are characterized by much more real world uncertainty, complexity, and rigor than those typically found in traditional I.S. classes.

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

Our initial research with the I.S. Instructional Model suggests that it can achieve the objective of modeling real-world systems development practices by applying rigorous underlying formalisms. This model appears to us to be a solid implementation in form, function, and spirit of the DPMA IS'90 guidelines.

Future Research in this area should focus on assessing student achievement after completing a program of this nature as compared to student achievement in traditional I.S. programs. This assessment should be both quantitative and qualitative. In addition, instructor satisfaction after participating in this program should be evaluated to assess how the I.S. Instructional Model can be enhanced.
### Table 1: Instructional Difference Between the Traditional and the I.S. Instructional Model

<table>
<thead>
<tr>
<th></th>
<th>Traditional Model</th>
<th>I.S. Instructional Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor knowledge of subject matter</td>
<td>compartmentalized</td>
<td>integrated</td>
</tr>
<tr>
<td>Instructor preparation</td>
<td>initial comprehensive course-length preparation with periodic updates</td>
<td>continuous, short-term consultative type of preparation</td>
</tr>
<tr>
<td>Instructor effort</td>
<td>individual</td>
<td>collaborative</td>
</tr>
<tr>
<td>Instructor role</td>
<td>pedagogue</td>
<td>facilitator</td>
</tr>
<tr>
<td>Instructor focus</td>
<td>subject driven</td>
<td>process driven</td>
</tr>
<tr>
<td>Instructor time</td>
<td>class and some individual consultation</td>
<td>considerable amount of group consultation</td>
</tr>
</tbody>
</table>

### Table 2: Project Differences Between the Traditional and the I.S. Instructional Model

<table>
<thead>
<tr>
<th></th>
<th>Traditional</th>
<th>I.S. Instructional Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duration of projects</td>
<td>generally contained within semester or term</td>
<td>projects are long-term; students may work on different segments in different courses</td>
</tr>
<tr>
<td>Complexity of projects</td>
<td>generally characterized by minimal interdependencies, external changes, constraints, etc.</td>
<td>real-world projects requiring interaction with actual users and approval processes</td>
</tr>
<tr>
<td>Goal of projects</td>
<td>demonstration of subject matter understanding</td>
<td>error-free systems that can be implemented and used</td>
</tr>
</tbody>
</table>

### REFERENCES


4. DPMA Model Curriculum for a Four-Year Undergraduate Degree, DPMA, 1990.


AUTHORS' BIOGRAPHIES

Shirley A. Becker is an assistant professor of computer science and information systems at The American University. Her research interests include system analysis and design, system modeling using Petri-nets, Cleanroom system engineering, and the group dynamics of system development. 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-supported cooperative work, hypermedia, and the group dynamics of system development. He received a M.S. in information systems from the The American University and is completing a Ph.D. He is a member of IEEE Computer Society and ACM.

Larry R. Medsker is a professor and chair of the computer science and information systems faculty in the College of Arts and Sciences at The American University. His research interests are in expert systems and neural networks, recently concentrating on hybrid systems. He received his Ph.D. in Physics from Indiana University. He is a member of ACM, IEEE Computer Society, and AAAI.

EDSIG MEMBERS' BITNET E-MAIL ADDRESSES

<table>
<thead>
<tr>
<th>Member Name</th>
<th>ID Number</th>
<th>Node Name</th>
<th>EDSIG Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Jagdish C. Agrawal</td>
<td>F60C029</td>
<td>SAKSU00</td>
<td>Review Board</td>
</tr>
<tr>
<td>King Saud University</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dr. Marvin Albin</td>
<td>MARVIN</td>
<td>MOREKYPR</td>
<td>Case Study Competition</td>
</tr>
<tr>
<td>Morehead State University</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prof. Victor M. Barlow</td>
<td>BARLOW</td>
<td>PURCCVM</td>
<td>Review Board</td>
</tr>
<tr>
<td>Purdue University</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prof. John Boggess</td>
<td>BOGGESS</td>
<td>PURCCVM</td>
<td>Member</td>
</tr>
<tr>
<td>Purdue University</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dr. Jim Caristi</td>
<td>MATHJC</td>
<td>VALPO</td>
<td>Member</td>
</tr>
<tr>
<td>Valparaiso University</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dr. Donald R. Chand</td>
<td>DCHAND</td>
<td>BENTLEY</td>
<td>Review Board</td>
</tr>
<tr>
<td>Bentley College</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dr. Eli Boyd Cohen</td>
<td>ELI</td>
<td>bradley.edu</td>
<td>Secretary &amp; Journal Hardware/Software Review Editor</td>
</tr>
<tr>
<td>Eastern New Mexico University</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

continued on page 32