Information Systems Education: What's missing?

Paul H. Rosenthal
Information Systems Department,
California State University, Los Angeles
Los Angeles, CA 90032-8132
prosent@calstatela.edu

Abstract

We are doing a good job of teaching IS technology and project management but are omitting implementation planning. We need to teach our users and professionals how to answer the following critical questions for our mission-critical transaction processing applications (TPS).

- Why does it cost so much?
- How long does it take-Why does it take so long?
- What makes our applications systems so complex?

This presentation discusses pedagogical and presentation structures that will focus more attention on these planning-oriented questions.

Keywords: enterprise systems, transaction processing systems, physical systems design, justifying information systems, estimating information systems.

1. INTRODUCTION

Based on my fifty-six years of experience teaching IS, we are doing a good job of teaching IS technology and project management, while almost omitting implementation planning. We need to teach IS users and professionals how to answer the following critical implementation questions for our mission-critical transaction processing applications (TPS).

- Why does it cost so much?
- How long does it take-Why does it take so long?
- What makes our applications systems so complex?

The theme of this presentation is therefore – that we move away from our current obsession with personal productivity and entertainment systems (e.g. Web 2.0) and back to where we should be – improving the productivity of the US economy through teaching the planning of enterprise-level business productivity systems (e.g. operations and management-oriented TPS systems). This presentation discusses pedagogical and presentation structures that will focus attention on these implementation planning-oriented questions.

2. SCOPE OF ENTREPRISE SYSTEMS

Enterprise level operations-oriented applications are at the core of the information and technology systems' (IS/IT) impact on organizations. In the typical medium sized to large business organization, they constitute the majority of IS/IT funding requirements, sometimes as much as 80%. A typical MIS text's view of the structure of enterprise systems is illustrated by the Figure 1 diagram extracted from Laudon and Loudon's 10th edition text (Laudon, 2007, pg. 60).

This view, while interesting, is not detailed enough for proper understanding of the types and relationships among operational,

decision support and personal productivity applications as might be shown in Figure 2.

Most IS intellectual contributions are currently directed toward the managerial support applications (e.g., decision and people-oriented applications) since they are more interesting and involve smaller, more easily understood systems. But the big money and major productivity impact is with enterprise-level transaction processing systems (operations oriented applications).

From an IS education oriented viewpoint, I believe that there are three major planning and design areas that are not being properly addressed, and will be stressed in this paper.

- Recognition of the Complexity and Importance of Transaction Processing Systems
- The Need for a Physical Systems Design Methodology understandable by all Stakeholders
- The Justification and Costing of IS/IT Projects

3. SCOPE OF TRANSACTION PROCESSING SYSTEMS

The initial step in answering the complexity question is to teach the true scope of TPS applications. The true scope and complexity of modern integrated transaction processing application systems is shown in Figure 3. The figure presents the overall scope of a typical administrative-oriented TPS application. It shows the interrelationships of core TPS online and batch processing with its dependant MIS, DSS, ESS, and interfacing systems.

"Today's transaction processing systems no longer provide discretionary support to the enterprise-*they are the enterprise*. They enforce decisions, dictate workflow, and optimize profitability" (3i Infotech, 2009). For many organizations, such as banks, they are the product delivery system's information resource. Their data controls and maintains the interfaces with customers and vendors.

Many enterprises spend well over half their development and operations budgets on their TPS applications. Their characteristics, design and implementation should be stressed in enterprise oriented IS/IT education. We need extensive research in the value, costs, and benefits of these multi-million dollar systems, and their impact on US productivity.

4. PHYSICAL SYSTEMS DESIGN METHODOLOGY

The complexity question's answer is to expand our systems analysis and design curriculum to include physical-level design. This paper proposes a TPS physical design approach that is easily understood by all stakeholders, and easily used by programmer analysts during implementation. As shown in Figure 4: The Design Process, a physical design is created from a DFD based logical design, by separating processes and data stores

- by time (daily vs. monthly, day vs. night ...),
- by place (client or server, centralized vs. distributed...), and
- by online vs. batch, and manual vs. automated.

None of these design decisions are fully illustrated in the systems analysis and MIS textbook in our IS user and IS professional's courses. Additionally, proper separation of data flow vs. paper flows, and people's actions vs. computer processes is almost never maintained.

Figure 5: A Physical Level Design Example presents an overall physical design approach of a country club restaurant using VISIO available symbols. The application is modularized across time and should allow programmers to produce a well structured program. Students presented with this type of chart have been easily able to create the four detailed program designs needed to implement the system. This level of physical charting is the step needed between logical designs and programming.

The key to the effectiveness of this methodology (as illustrated in Figure 5) is the inclusion in the design of both manual and automated procedures, and the separation of processes by time and place of actions.

5. JUSTIFICATION OF INFORMATION SYSTEMS

How do we answer the question on cost and scheduling? Our Systems Analysis and design, and MIS texts must stop ignoring the justification and cost/benefit analysis of enterprise information systems. For example, Martin's text (popular for MBA courses), has no entry in its index for justification, pricing, scheduling, or cost estimating Brown, of information systems (Brown, 2009). A basic overview of both the managers' role of system justification (including benefits, costs, and risks) and the IS professionals role of infrastructure and software cost estimating and scheduling, must be extensively covered.

The following justification policy statement is extracted from OMB Directive M-97-02 (Raines, 1996)

> "Demonstrate a projected return on the investment that is clearly equal to or better than alternative uses of available vendor resources. Return may include: improved mission performance; reduced cost; increased quality, speed, or flexibility; and increased customer and employee satisfaction. Return should be adjusted for such risk factors as the project's technical complexity, the organization's management capacity, the likelihood of cost overruns, and the consequences of under- or non-performance."

A paper on IT investment strategy (Gunasekaram, 2001, pg. 354) presents A Model for Investment Justification in IT Projects that suggests the use of the following justification factors.

Financial Tangibles
Budgets
Priority of Investment
ROI
Product Cost
Market Research
Alternate Technology
Profit Level
Revenue

Non-Financial Tangibles
Lead-time
Inventory
Labor Absence
Defective rate of Products
Set-up Time

Intangibles
Competitive Advantage
Service to Society
Job Enrichment
Quality Improvement
Improve Customer Relationships
Enhance Confidence
Securing Future Business
Risk of Not Investing in IT
Teamwork
Good Image

Our IS education must approach these ever-present user and CIO questions such as

- Why will it cost so much and take so long?
- How can I be sure this will be one of the 50% that succeed rather than the 50% that fail?

A special comment is needed on software estimating. The major increase in package usage has reduced the impact of the lack of our usage of the tools available in this area. Tools such as Function Point techniques are seldom taught or used.

Major work is needed in the justification of major enterprise applications, culminating in a monograph that can serve as a basis for chapters on 'Justification' in all of our relevant texts.

6. ESTIMATING

The process of estimating, while technically part of justification, deserves special consideration since it is the nemesis of IS planning and project management. Brooks in his classic *The Mythical Man Month* (1995) speaks to "gutless estimating" and recommends "stick to your estimates."

> "Observe that for the programmer, as for the chef, the urgency of the patron may govern the scheduled completion of the task, but it cannot govern the actual completion. An omelet, promised in two minutes, may appear to be progressing nicely. But when it has not set in two minutes, the customers have two

> choices—wait or eat it raw. Software customers have had the same choices.
>
> The cook has another choice; he can turn up the heat. The result is often an omelet nothing can save—burned in one part, raw in another.
>
> Now I do not think software managers have less inherent courage and firmness than chefs, nor than other engineering managers. But false scheduling to match the patron's desired date is much more common in our discipline than elsewhere in engineering. It is very difficult to make a vigorous, plausible, and job- risking defense of an estimate that is derived by no quantitative method, supported by little data, and certified chiefly by the hunches of the managers.
>
> Clearly two solutions are needed. We need to develop and publicize productivity figures, bug-incidence figures, estimating rules, and so on. The whole profession can only profit from sharing such data.
>
> Until estimating is on a sounder basis, individual managers will need to stiffen their backbones and defend their estimates with the assurance that their poor hunches are better than wish-derived estimates."

A widely available concise coverage of software project estimation can be found in Pressman's Software Engineering text (2005, Ch. 23). The core of the Function Point technique illustrated in the book involves the estimation of an application's software development cost using the type chart shown in Figure 6.

7. CONCLUSION

We need to raise the level of content of IS curriculums so that our graduates will be able to specify, estimate, evaluate, design, and implement high quality and successful systems, and continue to reduce our industry's project failure rate (Rosenthal & Park, 2009).

It is also worth mentioning that the Information Systems field needs extensive publicity. For example, most personnel departments still do not know the difference between Information Systems and Computer Science, and incorrectly believe that CS graduates are qualified to specify, design and implement business-oriented information systems.

8. REFERENCES

3i Infotech (2009). Enterprise Solutions for Payment & Imaging (ESPi) – An Overview. http://www.3i-infotech.com/content/payment_solutions/espi_overview.aspx

Brooks, Frederick P. (1995). The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition (2nd Edition). Addison-Wesley Professional, Pearson PLC.

Brown, DeHayes, Hoffer, Martin, Perkins (2009). Managing Information Technology: Sixth Edition. Pearson Prentice Hall.

Gunasekaran, A., Love, P., Rahimi, F., Miele, R. (2001). "A Model for Investment Justification in Information Technology Projects." International Journal of Information Management. Elsevier, 21 pp. 349-364.

IFPUG. Function Point Counting Practices Manual. International Function Point Users Group, Princeton, NJ. http://www.ifpug.org/publications/manual.htm

Laudon, Kenneth C., Laudon, Jane P. (2007). Management Information Systems: Managing the Digital Firm Tenth Edition. Pearson Prentice Hall, New Jersey.

Office of Management and Budget (OMB, 1996). Funding Information Systems Investments. www.whitehouse.gov/omb/**memo**randa/m**97-02**.html

Pressman, Roger S. (2005). Software Engineering: Practitioner's Approach 6th Edition. McGraw-Hill Irwin, New York.

Rosenthal & Park (2009). "Physical Level Systems Design**:** A Methodology for Inclusion in our Systems Analysis and Design Textbooks." Information Systems Education Journal, http://isedj.org/7/54/

Rosenthal & Park (2009). "Managing Information Systems Textbooks: Assessing their Orientation toward Potential General Managers". Journal of Issues in Informing Science and Information Technology (IISIT), V6, pp 241-255. http://iisit.org/Vol6/IISITv6p241-255Rosenthal584.pdf

9. APPENDIX

Figure 1: Enterprise Application Architecture

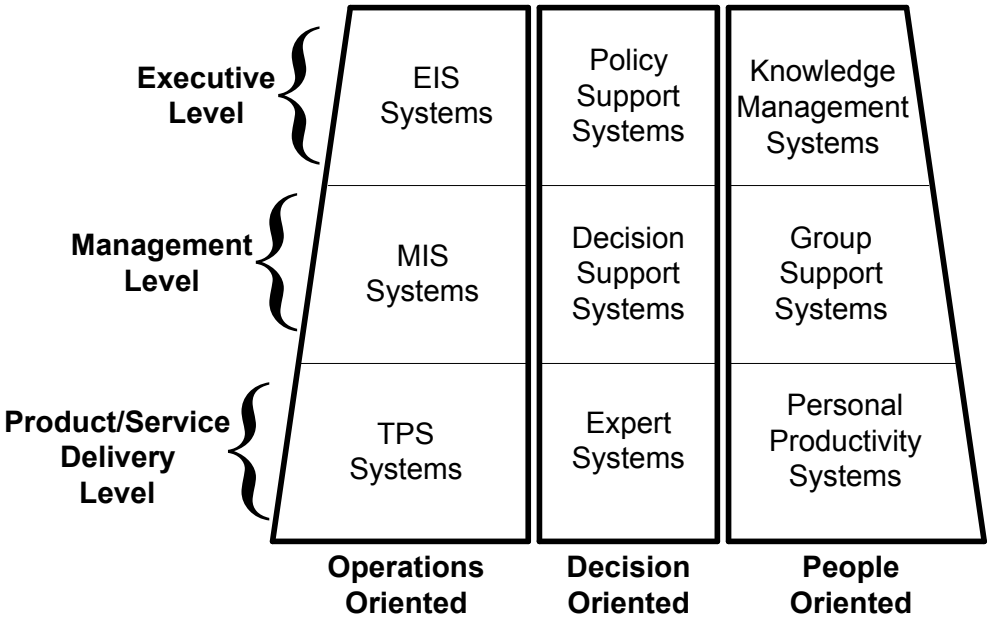

Figure 2: Enterprise Applications Classification Chart

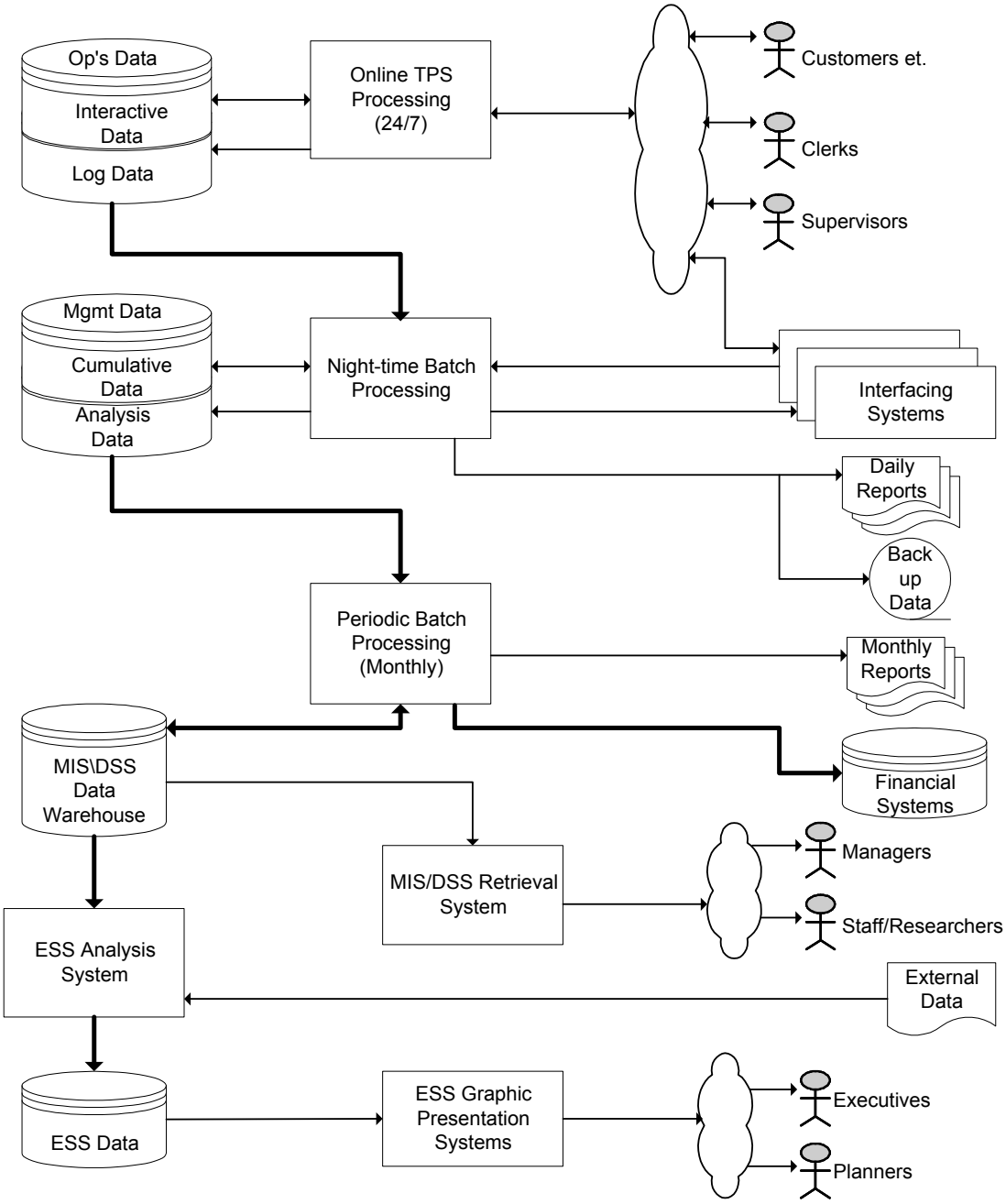

Figure 3: Structure of Transaction Processing Systems

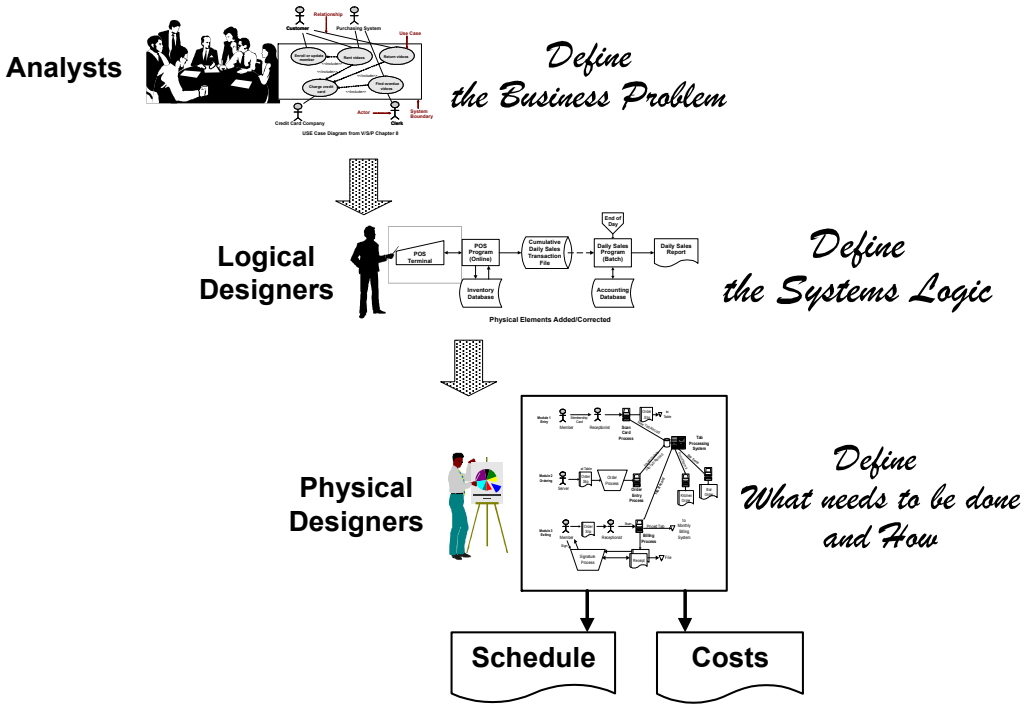

Figure 4: The Design Process

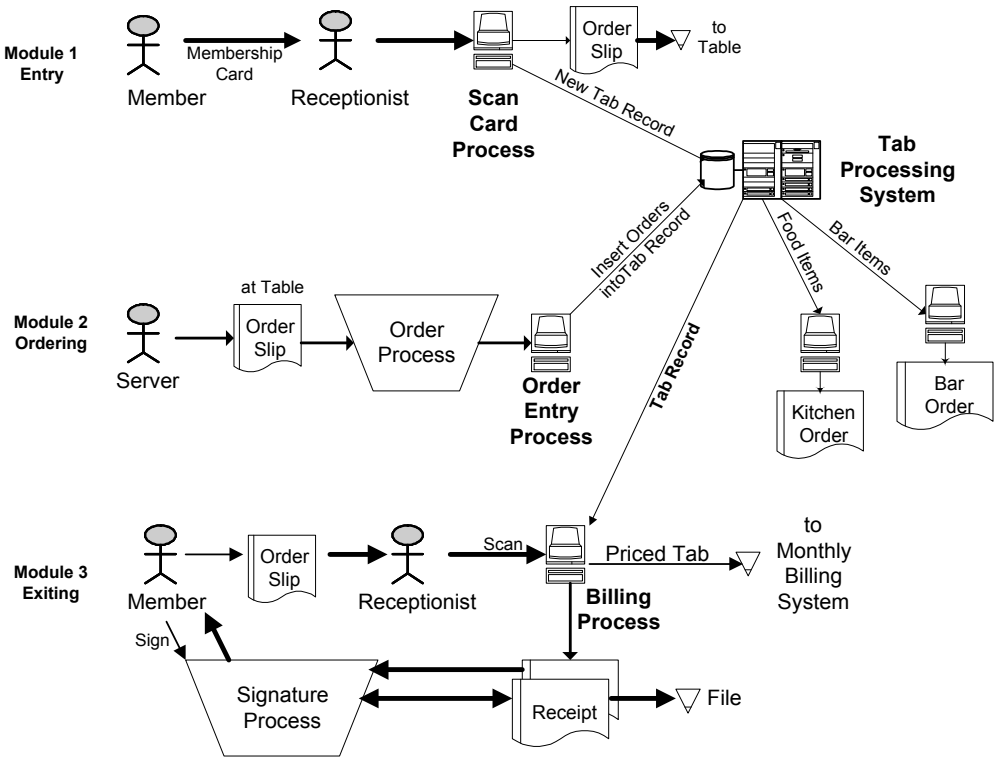

Figure 5: A Physical Level Design Example (Restaurant)

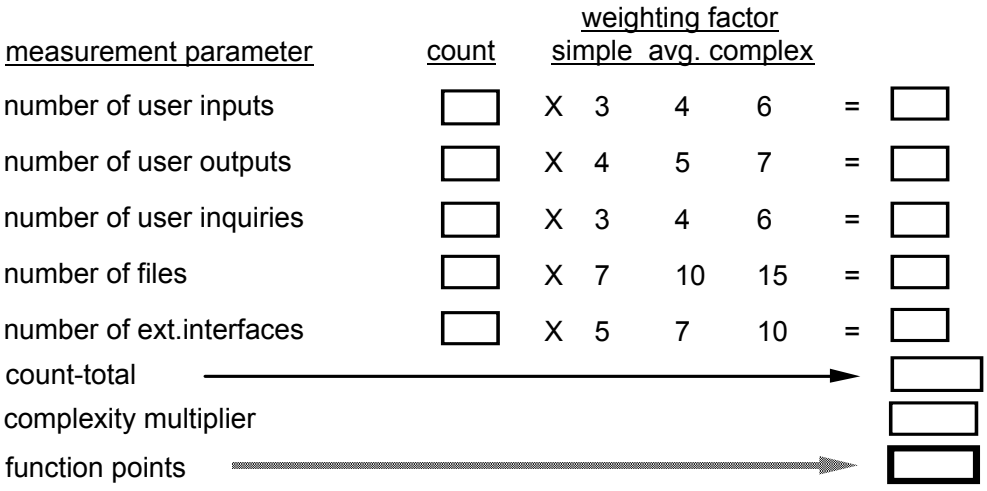

Figure 6: Software Costing Worksheet: Analyzing the Information Domain