SHIFT THE SUBJECT OF SYSTEM ANALYSIS AND DESIGN FROM CONSTRUCTION TO ACQUISITION

Shouhong Wang
Department of Marketing/Business Information Systems
University of Massachusetts Dartmouth
Dartmouth, MA 02747-2300, USA <swang@umassd.edu>

and

Hai Wang
Department of Finance and Management Science
Saint Mary's University
Halifax, NS B3H 2C2, Canada <hwang@smu.ca>

Abstract

Commercialized business application software packages and ERP systems have been widely used to implement business information systems. The major tasks of business information system analysts and designers have been shifted from system construction to system acquisition. This paper proposes the subject of information systems acquisition for the information SAD course. It suggests that the theme of information SAD for business students shall be system acquisition analysis and decision making. It also examines the issues of incorporating system acquisition into the SAD textbook and teaching system acquisition.

Keywords: Systems analysis and design, system construction, system acquisition, analytical hierarchy process.

1. INTRODUCTION

Information systems analysis and design (SAD) lies in the core of the information systems discipline. The techniques and approaches of SAD are continually renovated. About fifteen years ago, SAD projects were more likely to place the focal point on the use of databases and fourth generation languages to implement real business information systems. Gradually, systems users and consultants found that commercialized business application software packages were readily available in the software market. According to (Wang 2005), at least 90% of business applications can be implemented by using ERP systems or off-the-shelf software packages.

As a result of the proliferation of commercialized business applications software, systems design and implementation are no longer the major tasks for most business information technology professionals, but serving the management to perceive strategy value of software systems has become crucial for systems development (Jurison 2000). In this view, the theme of SAD for business enterprises has been shifted from system *construction* to system *acquisition*. As illustrated in Figure 1, in the traditional SAD cycle the software system is the ultimate product of SAD. However, in the system acquirement analysis cycle the system analysts choose commercialized software packages that best matches their system requirements.

This paper is to propose the core subject of systems acquisition for the SAD course for business information system majors, analyze how to incorporate system acquisition into the SAD textbook, and how to effectively teach students to adapt to the new SAD environment.

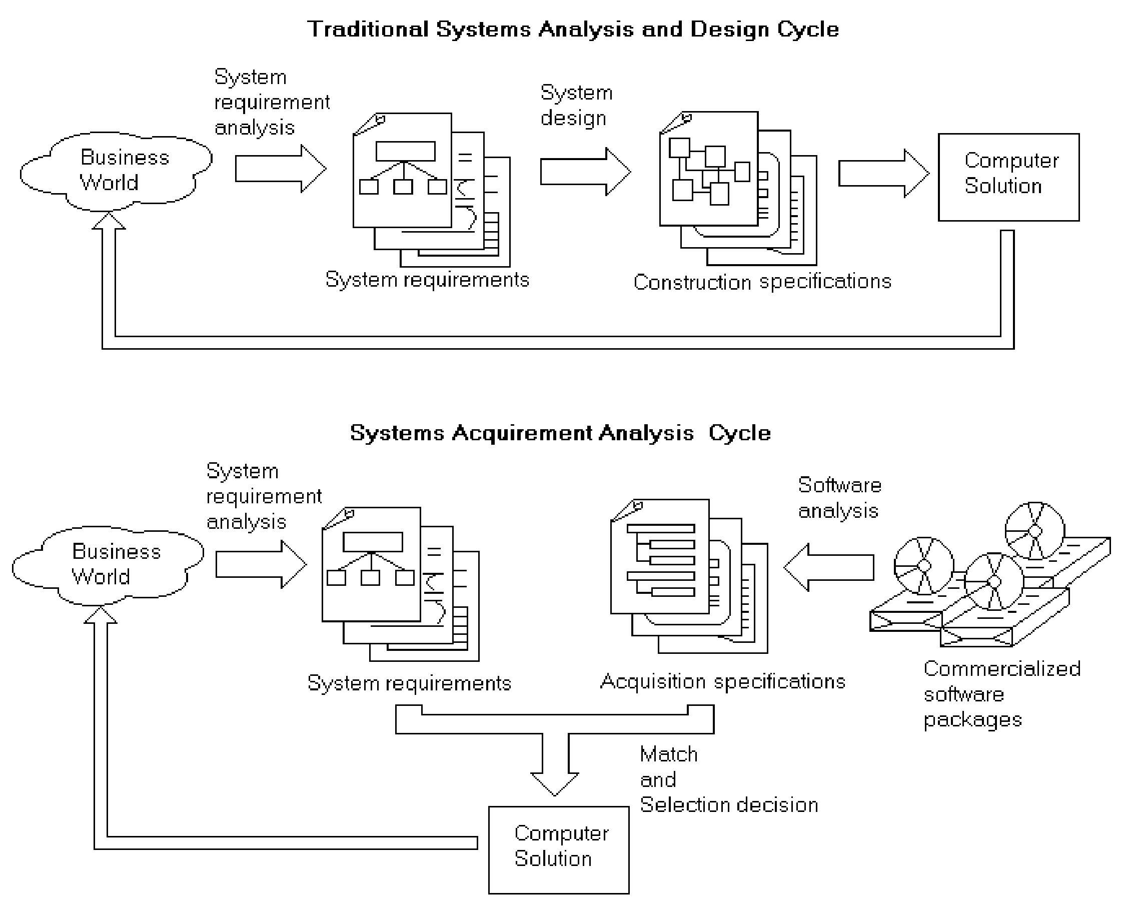
Traditional Systems Analysis and Design Cycle
System requirement analysis
Business World
System requirements
System design
Construction specifications
Computer Solution
Systems Acquirement Analysis Cycle
System requirement analysis
Business World
System requirements
Acquisition specifications
Software analysis
Commercialized software packages
Match and Selection decision
Computer Solution

Figure 1. The Two Systems Analysis and Design Cycles

2. ASPECTS OF INFORMATION SYSTEMS FOR ACQUISITION ANALYSIS

The SAD textbooks currently commonly used in business schools (e.g.,(Kendal & Kendal 2004; Dennis *et al.* 2004; Whitten *et al.* 2004)) place emphasis on systems construction, while including material for systems acquisition analysis. The common subjects of systems construction analysis and system acquisition analysis are the *derivative* system aspects, including cost/benefit analysis, hardware and networking analysis, vendor and technology advance analysis, and security considerations. However, the most important software *functional* aspects for systems acquisition should be analyzed from the user's view rather that from the constructor's view. This point should be reflected in the business SAD course. Next, we discuss the four most important software system functional aspects for system acquisition analysis; they are: business operations, user-computer interface, user-perceived input and output, and business rules.

2.1. Business Operations - Tasks, Processes, and Steps

Business operations implemented by the application software systems are the major concern for organizations (Kendall 1994). Recently, the term *use case* (Jacobson *et al.* 1992), which means about the same as *business scenario* (McGraw & Harbison 1997) does, became popular since its use in UML. In fact, the terms business process, business function, business scenario, and use case have not been rigorously differentiated in the information systems field. The term *business operation* is adopted here because it is more business user-oriented, and is more flexible to use. In the view of the traditional structured modeling approach (e.g., (DeMarco 1978)), which is simple and commonly adopted, a business operation is a hierarchy of sub-operations. Using simple terminology for systems acquisition analysts, it is proposed that a business operation supported by a software system has three levels of sub-operations: tasks, processes, and steps, in accordance with the simplest group-individual structure in business.

(1) *Task* - A business task is a set of business processes performed by a group of actors through the interaction between the actors and the software system to accomplish a specific outcome.

The specifications of a task shall describe the business functionality of the system in accomplishing the task and the type of the actors. Here, actor is referred to a particular role played by the user(s).

(2) *Process* - A business process is a set of steps performed by a single actor through the interaction between the actor and the software system to carry out the associated task. The specifications of a process shall describe the accomplished functionality for the particular actor.

(3) *Step* - A specific action performed by an actor through the interaction between the actor and the software system in carrying out the associated process.

2.2. User-Computer Interfaces

As application software became widespread, the human-computer interaction is one of the most important aspects in software specifications (Diaper & Addison 1992). It is so important because systems development specifications, such as the data flow diagram and UML, emphasize descriptions of functional and data requirements within the context of software engineering, but not within the context of software usability for users (Sutcliffe 1989; Wang 1995). To determine whether an application software package fits the business requirements, one must specify user-centered cognitive aspects of usability (Anonymous 1993; Benyon 1992; Diaper 1989; Harrison & Monk 1986).

A user-computer interface is the part of the software system that allows the user to interact with the system in carrying out the tasks. It includes the screen displays that provide navigation through the software system, as well as the screen displays that capture or generate data (Wang 1995).

Conceptually, there are three basic types of user-computer interfaces: navigation, data capture coupled with decision, and data receipt coupled with decision (Dennis & Wixom 2003).

(1) *Navigation* - A navigation interface is associated with a business process. It provides menus and command buttons that allow the user to proceed through the subsidiary steps.

(2) *Data Capture Coupled with Decision* - An interface for data capture coupled with decision provides forms and command buttons that allow the user to input data and make a business decision or move to another step.

(3) *Data Receipt Coupled with Decision* - An interface for data receipt coupled with decision displays/prints data for the user and allows the user to make a decision or move to another step.

2.3. User-Perceived Inputs and Outputs

In the early ages of the information systems field, inputs and outputs of processes were considered to be central components of SAD in almost every SAD approach (Ballou & Pazer 1985; Carey & McLeod 1988). The typical one of input and output driven approaches is HIPO (hierarchy plus input, process, output) (Stay 1976). Later, research (Srinivasan 1985) indicated that user-perceived inputs and outputs, not system internal inputs and outputs, are the major measures of effectiveness of systems. Research of contemporary object-oriented systems analysis (Wang 1996) and information system planning (Li & Chen 2001) has also confirmed this finding.

2.4. Business Rules

Business rules are constraints or guidelines for business operations. They specify the relationships between an anticipated condition and expected actions/outcomes. Using the currently available computer techniques, business rules are implemented through coded decision procedures or data models. Yet, there is a lack of systematic techniques for mapping business rules and the software system onto each other (Amghar *et al.* 2000). As the perspectives of business rules are crucial for software systems acquirement, important business rules implemented in the software system must be described in an explicit way (Hale et al. 1999).

Workflows are specified in the business operation and interface specifications, and are not considered to be business rules. Here, business rules are referred to those constraints associated to a task or a process.

These four important software system aspects along with important system derivative aspects for system acquisition analysis can be organized into a hierarchy, as shown in Figure 2.

2.5. Software System Aspects That Are Less Important for Acquisition Analysis

The above set of system aspects de-emphasizes several system aspects for software construction, as discussed below.

Construction specifications: The software industry has various system specification instruments with *de facto* standards for software development, such as data flow diagrams (DeMarco 1978; Gane & Sarson

1979), UML (Rumbaugh *et al.* 1999), and entity-relation diagrams (Chen 1976). However, these instruments are used to describe the deep structures and detailed aspects of software systems for construction, instead of the features of software products. Business people would like to have explicit specifications about the business process that can be carried out by the software package, rather than the specifications for the construction of the software system (Wang 2005). This is similar to the fact that consumers of computer hardware or cars never want to review the manufacturing blueprints in making a purchase decision.

System decomposition: In SAD, system decomposition is one of the important issues. This is because the system modules are the construction units for the software development. On the other hand, the software users are not particularly concerned with these construction units, as long as the system supports the required business operation. For instance, in object-oriented SAD, objects are the system modules. However, few software buyers need to understand these object modules. Specifically, a system acquisition analysis shall be based on the users' view of the system instead of the designer's view of the system.

Data modeling: Data models are certainly important for systems construction and software implementation. However, reviewing data model would involve excessive efforts for a system analyst. Also, conceptual data modeling techniques vary depending upon systems development tools (Topi & Ramesh 2002). For instance, the traditional entity-relationship model and object-oriented models use different semantics at the conceptual level. A system acquisition analyst may not familiar with the particular data modeling method used for a software system. In fact, few software producers provide data models for their consumers while marketing their products.

3. SYSTEM ACQUISITION DECISION MAKING

In addition to the emphasized system aspects, the requirements for decision making skills are different between the systems acquisition and systems construction. Unlike systems construction, systems acquisition analysis must involve an intensive decision process to choose one software system among the alternatives. Paradoxically, regardless the closeness of information systems and management science, decision methodologies used for system acquisition are commonly missing in the SAD textbooks.

As the system aspects for systems acquisition are organized into hierarchies, analytical hierarchy process (AHP) (Saaty 1980) is the most feasible, established, and widely applied method in this case. AHP is a multi-attribute decision-making technique through prioritizing the alternatives. In fact, many studies have reported the use of AHP in ERP adoption analyses (e.g., (Roldan et al. 2002; Teltumbde 2000). The decision making process for system acquisition using the AHP technique includes the following steps.

Step 1: Construct a hierarchy of system aspects for the system acquisition (Figure 2).

Step 2: Starting from the top of the hierarch, for each sub-tree of the hierarchy, conduct the pairwise comparison to reveal the comparative importance between the two aspects.

Step 3: Using the principal eigenvector of the pairwise comparison matrix manipulated by scaling ratio, find the comparative weight among the aspects for the sub-tree.

Step 4: Repeat the comparisons from top of the hierarchy until all relative weights have been determined.

Step 5: For each of the system alternatives, assign the values to each of the hierarchical system aspects (depicted in Figure 2).

Step 6: Based on the relative weights of the system aspects and the values of the system aspects for each systems alternative, calculate the score of each system alternative. The system with the highest score will be the best decision for system acquisition.

Computerized AHP algorithm is easy to implement. Practically, one can use commercialized AHP software (e.g., Expert Choice (EC 2003)) for system acquisition decision making.

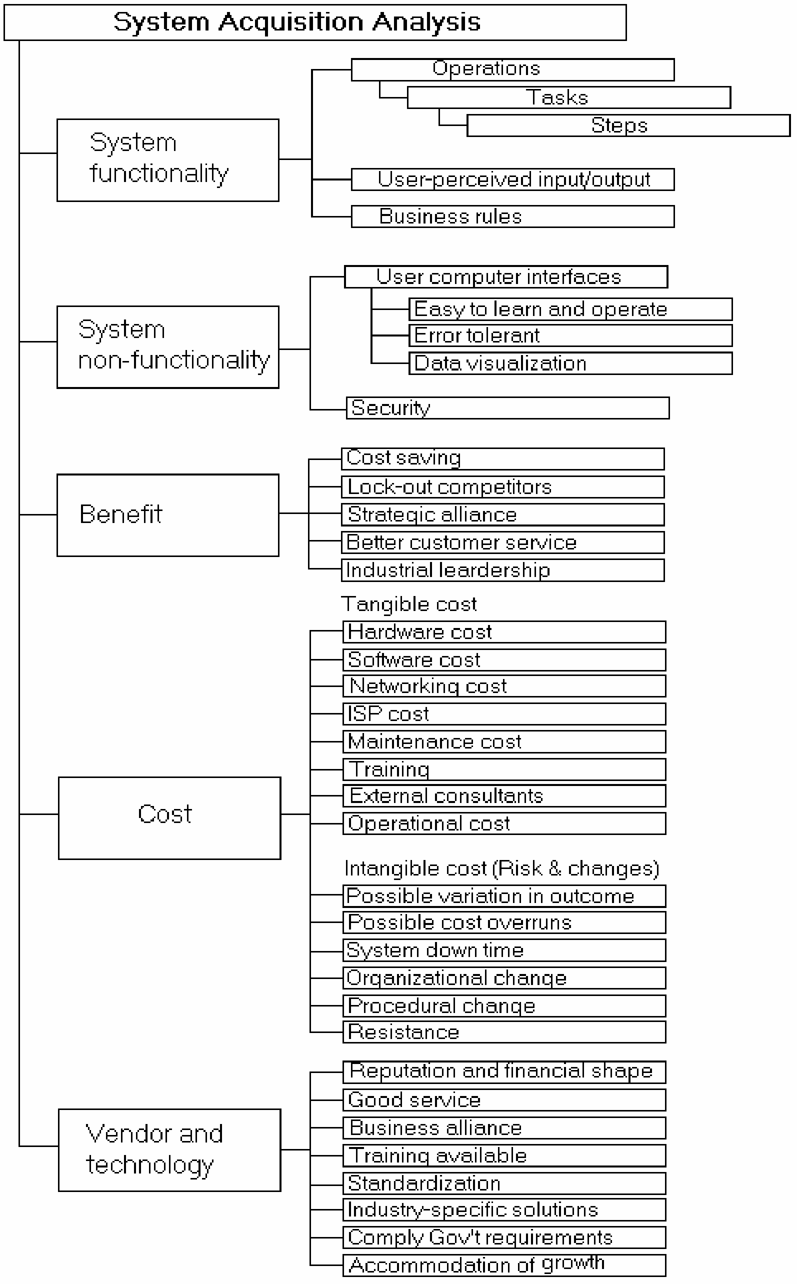

Figure 2. Systems Acquisition Analysis Hierarchy

4. TEACH SYSTEM ACQUISITION

4.1. Incorporate System Acquisition into the SAD Textbook

As analyzed in Section 2, the concept of system acquisition is significantly different form system construction. To provide better teaching material for students, the SAD textbook must incorporate the following components related to the aspects of system acquisition.

(1) System specifications for system acquisition, which include business operations (tasks, processes, and steps), user-computer interface requirements, user-perceived inputs and outputs, and business rules implemented by the information system.

(2) System acquisition decision making techniques, which include AHP and its affiliation methods such as alternative matrices and cost/benefit analysis.

(3) Real-world system acquisition cases that clearly apply the system acquisition concepts.

4.2. Strategies for Teaching System Acquisition

Based on our experiences of teaching system acquisition in the SAD course, we consider the following teaching strategies are effective.

Clearly define the differences between system construction and system acquisition: We need to compare the two system development approaches throughout the entire SAD course. While teaching concepts of system construction, we make it clear that those concepts might not be applicable for system acquisition, as discussed in Section 2. On the other hand, when we bring about system acquisition concepts which are missing in the current SAD textbook, we point out that those concepts are unique to system construction.

Exercise system acquisition through course projects: When students learn SAD from lectures, they often remember little more than lists of keywords. On the other hand, using the course project approach, students go out, find organizations, identify system development opportunities for the organizations, determine system requirements for them, and initiate system acquisition plans. The teaching philosophy of project courses is that people cannot learn without doing (Wang & Ariguzo 2004).

Utilize the Internet and reach realistic solutions: The system acquisition courses project is to teach SAD beyond theoretical frameworks. It allows the instructor to disseminate knowledge through handling real-world cases, and encourage students to further exercise. For instance, the textbook can never tell how to find a software package for a small car repair shop. The instructor shall provide pointers to those practical solutions, and demand students to further explore the Internet. This teaching method invites students to develop their problem-solving skills.

5. SUMMARY

The proliferation of ERP systems and business application software packages has introduced new tasks of acquirement analysis for the SAD field. To facilitate the communication between system acquisition analysts and software builders, application software specifications for systems acquisition analysts must be users-centered instead of builders-centered. This paper proposes core components of SAD for system acquisition. It suggests that the four important functional aspects of software systems should be analyzed based on the user's perspective; they are business operations, user-computer interfaces, user-perceived inputs and outputs, and business rules. These functional aspects along with derivative aspects of the alternative systems are organized into hierarchies for system acquisition analysis. These hierarchies of system alternatives can be analyzed for acquisition decision making using the AHP methods.

Furthermore, this paper has sketched how system acquisition can be incorporated into the SAD textbook, and provides several guidelines for teaching students to adapt to the new SAD environment.

In the history of management information systems, SAD methods have been dominated by computer software builders-centered approaches. On the other hand, the fast growth of ERP systems and commercialized business software packages on the market demands concise and precise business-centered acquisition approaches. The proposed subject of SAD will enhance the business students' systems acquisition skills. We believe the systems acquisition analysis needs to be emphasized in the SAD course for business information system majors.

6. ACKNOWLEDGEMENTS

The comments of an anonymous reviewer has made valuable contributions to the revision of the paper.

REFERENCES

Amghar, Y., M. Meziane & A. Flory, 2000. "Using business rules within a design process of active databases." *Journal of Database Management*, 11(3), pp.3-15.

Anonymous, 1993. "Two communities, two languages." *Communications of the ACM*, 36(4), p.113.

Ballou, D. P., & H. L. Pazer, 1985. "Modeling data and process quality in multi-input, multi-output information." *Management Science*, 31(2), pp.150-162.

Benyon, D. 1992. "The role of task analysis in systems design." *Interacting with Computers*, 4(1), pp.102-123.

Carey, J. M., & R. McLeod, Jr. 1988. "Use of system development methodology and tools." *Journal of Systems Management*, 39(3), pp.30-35.

Chen, P. P. 1976. "The entity-relationship model - toward a unified view of data." *ACM Transactions on Database Systems*, 1(1), pp.9-36.

DeMarco, T. 1978. *Structured Systems Analysis and Design*, New York, NY: Yourdon.

Dennis, A., & B. H. Wixom, 2003. *Systems Analysis and Design*, 2nd Edition, New York, NY: John Wiley & Sons.

Dennis, A., B. H. Wixom, & D. Tegarden, 2004. *Systems Analysis and Design with UML Version 2.0*, New York, NY: John Wiley & Sons.

Diaper, D. 1989. "The discipline of HCI." *Interacting with Computers*, 1(1), pp.3-5.

Diaper, D., & M. Addison, 1992. "Task analysis and systems analysis for software development." *Interacting with Computers*, 4(1), pp.124-139.

EC, Expert Choice, 2004. <http://www.expertchoice.com>. [Accessed August 25, 2004].

Gane, C., & T. Sarson, 1979. *Structured Systems Analysis: Tools and Techniques*, Englewood Cliffs, NJ: Prentice-Hall.

Hale, D., S. Sharpe, & J. E. Hale, 1999. "Business - Information systems professional differences: Bridging the business rule gap." *Information Resources Management Journal*, 12(2), pp.16-25.

Harrison, M. D., & A. F. Monk, (eds.) 1986. *People and Computers: Designing for Usability*, UK: Cambridge University Press.

Jacobson, I., M. Christerson, P. Jonsson, & G. Overgaard, G. 1992. *Object-Oriented Software Engineering: A Use-Case Driven Approach*, Reading, MA: Addison-Wesley.

Jurison, J. 2000. "Perceived value and technology adoption across four end user groups." *Journal of End User Computing*, 12(4), pp.21-28.

Kendall, J. E. 1994. "End user reengineering: Breaking the rules systems developers wrote." *Journal of End User Computing*, 6(2), pp.24-26.

Kendall, K. E. & J. E. Kendal, 2004. *Systems Analysis and Design*, 6th Ed. Upper Saddle River, NJ: Pearson Prentice Hall.

Li, E. Y., & H. G. Chen, 2001. "Output-driven information system planning: A case study." *Information & Management*, 38(3), pp.185-199.

McGraw, K., & Harbison, K. 1997. *User-Centered Requirement: The Scenario-Based Engineering Process*, Mahwah, NJ: Lawrence Erlbaum Association.

Roldan, M. D. G., A. R. Zamora, A. R. and F. D. Amores, 2002. Assessing Enterprise Resource Planning (ERP) Adoption in the Philippines, in L. Hossain, J. D. Patrick, and M. A. Rashid (Ed.) *Enterprise Resource Planning: Global Opportunities and Challenge,* Hershey, PA: Idea Group Publishing.

Rumbaugh, J., I. Jacobson, & G. Booch, 1999. *The Unified Modeling Language Reference Manual*, Boston, MA: Addison-Wesley.

Saaty, T. L. 1980. *The Analytic Hierarchy Process: Planning, Priority Setting*, New York: McGraw Hill.

Srinivasan, A. 1985. "Alternative measures of system effectiveness: Associations and implications." *MIS Quarterly*, 9(3), pp.243-253.

Stay, J. F. 1976. “HIPO and integrated program design.” *IBM Systems Journal*, 15(2), pp.143-154.

Sutcliffe, A. 1989. "Task analysis, systems analysis and design: symbiosis or synthesis?" *Interacting with Computers*, 1(1), pp.6-12.

Teltumbde, A. 2000. "A Framework for Evaluating ERP Projects." *International Journal of Production Research*, 38(17), pp.4507-4502.

Topi, H., & V. Ramesh, 2002. "Human factors research on data modelling: A review of prior research, an extended framework and future research directions." *Journal of Database Management*, 13(2), pp.3-19.

Wang, S. 1995. "Object-oriented task analysis." *Information & Management*, 29(6), pp.331-341.

Wang, S. 1996. "Toward formalized object-oriented management information systems analysis." *Journal of Management Information Systems*, 12(4), pp.117-141.

Wang, S. 2005. "Business Software Specifications for Consumers: Towards a Standard Format." *Journal of Organizational and End User Computing*, 17(1), pp.23-27.

Wang, S. & G. Ariguzo, 2004. "Knowledge Management through the Development of Information Schema." *Information & Management*, 41(4), 2004, pp.445-456.

Whitten, J. L., L . D. Bentley, K. C. Dittman, 2004. *Systems Analysis and Design Methods*, Boston, MA: McGraw-Hill.