INFORMATION TECHNOLOGYAND SYSTEMS - I
SYSTEMS ANALYSIS AND DESIGN:
SHOULD WE BE RESEARCHING WHAT WE TEACH?

Akhilesh Bajaj
University of Tulsa

Dinesh Batra
Florida International University

Alan Hevner
University of South Florida
ahevner@coba.usf.edu

Jeffrey Parsons
Memorial University of Newfoundland

Keng Siau
University of Nebraska-Lincoln

ABSTRACT

A guiding premise of academic scholarship is that knowledge gained from first-hand research experience is disseminated to students via the classroom. However, that valuable connection is lost when professors are not researching what they teach. In this paper, we explore issues of mismatch between teaching and research in the Information Systems (IS) discipline. Specifically, while systems analysis and design (SA&D) is an integral topic in IS curricula, this topic is the research specialty of few IS professors. This situation is reflected by the low number of research publications in this area; particularly in the leading mainstream IS journals. We characterize the gap between teaching and research in SA&D, offer possible explanations for this gap, suggest avenues to better understand and enhance SA&D research via the design science paradigm, list a number of areas in SA&D in which there is ample need and opportunity for high quality research, and show through an example how a research mindset can be incorporated in a graduate level SA&D course.

Keywords: systems analysis and design, IS research, IS teaching, design science

I. INTRODUCTION

Systems Analysis and Design (SA&D)[1] is an important topic that is taught in almost all undergraduate and graduate programs in IS. It appears as a core course for IS majors and minors in the IS 2002 model undergraduate curriculum [Gorgone et. al, 2003] and as part of the core of the MSIS 2000 model graduate curriculum [Gorgone et al. 2003]. In this rapidly changing core area, one should expect modern textbooks and curricula to incorporate sound research on methodologies, techniques, and leading practice. Yet, there appears to be a relatively low level of SA&D research during the past decade, relative to the needs of the teaching and practice communities. We term this shortfall the teaching-research gap in SA&D.

In this paper, based on a panel presented at the 2004 Americas Conference on Information Systems (AMCIS) in New York City, we examine the dimensions, possible origins, and consequences of the SA&D teaching-research gap. The paper is organized as follows. Section II begins with an analysis of the gap and identifies some factors that may contribute to it. Section III argues that SA&D research often follows a design science paradigm [Hevner et al., 2004]. As an example, we discuss how research on SA&D concepts, models, methods, and instantiations can be described by key components of the design science research framework. We then identify some important open research questions in SA&D. Section IV proposes some strategies that may help close the teaching-research gap. Section V provides an example illustrating the effective use of relevant research in teaching a graduate level SA&D course. Section VI concludes the paper by calling on the IS research community to narrow this gap.

II. THE SA&D TEACHING-RESEARCH GAP

The analysis and design of information systems is arguably at the center of the MIS discipline. SA&D is a required course in almost all university IS curricula. Moreover, SA&D is perhaps the only course whose core topics are seldom covered in other disciplines, such as management, marketing, finance, or economics. Some content overlaps with software engineering courses in computer science curricula, although the focus there is largely technical with little attention on the management of the software development process. Therefore, we would expect to see a sizeable quantity of research on these topics in top IS journals. However, recent data suggest that research in SA&D is under-represented. Vessey, Ramesh, and Glass [2002] find that, in the period between 1995-2000, research on SA&D (including database topics) accounted for only 7% of the total research in terms of the number of papers in a set of leading IS journals. Moreover, only 3% of faculty listed in the ISWorld Faculty Directory (http://www.isfacdir.org/) list SA&D as a research interest. In contrast, 22% of faculty express interest in teaching "analysis and logical design."[2] These data leave little doubt that the teaching-research gap is real.

The scarcity of research in SA&D is puzzling given that over the last decade the area continued to evolve with exciting new developments. As an illustration to which we will return, at present there are three general system development approaches, each with several variations, in teaching SA&D:

- the traditional Systems Development Life Cycle (SDLC) methods;

- Object-Oriented Systems Analysis & Design (OOSAD) methods; and

---

[1] We include topics related to database design. These topics may be covered in SA&D or database courses, both of which are core elements of IS curricula.

[2] The data for "database" interests are similar, with 12% indicating this subject as a teaching interest, but only 1.4% indicating a research interest in that area. Note that multiple entries are allowed for both teaching and research interests on the ISWORLD Faculty Directory, so there is likely some overlap between those selecting SA&D and database as teaching and/or research interests.

- agile methods.

The absence of a level of SA&D research concomitant with developments in the field suggests that teaching in the area is not generally informed by rigorous academic research. Rather, it seems likely that faculty members who teach these courses are choosing topics and content based on practitioner literature, textbooks, and the perceived industry popularity of the different approaches. Moreover, it is possible that flawed approaches are being taught.

As suggested by Stone [1978], practice alone cannot provide us the "ways of knowing" for teaching effective tools, techniques, and methods. Cohen and Nagel [1934] list four ways of knowing: *tenacity, authority, intuition*, and *science*.

- Tenacity is the tendency to continue to believe a proposition through habit or inertia. For the SA&D teaching area, tenacity translates to a tendency to continue to use syllabi and knowledge regarding a particular SA&D approach that was acquired by the educator early in a career and was delivered to students successfully by the educator for an extended period of time.

- Authority involves appealing to some highly respected source to substantiate the views held. Given the gap between SA&D research and teaching, authority implies an educator's use of an SA&D approach that is based on their belief in the authority of a practitioner guru, a textbook author, or de facto industrial use.

- Intuition relies upon the appeal to "self-evident propositions". For example, the popularity of pair programming in an agile methodology may be based more on intuition than on research evidence.

- Science, the fourth way of knowing, aims at objective knowledge based on rigorous academic research. Given the teaching-research gap in SA&D, it is probable that educators' choices are influenced as much by tenacity, authority or intuition as by science.

Why does the teaching-research gap exist in SA&D? We offer several possible factors as explanations for consideration:

1. *Publication Outlets:* The top-ranked journals in IS focus mainly on behavioral science topics. Thus, SA&D research tends to appear in journals outside the IS mainstream. Many academics are motivated to publish in the top journals for career advancement. The lack of SA&D research in the leading IS journals may discourage researchers (particularly doctoral students and new faculty) from pursuing these topics.
2. *Business School Environment:* Most IS programs in North America are housed in business schools. SA&D research is frequently closer to what Hevner et al. [2004] term design science as opposed to more IS mainstream behavioral science. Thus, this research may not be easily understood or appreciated by faculty in other business disciplines such as economics, marketing, finance, or organizational behavior. This situation poses a potential problem when it is time for tenure and promotion, and may discourage SA&D research by IS faculty.
3. *Inadequate Training of PhD Students:* From the authors' experiences, few IS PhD students seem to have the technical background to pursue SA&D research. Many schools do not require doctoral students to take more than one introductory course in technical topics, such as SA&D, database systems, and telecommunications. Thus, they are not introduced to the most recent ideas in these areas. Often, doctoral students must take courses outside of the business school to obtain this content. Based on our observations and conversations with numerous IS academics, it appears that seminar classes on SA&D are missing in many PhD programs. The small population of IS researchers in the SA&D area results in few PhD students trained to do such research, leading to a negative feedback loop.
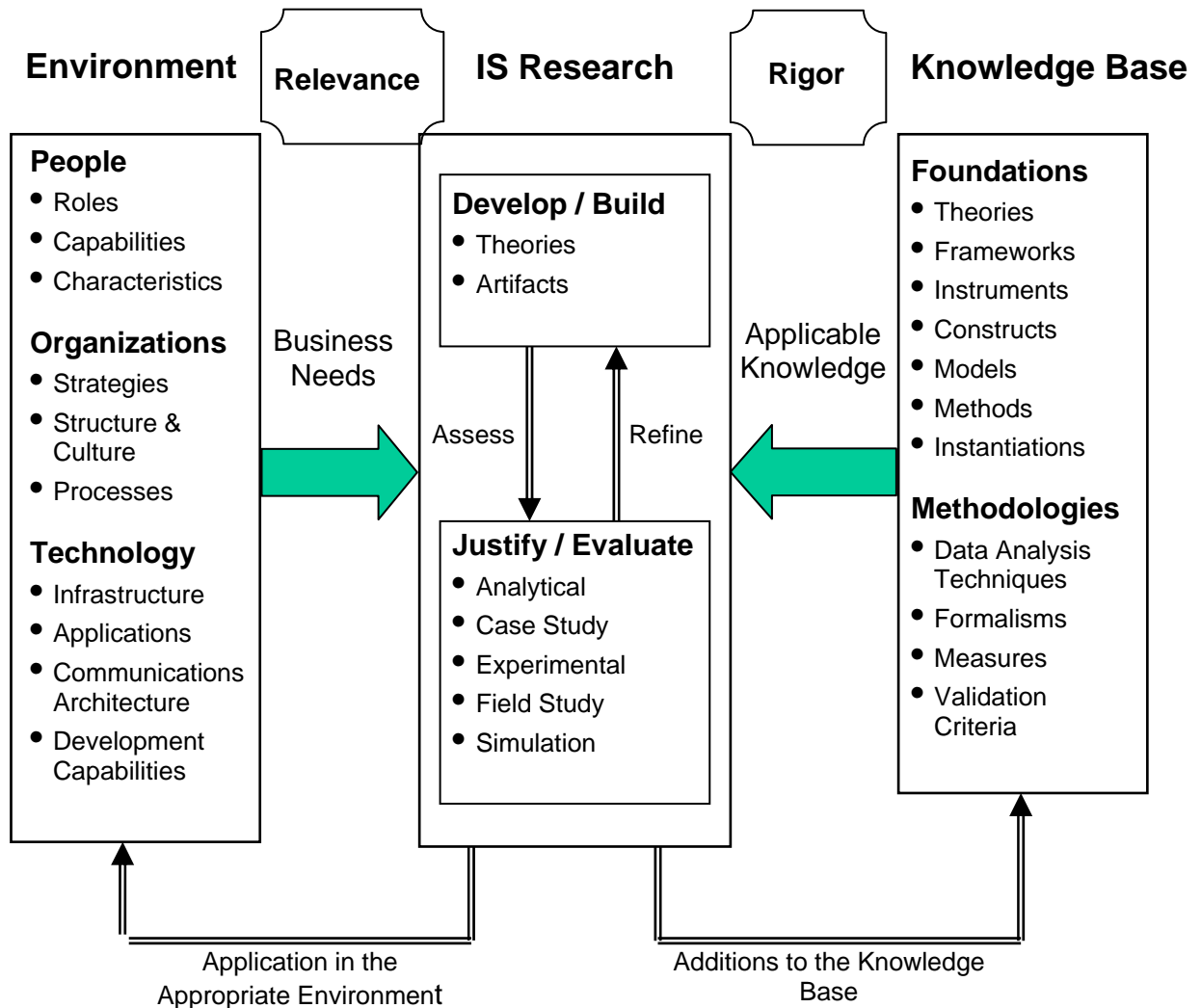
4. *SA&D* is perceived as a well known practice: Some may claim that further research in SA&D only adds marginally to existing knowledge. Yet, as we show in Section III many fundamental SA&D issues in today's environment are unresolved.   For example, how do basic principles of one system development approach differ from that of another? More importantly, why do organizations continue to conclude that billions of dollars that are spent on IT every year are wasted [Robey, 2001]? We lack answers to fundamental questions such as: "Under what conditions is a given SA&D methodology effective?" We believe faculty members teach this course largely based on what appears in the textbooks. Since there is inadequate research in the area, SA&D textbooks rely significantly on practitioner literature, much of which is based on anecdotal observation.   While practitioners are to be commended for their contributions, one goal of applied academic research is to test practitioner observations and claims systematically. We can even argue that academic research should lead the way in this area (possibly in partnership with practitioners), and propose new analysis and design methods to enhance the success of software development (Section III).

5. *Alleged lack of rigor*: Often, it seems that SA&D research is expected to provide the same kinds of theoretical bases as behavioral science. However, SA&D research under the design science paradigm must be evaluated differently. The creation and evaluation of a design artifact are key issues to be judged in SA&D research. Thus, rigor in SA&D needs to be assessed using different criteria. Hevner et al. [2004] provide a robust checklist to ensure rigor in design science research.

III. DESIGN SCIENCE RESEARCH QUESTIONS IN SA&D

It is encouraging to note that SA&D contains many interesting and challenging research questions to be studied. In addition, there is a well-defined base of knowledge to draw on to execute design science research in the field. Thus, as shown in the IS research framework of Figure 1, the essential issues of relevance and rigor are satisfied. The framework describes the environment from which IS research questions arise, the knowledge base on which IS research draws, and the products of IS research.

SA&D touches on several areas of the IS research framework in Figure 1. In the knowledge base section, SA&D contributes by providing the models used to represent requirements and systems, and the methodologies used to develop systems drawing from several theories such as cognitive theories, frameworks [e.g., Markus, Majchrzak, and Gasser, 2002; Walls, Widmeyer, and El Sawy, 1992], and ontologies like Bunge-Wand-Weber (BWW) [Wand and Weber, 1993; Weber, 2003]. In the IS Research Section, the framework identifies artifacts as a product of IS research, where artifacts can range from initial system requirements, to formal representations of systems, and to actual software. SA&D feeds the knowledge base via the creation of several IS artifacts. SA&D research can employ various research strategies such as laboratory experiments, field study, case study, action research, simulation, and analytical methods. SA&D also touches the Environment section given that SA&D research can be done in an organizational environment and incorporates the effects of personal or organizational characteristics.   The framework provides a useful illustration for the fit of rigorous and relevant SA&D studies in the overall context of IS research.

With this understanding of how SA&D fits into the overall IS research framework, we can move to the description of several open research questions in SA&D. This survey is not comprehensive; the purpose is simply to illustrate that there are indeed a number of rich research opportunities in SA&D. First, we indicate the nature of SA&D research by discussing an example - the evaluation of existing SA&D models and methodologies, and the creation of new models and methodologies. This brief survey provides a flavor of the design science elements of SA&D research. We also mention limitations of this line of research. This is followed by a discussion of additional interesting research issues that can be pursued by SA&D researchers.

Source: [Hevner et al., 2004][3] © University of Minnesota 2004. Reprinted by Permission

Figure 1: The IS Research Framework

EVALUATION OF CONCEPTUAL MODELS: AN SA&D RESEARCH EXAMPLE

The evaluation of conceptual modeling methods maps well with the design research framework presented in Figure 1 and the previous subsection. Several theories, frameworks, ontologies, and research models are employed, and the line of research is generally based on experimental work. Data analysis techniques and measures are developed, and selected dependent and independent variables are used.

A survey of the literature on the evaluation of modeling methods shows several desirable attributes for conceptual modeling methods, which were used as dependent variables in past empirical studies. These attributes include

---

[3] Copyright @ 2004, Regents of the University of Minnesota. Used with permission.

Systems Analysis and Design: Should We Be Researching What We Teach? by A. Bajaj, D. Batra, A. Hevner, J. Parsons, and K. Siau

1. the adequacy or completeness of the modeling method in being able to represent the underlying reality [Amberg, 1996; Bajaj and Ram, 1996; Brosey and Schneiderman, 1978; Kramer and Luqi, 1991; Moynihan, 1996; Siau, 2004],

2. the readability of the modeling method's schemas [Hardgrave and Dalal, 1995; Shoval and Frummerman, 1994], and

3. how easy it is to use the modeling method to *represent* requirements [Bock and Ryan, 1993; Kim and March, 1995; Shoval and Even-Chaime, 1987; Siau and Cao, 2001].

Independent variables include modeling grammar, designer/user individual differences, and task. The modeling grammar variable may consider grammars such as entity relationship diagrams, class diagrams, relational schemas, use cases, or data flow diagrams, as well as variants of a grammar motivated by different theoretical considerations. Designer/user variables may consider the level of experience and familiarity of the subjects with the conceptual model used. Readers who are more experienced in the underlying conceptual model are thought to perform better at interpreting the schemas as well. In most studies, using subjects with similar backgrounds for all treatment levels has controlled this variable. Task variables include problem solving, comprehension, and recall. Past studies attempted to control for the level of familiarity with the domain by employing domains that are reasonably familiar or completely unfamiliar to all subjects, and further, by randomly allocating subjects across treatment levels. A random allocation reduces the likelihood of small differences in domain familiarity among subjects in different treatment levels. Another task variable is the underlying complexity of the requirements for a particular situation, where a more complex set of requirements is harder to reconstruct than a simpler set. Such complexity is controlled by using the same requirements case across treatments [Juhn and Naumann, 1985; Kim and March, 1995; Peleg and Dori, 2000], or studying several levels of complexity [Batra and Wishart 2004].

There are several other similar studies. What is common among them is the use of the laboratory method. This research strategy is appropriate for usability studies. The studies typically were rigorous and generally used theories, frameworks, and ontologies such as the BWW ontology [Wand and Weber, 1993], GEMS model [Reason, 1990], classification theory [Parsons, 2003], cognitive distance [Hutchins, Hollan, and Norman, 1986], ACT [Anderson, 1996], cognition [Newell, 1987], and cognitive mapping [Eden, 1988] among others. The studies employed similar grading schemes, and instruments such as perceived usefulness and perceived ease of use [Davis, 1989], and generally used novice designers as subjects.

Thus, the evaluation of conceptual models is a line of research that maps well with the design science research framework. The mapping also shows that the line of research is not complete. Although the studies are high in internal validity, the generalizability of the findings is limited. External validity needs to be improved with studies that employ practitioners and realistic-sized applications. In addition, a mix of research methodologies may be useful. Findings from laboratory studies need to be validated using survey and case methods.

EMERGING RESEARCH AREAS IN SA&D

Conceptual modeling, discussed in the previous subsection, is just one of the many research areas in SA&D in which several challenging research issues wait to be investigated. The field of SA&D is at a crossroads today as we witness methodologies taking diverse positions. On one hand, there are mature methods based on the disciplined, structured approach which focus on detailed planning and analysis. On the other hand, emerging agile methodologies are challenging the core concepts of the conventional approaches. In between these approaches are iterative methodologies that try to balance both approaches. Instructors continue to teach what the books offer – sometimes all three approaches – without questioning or answering under what conditions each of the three approaches – structured, iterative, or agile – should be employed.

While practitioner guidelines exist (e.g., [Royce 1998]), only rigorous research can provide answers to the issue of what Boehm and Turner [2004] call the balance between discipline and agility. Although one may be inclined to state that the IS industry seems to adopt the iterative approach, empirical evidence of this trend is lacking, as is evidence of whether or not it leads to improvements in systems development. The iterative approach seems like a fair compromise between the planning- and analysis-heavy SDLC, and the lightweight agile approaches that focus on coding. However, speculation in the area can be dangerous. It is also possible that we can merge the best principles in a new approach. Or, as suggested by Boehm and Turner [2004], each approach may be suitable for a range of applications depending on the risk characteristics of the projects.

Although the iterative approach seems to hold promise, it needs to be researched given that its project management is quite complex. We need to study how such an approach will work in today's distributed development environment where the implementation might take place in an offshore site, which adds another layer of complexity. Such research would require field, case, and action research based research strategies.

Another issue with the iterative and agile approaches is their emphasis on exploring solutions early in the life of the project. Although this is a worthwhile method of testing feasibility, it does raise the possibility of bias because of the anchoring phenomenon [Gilovich, Griffin, and Kahneman, 2002; Parsons and Saunders, 2004]. For example, Kroll and Kruchten [2003] recommend at least one solution at the inception stage and performance testing at the elaboration stage of requirements elicitation.

UML

Today, the most popular modeling language is the Unified Modeling Language (UML), which continues to evolve with the addition of new concepts. A typical syllabus on SA&D includes some coverage of UML. Research on its usability is limited although a few recent studies begin to address this question, showing strong differences among practitioners in the perceived value of different UML components [Dobing and Parsons, forthcoming] and limitations [Shen and Siau, 2003]. Such research can help SA&D instructors focus on the more important areas of UML.

Modeling

It is assumed that models, whether based on UML or some other grammar, are quite useful. This assumption is not unfounded since many of us cannot visualize how a system can be developed without models. However, a tricky issue is whether modeling continues to be useful after the system is developed. Several studies show that the maintenance cost over the lifetime of a system can be as high as 80% of the total system cost. It is likely that, in practice, maintenance is more of a matter of coding and 'fixing the problem'. Over a period of time, the models and code can become inconsistent. This inconsistence can spell trouble in an industry that historically witnessed fairly high employee mobility and turnover. The cost and benefits of keeping models and code consistent is an important research issue, which can provide useful insights for both teaching and practice.

Object Oriented Development

An important facet of the object-oriented development approach is the notion of use cases, which are generally taught in the typical SA&D course. However, use cases capture only about a third of requirements [Rosenberg and Scott, 1999] and prescriptions for their use vary greatly [Dobing and Parsons, 2000]. Given the increasing importance of the use case method of requirements elicitation, an approach is needed that integrates use cases with the remaining pieces of requirements [Siau and Lee, 2004]. One way to come up with such an approach is to conduct action research. Although we teach use cases in the classroom, little is known about the granularity of use cases. Cockburn [2000] discusses five levels of granularity – cloud, kite, sea level, fish, and clam. The cloud level represents a high level of abstraction, while the sea-level represents a desirable level of abstraction with adequate detail about the use case. A research

study of the usefulness of each of the five granularities of the use cases (*e.g.,* how practical are the finer levels of fish and clam?) can inform teaching in this area.

Agile Methods

The advent of agile methods with its emphasis on pair programming and a close-knit software development team raises the issue of communication among team members. For example, agile methods rely on sharing tacit knowledge; however, whether this practice scales well is a challenging question. Conversely, empirical work needs to be done to verify if high discipline organizations using conventional approaches [Ahern, Clouse, and Turner, 2003] actually achieve success in today's volatile and budget-conscious environment.

Distributed Development Environments

As outsourcing and offshoring become increasingly important, distributed software development environments are likely to become the rule rather than the exception [Davies, 2004]. This change may lead to the formation of teams, both physical and virtual, from organizations that differ in culture (e.g., institutional,   regional, ceremony, capability). Culture differences may result in interesting and research worthy dynamics. Specifically, the management of outsourcing software development, especially offshore, is an important research topic. Another issue is developing systems by synthesizing components as the process becomes more open and less proprietary, similar to other engineering disciplines.

IV. ADDRESSING THE TEACHING-RESEARCH GAP IN SA&D

What can be done to narrow the teaching-research gap in SA&D and to elevate the visibility of SA&D research in IS? It is impractical to believe this gap will disappear overnight. However, we believe the following concrete steps can be taken within the MIS research community as a start.

DESIGN SCIENCE JOURNALS

Journals that publish top quality design science research should be recognized as equivalent to existing top-tier IS journals. Design science research in SA&D is published most frequently in computer science journals from the IEEE Computer Society and ACM. Recognizing these journals, however, poses a number of problematic issues.

First, these journals are outside the MIS discipline (and the business disciplines) and their editorial boards consist mainly of computer science and software engineering researchers. This situation hinders the establishment of a legitimate technical research area whose focus is IS.

Second, many of the IEEE and ACM transactions are slow in the review process. This may be acceptable to computer science researchers because the computer science field counts conference publications heavily for tenure and promotion. However, conference papers typically carry little or no weight in IS departments or business schools.

Third, IEEE and ACM transactions are very specific and narrowly focused towards different areas in computer science. For example, the *IEEE Transactions on Software Engineering*, the *IEEE Transactions on Knowledge and Data Engineering*, the *ACM Transactions on Software Engineering and Methodology*, the *IEEE Transactions on Professional Communications*, and the *ACM Transactions on Database Systems* each service a specific area of inquiry*.* Unless we regard all the IEEE and ACM transactions as top-tier MIS journals, we will be forced to prefer some and thus will emphasize one research area over another. Fortunately, many business schools do consider publications in IEEE and ACM transactions as top-tier publications (even though IEEE and ACM transactions may not be listed "officially" on the top-tier journals list of a school). Nevertheless, candidates coming up for tenure or promotion typically must justify these IEEE and ACM transactions as top-tier journals since they are not on the official lists, which can be a political and subjective process. Thus, while IEEE and ACM transactions are regarded by

many in the MIS community as top journals in the MIS area, there is a need for one or more design/technical MIS journals that belong to the MIS community and are regarded as top-tier MIS journals by the MIS community. The editorial board of these journals should mainly be MIS researchers and the journals should be receptive of high-quality papers from the wide range of requirements, analysis, design, and technical research in MIS.

RECEPTIVENESS OF IS JOURNALS TO SA&D RESEARCH

At the same time, we need to encourage existing top-tier journals in IS to be more open and receptive of SA&D research. This goal may present initial difficulties as most existing journals already established their niche areas of focus and set up their editorial boards to further support their areas. However, we believe this approach is feasible. Some top journals seem to be more receptive of SA&D articles when their chief editors themselves perform research in the area. For this to change to be realized, an increased number of Senior and Associate Editors need to be appointed to the editorial boards of different IS journals that are knowledgeable and appreciative of SA&D research. The editorial board members also need to recognize that the writing style and the components of a paper on SA&D research may be different from papers in other IS areas. SA&D researchers may hesitate to submit to top-tier MIS journals because of the concern that the contribution of a design science research paper will be mitigated in the review process.

One possibility is for the current top journals in IS to be split into different departments. For example, *Management Science* is organized in several departments, and *IEEE Transactions of Systems, Man, and Cybernetics* contains three parts (in a sense, three different journals). It may be possible for highly regarded IS journals such as *MIS Quarterly* and *Information Systems Research* to pursue similar strategies to cater to different groups in the IS area.

REPRESENTATION OF SA&D RESEARCH IN IS CONFERENCES

SA&D and other technical research areas need to increase their share of appearances at IS conferences such as the *International Conference on Information Systems* (ICIS). The onus here is on SA&D researchers to send their best work to ICIS. But again, track chairs, program chairs, and reviewers must be aware of the different criteria for evaluating design science research. There are signs this situation is improving with the occasional inclusion of tracks for topics related to SA&D in recent years, although these gains are not consistent.

REQUIRED DOCTORAL COURSES AND SEMINARS IN TECHNICAL TOPICS

Many IS curricula are weak in advanced courses in technical topics. Doctoral students report that they often must take classes offered by other colleges in the university to receive advanced knowledge in information systems development, database systems, and telecommunication topics. Doctoral students typically obtain their training in research and their dissertation ideas from seminar classes. A doctoral technical seminar should train students to perform research in the design science paradigm and expose them to top-quality IS design science research. To stimulate more research in SA&D by doctoral students, a larger number of advanced topic and seminar classes in SA&D and other design research IS areas need to be offered.. If more students pursue SA&D research, there will be more faculty researchers in SA&D, which in turn will result in more doctoral students pursuing these areas.

RECOGNIZE AND BUILD ON EXISTING SA&D RESEARCH

SA&D researchers cannot just stand by and hope things will change for the better. Lately, concerted and coordinated effort and activities by SA&D researchers to promote their research areas increased. Conferences such as the *Workshop on Evaluation of Modeling Methods in Systems Analysis and Design* (EMMSAD) and the *Workshop on Information Technology and Systems* (WITS) provide outlets for SA&D papers, and are helping to promote the research areas. Journals such as *Information Systems* and *Journal of Database Management* are receptive of SA&D research papers. A special joint theme issue of *Journal of AIS* and

Communications of AIS is being assembled. Societies such as the AIS SIGSAND (Special Interest Group on Systems Analysis and Design) are promoting and encouraging research in SA&D areas.

A rich tradition of international research exists in the area of SA&D. Examples include the formation of the IFIP 8.1, the EMMSAD workshop linked to the CAiSE conference, the significant percentage of papers in CAiSE dealing with SA&D, and the international reach of the ER conference. The *Requirements Engineering* journal, edited traditionally out of Europe, is exclusively focused on systems analysis. Several SA&D related articles appear in *Information Systems* and *Information Systems Journal*, high quality journals based in Europe. The *Journal of Database Management*, based in the USA, also publishes systems analysis and design research on a regular basis. We hope that these trends will continue to stimulate interest in SA&D research.

V. AN EXAMPLE OF INTRODUCING RESEARCH IDEAS INTO SA&D TEACHING

The objective of this section is to demonstrate the successful integration of industry best practices and research ideas into an advanced graduate-level SA&D class. A class of this type has been offered by Alan Hevner for the past twenty years at the Universities of Maryland and South Florida.   The course attracts masters and doctoral level students with a wide variety of backgrounds and experiences in the IS field and, in particular, software development. As a prerequisite, students must complete a basic course in SA&D, or show significant experience in industrial software development projects.

To illustrate the effective integration of practice and research in the course, several important selected topics in the curriculum are presented briefly; software development methods, system specification, technical reviews, system quality and student reports on hot topics. No course textbook is used. Student readings come from both the research literature and the industry trade press. The key goal is for students to understand the relationships and synergies between the industry best practices and the active research in each of the topics presented. Students are encouraged to challenge current practice with insights and ideas on how practice can be improved in terms of software product quality and software development productivity.

DISCIPLINED AND AGILE SOFTWARE DEVELOPMENT METHODS

Disciplined software development process models, such as the waterfall and V models, are presented alongside more agile process models. Any software development project must evaluate the tradeoffs between discipline and agility for its own unique context, resources, and project objectives. Current best practice thinking on process modeling is exemplified by the recent text by Boehm and Turner [2004] that recommends capturing information on project risk over several important dimensions and selecting an effective process model based on project risk analysis. As identified in Section III, research is needed to better identify and evaluate the key determinants of a software development project to enable the project team to select the best development method.

SOFTWARE SYSTEM SPECIFICATION AND DESIGN

The use of software models to specify and design software systems displays a significant gap between practice and research. Current best practices in design include UML techniques and automated integrated computer aided software engineering (ICASE) systems that support recent modeling tools. Thus, students in the course are presented an overview of UML [Fowler, 2004] and receive hands-on exposure to an industrial strength ICASE system. It is well recognized that current design methods lack a rigorous base of theory and software development as practiced today is more a craft than an engineering discipline. Thus, in the course, students are given several readings on research that potentially provides a more rigorous foundation to the engineering of software systems. In particular, the concepts of Cleanroom software engineering

are presented [Prowell et al., 1999]. An active research initiative to extend these ideas to network-centric system development acquaints students with the future challenges of software development [Linger et al., 2002].

TECHNICAL REVIEWS/INSPECTIONS

The effective use of technical reviews and inspections of artifacts throughout the software development process is one of the most important best practices in the industry. Students are instructed on the IEEE Standard 1028-1988 inspection process also known as the Fagan inspection process. They become familiar with some of the research that demonstrates the benefits of inspections on text documents such as source code [Porter, Mockus, and Votta, 1998]. Further, students read recent research that investigates the use of inspections on graphic design models, which are becoming more prevalent for representing analysis and design artifacts [Hungerford, Hevner, and Collins, 2004].

SYSTEM QUALITY / METRICS

An important overarching theme of the SA&D course is the issue of quality. What does it mean to develop a quality software system? How do we measure the quality of software? An excellent discussion of software quality and the industry best practices that lead to quality software are presented in McConnell's book on Rapid Development [McConnell, 1996]. This material is complemented by several research papers on the importance of discovering a full set of quality attributes (e.g., performance, reliability, usability) as part of requirements analysis [LeRouge et al., 2004] and the effective use of metrics in the verification of software quality [Hevner, 1997].

STUDENT HOT TOPIC REPORTS

In the limited time of a one-semester course it is impossible to cover all of the important topics in the SA&D field. Therefore, students are given the opportunity to select a hot topic of special interest to them and to explore that topic in depth. In addition to discovering the current industry best practice, the students are challenged to recognize the deficiencies of current practice and to explore active research projects on the topic in academia or industry. The following is a sample of the hot topics covered during the past several course offerings:

- Open Source Software Development

- Offshore Outsourcing of Software Development

- Medical Information Systems and HIPAA Impacts

- Integrating New Technologies into Software Systems (e.g., RFID, GIS/GPS, Voice Recognition, Animation)

- Wireless and Mobile Environments

- Grid Computing

- Computer Forensics

- Web Services and Service Oriented Architectures

COURSE SUMMARY

Effective education in SA&D must be informed by industry best practices and future-looking research investigations. The SA&D course discussed in this section demonstrates the synergies achieved when instruction on a topic integrates both practice and research in its course coverage. Students come away with not only practical skills knowledge but also a desire to challenge the status quo and to apply innovative research ideas into their software development projects. We believe that an instructor who is an active researcher on SA&D topics is better positioned to inspire students in this way than is an instructor without an active SA&D research agenda.

VI. CONCLUSION

The teaching-research gap in systems analysis and design topics is alarming. SA&D lies at the heart of the discipline and forms part of the core of most IS curricula. In addition, the failure rate of systems development projects remain high and the world of SA&D practice is one of constantly evolving methods and approaches. Despite these realities, the quantity and quality of research on SA&D in IS remains low. As a result, SA&D teaching is often based on principles and practices that are of questionable validity. We believe this situation should be of serious concern to both IS academics and practitioners.

In this paper, we identify a number of factors that may contribute to the teaching-research gap, outline a selection of current topics in SA&D that merit research, and propose a set of approaches that should help narrow the gap. We also offer an example of how research issues and questions can be integrated in teaching SA&D at the graduate level. Our goal is to motivate critical thinking and spur future researchers. We call on the IS academic community to devote more attention to advancing knowledge in SA&D through rigorous research on both foundational issues and topics of current interest among practitioners and to transition the value of this research into the teaching of students in SA&D classes.

ACKNOWLEDGEMENT

REFERENCES

Ahern, D. M., A. Clouse, and R. Turner (2003) *CMMI® Distilled: A Practical Introduction to Integrated Process Improvement*, *The SEI Series in Software Engineering*. Reading, MA, Addison-Wesley, Inc.

Amberg, M. A. (1996) "A Pattern Oriented Approach to a Methodical Evaluation of Modeling Methods", *Australian Journal Of Information Systems* (4)1, pp. 3-10.

Anderson, J. R. (1996) *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.

Bajaj, A. and S. Ram (1996) "A Content Specification for Business Process Models", *Australian Journal of Information Systems* (4)1, pp. 22-31.

Bock, D. and T. Ryan (1993) "Accuracy in Modeling with Extended Entity Relationship and O-O Data Models", *Journal of Database Management* (4)4, pp. 30-39.

Boehm, B. and R. Turner (2004) *Balancing Agility and Discipline: A Guide for the Perplexed*. Reading, MA, Addison-Wesley, Inc.

Brosey, M. and B. Schneiderman (1978) "Two Experimental Comparisons of Relational and Hierarchical Database Models", *International Journal of Man Machine Studies* (10), pp. 625-637.

Cockburn, A. (2000) *Writing Effective Use Cases*. Reading, MA**,** Addison-Wesley.

Cohen, M. R. and E. Nagel (1934) *An Introduction to Logic and Scientific Method*. New York: Harcourt Brace Jovanovich.

Davies, P. (2004) *What's this India Business? Offshoring, Outsourcing and the Global Services Revolution.* London, Nicholas Brealey.

Davis, F. D. (1989) "Perceived Usefulness, Perceived Ease of Use and End-User Acceptance of Information technology", *MIS Quarterly* (13), pp. 318-339.

Dobing, B. and J. Parsons (2000) "Understanding the Role of Use Cases in UML: A Review and Research Agenda", *Journal of Database Management* (11)4, pp. 28-36.

Dobing, B. and J. Parsons (forthcoming) "How the UML is Used", *Communications of the ACM.*

Eden, C. (1988) "Cognitive mapping", *European Journal of Operational Research* (36), pp. 1-13.

Fowler, M. (2004) *UML Distilled. Reading, MA,* Addison-Wesley, Inc.

Gilovich, T., D. Griffin, and D. Kahneman (eds.) (2002). *Heuristics and Biases.* New York: Cambridge University Press.

Hardgrave, B. C. and N. Dalal (1995) "Comparing Object Oriented and Extended Entity Relationship Models", *Journal of Database Management* (6)3, pp. 15-21.

Gorgone, J. et al. (2000) "MSIS 2000 Model Curriculum And Guidelines For Graduate Degree Programs In Information Systems" Communications of AIS, (3)1 http://www.aisnet.org/Curriculum/msis2000.pdf

Gorgone, J. et al. (2003) IS 2002 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, *Communications of AIS* (11)1 http://www.aisnet.org/Curriculum/IS2002-12-31.pdf

Hevner, A. (1997) "Phase Containment Metrics for Software Quality Improvement", *Information & Software Technology* (39)13, pp. 867-877.

Hevner, A., S. March, S. Ram, and J. Park (2004) "Design Science Research in Information Systems", *MIS Quarterly* (28)1, pp. 75-105.

Hungerford, B., A. Hevner, and R. Collins (2004) "Reviewing Software Diagrams: A Cognitive Study", *IEEE Transactions on Software Engineering* (30)2, pp. 82-96.

Hutchins, E. L., J. D. Hollan, and D. A. Norman (1986). "Direct manipulation interfaces". in D. Norman and S. Draper (eds.) *User-centered System Design.* Hillsdale, NJ: Lawrence Erlbaum, pp. 87-124.

Juhn, S. and J. D. Naumann (1985) "The Effectiveness of Data Representation Characteristics on User Validation", *Proceedings Of International Conference on Information Systems*, Indianapolis, IN.

Kim, Y.-G. and S. E. March (1995) "Comparing Data Modeling Formalisms", *Communications of the ACM* (38)6, pp. 103-113.

Kramer, B. and Luqi (1991) "Towards Former Models of Software Engineering Processes", *Journal of Systems and Software* (15), pp. 63-74.

Kroll, P. and P. Kruchten (2003) *The Rational Unified Process Made Easy*: Reading, MA, Addison-Wesley, Inc.

Linger, L. et al. (2002) Flow Service Quality (FSQ) Engineering: Foundations for Network Systems Analysis and Development: Pittsburgh, PA:Carnegie Mellon University Software Engineering Institute, Technical Note CMU/SEI-2002-TN-019.

Mantha, R. W. (1987) "Data Flow and Data Structure modeling for database requirements determination: a comparative study", *MIS Quarterly* (11)4, pp. 531-545.

Markus, M. L., A. Majchrzak, and L. Gasser (2002) "A Design Theory for Systems that Support Emergent Knowledge Processes", *MIS Quarterly* (26)3, pp. 179-212.

McConnell, S. (1996) *Rapid Development: Taming Wild Software Schedules*: Redmond, WA:Microsoft Press.

Moynihan, A. (1996) "An Attempt to Compare OO and Functional Decomposition in Communicating Information System Functionality to Users", *Proceedings Of Workshop on Evaluation of Modeling Methods in Systems Analysis and Design: CAiSE*96.*

Newell, A. (1987) *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Parsons, J. (2003) "Effects of Local Versus Global Schema Diagrams on Verification and Communication of Conceptual Data Modeling", *Journal of Management Information Systems* (19)3, pp. 155-183.

Parsons, J. and C. Saunders (2004) "Cognitive Heuristics in Software Engineering: Applying and Extending Anchoring and Adjustment to Artifact Reuse", *IEEE Transactions on Software Engineering* (30)12, pp. 873-888.

Peleg, M. and D. Dori (2000) "The Model Multiplicity Problem: Experimenting with Real Time Specification Methods", *IEEE Transactions on Software Engineering* (26)6, pp. 1-18.

Porter, A. H., A. Mockus, and L. Votta (1998) "Understanding the Sources of Variation in Software Inspections", *ACM Transactions on Software Engineering and Methodology* (7)1, pp. 41-79.

Prowell, S. et al. (1999) *Cleanroom Software Engineering: Technology and Process.* Reading, MA, Addison-Wesley, Inc.

Reason, J. (1990) *Human Error*: Cambridge University Press.

Robey, D. (2001) "Blowing the whistle on troubled software projects", *Communications of the ACM* (44)4, pp. 87-93.

Rosenberg, D. and K. Scott (1999) *Use Case Driven Object Modeling with UML : A Practical Approach*: Addison-Wesley Professional.

Rouge, C. L. et al. (2004) "Telemedicine Encounter Quality: Comparing Patient and Provider Perspectives of a Socio-Technical System", *Proceedings Of 37th Annual Hawaii International Conference on System Sciences (HICSS)*, Hawaii.

Royce, W. (1998) *Software Project Management: A Unified Framework*. Reading, MA, Addison-Wesley, Inc.

Shen, Z. and K. Siau (2003) "An Empirical Evaluation of UML Notational Elements Using Concept Mapping Approach", *Proceedings Of International Conference on Information Systems (ICIS)*, Seattle.

Shoval, P. and M. Even-Chaime (1987) "Database Schema Design: An Experimental Comparison Between Normalization and Information Analysis", *Database* (18)3, pp. 30-39.

Shoval, P. and I. Frummerman (1994) "OO and EER Schemas: A Comparison of User Comprehension", *Journal of Database Management* (5)4, pp. 28-38.

Siau, K. (2004) "Informational and Computational Equivalence in Comparing Information Modeling Methods", *Journal of Database Management* (15)1, pp. 73-86.

Siau, K. and Q. Cao (2001) "Unified Modeling Language (UML)-A Complexity Analysis", *Journal of Database Management* (12)1, pp. 26-34.

Siau, K. and L. Lee (2004) "Are Use Case and Class Diagrams Complementary in Requirements Analysis: An Experimental Study on Use Case and Class Diagrams in UML", *Requirements Engineering* (9)4, pp. 229-237.

Stone, E. (1978) *Research Methods in Organization Behavior*. Glenview, Ill.: Scott, Foresman and Co.

Vessey, I., V. Ramesh, and R. Glass (2002) "Research in Information Systems: An Empirical Study of Diversity in the Discipline and its Journals", *Journal of Management Information Systems* (19)2, pp. 129-174.

Walls, J. G., G. Widmeyer, and O. E. Sawy (1992) "Building an Information System Design Theory for Vigilant EIS", *Information Systems Research* (3)1, pp. 36-59.

Wand, Y. and R. Weber (1993) "On the Ontological Expressiveness of Information Systems Analysis and Design Grammars", *Journal of Information Systems* (3), pp. 217-237.

Weber, R. (2003) "Conceptual Modeling and Ontology: Possibilities and Pitfalls", *Journal of Database Management* (14)3, pp. 1-20.

ABOUT THE AUTHORS

Akhilesh Bajaj is Associate Professor of MIS, at the University of Tulsa. He received a B. Tech. in Chemical Engineering from the Indian Institute of Technology, Bombay in 1989, an MBA from Cornell University in 1991, and a Ph.D. in MIS (minor in Computer Science) from the University of Arizona in 1997. Dr. Bajaj's research deals with the construction and testing of tools and methodologies that facilitate the construction of large organizational systems, as well as studying the decision models of the actual consumers of these information systems. His articles appear in *Management Science, IEEE Transactions on Knowledge and Data Engineering, Information Systems and the Journal of the Association of Information Systems.* He is on the editorial board of several journals. His research has been funded by the department of defense (DOD) via the Software Engineering Institute at Carnegie Mellon University. He teaches graduate and undergraduate courses on basic and advanced database systems, management of information systems, and enterprise wide systems.

Dinesh Batra is Associate Professor at Florida International University. He received his Ph.D. from Indiana University. His research appears in *Management Science, Communications of the ACM, International Journal of Human Computer Systems, Journal of MIS, European Journal of Information Systems, Computers and OR, Information and Management, Journal of Database Management,* and others. His research interests are in usability and cognitive issues in database design and use, data modeling, and knowledge-based systems. He is currently serving as the President of the Special Interest Group on Systems Analysis and Design (SIGSAND). He serves in the editorial boards of the *Journal of Database Management, and Information Systems Management.* He is a co-author of the textbook *Object Oriented Systems Analysis* and *Design*.

Alan R. Hevner is an Eminent Scholar and Professor in the Information Systems and Decision Sciences Department in the College of Business Administration at the University of South Florida. He holds the Citigroup/Hidden River Chair of Distributed Technology. Dr. Hevner's areas of research interest include information systems development, software engineering, distributed database systems, healthcare information systems, and telemedicine. He is the author of over 120 research papers on these topics. He consults for a number of Fortune 500 companies. Dr. Hevner received a Ph.D. in Computer Science from Purdue University. He held faculty positions at the University of Maryland and the University of Minnesota. Dr. Hevner is a member of ACM, IEEE, AIS, and INFORMS.

Jeffrey Parsons is Professor of Information Systems and Associate Dean (Research) in the Faculty of Business Administration at Memorial University of Newfoundland. He received a Ph.D. from The University of British Columbia in 1992. His research interests include systems analysis and design, database management, and the Semantic Web. His research is published in such journals as *Management Science*, *Communications of the ACM*, *ACM Transactions on Database Systems*, *Journal of Management Information Systems*, and *IEEE Transactions on Software Engineering*. He is a member of the editorial boards of *Journal of the Association for Information Systems* and *Journal of Database Management*, and is co-guest editor of a forthcoming special theme issue of JAIS and CAIS focusing on research in systems analysis and design.

Keng Siau is a Professor of Management Information Systems (MIS) at the University of Nebraska-Lincoln (UNL). He is the Editor-in-Chief of the *Journal of Database Management* and the Book Series Editor for *Advanced Topics in Database Research*. Dr. Siau is the author of over 70 refereed journal articles that appear in journals such as *MISQ, CACM, IEEE Computer,*

Information Systems, IEEE Transactions on Systems, Man, and Cybernetics, IEEE Transactions on Professional Communication, IEEE Transactions on Information Technology in Biomedicine, and *IEEE Transactions on Education.* In addition, he has published over 80 refereed conference papers, edited/co-edited 10 scholarly and research-oriented books, and written more than 15 scholarly book chapters. He served as the Organizing and Program Co-chairs of the *International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design (EMMSAD)* (1996–2005).