



ISSN: 1545-679X

# Information Systems Education Journal

Volume 4, Number 62

<http://isedj.org/4/62/>

August 24, 2006

In this issue:

## Using a “Real” Systems Development Project to Enrich a Systems Analysis and Design Course

**Janet Helwig**

Dominican University  
River Forest, IL 60305 USA

**Abstract:** Without a real problem to solve, a Systems Analysis and Design course typically covers systems development from a theoretical perspective, and emphasizes how to develop graphical models (e.g., use case diagrams, data flow diagrams, ERD's, and UML diagrams) to document requirements and design. However, this course can provide a richer and more meaningful learning experience for students by providing them the opportunity to work as a team to build a real system for real users and thereby experience the “systems development process” – gathering and documenting requirements, designing, building, testing, and deploying. With semester time constraints, though, even a small project may prove unwieldy. Using an architected, rapid application development (ARAD) tool to speed development can help by delivering a well-engineered application architecture and generating much of the needed code. This paper explores the author's experience teaching this course with a systems development project serving as the main learning activity, look at why providing a real project is invaluable for students, yet challenging for all invested parties, and also discuss how using a commercial ARAD tool enabled this approach.

**Keywords:** systems development project, teaching methods, Systems Analysis & Design, ARAD tool, application architecture

---

**Recommended Citation:** Helwig (2006). Using a “Real” Systems Development Project to Enrich a Systems Analysis and Design Course. *Information Systems Education Journal*, 4 (62). <http://isedj.org/4/62/>. ISSN: 1545-679X. (Also appears in *The Proceedings of ISECON 2005*: §2314. ISSN: 1542-7382.)

This issue is on the Internet at <http://isedj.org/4/62/>

The **Information Systems Education Journal** (ISEDJ) is a peer-reviewed academic journal published by the Education Special Interest Group (EDSIG) of the Association of Information Technology Professionals (AITP, Chicago, Illinois). • ISSN: 1545-679X. • First issue: 8 Sep 2003. • Title: Information Systems Education Journal. Variants: IS Education Journal; ISEDJ. • Physical format: online. • Publishing frequency: irregular; as each article is approved, it is published immediately and constitutes a complete separate issue of the current volume. • Single issue price: free. • Subscription address: [subscribe@isedj.org](mailto:subscribe@isedj.org). • Subscription price: free. • Electronic access: <http://isedj.org/> • Contact person: Don Colton ([editor@isedj.org](mailto:editor@isedj.org))

### 2006 AITP Education Special Interest Group Board of Directors

Stuart A. Varden Pace University EDSIG President 2004		Paul M. Leidig Grand Valley State University EDSIG President 2005-2006		Don Colton Brigham Young Univ Hawaii Vice President 2005-2006	
Wendy Ceccucci Quinnipiac Univ Director 2006-07	Ronald I. Frank Pace University Secretary 2005-06	Kenneth A. Grant Ryerson University Director 2005-06	Albert L. Harris Appalachian St JISE Editor	Thomas N. Janicki Univ NC Wilmington Director 2006-07	
Jens O. Liegle Georgia State Univ Member Svcs 2006	Patricia Sendall Merrimack College Director 2006	Marcos Sivitanides Texas St San Marcos Chair ISECON 2006	Robert B. Sweeney U South Alabama Treasurer 2004-06	Gary Ury NW Missouri St Director 2006-07	

### Information Systems Education Journal 2005-2006 Editorial and Review Board

Don Colton Brigham Young Univ Hawaii Editor		Thomas N. Janicki Univ of North Carolina Wilmington Associate Editor		
Samuel Abraham Siena Heights U	Tonda Bone Tarleton State U	Alan T. Burns DePaul University	Lucia Dettori DePaul University	Kenneth A. Grant Ryerson Univ
Robert Grenier Saint Ambrose Univ	Owen P. Hall, Jr Pepperdine Univ	Jason B. Huett Univ W Georgia	James Lawler Pace University	Terri L. Lenox Westminster Coll
Jens O. Liegle Georgia State U	Denise R. McGinnis Mesa State College	Therese D. O'Neil Indiana Univ PA	Alan R. Peslak Penn State Univ	Jack P. Russell Northwestern St U
Jason H. Sharp Tarleton State U		Charles Woratschek Robert Morris Univ		

EDSIG activities include the publication of ISEDJ, the organization and execution of the annual ISECON conference held each fall, the publication of the Journal of Information Systems Education (JISE), and the designation and honoring of an IS Educator of the Year. • The Foundation for Information Technology Education has been the key sponsor of ISECON over the years. • The Association for Information Technology Professionals (AITP) provides the corporate umbrella under which EDSIG operates.

© Copyright 2006 EDSIG. In the spirit of academic freedom, permission is granted to make and distribute unlimited copies of this issue in its PDF or printed form, so long as the entire document is presented, and it is not modified in any substantial way.

# Using a “Real” Systems Development Project to Enrich a Systems Analysis and Design Course

Janet Helwig  
[jhelwig@dom.edu](mailto:jhelwig@dom.edu)  
Department of Math & Computer Science  
Dominican University  
River Forest, IL 60305, USA

## ABSTRACT

Without a real problem to solve, a Systems Analysis and Design course typically covers systems development from a theoretical perspective, and emphasizes how to develop graphical models (e.g., use case diagrams, data flow diagrams, ERD's, and UML diagrams) to document requirements and design. However, this course can provide a richer and more meaningful learning experience for students by providing them the opportunity to work as a team to build a real system for real users and thereby experience the “systems development process” – gathering and documenting requirements, designing, building, testing, and deploying. With semester time constraints, though, even a small project may prove unwieldy. Using an architected, rapid application development (ARAD) tool to speed development can help by delivering a well-engineered application architecture and generating much of the needed code. This paper explores the author's experience teaching this course with a systems development project serving as the main learning activity, look at why providing a real project is invaluable for students, yet challenging for all invested parties, and also discuss how using a commercial ARAD tool enabled this approach.

**Keywords:** systems development project, teaching methods, Systems Analysis & Design, ARAD tool, application architecture

## 1. INTRODUCTION

Historically, there have been a number of different approaches to teaching systems analysis and design. This is the natural result of professors setting varied learning objectives for the course. As documented by Raymond McLeod, Jr. (1996), course objectives have included (but are not limited to) teaching the system life cycle, providing a survey of systems concepts and methods, teaching tools of analysis and design, and teaching systems development methodologies. However, a majority of courses use an approach with some form of experiential learning, varying from simple exercises to case studies to real world system development. As author George Marakas (2006) notes, “when students are *doing* systems analysis they develop a richer understanding

of the concepts, activities, tools, and techniques that are used daily by the professional systems analyst.”

The real world system development approach can be a viable and preferred method for teaching this course. It gives students a real project and running case where they can apply the tools and techniques they are studying and witness first hand how the systems development process really works. Further, they have the added benefit of developing communication skills through the process of interviewing and listening to clients. Finally, their work as part of a development team can foster growth in interpersonal skills. For the teacher, the inherent limitations of a finite semester, and variations in student capabilities and available projects are challenges to be addressed.

Course learning objectives at our university currently include understanding the work, methods, and tools of systems analysts and software architects, appreciating the importance of employing a development methodology, understanding the system development process, and learning how, why, and when to create deliverable artifacts. To meet these learning objectives, the class most recently worked on a real life system development project to develop a registration and billing system for a local youth swim team.

## 2. WHY BUILD A REAL SYSTEM?

The benefits of this approach become apparent when students are confronted with the realities of developing a system for an actual client. When working with a real client, students experience what systems analysis is all about. They learn how challenging it is to determine scope and to establish business requirements. They experience first hand that the client cannot always communicate his needs, or in a more basic sense, the client initially does not fully understand exactly what he wants. This impresses on students how vital it is to have the client/user available on a regular basis, and how much of the job involves working closely with the client.

Students are also impressed by how iterative and time-consuming requirements analysis is. It can take a long time to reach the point where the development team has a solid understanding of what the user wants and what the system needs to be able to do. Again, it isn't possible to really do requirements analysis work without an actual client to interview. My class spent a few weeks working with our client to develop use cases describing how she wanted to interact with the system. During this process, students were struck by how frequently a user can change her mind in defining requirements. As one student noted, "we've seen project requirements evolve throughout the project, which makes it hard to complete the construction phase on time and to feel comfortable that we've ultimately constructed a system that reflects what our client wants." Students come to understand the reality of requirements creep and out of necessity, define methods to manage scope evolution. They also see the value of the proposal, signed off by the client, in helping to contain

user-requested modifications and additions to the original agreement.

While many textbooks, including the Whitten, Bentley, and Dittman textbook (2004) currently used in my class, cite how important interpersonal and communication skills are to the systems analyst, working on a real project cements it in the minds of students. They learn the necessity of being active listeners and effective negotiators in client meetings. Importantly, they have repeated opportunities to improve their skills in this area, identified in IS 2002 - Model Curriculum for Undergraduate Degree Programs in Information Systems as capabilities expected for IS program graduates. Students also come to recognize the importance of working effectively with other team members and the complexities of working as a team: team members have various levels of skill and aptitude, a range of personalities, and varying levels of commitment to the project, yet project success depends on contributions from all team members. This gives students the incentive to work synergistically as a team. As a final justification for providing this component to a Systems Analysis and Design class, Ferguson's (2005) survey of employers found that teamwork and communication skills, as well as real-world experiences, are highly and increasingly valued by employers of entry-level application developers. Specifically, of 22 skill areas employers were asked to rank, real-world experiences, oral communication, and teamwork skills were in the top 5, both when ranked by average importance of that background/skill and when ranked by average change in perceived importance of that background/skill over the last three years.

Working on an actual systems development project clarifies to students the value of having and using a methodology. Students quickly appreciate the structure a methodology provides. That is, at each moment in the systems development process, a methodology tells them what the next step is and how to go about doing that next step. Early in the semester, students have little understanding of why using a methodology is important. By the end of the semester, all students are glad to have the framework the methodology provides.

A real project allows students to apply their knowledge of how to build models and de-

velop specifications to a real situation. Rather than doing textbook exercises, students had the advantage of being able to question a real client when gaps in their understanding led to uncertainty about how to complete detailed specifications and design models. Student feedback noted that this was appreciated and added to learning. They experienced first hand the value that a prototype can provide in helping clients figure out what they want. Another benefit of working on a project end to end is that it clarifies for students when to develop specific models and artifacts, as well as justifying the value of those artifacts when they are later referenced. For example, we kept returning to our signed proposal when scope questions came into play. Similarly, use cases, developed early in requirements analysis, guided us in our initial prototyping, performed during systems design. Students realized that some of the use cases needed slight modifications as a consequence of feedback received from our prototyping efforts. So students also saw first hand the interdependence of the artifacts produced.

Finally, developing a product that answers a real need and helps a real person is inherently satisfying and motivating. Students who worked on the swim team system were truly engaged in the work, and motivated to deliver a quality solution for a client they had come to like and respect. Several students independently stated that they had invested more effort and time in this class than other classes because they were building a real product for a real client. Student learning in this class, evidenced via test results, was much deeper than achieved in my previous classes that did not use a real project approach. I am confident that students took away much more from the class than they would have without the project component.

### 3. CHOOSING THE RIGHT PROJECT

The single largest constraint to any project is time. In the classroom, this will usually mean the length of a semester. Projects that do not have a reasonable chance for completion in this period of time should not be used. Still, this does not mean that an overly ambitious project should be discarded. For example, a project may be shortened to only go through the analysis and design phases and not to the completion

of a fully operational system. While abbreviating the project in this way may not lead to the same satisfaction as delivering a final product, the client may be perfectly content with the results to this point. The key is that realistic expectations are established at the outset. Consequently, it is critical that the instructor evaluate the project before the course begins, and work out with the client in advance what can be realistically accomplished in the allotted time. In this way, the instructor can establish reasonable client expectations.

In the previously mentioned development of the registration and billing system for a local swim team, students carried the product through to the delivery of a working system, albeit in a Beta form. The report back from the client was favorable, even though the system was not deployed. The planning process allowed her to examine her business more critically, and implement beneficial changes to her own business processes. Had a fully operational system been initially promised, it is doubtful that the client would have had the same favorable impression.

Another important aspect of choosing a project is the anticipated involvement from the client. Students are likely to need more time understanding business problems than an experienced professional, because students generally do not have a mature understanding of typical business processes. Potential clients need to be evaluated based on their willingness and ability to patiently meet with students on a regular basis. The swim team client was used to patiently working with young people since she was the swim team's head coach. Additionally, since the team met in the evenings, she was in a position to spend time during the workday with the students.

Many clients will have extremely busy schedules that will not allow for the regular meeting times needed by students. The client who is too busy one week, and offers to defer meetings for a week or two will find that the semester quickly slips by without adequate progress.

A potential source for projects is the institution itself. For example the following semester's Systems Analysis and Design class worked on developing a system to be used in our university's School of Education, which will allow for the tracking of student

progress toward certain state mandated requirements. The client for this project, a fellow faculty member, is again familiar with working with students and very aware of the constraints of the university's semester system.

A wildcard to the process is the composition of the class itself. Since the project needs to be arranged before the term begins, it is very difficult for the teacher to know the strengths and weaknesses of the incoming class. While little can be done to shape the composition of the class, the teacher needs to quickly assess the class's capabilities, and if necessary revise the objectives of the project.

#### **4. SPEEDING DEVELOPMENT WITH ARAD TOOLS**

Since the analysis and design phases of the project are the focus for any Systems Analysis and Design course, they can and should consume a large chunk of the course term. In order to create a quality working system in a short timeframe, we benefited from the use of an ARAD tool. As defined by Blechar and Hotele of Gartner Group (2004), ARAD tools include "pre-built J2EE and .NET frameworks, architectural and design patterns, and pre-built technical components that can be customized by technical architects and then used to generate between 60 percent and 85 percent of the code of most applications prior to a professional programmer adding detailed business logic ... Because they can generate most of the code pertaining to the technical architecture and ensure compliancy to the predefined frameworks and patterns, the resultant applications will generally be quicker to deploy, better performing and of higher quality than hand-coded SODA applications [service-oriented development of applications]." While several commercial ARAD tools are available (see references for web-based information on Computer Associates' AllFusion Plex, Compuware's OptimalJ, and IBM's Rational Rapid Developer), we used Accelerator, an ARAD product developed by Modularis, Inc. The Accelerator tool includes an application generator for the Microsoft platform (Visual Studio.NET and SQL Server) called Architect. Architect generates a layered application based on an "entity object model". This model can be directly created

from the information contained on a UML class structure diagram.

We used this ARAD tool because of its ability to speed the development process, to generate a quality starting-point system, and to provide a platform to discuss system architecture and design. In our case the application architecture inherent in the Accelerator-generated system provided an opportunity to discuss the meaning and importance of application architecture in context. Having actual system code to study made it much easier to teach application architecture concepts. For example, students saw that by simply adding references to generated classes, they received access to all the function of those classes. Further, as with the Accelerator product, commercially viable ARAD tools are likely to utilize current state of the art software engineering, giving students exposure to expert software design. In our case, students had exposure not only to expert design but also to expert designers. We were fortunate to have access to Modularis software architects, who met with students on multiple occasions. Besides training students how to use the product from both a design and programming perspective, they also provided guest lectures focused on service-oriented architecture (SOA), emphasizing the need for well-engineered architecture in facilitating reuse and extensibility, and in use of their own systems development methodology, tailored for use with the Accelerator tool.

Accelerator gives developers the choice of generating code in either C# or VB.NET. The generated source code is open and extensible. What's more, the generated code is well commented and elegantly written - a good model for students to see and emulate. For example, "Try ... Catch... Finally" blocks are generated for error catching. Standard comment headings are created for each function. Comments within functions are plentiful as well. Code to support business rules can be easily added to the business logic layer. Further, the generated user interface supports testing and data creation even if a separate user interface is ultimately created. In our swim team project, we created a Windows application front end for our system. A teaching moment occurred when students needed the newly created interface layer to make service requests to another application tier. Students

saw how little code was needed to establish the communication between layers.

Accelerator also delivers role-based security, audit trail, and error and exception management features. Again, seeing how these elements are incorporated into systems generated by the product provides an opportunity to talk about why these features are essential in modern information systems.

In summary, use of the Accelerator product enabled students to create a working, maintainable and extensible system quickly based on their system analysis and design work. Without it, we would not have been able to create an operational system within our course term. As the Accelerator literature puts it, "With Accelerator, developers are free to focus more on design, business logic, and user interface implementation--the elements of an application most important to your business." (Modularis, 2005) In addition, the product facilitated discussion of other topics appropriate to systems analysis and design, such as application architecture and non-functional system requirements, including the need to provide role-based access to system functions and data, and audit trail logging. Finally, students learned that using such a tool to support systems development can make custom development a viable alternative to buying a package.

### 5. UNIQUE COURSE ISSUES

The real world project approach to teaching a systems analysis and design course yields a number of unique problems for the teacher. First, creating a pipeline of new and viable projects can be a challenging task. Clearly the teacher needs to be identifying suitable projects and users well in advance of the semester, so as to allow the class as much time as possible to complete the project. In our case we began working with the client during the second week of class. This was only possible because the project had been defined in meetings with the client well in advance of the semester.

Second, to keep the pipeline filled with projects, the teacher needs to develop a reputation for usable deliverables. Future prospects are more willing to devote the time needed for their project if they believe that there is a good chance that a real solution will be delivered as promised.

Third, the teacher needs to be in regular contact with the teams to make sure that they are staying focused and keeping the agreed timeline. Further, conflicts within the team need to be addressed and mediated as they arise, so that students stay positive about the team project. As an aside, it should be noted that this course, with the project component, has been taught to two separate classes meeting once per week and three times per week. Predictably, classes that meet more frequently were easier to monitor.

I anticipated grading to be an issue. However, this was much easier than I expected. In addition to my close monitoring of the team's progress and my presence in all meetings with the client, I received feedback about individual team members from the client. Also, as a final assignment, team members (four to a team) evaluated other team members using an evaluation template provided to them. Their peer evaluations were not shared directly with each other, but became part of a final performance evaluation that I prepared for each student. Through this process, individual student grades became easy to determine.

### 6. CONCLUSION

It has long been recognized that systems analysis and design is best learned by doing. Teaching a class that provides an experiential project approach enriches the learning experience by enabling students to actually *do* systems analysis and design work. The semester system creates constraints, but they can be overcome by proper planning, setting realistic expectations, and using good development tools. While this approach requires an extra degree of planning by the teacher, it is worth it to see students emerge with a richer understanding of systems analysis and design.

### 7. REFERENCES

- Blechar, M. and M. Hotle (2004) "ARAD Methods and Tools Improve Productivity and ROI." Gartner Group, <<http://www.compuware.fr/pdf/ROIstudy-ARAD-Gartner-Blechar.pdf>>. [Accessed 8/24/2005].
- Computer Associates (2005) "AllFusion Plex: Pattern-Based Application Development

- in Half the Time.”  
<<http://www3.ca.com/Solutions/Product.asp?ID=258>>. [Accessed 8/24/2005].
- Compuware (2005) “Compuware OptimalJ: Delivering reliable J2EE applications faster.”  
<<http://www.compuware.com/products/optimalj/>>. [Accessed 8/24/2005].
- Ferguson, E. (2005) “Changing Qualifications for Entry-Level Application Developers.” *The Journal of Computing Sciences in Colleges*, 20 (4), pp. 106-111.
- IS 2002 (2002) “IS 2002 - Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems.” Association for Computing Machinery (ACM), Association for Information Systems (AIS), and Association of Information Technology Professionals (AITP).  
<<http://www.acm.org/education/is2002.pdf>>. [Accessed 8/24/2005].
- Marakas, George M. (2006) *Systems Analysis & Design: An Active Approach*. McGraw-Hill Irwin.
- McLeod, Jr., Raymond (1996) “Comparing Undergraduate Courses in Systems Analysis and Design.” *Communications of the ACM*, May 1996. Vol 39, No 5, pp. 113-121.
- Modularis, Inc. (2005) “Product Overview – Modularis Accelerator an Architected-RAD Environment.”  
<<http://www.modularis.com/products>>. [Accessed 8/24/2005].
- Noonan, Joseph (2004) “IBM Rational Rapid Developer: More control for architects over code, design, and deployment.”  
<<http://www-128.ibm.com/developerworks/rational/library/2100.html>>. [Accessed 8/24/2005].
- Whitten, J., L. Bentley, and K. Dittman (2004) *Systems Analysis and Design Methods*, 6th edition. McGraw-Hill Irwin, Boston.