A COLLABORATIVE EXPERIENTIAL FRAMEWORK APPROACH TO THE SYSTEMS ANALYSIS AND DESIGN AND THE PROJECTS IN INFORMATION TECHNOLOGY COURSE SEQUENCE

ABSTRACT

Typically, the Systems Analysis and Design course requirements for majors in information systems have included either a single course that combines systems analysis (logical design) and systems design (physical design) concepts, or a two-course sequence delivered over two semesters. A problem with offering the systems analysis and systems design course content has been the limitation of "instructor-friendly" teaching tools and with textbook support, coupled with the semester time constraints for completing a project beyond the logical design and partial implementation stages. Students generally do not experience the actual construction and implementation of a system from the design specifications. This paper proposes a two-semester, two-course sequence that incorporates the analysis, design, and implementation of a system using a collaborative experiential learning framework following the Unified Process development model.

INTRODUCTION

The Department of Information Systems at Virginia Commonwealth University requires students majoring in Information Systems to take twenty-seven hours of coursework in the discipline. Core courses in the major cover information technology and systems topics in hardware and software design, programming languages, systems analysis and systems design, database systems, and data communications. Students concentrate in tracks that allow them to gain in-depth knowledge in the areas of applications development (e.g., additional programming courses), network administration (e.g., additional communications and network design courses), and information engineering (e.g., additional courses in organizational systems and reengineering). This set of core and elective courses are fundamentally the same for most information systems programs in schools of business that approach the curriculum from a more technical perspective. The advantage of this approach is that students are technically prepared to enter the information technology profession with core proficiencies of programming, systems development, database, and communications, and a specialty area that

enhances one aspect of their core competencies. Students are not only prepared to 'talk-the-talk', but also 'walk-the-walk'.

THE ROLE OF INFO 361 AND INFO 465: SYSTEMS ANALYSIS AND DESIGN AND PROJECTS IN INFORMATION SYSTEMS

At our School, the courses that introduce students to doing systems analysis, systems design, and developing a new application from a set of systems design specifications, are *INFO 361: Systems Analysis & Design*, and *INFO 465: Projects in Information Systems* courses. In INFO 361, students are taught various systems life cycle approaches to systems analysis and design. Much of the emphasis is on learning a modern CASE tool such as Rational Rose to employ Universal Modeling Language (UML) models that graphically represent the logical design of a systems solution for a guided project case scenario. Starting with an initial scenario, these tools permit students to create use-case diagrams, sequence and collaboration diagrams, class diagrams, state charts and activity diagrams as they graphically depict the system for the user. The course is one semester in length and requires students to submit approximately ten assignments, mostly requiring specific UML models of the system under investigation. Each assignment is graded, and a "possible" solution is posted on Blackboard for students to compare their work. Students use this solution to correct any problems in their initial models. A final assignment, weighted as two regular assignments, is used to have students compile the final versions of their previously submitted assignments in a coherent report form. This assignment is graded more rigorously as students have had the opportunity to discuss better design solutions with the instructor. An additional benefit for having the students submit the most correct set of models is to prepare them for the use of these specifications in the INFO 465 projects course. Familiarity with the project's set of specifications allows continuity and a minimum of lost start-up time from the carry-over as they begin the projects course.

A COLLABORATIVE FRAMEWORK: RATIONALE AND BENEFITS

Learning how to build information systems requires students to perform those activities that lead to a functioning application. Too often students are taught only the conceptual features and functionality that the final working application will display. The missing element in this formula for retained

learning is the understanding of the mechanics of building the logical design features/functions into a working prototype. Anecdotal observations from years of teaching systems analysis, design, and implementation indicate graduates lack the level of confidence as new hires in their ability to successfully participate and make a real contribution in a real system design project. Our students tend to miss out on the "nuances" that are difficult to experience in a classroom environment. Their creativity and innovation in devising applications are not at a sufficient level for them to take pride in the work they have performed. The classroom projects are not "fun"; learning is piecemeal and disjointed, as only parts of the total picture are seen, and ownership of the project is lacking. In other words, there needs to be involvement by the students in activities at a level that complements the typical classroom instruction.

Our collaborative and experiential framework is an approach for offsetting the deficiencies reflected in the traditional structure of the analysis, design, and implementation of information systems coursework. The approach incorporates the synthesis of conceptual analysis and design with the pragmatics of using CASE tools to facilitate a full project development. This framework is predicated on using a project 'engine' that will provide the core data and processes that the student developed applications can "hook into." The hooks will allow both students and instructor to focus on making the requirements operational on a more knowledgeable basis. The framework will support both an incremental and iterative process, which is more real world oriented. Students better visualize the mistakes they make, and solutions will become more robust and accurate, building students' confidence in the process. The strength of the framework relies on the Unified Process methodology, the core engine, and a collaborative experiential learning approach. The coordination of these elements can be facilitated by a multitude of diverse project scenarios.

Using the Unified Process [1] as the principal development paradigm, the INFO 361 course focuses on the requirements, analysis, and design workflows; whereas, INFO 465 focuses on the design and implementation workflows, as shown in Figure 1. The concentration of the topics and work effort in INFO 361 and INFO 465 is graphically depicted with an overlay of the Unified Process model showing the Inception, Elaboration, Construction, and Transition phases. This represents only a guideline for instructors as they prepare for the courses.

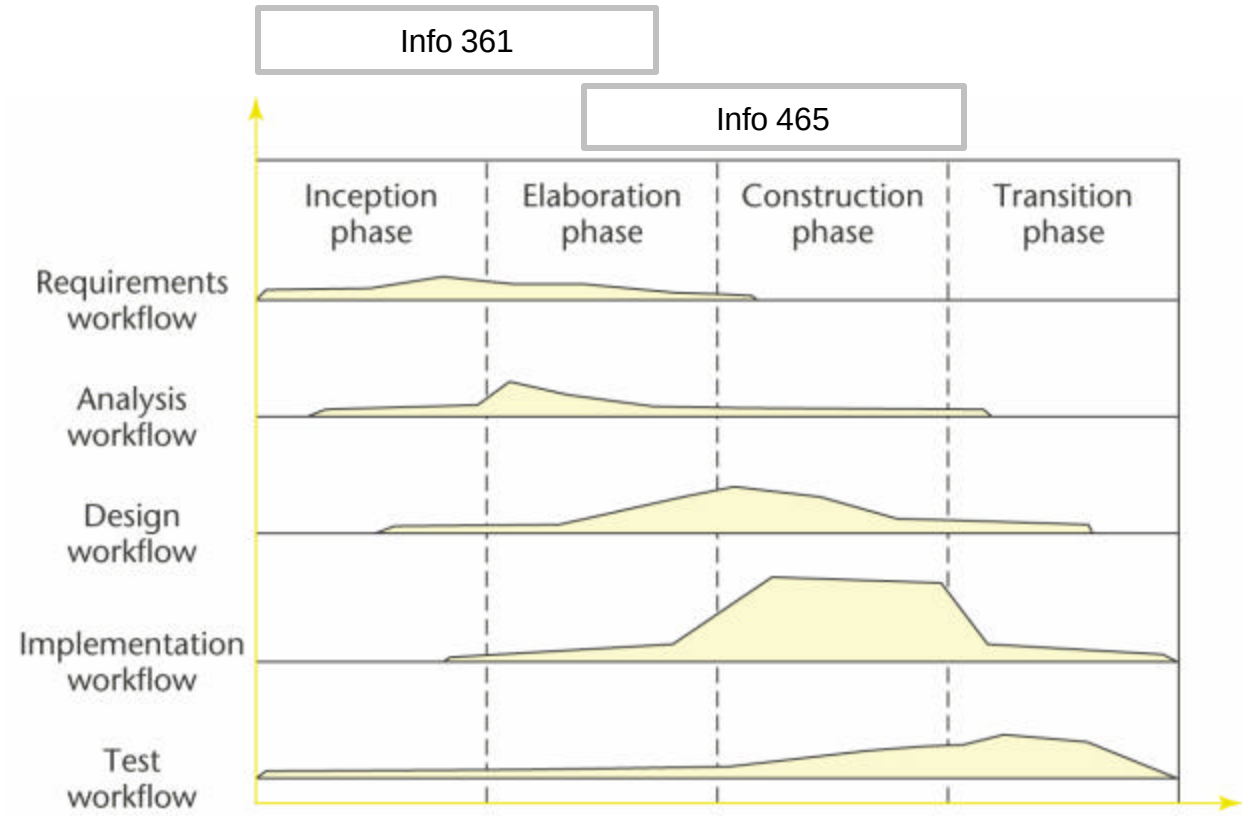

Figure 1: The Unified Process Phases & Workflows and INFO 361 and 465 Course Sequence

IMPLEMENTING THE PROPOSED FRAMEWORK

The framework is introduced to students in INFO 361 through 'engineered' or disguised real-world businesses that require an information system. The case presents the need for the development of an information system to support a business opportunity or to enhance functionality of an existing system. An e-business enterprise environment is used to provide the basis for the information architecture and infrastructure requirements. This approach is designed to complement the selected Systems Analysis and Design textbook. Most existing courses in systems analysis and design do well in developing the conceptual model to identify system requirements, but fall short in the Elaboration and Construction Phases that include the physical design, implementation and testing of a completed project. The goal of the new approach is for the students to complete a system of sufficient complexity and detail to challenge and enhance their system development knowledge and skills. Thus when students interview

with employers, they can document their level of skill by including their senior project in their portfolio.

The enterprises chosen for the framework's case studies are instructor-selected and range from simple retail merchants to more complex and specialized businesses such as manufacturers, public agencies, and professional offices (dentists, engineers, and contractors). Together, their needs describe a range of small-to-large EDI/e-business environments.

Course materials provide sample input screens, output screens and reports, and other infrastructure requirements as related to specific technologies involved (e.g., Web applications). The framework provides flexibility for the instructor to include new information technologies such as wireless communications and networks. For many students this set of courses will be their first practical exposure to 'business processes' that include concepts such as issuing purchase orders, recording and tracking sales orders, invoicing, inventory control, credit checks, transaction posting, and decision analysis models. Instructors are encouraged to identify short, summary-styled handouts or references to reacquaint students to these concepts as they begin to incorporate them as part of the enterprise model.

Prototype applications that satisfy the stated requirements for the enterprises described in the case studies are demonstrated and presented in class as both system and database design issues are considered in the case study. Prototypes are provided to the instructor in VisualBasic.NET or similar format, along with a 'solution set' that includes detailed design models to be used in the project. Students are drawn into the framework methodology, as a major component of their performance in the course relies on their ability to master the case tools as they learn UML and the Unified Process methodology. This approach engages the students as they complement the learning of the systems development process with the intrigue of becoming proficient in the use of contemporary modeling and prototyping tools.

Students' projects in INFO 361 exercise learned UML modeling skills to represent the system requirements derived from the case scenario, and 'discoveries' through discussions with the instructor. These models and the supporting documentation are the basis for the System Proposal at the end of the

Elaboration phase. The proposal includes requirements definitions, economic, organizational and technical feasibility analyses, and Use Case, Sequence, Class, State Chart, and Activity diagrams depicting the results of the analysis and design iterations of the Inception and Elaboration Phases. In addition, it includes a project management plan with a work break-down structure and estimated durations, a Gantt Chart, a network (PERT) diagram, and a resource sheet. A project budget is generated from resource allocation decisions and a baseline is established for the project. Figure 2 illustrates the implementation of the approach.

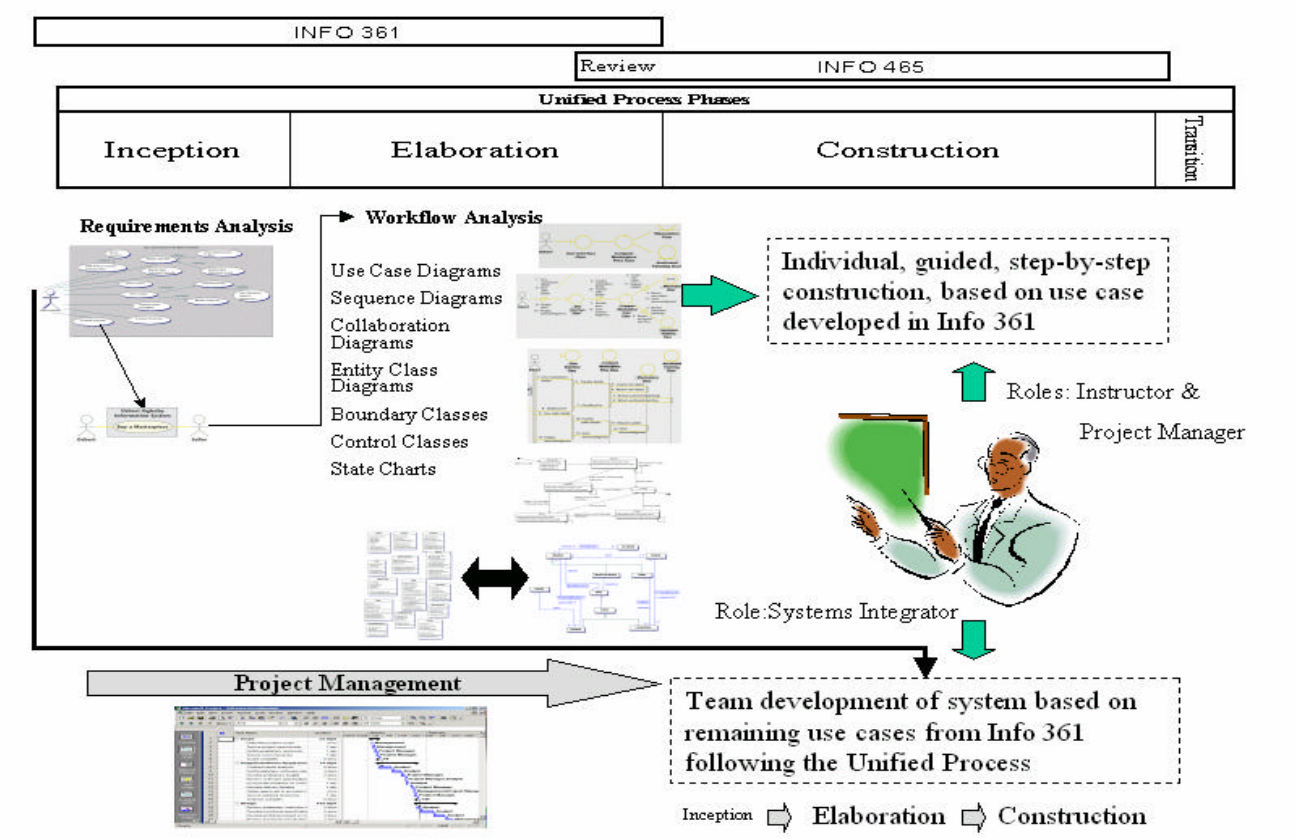

Figure 2. Framework Implementation

Students begin by developing a user request, which stipulates the initial system requirements in terms of desired functionality. Following a discussion and analysis of the user request, and the initial set of functional user requirements, a final set of requirements is agreed upon. A Use Case diagram is created that depicts the necessary use cases that represent the final set of requirements. The diagram shows sufficient *include* and *extend* stereotypes that demonstrate to students the first concept of

reusability, and *normal* and *exception* flow-of-events. The instructor selects one of the user requirements as the basis for all the semester assignments. The associated use cases for realizing that requirement is then used to develop the subsequent dynamic and static UML models in the analysis and design workflows. This guided approach focuses on only one major use case.

To manage this part of the course, a series of team assignments are posted with specific deliverables and weighting factors for performance measurement. These guidelines provide a technique for structuring the project around a team approach. Upon compiling final versions of the assignment deliverables, each student uses these team assignments to finalize the individual System Proposal. In-class individual and team exercises are used to develop student skills in using UML to create accurate models. Midterm and final exams are given to test individual student content knowledge. What is learned through the project will be reinforced when their designs are used as a starting point to construct an application in INFO 465 the following semester.

At the beginning of the following semester, the framework is presented to students in the projects course (INFO 465) in the form of rough design models, prototype screens, forms, and output reports, a database scheme, and software components. These "pieces of the pie", coupled with the requirements documented in the System Proposal completed in INFO 361 are used as a starting point to build an application to meet the requirements specifications detailed in the System Proposal based on the enterprise infrastructure in the case assigned the previous semester. The students will be working with an enterprise engine that will allow "hooks" for the students to operationalize their design specifications.

The courseware and project approach allows students to:

- Learn a new software environment and adapt their prototype to handle basic requirements for their enterprise;
- Add functions for basic order entry, accounting, and reporting;
- Create screen interfaces, which add functions to support EDI trading functions for catalog, sales, invoicing, and purchasing among the class' EDI trading community.

At the outset of INFO465 there is a review of UML documents and business processes introduced earlier in INFO361. The requirements for the Case Study are detailed, and students continue the elaboration process for the use cases they have chosen or been assigned. Deliverables from this activity are UML Use Case diagrams supported with details for each Use Case.

Figure 3 shows a student's Use Case diagram modified to show EDI interfaces with other 'engines' operated by the instructor and other students in the class. This is an example of an individual student's project, to support a case study for a small, mercantile enterprise operating in the class' EDI trading community.

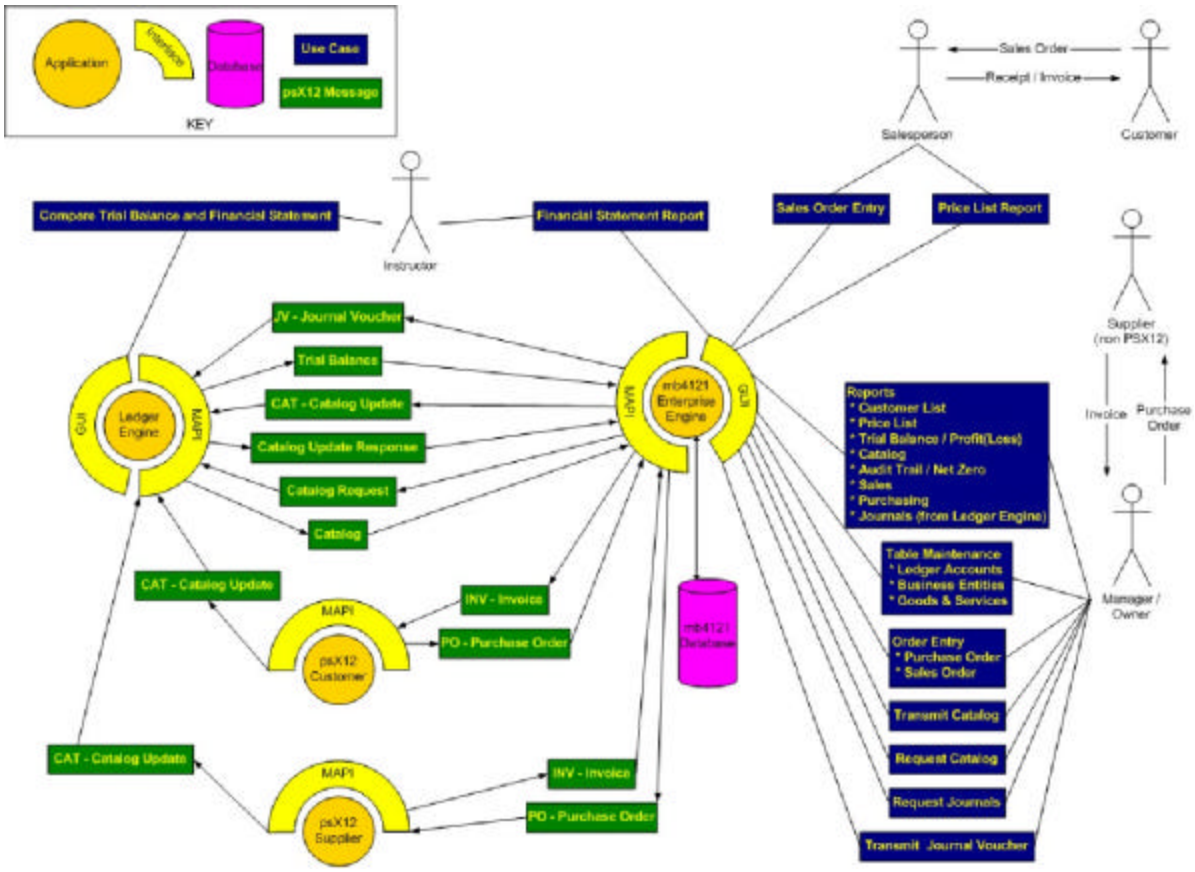

Figure 3: INFO 465 Use Case Diagram Example

To help with the details of the elaboration and construction phases, the database and objects presented earlier in the framework are reviewed in class and become the focus for business process modeling on the board. The database structure provided in the framework can be easily adapted to most case studies and provides a reference for classroom modeling of business processes. Figure 4 shows an example of a student's adaptation for a small enterprise selling books.

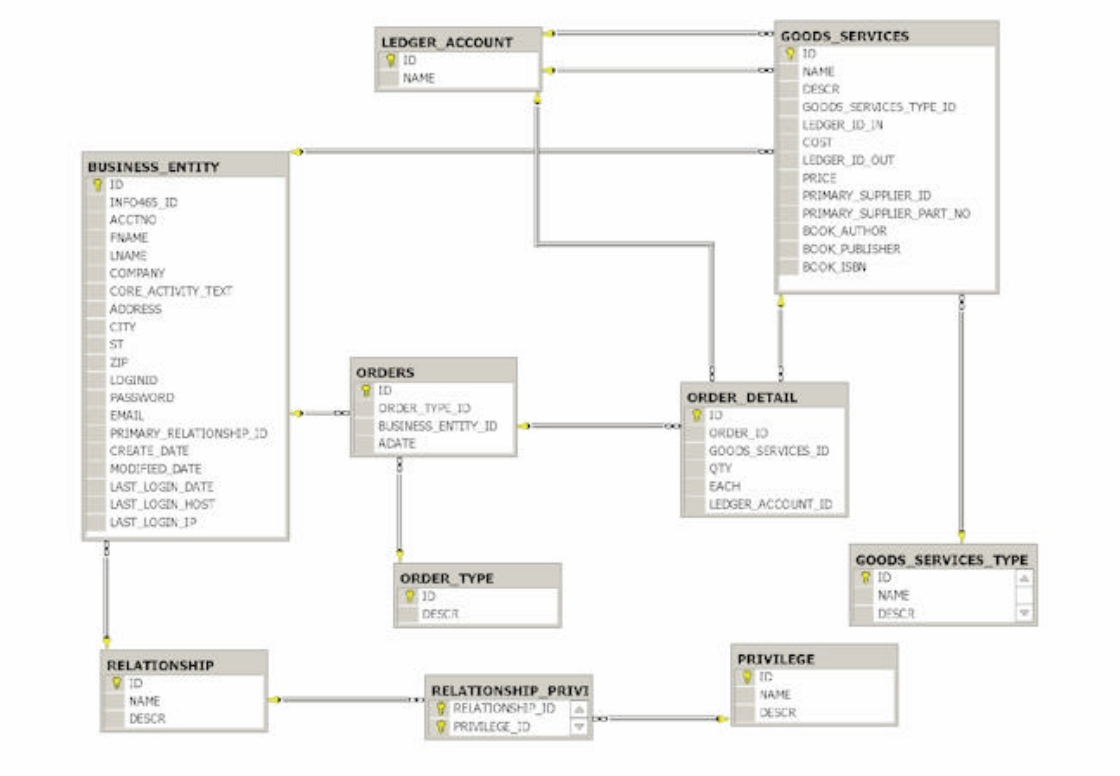

Figure 4: INFO 465 Class Diagram Example

Another early exercise asks for examples of the tables populated with details reflecting business activity, accompanied by a Trial Balance or other financial report prepared with reference to the details. Figure 5 shows a student's Details, from a database similar to the above, and a simple financial statement prepared from them.

id	order_id	goods_services_id	quantity	each	ledger_id	extended_amount
201	101	52	-1	20,000.00	3000	-20,000.00
202	101	51	1	20,000.00	1000	20,000.00
203	102	52	-1	10,000.00	3000	-10,000.00
204	102	51	1	10,000.00	1000	10,000.00
205	103	61	1	50.00	5061	50.00
206	103	51	-1	50.00	1000	50.00
207	104	59	1	2000.00	5050	2000.00
208	104	51	-1	2000.00	1000	2000.00
209	105	60	1	70.00	5060	70.00.00
210	105	51	-1	70.00	1000	70.00.00
211	106	58	1	2200.00	1070	2200.00
212	106	51	-1	2200.00	1000	2200.00
213	107	53	200	4.00	5000	800.00
214	107	54	500	2.00	5000	1,000.00
215	107	55	300	6.00	5000	1,800.00
216	107	56	500	4.00	5000	2,000.00
217	107	51	-1	5,600.00	1000	5,600.00
218	108	53	2	6.00	4000	12.00
219	108	51	1	12.00	1000	12.00
220	109	55	1	8.00	4000	8.00
221	109	56	2	6.00	4000	12.00
222	109	51	1	20.00	1000	20.00
223	110	54	5	3.00	4000	15.00
224	110	51	1	15.00	1000	15.00

Financial reports derived fromthese data:

Account			
1000	Cash	20,127.00	
1020	Furniture & Equipment	2,200.00	

	Assets	22,327.00	
3000	Stock		30,000.00
3999	Retained Earnings		(7, 673.00)

	Equity		22,327.00
			========
	Liabilities + Equity		22,327.00

Profit/Loss:

4000	Sale of Goods		$ 47.00
5000	Cost of Goods	5,600.00	
5050	Rent	2,000.00	
5060	Utilities	70.00	
5061	Telephone	50.00	

	Expenses	7,720.00	
	(Loss)	(7,673.00)	

Figure 5: Info 465 Table Details Example

As review is completed and the student developers' focus turns to construction, the deliverables focus on software. Figure 6 depicts an example of a student's user interface for a small enterprise showing the core functionality of a project built with using the framework, from order entry at the top to ledger accounts at the bottom.

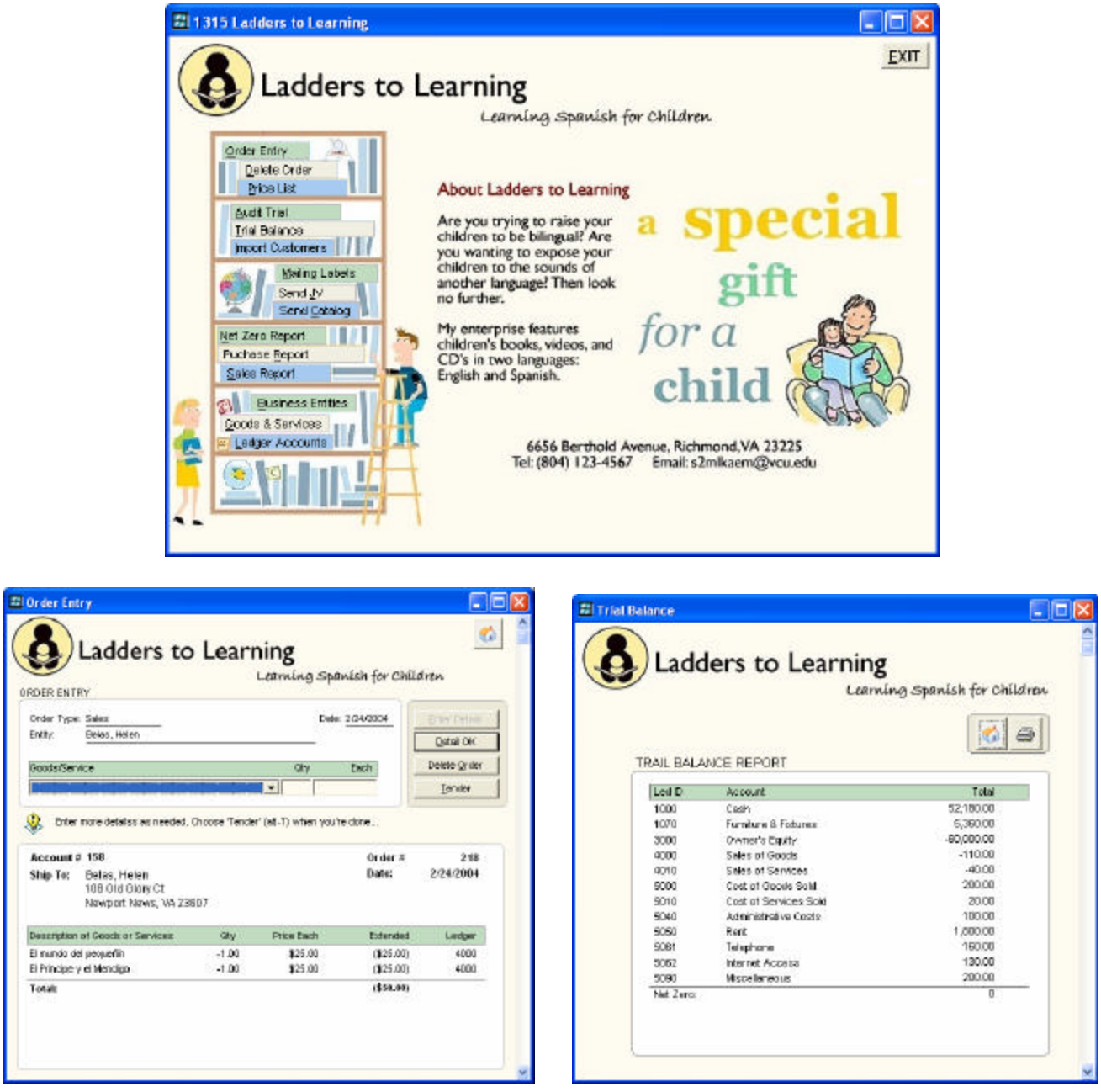

Figure 6: INFO 465 Interface Example

The exercises provide students with the opportunity to work with X12-like EDI documents for cataloging, purchasing, invoicing, fund transfers among the class' EDI trading community. And, the requirements include reporting activity to a central 'ledger engine' operated by the instructor. Figure 7 shows an example Journal Voucher transaction summarizing a day's activity for a student's enterprise.

```
From: "Ledger Engine" <ledgerengine@info465.net>
To: <bantamax@attbi.com>; <ledgerengine@info465.net>
Sent: Monday, April 14, 2003 3:53 AM
Subject: !*!*JV response from Ledger Engine
> Received from Enterprise #0000 Brand Now
> Dated 20030414 Time 035248
>
> JV*0000*20030414
> NET*3000*-2500000
> NET*1040*11000
> NET*5050*50000
> NET*4000*-394100
> NET*1000*2489700
> NET*5000*263400
> NET*5030*80000
> JVE*9
> Previous Journal for this date has been backed out of the Ledger
>
> Ledger Trial Balance as of this JV Transaction:
>
> 1000  Cash                        24897.00
> 1040  Equipment                     110.00
> 3000  Owner's Equity             (25000.00)
> 4000  Sales of Goods              (3941.00)
> 5000  Cost of Goods Sold           2634.00
> 5010  Cost of Services Sold           0.00
> 5030  Wages                        800.00
> 5050  Rent                         500.00
>                                 ------------
>                                        0.00
```

Figure 7: INFO 465 Journal Voucher Example

The class website posts trial balances for all the class' enterprises. Improbable trial balances are displayed in red. Students are motivated to have their enterprise show 'in the black'. An example is shown in Figure 8.

3333 PsychTek:	1000 Cash	8588.00	
	1040 Equipment	2000.00	
	3000 Owner's Equity	(10000.00)	
	4000 Sales of Goods	(2520.00)	
	4010 Sales of Services	(890.00)	
	5000 Cost of Goods Sold	200.00	
	5010 Cost of Services Sol	320.00	
	5030 Wages	252.00	
	5040 Administrative Costs	1000.00	
	5050 Rent	1000.00	
	5062 Internet Access	50.00	
	Net:	0.00	
5869 Action Auto Tint:	1000 Cash	(5340.00)	
	1040 Equipment	45.00	
	3000 Owner's Equity	5000.00	
	4000 Sales of Goods	125.00	
	4010 Sales of Services	75.00	
	5000 Cost of Goods Sold	55.00	
	5010 Cost of Services Sol	40.00	
	Net:		0.00

Figure 8: INFO 465 Example of Trial Balances

FRAMEWORK STRENGTHS

Using this approach provides the following strengths:

- Learn current systems development tools
- Learn the UML system design paradigm
- Experience the complete Systems Development Life Cycle
- Integrate real business process concepts
- Learn e-business modeling concepts
- Simulate real-world project team participation
- Simulate the iterative and incremental nature of the Unified Process
- Learn Project Management skills and tools
- Integrate programming, database management, data communications, and hardware/software skills

FRAMEWORK LIMITATIONS & CONSTRAINTS

Using this approach has a series of potential limitations:

- Hardware/software requirements
- Technical orientation
- New case development required each year
- Part-time students participation

REFERENCES

[1] Jacobson, I., Booch, G., & Rumbaugh, J. The Unified Software Development Process. Addison Wesley, 1999.