Introducing a Graduate Research Problem to a Junior Level Class:  
A Successful Experience

1Ghassan Alkadi, Theresa Beaubouef 2
Southeastern Louisiana University
1galkadi@selu.edu, 2theaubouef@selu.edu

ABSTRACT
Formulating a research project to be designed and implemented by a junior level class during a short time period is a formidable undertaking. In this paper the authors discuss the problem of rough querying of databases presented to the information systems course and the special issues involved with the design and development of such a system. Of particular note was the greater emphasis in this project on research, and the special challenges involved with exposing undergraduates to graduate-level research concepts and tasks. The students were asked by the instructor to research and learn Database Connectivity, SQL Syntax, Microsoft C# Syntax, Graphical-User Interface Controls, and Database Structure and Design and the .Net framework. The students had to elicit the software requirements from the client at different meetings. Moreover, the students worked on their communicational and oral skills because they had to present their final product to the client and provide ample documentation.

Keywords: Rough Querying, Research Project, Oral and Communicational Skills

1. INTRODUCTION

As part of an information systems course it is important that students develop the knowledge and skills to formally design and implement a nontrivial information system. In our course students are first exposed to concepts and practices, and begin by working on small projects to reinforce what they have learned to prepare them for their final projects. It is this final project that is most important and is the subject of this paper. The final project involves some real world application for a client, and although the instructor is knowledgeable about the basics of the project in order to ensure its appropriateness for the course, he has no part in the formulation of the problem or the specification of the requirements. For the work discussed here, the client happened to be another faculty member.

The instructor of the information systems course asked “the client” to present the students with a research problem that could be programmed within five weeks of the semester. The problem involved rough querying of databases, and the students were required to develop from scratch a rough querying information system. The students were asked to elicit the requirements from the client and produce a Software Requirements Specification (SRS) document and to make sure that the customer was in full agreement with it and signed off on it. There were several challenges in the process of developing such a system. To name a few: (1) researching the problem, (2) learning software tools, such as MS LINQ, and (3) learning MS C# and the .NET framework.

The client played the role of a non-programmer and asked for vague requirements at the beginning to entice the students to do more research. Not only were the students required to code construct the system at hand, but to produce other important documents and manuals for the user. The students were grouped into four teams and were asked to compete against each other in producing the best system. Each group was asked to create a Webpage that updates both of the instructors with the latest updates. Notwithstanding the effort required of the students, the majority were ecstatic about the opportunity to start immediately. A survey was conducted at the end of the semester which elicited students’ opinions of how the class fared was very favorable. The following paragraphs discuss the problem and the effects of it on the students’ knowledge, whether programmatically or research based. Later sections explain in greater detail the way the course was conducted, the results, and the benefits obtained from the course.

2. THE PROBLEM AT HAND

Information systems are everywhere in society, and businesses and organizations invest significant time, personnel, and financial resources into these systems. Because of this, it is imperative that the maximum use possible is obtained from databases and information systems. An information system, by its name, provides a higher level of product from data than a simple database; however, most information systems today continue to lack the ability to manage uncertain and imprecise data or to incorporate some type of uncertainty management into the querying and access of crisp data.

Several techniques have been proposed throughout recent years to try and overcome this deficiency, and various mathematical theories for the management of uncertainty proposed such as probability, possibility, interval logic, and fuzzy and rough set theories. Models have been developed that incorporate uncertainty management into the underlying data model, whether it is relational, object-oriented, or a combination
of the two (Beaubouef, Petry, & Buckles 1995; Buckles, and Petry 1982). Another approach has been to apply uncertainty management in the querying of crisp data, leaving the underlying model unchanged (Anvari, and Rose 1987; Bose, and Pivot 1991).

It is this approach that is taken in this project: an information system that incorporates the rough querying of crisp data for a relational database (Beaubouef, and Petry 1995). Rough set techniques are built on top of an information system, and user interfaces designed for incorporating these concepts. In this manner, our system is better able to model some enterprise in the real world since it incorporates uncertainty management techniques, yet it makes use of existing databases without changing the underlying structure of the data.

The basic concepts of rough sets are briefly described as follows:

Rough set theory is a mathematical formalism for representing uncertainty. An approximation region in rough sets partitions some universe into equivalence classes (Pawlak 1982). This partitioning can be adjusted to increase or decrease its granularity, to group items together that are considered indiscernible for a given purpose, or to “bin” ordered domains into range groups.

Any finite union of these elementary sets is called a definable set. A rough set $X \subseteq U$, however, is defined in terms of the definable sets by specifying its lower ($RX$) and upper ($\bar{R}X$) approximation regions:

$$RX = \{x \in U \mid [x]_R \subseteq X\}$$

and

$$\bar{R}X = \{x \in U \mid [x]_R \cap X \neq \emptyset\}.$$

$RX$ is the positive region, $U - \bar{R}X$ is the negative region, and $\bar{R}X - RX$ is the boundary or borderline region of the rough set $X$, allowing for the distinction between certain and possible inclusion in a rough set.

Rough set techniques may be incorporated into the querying through the use of user-defined equivalence classes and rough predicates. This method is flexible in that the user may change the specifications as necessary without changing any of the actual application data.

The first specifications to be made are the indiscernibility relations. An indiscernibility relation, which is required for every domain, partitions the domain into equivalence classes of attribute values which are indiscernible from one another. In crisp data, each value belongs to an equivalence class containing only one member, the value itself. If no indiscernibility relation is defined for some domain, the data within the domain is considered crisp, and values are automatically partitioned into singleton equivalence classes.

Rough set theory is well developed and formalized, and the literature on rough sets is quite vast. In order to fully understand this theory, one must have a good understanding of mathematics, logic, and proofs. The basics of the rough set theory are easy for an average college junior to comprehend, but anything beyond the basic concepts requires mathematical maturity, diligent study, and research comparable to that done in graduate study. This was one of the challenges of this project, since the students in the course are mostly juniors.

Another challenge is that students in this course have not been exposed to very much formal database terminology or database query languages, having not yet taken the senior level database course. Therefore, students in the course were required to learn much of the database material on their own.

Recognizing the challenges associated with this project, the instructor appointed one member of each group to be in charge of research. This person spent considerable time trying to understand the literature on rough sets and the mathematical constructs, and determined which areas were relevant to the current project. He then related his understanding to the other group members during the design and development project and resolved questions as they arose.

3. HOW THE COURSE WAS CONDUCTED

At the beginning of the semester, students were grouped into teams and assigned group leaders to handle the tasks assigned by the instructor. The instructor had also assigned a project manager to oversee the work of all the students when they were meeting in the majors’ lab and to answer any technical question that they may have had. The instructor almost always chooses a student that had previously taken the class and performed well in it to be the project manager. Each group was assigned a list of topics of research. Examples of the topics included Database Connectivity, SQL Syntax, Microsoft C# Syntax, Graphical-User Interface Controls, and Database Structure and Design. Obviously, the students had no idea about the language to be used to develop the software system until they saw the list of research topics. Each group had to orally present their research work to the rest of the class. In addition, they were required to post their work on the class discussion board to exchange information.

After the research phase was over, the second phase began with Phase I. New teams were created for Phase I and new team leaders assigned. The performance of each team leader and member was evaluated by the students in their respective groups and by the instructor, who would use these observations later on to determine who would be the leader in the final phase. Two weeks were given as the due date for code constructing a human resource software system that basically handled employees’ records that include many fields. The students were not only graded on satisfying the requirements imposed by the instructor, but also on the GUI interface and its user friendliness.

The instructor insinuated that the students were to compete, making the pace of this phase quite frantic. The majority of the students did not know anything about C#, nor did they know anything about coding together in a group environment, so team dynamics was a new phenomenon to all of them. Hence, the learning curve for them was a steep one. Amazingly, three out of the four groups did very well. One group did manage to satisfy the
The majority of the requirements but failed to have a good GUI.

The second phase was a more involved one. The requirements became more difficult since this phase involved additional functions, such as administrators in addition to regular users. The teams were shuffled again, with the intent of creating groups with totally new team members and group leaders that hadn’t yet served in this capacity. The instructor allowed the students to use any software developed earlier in Phase I. Phase II also had to be complete in two weeks. C# remained the language to use for developing the software. To the instructor’s astonishment, one group opted to start from scratch even though the instructor tried his best to sway them from doing so. The other groups used the code that was posted on the discussion board. The GUI this time around was very good across the board. There were a few requirements not met because of their complexity, namely scheduling of employees’ working hours for an upcoming week or month in a calendar. Additionally, applying over time rates for part time and full time employees was a complicated process considering that a 401 K is involved with tax rates.

After the presentation of Phase II, the students were very anxious to know who the client would be for the final phase and what type of requirements they would be facing. The final phase allowed five weeks for delivery of the final product. The instructor asked a colleague, the “client”, to come to the class and provide the sketchy requirements for the students. The groups were shuffled again and the selection of the group leaders depended on two important characteristics: leadership and communication skills. Programming skills are also, of course, highly regarded. The instructor advised all group leaders to assign one member of each group to dedicate significant time to research, a member that was proficient in math and logic. Moreover, the group leaders had to communicate with the client to set up meetings to thoroughly discuss the requirements and to resolve any questions regarding the problem at hand. The groups had to create WebPages to keep the instructor and the customer abreast of all the updates (Southeastern 08). The webpage also listed the requirements, along with a biography, contact information and personal picture of each group member. Such features tremendously helped other group members, other teams, and the instructor and his client. All of the groups chose C# and the .NET framework to construct the code (Microsoft Visual Studio 2008). They also chose to use MS LINQ (Microsoft LINQ 2008).

4. SUCCESSFUL RESULTS AND DELIVERABLES

The four groups presented their work on the due dates assigned for them. They showcased their PowerPoint presentations and the software systems they developed. The groups were evaluated by the client as well as the instructor for meeting the requirements, graphical user interface and user friendliness. Each group member was evaluated individually by the group leader and other group members. Code construction contribution was another main factor in determining final grades. The group leader was evaluated by his members and the instructor. All of the software systems were successful in delivering the majority of the specifications. Not a single group failed in delivering the software system required. (Southeastern 08) shows the links for each project and the team members that were involved in creating them.

Specific deliverables that were required by the client included:

1. Welcome Packet:
   1.1. Flyer that contains a photo of group and a description of individual and group strengths/qualifications
   1.2. Letter
2. Software Package (generic system)
   2.1. Must run on a PC
   2.2. Complete User and System Manuals
   2.3. Detailed documentation on how things were developed and why a specific language was chosen.
   2.4. Must have a good GUI
3. A research paper that may be submitted as an undergraduate paper
SURVEY AND EVALUATION FORMS

A sample of the evaluation form that was used in the class is shown below.

<table>
<thead>
<tr>
<th>Member Name</th>
<th>Leader</th>
<th>Member 1</th>
<th>Member 2</th>
<th>Member 3</th>
<th>Member 4</th>
<th>Member 5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Productivity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quantity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reliability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Team Membership</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cooperation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attitude</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Individuality</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creativity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive Feedback</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaptability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conscientiousness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following evaluation pertains to Team Leaders only:

- Communications:
  - Listening
  - Persuading
- Organizational Skills:
  - Planning
  - Goal Setting
- Team-Building Skills:
  - Empathy
  - Motivation
- Leadership Skills:
  - Setting Example
  - Energy
  - Vision
  - Delegating
  - Positive Outlook
- Coping Skills:
  - Flexibility
  - Creativity
  - Patience
  - Persistence
- Technological Skills:
  - Experience
  - Project Knowledge

The submission of these mandatory forms helped significantly in resolving any problems that may have arisen between students. Another survey was conducted to elicit feedback from the students to see how this class helped them with their programming and communication skills. The survey is shown below:

1. What team size do you like be part of?
   a. 3
   b. 4
   c. 5
   d. Other? _____

The following pertains to the role of team manager:

2. When you were a team leader, which of the following was the most important to you? The most difficult? The most rewarding? Why?
   a. Spend ample time to help the team set goals to finish on time
   b. Provide encouragement when a team member is discouraged
   c. Decision making
   d. Keep the team focused on the task at hand

3. How did you resolve conflicts between team members?
   a. Let the project manager resolve it
   b. Let the instructor resolve it
   c. Try your best to resolve the conflict without neither the project manager nor the instructor know
   d. Do nothing

4. How much support do you provide?
   a. Encourage team members to make decisions on their own
   b. Focus on developing teamwork
   c. Reduce the level of direction to a minimum

5. How do you delegate tasks to your members?
   a. Provide minimal direction
   b. Merely monitor milestones and code contribution
   c. Allow each member the freedom to choose any task he/she deem appropriate

6. How much training did you provide your team members?
   a. None     b. Minimal     c. Maximal

7. How much input did you require of your team members?
   a. None     b. Minimal     c. Maximal
The following pertains to the language, platform, etc.
8. How comfortable were you with the chosen language by your leader?
   a. Very
   b. Moderately
   c. Slightly
   d. Not all
9. How comfortable were you with the IDE, platform, and .Net framework?
   a. Very
   b. Moderately
   c. Slightly
   d. Very
10. How much improved are your communication skills after the course?
    a. Very
    b. Moderately
    c. Slightly
    d. Not all
11. Has your ability to work and communicate with others improved after taking this course?
    a. Very
    b. Moderately
    c. Slightly
    d. Not all
12. Has your knowledge base increased after taking this course (programming as well as understanding the customer’s requirements)
    a. Very
    b. Moderately
    c. Slightly
    d. Not all

Please discuss what you consider to be the major strengths/benefits of the course and address any concerns you may have. (Use the back of sheet if necessary.)

The feedback was exceptionally good. The majority of the students said that their communicational and programming skills have increased considerably. They were delighted that they were exposed to the .Net framework and C# language. Also, they were thrilled that they could tackle a graduate problem despite the fact that they are still juniors.

5. CONCLUSION AND FUTURE WORK

In summary, this course was one of the most successful classes the instructor has ever taught. As a matter of fact, the Student Opinion of Teaching evaluation for this class was completed, and the results of the evaluation were outstanding. The vast majority of the students concluded that this class was a great learning experience for them. The “client,” who is a colleague, was accurate in proposing to the instructor that the undergraduates should be exposed to a graduate level research problem. The instructor of this course will collaborate with the client to introduce future students to similar research problems, expecting them to design and construct software for robust solutions.

REFERENCES


Southeastern Louisiana University, CMPS 383 Project Site, http://bellsouthpwp.net/d/a/dafre/portal.htm