Enabling Group Work: Developing an Automated Evaluation System that Collects, Manages, and Analyzes Student Performance Data

Shaun M. Lynch, Ph.D.
Department of Mathematics and Computer Science
University of Wisconsin-Superior
Superior, WI 54880
slynch@uwsuper.edu

Abstract

Group activities have the potential to enhance student learning by stimulating individuals to explore complex issues and develop creative solutions through teamwork and interpersonal relationships. Evaluating performance, however, is very challenging and often involves collecting, organizing, storing, and retrieving significant amounts of data to measure the final product and the processes used to achieve it. In Spring Semester 2009, the author incorporated group exercises into the Management Information Systems course at the University of Wisconsin-Superior. Class time alternated between traditional lecture and group activities. Group exercises were designed around various themes that emphasized a broad range of student skills and to create a diverse learning experience. This paper presents a brief background of the effort, approach used to evaluate group work, and an automated evaluation system designed to enable group work in the classroom. The discussion includes the system's basic design and implementation and concludes with lessons learned.

1 Introduction

Group activities have the potential to enhance student learning by stimulating individuals to explore complex issues and develop creative solutions through teamwork and interpersonal relationships. However, evaluating group performance is very challenging and often involves measuring the final product and the processes used to achieve it. Metrics that define team member participation, contribution, and interaction as well as the procedures employed often contain qualitative elements and must be evaluated empirically based on data gathered using questionnaires. Incorporating multiple group activities in a single course only amplifies the logistics of collecting, managing, and analyzing data thus increasing the overhead needed to implement group-based pedagogies in the classroom.

In Spring Semester 2009, the author incorporated group exercises into the Management Information Systems course at the University of Wisconsin-Superior. Class time alternated between traditional lecture that covered important concepts and group activities that focused on their application. Exercises were designed around various themes that emphasized a broad range of student skills and to create a diverse learning experience. In all, the class contains ten distinct group exercises. The methods used to gather performance data evolved from a manual form-based system to an automated system that employed a client application developed and deployed using Microsoft's .NET Framework and a centralized database.

This paper presents the outcomes of this experience. To begin with, a brief background of the course is presented along with the rationale to incorporate group work and the expected outcomes. Next, an overview of the process is provided that identifies shortcomings of the initial form-based system leading to the design and implementation of the automated evaluation system. The discussion also includes pertinent design considerations and approaches used to grade group exercises. Finally, the overall effort is reviewed with attention to the obstacles and successes encountered along the way. The paper closes with commentary on the approach and inherent challenges of designing and implementing an automated evaluation system that supports group activities in the classroom.

2 Background

2.1 Course Background

ITS 342 Management Information Systems is an elective for students pursuing one of several concentrations in the Business Administration major. The course is part of the Information Technology and Systems program housed in the Department of Mathematics and Computer Science. A single section is offered during spring semester and normally enrolls between 15 and 25 students.

ITS 342 is a survey course that exposes students to a breadth of systems and processes used to gather, manipulate, analyze, and manage information in modern business and service organizations. Concepts range from hardware, software, database, and networking technology to using information to create knowledge and make decisions to enterprise systems that drive the supply chain, business operations, and marketing efforts [1].

For most traditional college students, unfortunately, these concepts are often disconnected from their everyday experience since many have little or no exposure to business operations. In reality however, most students use and benefit from these systems as consumers but fail to notice them whether shopping at the grocery store or placing an order online. Therefore, the primary challenge of any introductory class in Management Information Systems (MIS) is drawing student attention to these systems and making the connection between abstract system concepts and real-world practices.

2.2 Group Work Rationale

The idea of incorporating group projects into ITS 342 took root after the author attended a faculty development workshop on group work in the classroom [2]. The workshop presented group work in a broader context that included issues related to organizing teams, managing activities, and team member accountability. However, the real potential of group work rests in the semi-structured nature of activities and the ability to introduce projects that draw on existing student knowledge and experience to make connections with concepts introduced in the classroom (For further reading see [3]).

In addition, group projects give the students a chance to explore concepts through a variety of hands-on exercises. Creating concept maps, discussing current events, debating pros and cons of an issue, reviewing case studies, engaging in "information technology scavenger hunts", and constructing poster boards adds variety to the classroom experience and stimulates different perspectives. Students also become active participants in their own learning experience and must share ideas, articulate their own experiences, and work collectively to solve mutual problems.

There are tradeoffs associated with incorporating systematic group work into a course, however. First, group activities compete with other teaching practices for the finite number of contact hours in a three credit course. Second, course material must be synchronized so that group projects coincide with lectures. Third, class resources need to serve both lecture and group pedagogies. Nonetheless, a philosophy of "less is more" suites this approach since students spend more time on fundamental concepts and connecting those concepts to their own experience.

2.3 Expected Outcomes

Several factors led to the decision to implement group work into ITS 342 and influenced the expected outcomes for the project. To begin with, the Department of Business and Economics updated course requirements for the Business Administration curriculum and

assigned ITS 342 as an elective. Changing the designation of a course from *required* to *elective* immediately changes the type of student interested in taking the class as well as the enrollment. Although class numbers decreased by half, those that enrolled were significantly more motivated and interested in the subject. In addition, elective courses enjoy considerably more latitude when it comes to content than required courses that serve all students in a particular major. In this respect, group work offered a way to capitalize on reduced class numbers and as a means to build on the interest students already had in the subject.

Next, the decision to incorporate group work was based on professional curiosity and the desire to try new pedagogies. Implementing group work as the dominate pedagogy is not a trivial decision and entails fundamentally redesigning the course. Nonetheless, periodically revamping a course keeps the material fresh and often stimulates insights into the fundamental nature of the classroom experience. The challenge lay in tailoring group work concepts and principles to this particular class and translating theory into practice.

Finally, the ability of students to concentrate and focus in the classroom has changed. Millennials are regularly described as the multitasking generation [4]; however, they often have short attention spans and suffer from frequent distractions in the classroom. The personal technology they use for socializing, gathering information, and entertainment provides instant gratification but seems to rob them of their ability to focus on material presented in a lecture. To compound the problem, the author has observed that students are under intense economic pressures that lead to long working hours, sleep deprivation, poor nutrition, and a lack of physical activity. All these factors coalesce in the classroom and detract from the learning experience.

The expected outcomes of employing group work can be summarized as follows: First, provide students with a classroom experience that coincides with their interests and abilities. Second, experiment with new pedagogies and technologies to present content in fresh and lively manner. Third, utilize the learning skills students already possess upon entering the class. Fourth, create an active learning environment that promotes student involvement, contribution, and participation.

3 Evaluating Group Work

3.1 Initial Attempt

The initial attempt to evaluate group work entailed measuring student participation by group peers. The class contained 23 students organized into five groups. Ten group exercises were conducted over the course of the semester. Students were asked to complete the digital form shown in Figure 1 at the end of each exercise.

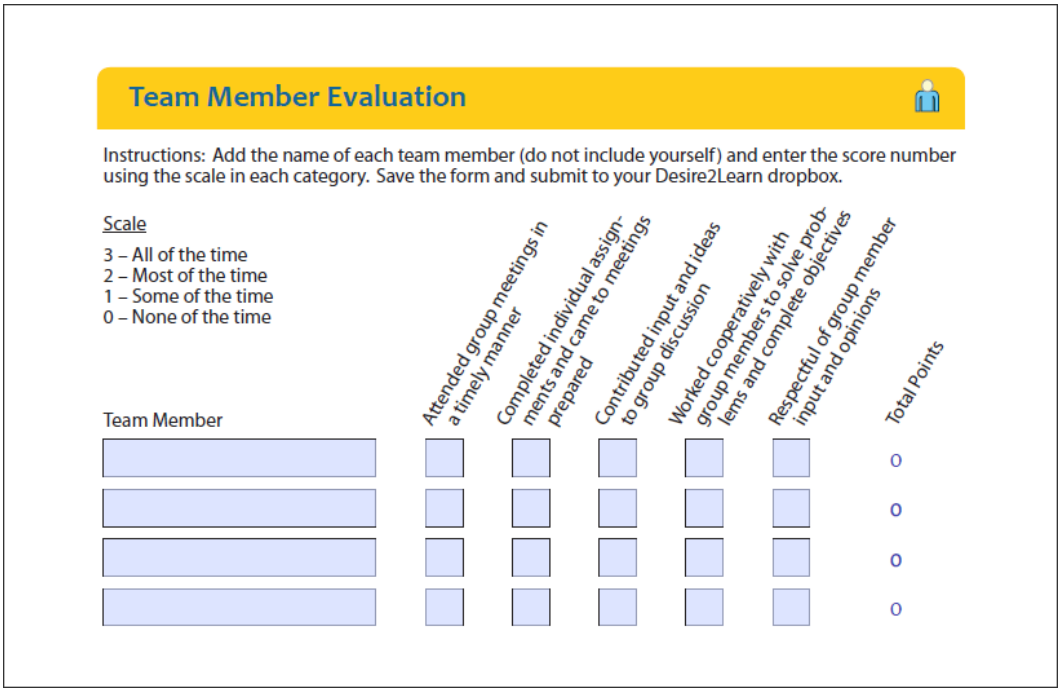

Figure 1: Team member evaluation instrument.

Adobe Illustrator was used to create the layout of the form and controls were added using Adobe Acrobat Professional. The form was configured so students could use Adobe Acrobat Reader to complete and save the document. Students entered the name of each team member and ranked each item according to the scale provided. Validation rules were used to enforce all entries except the team member name field and a calculated control tallied the *Total Points* field automatically.

Administering the form required relatively little effort. Students downloaded a blank form from the Desire2Learn course management system to their workstation. Once students completed the evaluation using Adobe Acrobat Reader, they saved the form and placed the file in their individually assigned drop box designated for the particular exercise. Approximately 230 evaluation forms were submitted in total containing the ratings of four team members on average (individuals did not evaluate themselves).

Next, student data was entered into a spreadsheet for analysis. Total points were entered in a matrix that cross-listed group members as shown in Figure 2. Each column recorded the total points assigned each team member (i.e. In the first column, Student A awarded 12, 14, 8, and 12 points to students B, C, D, and E, respectively). An average score was calculated across each row and entered as the student grade. Extracting the evaluation data consumed a great deal of time and despite efforts to streamline the process, nearly 920 values needed to be entered manually!

	Student A	Student B	Student C	Student D	Student E	Average
Student A		12	14	15	13	13.50
Student B	12		15	15	14	14.00
Student C	14	12		10	9	11.25
Student D	8	7	7		8	7.50
Student E	12	15	14	11		13.00

Figure 2: Scheme used to analyzing team member evaluations.

3.2 Shortcomings

Managing evaluation data is the primary challenge of employing group work pedagogies in the classroom. This problem severely limits the capacity to measure the outcomes associated with implementing group work and hinders effective assessment. Nonetheless, identifying the shortcomings of the process provides insight to ways to overcome the daunting task of managing data.

The first attempt previously described provided a simple way to evaluate student contributions to group work and laid the foundation for the process. However, it is useful to consider data management in a broader context that includes the basic activities of collecting, organizing, storing, and retrieving data. Breaking down the process by activity yields,

- Data Collection – Data was collected using a digital form that provided some data validation for ratings along with one calculated field that generated the total points. However, completing the form properly remained the responsibility of the evaluator.
- Data Organization – Data was organized discretely by evaluation form submitted by individual evaluators.
- Data Storage – Forms were stored individually in separate drop box on the Desire2Learn course management system.
- Data Retrieval – Data was retrieved manually by having to read the form and type values directly into a spreadsheet for analysis.

The most pronounced shortcoming was the inability to efficiently retrieve the evaluation data due to manually intense nature of effort. However, the manner in which the data was organized and stored contributed significantly to the problem making the process rigid and inflexible. In addition, the instrument used to collect the data relied heavily on the care and diligence the respondent to complete the form properly.

3.3 System Concept and Options

Ideally, an evaluation system designed for group work would provide an interface to enter data and the backend data tools that would organize, store, and retrieve data with minimal effort. At this point, the author began exploring various tools and capabilities looking for an existing product that could be adapted to the task. Of course, the product needed to readily available and free to use in an academic environment. Online survey applications seemed to be that natural choice for a project of this nature and a good place to start.

The search began with the survey tool incorporated in the Desire2Learn course management system. This initially provided the most promise due to the following factors:

- Students enrolled in the course would have immediate access to the survey instrument
- Confidentiality and privacy are assured since students would have to login to their individual accounts to complete the survey
- Questionnaire design tools provided a variety of question types, such as multiple choice, short answer, etc.
- Responses would be compiled in a form that could be manipulated using off-the-shelf productivity software
- Results could be posted and summaries distributed to individual students, groups, or the class

The product suffered from one intrinsic limitation however, the inability to tailor questions and responses to individual respondents based on group membership. Unfortunately, the author could not find a way to work around this limitation and expanded the search for a more suitable product.

Two online survey applications were next to be considered—a free service provided by SurveyMonkey.com and a University licensed application called Qualtrics. These products have commercial origins and provide sophisticated tools to create intricate questionnaires. However, they also lacked the ability to create questions tailored to individual respondents as needed leading to another dead end.

Exploring the various options available and not finding a suitable solution led to the option of creating a custom solution. The concept for the prototype would include an interface tailored to each participant based on the group they belong. Responses would be collected and sent to a centralized database. Data would be stored and organized in aggregate by logical entity. Finally, queries would be used to retrieve data for analysis.

4 Automated Evaluating System

4.1 Design and Construction

The automated evaluation system consists of three interconnected components: a client interface for the respondent, a centralized database that organizes and stores response data, and an administrative interface that consists of queries to extract data.

The *Group Exercise Evaluation* client interface application was developed in Visual Studio 2008 using Windows Presentation Foundation (WPF) and the .NET Framework. The core of the application consists of main window, an evaluation form, and a static class that contains all the methods needed to make transactions with the database.

The user interface consists of two windows. The *Group Exercise Evaluation* window shown in Figure 3 is displayed when the application launches and performs the following functions:

1. Identifies the respondent's username,
2. Displays a welcome message, and
3. Lists the exercises ready for evaluation

The user selects the particular exercise to evaluate from the list box and clicks the *Evaluate* button to continue.

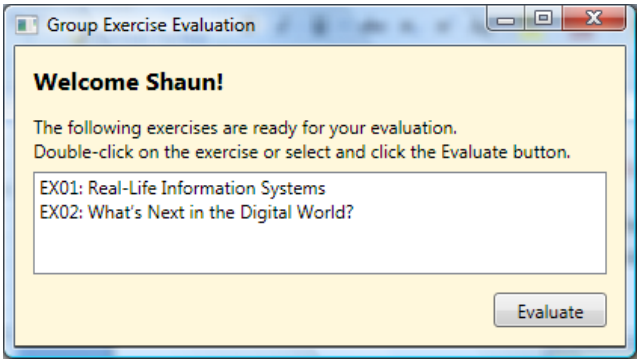

Figure 3: Group Exercise Evaluation window.

The *Evaluation Form* window is divided into two parts as shown in Figure 4. The top portion of the window contains fields for entering group data while the lower portion contains fields for entering member evaluations. The following functions are performed when the form is created:

1. Displays the exercise title
2. Creates a new group evaluation object and populates form with controls
3. Creates new member evaluation objects for each group member (except the respondent) and populates the form with controls

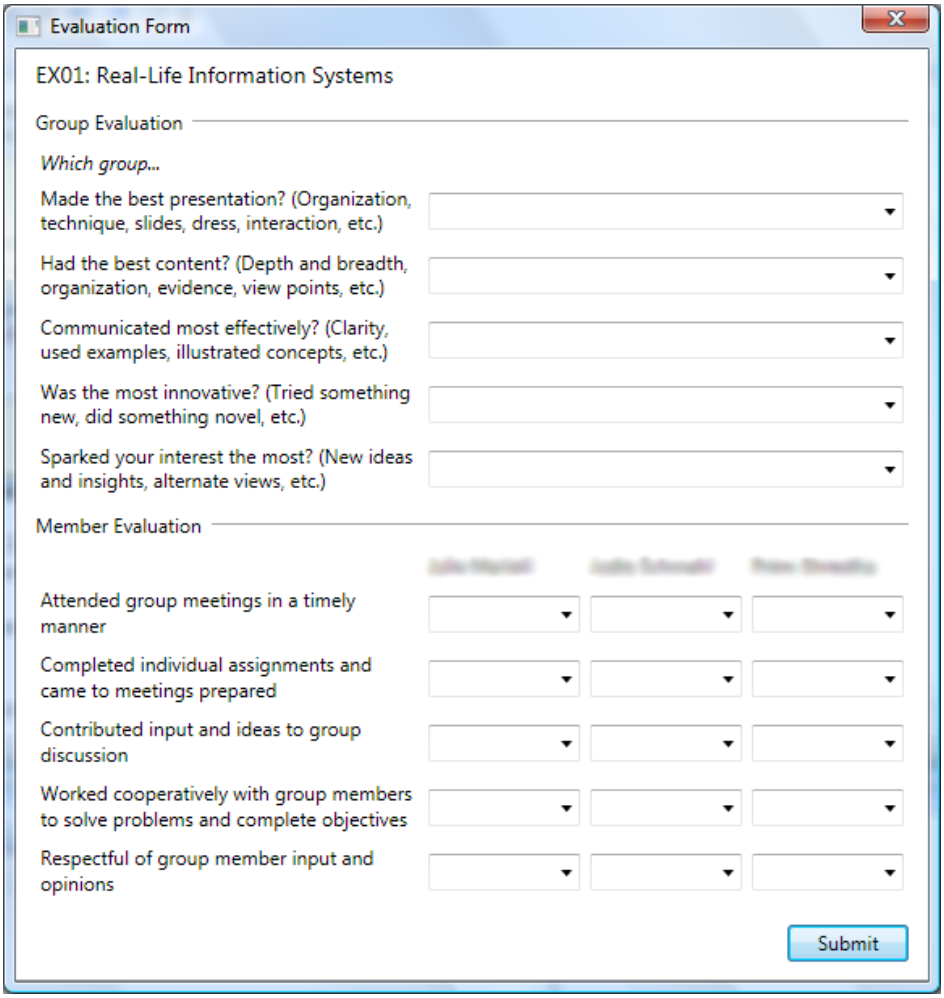

Figure 4: Evaluation Form window.

Drop-down list boxes are used for all fields and populated based on the group the respondent belongs. All items are shown in plain language. For instance, a group evaluation drop-down list box contains all the groups (except the one the respondent belongs) and the member evaluation drop-down list box contains descriptive terms *Never*, *Sometimes*, *Often*, and *Always*. Respondents must select a value since default values are not provided to prevent any initial bias.

When the respondent completes the evaluation form, clicking the *Submit* button will validate the form to ensure all fields are completed and send the responses to the database.

The application employs a static class that contains all the methods needed to interact with the database. The following methods are declared in the class:

- Load User – Retrieves the user's first and last name and create the *Student* object
- Load Assignment – Identifies which evaluations need to be completed and creates a list of *Assignment* objects

- Load Groups – Retrieves group information from all other groups except the one the respondent belongs and create a list of *Group* objects
- Load Member Evaluations – Identifies the other team members in the respondent's group and creates a list of *Member Evaluation* objects
- Store Evaluation – Stores the group evaluation data and member evaluation data for each team member, and updates the database with a timestamp indicating the update

ODBC connectivity is used since a Microsoft Access database is employed as the centralized data store. As the project evolves, this layer can be easily converted to a middle layer in a three-tier structure with a more sophisticated database solution.

The centralized database was created using Microsoft Access and contains only tables. The database model consists of seven entities as shown in Figure 5 and the principle business rules for the data model include:

- A student is a member of no more than one group
- A group can contain any number of students
- A student has no more than one evaluation per exercise

The EvaluationMemberData entity defines the structure that respondents evaluate team members. The EvaluationGroupData entity defines the structure that contains the team rankings.

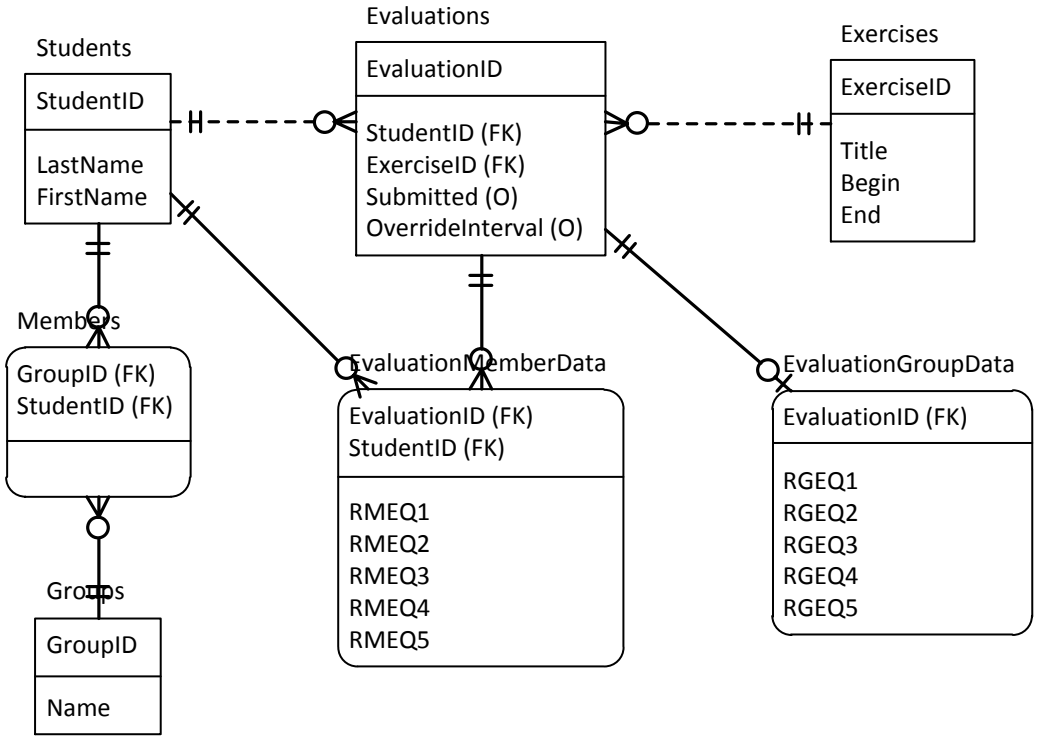

Figure 5: Entity-Relationship diagram

Finally, the system consists of an administrative interface that contains queries to retrieve data from the centralized database. The application was created using Microsoft Access and uses links to connect to the centralized database. The application consists of three queries as described,

- Submitted Evaluation – Creates a list of students in the class and whether or not they submitted an evaluation
- Team Member Evaluation – Employs a crosstab query to calculates the average contribution rating team members assign an individual
- Vote Tally – Tabulates the number of votes each group receives in each group evaluation category and the pool of votes available

The application also contains specialized functions written in Visual Basic for Access used in the queries and would also provide a location for other database objects, such as reports and forms, for the interface.

4.2 Implementation

The entire application—client interface, database, and administrative interface—only needed a managed file server on the university's domain to operate. A request was made to the Information Technology (IT) Department to create a shared folder on the public file server and assign group privileges. Once the shared folder was created, two subfolders were added. One folder contained the database and the other contained the installation files for the client application. Appropriate sharing privileges were assigned to both folders by the author.

Next, a brief installation guide was prepared for the students to install the application. The guide provided a URL to the shared folder containing the installation files and step-by-step instructions how to install and use the application. Documentation was posted on Desire2Learn for easy access by students enrolled in the class.

Clicking the setup icon in the installation folder installs the *Group Exercise Evaluation* client interface. The setup application employs Microsoft's ClickOnce deployment which installs applications on a per user basis on their respective workstation [5]. This approach is useful for low impact applications that require minimal user interaction to install. ClickOnce applications are self-contained and do not interfere with other installed applications or utilize local workstation resources. In addition, ClickOnce applications provide a mechanism for automatically updating a workstation with prerequisites and application upgrades.

Initially, there were some difficulties getting the client interface to function properly. Although the application would install correctly, an error message would post once the application opened and then proceed to close the program. However, this did not occur during the tests made before deploying the application. It was discovered that connecting to a Microsoft Access database on a shared folder using an ODBC connection string with an embedded password would trigger the error. The problem was traced back to the

permissions students accounts are assigned which are considerably more restrictive that faculty accounts. In the meantime, the password was removed to make the application operational.

Once the error had been resolved, the application worked seamlessly. Modest precautions were made to protect and backup the database while a more robust solution is developed. Students complete their evaluations within a few minutes at the end of each group exercise with minimal effort. Response data is stored and organized in the databases in a manner that is easy to manage. And, data can be retrieved quickly from the database for grading using the administrative interface.

5 Lessons Learned and Summary

It is as much an art as it is a science to successfully incorporate group work into a course. Although literature is available that discusses the advantages of group work, finding the tools to assess student contributions and learning remains challenging.

The author learned three important lessons from this effort. First, group work entails a fundamentally different way of thinking about a course. Systematic group work dramatically changes the nature of the class in both structure and how content is prepared and delivered. It requires instructors to focus on the material students really need to know and devise delivery methods that work across pedagogical boundaries. This often means thinking about the classroom experience in a different light.

Second, make incremental improvements once implemented. There will likely be numerous side effects that one cannot predict after a decision to implement systematic group work is made. Many of the side effects may be due to nuances in the approach or external factors, such as student attitudes, class size, or even when the class is offered. Making a series of correcting adjustments after a large structural change helps expose patterns and provides time to assess the effectiveness of the approach.

Third, use what works and don't let perfect be the enemy of good enough. There are always plenty of areas to improve upon—a new database application, better security, tighter integration, *et cetera*. Use what is available first and seek proof-of-concept with prototypes to determine how applications will actually be used. If developing new tools, piggyback off of existing projects or interests to make the most of the time committed to the effort.

Why is this topic important? Three reasons: First, implementing group work expands the repertoire of skills instructors bring to the classroom. Second, group work adds variety to a student's learning experience and encourages students to engage in their education. Third, assessing group work parallels broader assessment initiatives and highlights the challenges of assessing student learning. Group work is one of several pedagogies instructors can use in a classroom to promote effective use of class time.

Finally, areas for further work and development center upon building the tools for assessing student learning in exercises that utilize group work. The automated evaluation system developed up to this point is just a start and provides insight into the tools needed to incorporate group work into courses. Specific areas include: 1) a more robust database solution that provides essential security and backup protection, 2) a more flexible model that allows for different evaluation instruments over the course of a semester, and 3) a multitier architecture to modularize the design and promote the ability to upgrade components.

References

[1] University of Wisconsin-Superior (2009). *2010-2012 General Catalog.* Superior, WI.

[2] Group Work That Works (2009). Workshop: Center for Excellence in Teaching and Learning (CETL). University of Wisconsin-Superior, Superior, WI.

[3] Johnson, D. W., Johnson, R. T., and Holubec, E. J. (2009). *Circles of Learning: Cooperation in the Classroom.* Interaction Book Company, Edina, MN.

[4] Bridging the Generational Divide: Working Together to Teach Millennial Students (2009). Conference: The Collaboration for the Advancement of College Teaching & Learning, Bloomington, MN.

[5] ClickOnce Deployment for Windows Forms Applications (2011). Microsoft Corporation, http://msdn.microsoft.com/en-us/library/wh45kb66(v=VS.90).aspx.